HASSO-PLATTNER-INSTITUT
Fachgebiet Computergrafische Systeme

Design and Implementation of Non-Photorealistic Rendering Techniques for 3D Geospatial Data

Dissertation
zur Erlangung des akademischen Grades
"doctor rerum naturalium"
(Dr. rer. nat.)
in der Wissenschaftsdisziplin Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

von

Amir Semmo

Potsdam,
March 23, 2016

To my family.

Abstract

Geospatial data has become a natural part of a growing number of information systems and services in the economy, society, and people's personal lives. In particular, virtual 3D city and landscape models constitute valuable information sources within a wide variety of applications such as urban planning, navigation, tourist information, and disaster management. Today, these models are often visualized in detail to provide realistic imagery. However, a photorealistic rendering does not automatically lead to high image quality, with respect to an effective information transfer, which requires important or prioritized information to be interactively highlighted in a context-dependent manner.

Approaches in non-photorealistic renderings particularly consider a user's task and camera perspective when attempting optimal expression, recognition, and communication of important or prioritized information. However, the design and implementation of non-photorealistic rendering techniques for 3D geospatial data pose a number of challenges, especially when inherently complex geometry, appearance, and thematic data must be processed interactively. Hence, a promising technical foundation is established by the programmable and parallel computing architecture of graphics processing units.

This thesis proposes non-photorealistic rendering techniques that enable both the computation and selection of the abstraction level of 3D geospatial model contents according to user interaction and dynamically changing thematic information. To achieve this goal, the techniques integrate with hardware-accelerated rendering pipelines using shader technologies of graphics processing units for real-time image synthesis. The techniques employ principles of artistic rendering, cartographic generalization, and 3D semiotics—unlike photorealistic rendering—to synthesize illustrative renditions of geospatial feature type entities such as water surfaces, buildings, and infrastructure networks. In addition, this thesis contributes a generic system that enables to integrate different graphic styles—photorealistic and non-photorealistic—and provide their seamless transition according to user tasks, camera view, and image resolution.

Evaluations of the proposed techniques have demonstrated their significance to the field of geospatial information visualization including topics such as spatial perception, cognition, and mapping. In addition, the applications in illustrative and focus+context visualization have reflected their potential impact on optimizing the information transfer regarding factors such as cognitive load, integration of non-realistic information, visualization of uncertainty, and visualization on small displays.

Zusammenfassung

Geodaten haben sich zu einem natürlichen Bestandteil in einer steigenden Zahl von Informationssystemen und -diensten in der Wirtschaft, Gesellschaft und im Privatleben entwickelt. Virtuelle 3D-Stadt- und Landschaftsmodelle stellen hierbei insbesondere wertvolle Informationsquellen in einer Vielzahl von Anwendungen dar, wie z. B. in der Stadtplanung, Navigation, Touristeninformation und im Katastrophenschutz. Heutzutage werden diese Modelle oftmals detailliert dargestellt, um ein möglichst realistisches Bild zu vermitteln. Jedoch führt eine fotorealistische Darstellung, hinsichtlich einem effektiven Informationstransfer zum Betrachter, nicht zwangsläufig zu einer hohen Bildqualität, welche eine interaktive und kontextsensitive Hervorhebung von wichtigen oder priorisierten Informationen erfordert.

Ansätze in der nichtfotorealistischen Bildsynthese berücksichtigen insbesondere die Aufgabe eines Nutzers und Kameraperspektive, um Aspekte der Expressivität, Wahrnehmung und Kommunikation von wichtigen oder priorisierten Informationen zu optimieren. Das Design und die Umsetzung von Techniken der nichtfotorealistischen Bildsynthese für 3D-Geodaten sind jedoch mit einer Vielzahl von Herausforderungen konfrontiert, besonders dann, wenn die Geometrie, das Erscheinungsbild und thematische Daten interaktiv verarbeitet werden müssen. Infolgedessen stellt die programmierbare Architektur und parallelisierte Datenverarbeitung von Grafikprozessoren eine vielversprechende technische Grundlage zur Verfügung.

Diese Arbeit präsentiert Techniken der nichtfotorealistischen Bildsynthese, die den Abstraktionsgrad von Inhalten raumbezogener 3D-Modelle, entsprechend der Nutzerinteraktion und dynamisch-veränderbaren thematischen Informationen, berechnet und auswählt. Hierzu sind die vorgestellten Techniken in die hardwarebeschleunigte Rendering-Pipeline integriert, unter Verwendung der Shader-Technologie von Grafikprozessoren, um eine Echtzeit-Bildsynthese zu gewährleisten. Dabei werden Prinzipien der künstlerischen Darstellung, Aspekte der kartographischen Generalisierung sowie 3D Semiotik verwendet—im Gegensatz zur fotorealistischen Bildsynthese—um illustrative Darstellungen von raumbezogenen Feature-Typ-Entitäten zu synthetisieren, z. B. von Wasserflächen, Gebäuden und Infrastrukturnetzen. Darüber hinaus stellt diese Arbeit ein generisches System vor, welches die Integration verschiedener Grafikstile—fotorealistisch und nichtfotorealistisch—und ihren nahtlosen Übergang, entsprechend von Nutzeraufgaben, Kameraansichten und Bildauflösungen, ermöglicht.

Evaluierungen der in dieser Arbeit vorgestellten Techniken haben ihre Bedeutung im Bereich der Informationsvisualisierung von raumbezogenen Daten aufgezeigt, einschließlich Themengebiete der räumlichen Wahrnehmung, Kognition und Kartierung. Darüber hinaus haben Anwendungen im Bereich der illustrativen Visualisierung und Fokus-&-Kontext Visualisierung den potentiellen Einfluss dieser Techniken, in Bezug auf die Optimierung des Informationstransfers zum Nutzer, demonstriert, z. B. hinsichtlich der kognitiven Last, der Integration nichtrealistischer Informationen, der Visualisierung von Unsicherheiten und der Visualisierung auf kleinen Bildschirmen.

Acknowledgments

This thesis presents the results of my work on non-photorealistic rendering that I carried out as a research scientist at the Computer Graphics Systems Group of the Hasso Plattner Institute in Potsdam, Germany. I am very grateful to Prof. Dr. Jürgen Döllner, my adviser, for mentoring and giving me the opportunity to write this thesis in this exciting field of research. I would also like to thank Prof. Dr. Oliver Deussen from the University of Konstanz, Prof. Dr. Rüdiger Westermann from the Technical University of Munich, and Dr. Tobias Isenberg from INRIA-Saclay for agreeing to review this thesis.

Further, I like to thank my (former) colleagues Prof. Dr. Jan Eric Kyprianidis and Dr. Matthias Trapp for their collaborations and inspiring discussions on the numerous research projects. Your enthusiasm and excellent tutoring have been major driving forces throughout my studies.

Furthermore, I like to thank my colleagues for the stimulating research environment, in particular Daniel Limberger for his collaboration on the oil paint filter, Dr. Juri Engel for leading the joint user study on thematic color mappings, and Sebastian Pasewaldt for the joint work on multiperspective visualization. Thanks goes also to Rico Richter and Sebastian Pasewaldt for the positive working atmosphere while sharing an office, and Sabine Biewendt for her excellent organizational and administration support.

Finally, I owe my deepest gratitude to my family for their support, in particular my parents for their love and confidence throughout my studies.

This work has been partially funded by the German Federal Ministry of Education and Research (BMBF), Germany, within the InnoProfile Transfer research group "4DnD-Vis" (`www.4dndvis.de`), and in the Potsdam Research Cluster for Georisk Analysis, Environmental Change, and Sustainability (PROGRESS).

Potsdam, Germany, November 20, 2016 *Amir Semmo*

Contents

List of Figures

List of Tables

I

Introduction

Owing to the progress in interactive rendering and computer graphics hardware (Akenine-Möller & Haines 2008), the field of geospatial information visualization is confronted by new potentials and applications (M. Kraak 1989). The ongoing shift from static media to diverse interactive display technologies has a tremendous impact, in particular, on the design, production, and use of digital maps and map-related visualization (Buchroithner *et al.* 2000). Today, users are able to interactively adjust visualization artifacts within virtual environments (A. M. MacEachren *et al.* 1999) in real-time for an independent data exploration and analysis—providing essential means for *geospatial involvement* (Cartwright & Peterson 2007) and support of decision making.

Geospatial involvement significantly depends on how information is designed and cognitively processed (A. MacEachren 1995) to make the contents relevant within the information transfer (Brodersen 2007). A successful information transfer involves design aspects that address rational, emotional, and cognitive aspects (Montello 2002; Fabrikant *et al.* 2010) to provide a comprehensible visualization and reduce the memory load of a user. Over centuries, cartographers developed a plethora of illustration techniques and design principles (Robinson *et al.* 1995; M. Kraak & Ormeling 2003; Tyner 2010) to ensure a successful information transfer. Prominent examples are bird's-eye view maps of cities (Figure I.1).

■ *What is visualization? – "Computer-based **visualization** systems provide visual representations of datasets intended to help people carry out some task more effectively" (T. Munzner 2011).*

Yet, the design process has changed dramatically with the advancements in the (semi-)automatic acquisition and provision of high-detail 3D geospatial data, e.g., via remote sensing (Lillesand *et al.* 2014). Today, interactive 3D virtual environments are increasingly visualized using photorealistic graphics. Although a photorealistic rendering eases the mental mapping between visualization and reality, however, it does not automatically lead to high image quality with respect to an effective information transfer to a user (Ware 2004). By contrast, a design using abstract graphics promises to give more opportunities to emphasize important information (Durand 2002). The remainder of this chapter surveys some of this motivation and highlights the major contributions of this thesis.

■ *Virtual environments represent "interactive, virtual image displays enhanced by special processing and by non-visual display modalities, such auditory and haptic, to convince users that they are immersed in a synthetic space" (S. R. Ellis 1994).*

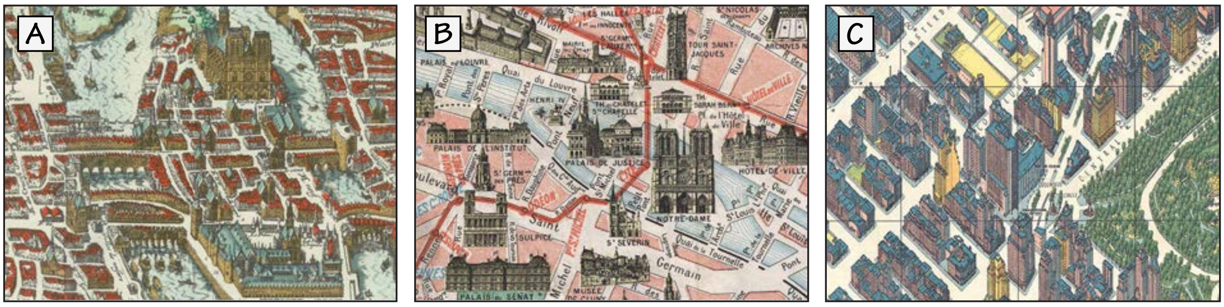


Figure I.1.: Three examples of hand-drawn maps depicting cities. (A) bird's eye view on Paris by Matthäus Merian (1615), (B) Nouveau Paris Monumental map (1932), (C) Bollmann map of Manhattan (1962).

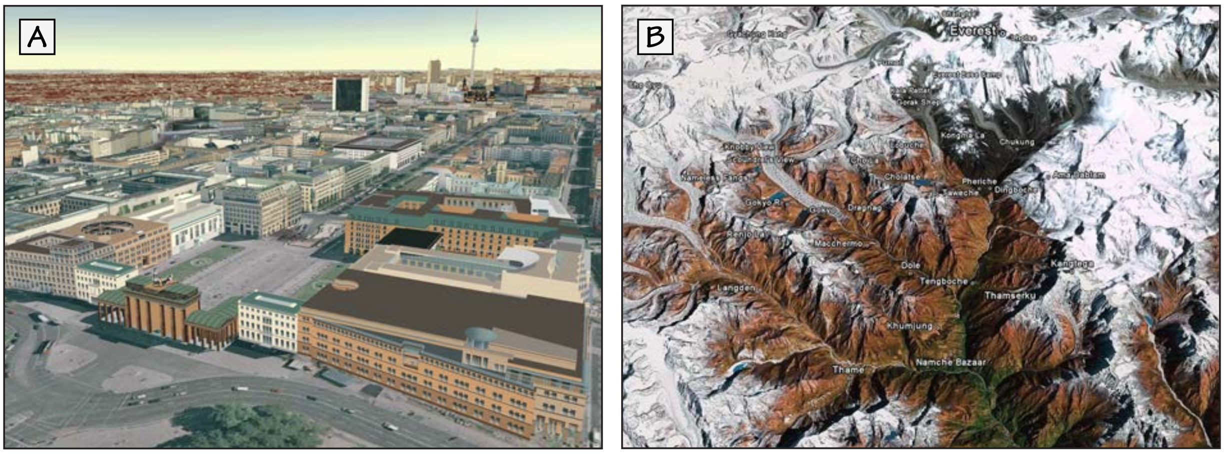


Figure I.2.: Two examples of 3D geovirtual environments. (A) 3D city model of Berlin (© Business Location Center), (B) landscape model for the Solu Khumbu Mount Everest region of Nepal (© Google Earth).

1. Limitations of Photorealistic Rendering

3D geovirtual environments (3DGeoVEs) represent key elements in a growing number of geospatial applications, systems, and services, e.g., dedicated to urban planning, navigation, tourist information, and disaster management (Biljecki *et al.* 2015). In particular, 3DGeoVEs represent and visualize geospatial data through interactive 3D graphics, based on the virtual-world metaphor, and are typically constituted by virtual 3D city and landscape models (Figure I.2):

> Virtual 3D city and landscape models represent spatial and geo-referenced data "by means of 3D geovirtual environments that basically include terrain models, building models, vegetation models as well as models of roads and transportation systems". (Döllner *et al.* 2006)

Popular examples of 3DGeoVEs are virtual globes, such as Google Earth or Bing Maps, that enable navigational and behavioral realism (Wood *et al.* 2005). They typically serve as general-purpose applications for the ubiquitous and interactive access, use and distribution of geospatial information.

■ *Photorealistic rendering uses techniques "that resemble the output of a photographic camera and that even make use of the physical laws being involved in the process of photography" (Strothotte & Schlechtweg 2002).*

Today, virtual 3D city and landscape models—as basic components of 3DGeoVEs—are characterized by inherently complex geometry, appearance, and thematic data of intrinsic heterogeneity, imprecision, and incompleteness. To this end, visualization techniques are required that leverage "the specific ways humans can perceive, cognitively process and think about spatial information" (J. Dykes *et al.* 2005) to help gain insights about geospatial phenomena, structures and relationships *hidden* in the data sets. Nonetheless, the visualization of 3DGeoVEs was and still is defined by the pioneering work of 3D computer graphics dedicated to simulate real-world phenomena with high fidelity using photorealistic rendering techniques (Durand 2002). Although photorealistic rendering is a natural way to provide an effortless and comprehensible understanding, and unbiased experience (Bishop 1994), it also comes with several drawbacks for a number of applications in geoinformation systems and services:

1. **Cognitive load:** Cluttered information displays as induced by photorealistic graphics, e.g., shadows and texture details (B. Gooch & A. Gooch

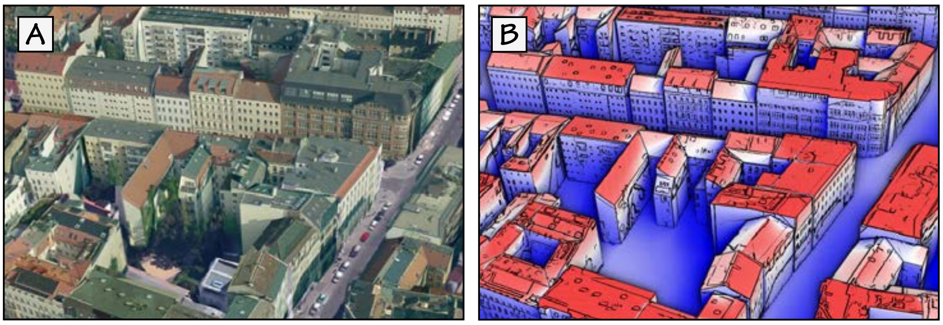


Figure I.3: Limitation of photorealistic rendering for integrating thematic information (solar potential).

(A) photorealistic

(B) proposed method: filtered textures with color mapping

2001), often lead to high cognitive load and affect a user's performance significantly, especially in exploration tasks (G. Ellis & Dix 2007). Instead, a visualization should provide just enough information as is required to carry out a given task (Norman 2013) and to ease the cognitive burden.

2. **Integration of non-realistic information:** Photorealistic rendering does not provide optimal means for the integration of non-realistic information (Figure I.3), e.g., "a photorealistic depiction of a virtual 3D city can hardly be augmented by thematic information because façade, roof, and street textures are massive image elements" (Döllner & Buchholz 2005b).

3. **Visualization of uncertainty:** There is general agreement that uncertainty is inherent to spatial data, e.g., the precision, completeness, and credibility of the underlying data, and affects the process and outcomes of the information analysis and decision making (A. M. MacEachren *et al.* 2005). Photorealism, however, is not conducive to visualize uncertainty because novice users seem to be susceptible to *naïve realism* by putting too much faith into photorealistic displays (Smallman & John 2005).

4. **Visualization on small displays:** Photorealism frequently leads to visual clutter when complex information is visualized with limited display resources, e.g., as is prevalent in smartphones and tablets. This problem is especially acute in 3D perspective views, where distant objects are mapped to increasingly fewer pixels, leading to *visual noise* (Jobst & Döllner 2008b).

5. **Memory resources:** Photorealistic rendering requires geometric and graphical detail to generate convincing results. However, processing massive data leads to high memory bandwidth—a sparse resource in service-oriented architectures (e.g., for remote visualization)—while abstract graphics tend to have a better compression ratio (Quillet *et al.* 2006).

With respect to the quality of the processed geospatial data, inherent limitations also become apparent. Because raw 3D geospatial data are characterized by their imprecision and incompleteness, a direct mapping to graphical representations—as is common in photorealistc rendering—may yield inconsistent and distracting visual artifacts. Here in particular, "the lack of sufficient detail is often frustrating and not convincing from a perception's and an observer's point of view" (Döllner & Buchholz 2005b). To this end, alternative graphical representations need to be found through a careful design that makes "visual distinctions as subtle as possible, but still clear and effective" (Tufte 1997) to optimize the information transfer to the viewer.

2. Design Aspects for 3D Geospatial Information Visualization

> "Clutter and confusion are not attributes of data
> — they are shortcomings of design." — *Edward Tufte*

Previous works evolve the collection of visual variables (Bertin 1981; Tufte 1983; Brewer 1994; Garlandini & Fabrikant 2009) to provide graphic semiology principles and optimize the geospatial information transfer mainly of 2D cartography. Because 3D geospatial visualization adheres to specific challenges such as occlusion, visual clutter, and the absence of map scales (Jobst & Döllner 2008b), the need of specialized design aspects has received considerable attention in the past years (Häberling *et al.* 2008; Jobst *et al.* 2008; Pegg 2012). Primary concerns of these works are mechanisms of independent graphic variables and their mutual impact (Jobst 2008). These variables follow the basic stages of 3D map production, i.e., modeling, symbolization, and visualization (Häberling *et al.* 2008), including psychological and physiological design aspects (Albertz *et al.* 1997; Buchroithner *et al.* 2000).

Researchers agree that the design process for the visualization of 3D geospatial data should account for application space, level of interactivity, and the audience of purpose (Hake *et al.* 2002), including the user's context and environment, skills and competence, and the purpose of visualization (Nivala & Sarjakoski 2003), e.g., the user's task. In 3D geospatial visualization, *level of abstraction* describes a key concept to consider these contextual constraints:

> Level of abstraction (LoA) refers to the spatial and thematic granularity at which 3D model contents *should* be represented. In particular, LoA extends level of detail (LoD), which is directed to decrease 3D scene complexity to optimize a system's rendering performance, to explicit visual abstraction.

In practice, the LoA is controlled by user interaction or other metrics, e.g., feature importance, to yield parameterizations of 3D semiotic variables that range from high-detail definitions, e.g., using unfiltered data, over symbolized (Häberling 1999) to completely abstract definitions (M. Kraak 1989; J. A. Dykes *et al.* 1999). Because the presentation of 3D model contents in perspective views can be manifold and complex, 3D objects need to be generalized to display relevant information in a clear and efficient manner (Petrovic 2003; Häberling *et al.* 2008). This raises the demand for rendering techniques that interactively highlight important or prioritized information in a context-dependent manner, i.e., to enable a user and media-dependent visualization. A technical foundation that fulfills these demands is laid with the field of *non-photorealistic rendering*.

3. Non-Photorealistic Rendering

Non-photorealistic rendering (NPR) is a particular domain of computer graphics that has been adopted and used in illustrative visualization (Viola *et al.* 2005; Rautek *et al.* 2008) to provide the hardware-accelerated means for reducing visual complexity and facilitating the expression, recognition, and communication of information aspects (B. Gooch & A. Gooch 2001; Strothotte & Schlechtweg 2002; Durand 2002). The main principles of NPR relate to creations of artworks (Ma

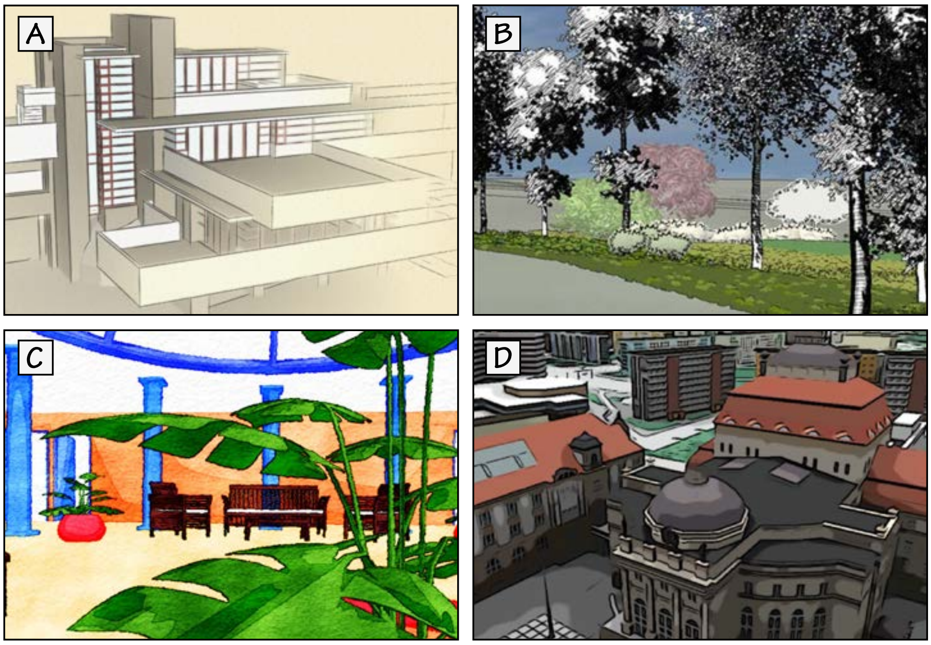


Figure I.4: Examples of non-photorealistic rendering techniques for 3D geospatial models.

(A) stylized focus pull effect (Cole *et al.* 2006)

(B) pen-and-ink illustration of a landscape (Coconu *et al.* 2006)

(C) watercolor rendering (Bousseau *et al.* 2006)

(D) cartoon rendering (Döllner & Kyprianidis 2010)

et al. 2002), where principles of abstraction and highlighting are used to evoke emotions and enhance communication aspects. Prominent examples are artworks from the Expressionism era at the beginning of the twentieth century. Therefore, NPR essentially encompasses aspects from the sciences of art, cognition, and psychology (Halper *et al.* 2003; Hertzmann 2010):

> Non-photorealistic rendering techniques comprehend depiction styles used in computer graphics that relate to regular artistic styles with the intention to communicate complex image contents by emphasizing important or prioritized information and omitting extraneous information.

The field of NPR is a highly active area of research, covering a wide range of techniques in image-based artistic rendering (Kyprianidis *et al.* 2013) and expressive rendering of 3D graphics (Isenberg *et al.* 2003; Rusinkiewicz *et al.* 2008; McGuire *et al.* 2010; Bénard *et al.* 2011). In addition, it inspired techniques used in scientific visualization, e.g., flow visualization (Brambilla *et al.* 2012). Evaluations also showed that NPR has the ability to direct a viewer's gaze to certain image regions (Santella & DeCarlo 2004; Cole *et al.* 2006). To this end, NPR not only marks an analogy to general artwork, but also relates to problems of graphical illustrations (Santella 2005), e.g., as it exists for uncertainty visualization to affect the user's level of confidence in the visualized information and the design process of 3D geospatial information visualization (Figure I.4). In this respect, NPR can assist users with a concise and unambiguous understanding by taking perceptional, cognitive, and graphical design issues into account.

The design and implementation of NPR techniques for 3D geospatial data, however, pose a number of challenges, especially when it comes to interactively processing inherently complex 3D geospatial data while representing model contents at the feature level (Jobst & Döllner 2008a). Here, a promising technical foundation is laid with the rapidly growing computing power and programmability of graphics processing units (GPUs).

4. Dissertation Plan and Contributions

This thesis aims at designing, implementing, and applying NPR techniques for 3D geospatial data. One particular goal is to identify general-purpose and feature-specific NPR techniques that enable geoinformation systems and services to compute and select the LoA of 3D geospatial model contents according to user interaction and dynamically changing thematic information. Thereby, the techniques should address the identified drawbacks of photorealistic rendering. From the perspective of 3D geospatial information visualization, this goal is reasoned by the following statement:

> **Thesis Statement**
>
> Non-photorealistic rendering represents an effective visualization category—situated between computer graphics and cartography—for the interactive, context-dependent visual design of complex 3D geospatial information.

Contributions

The validity of the thesis statement will be demonstrated by NPR techniques that interactively compute and select the LoA of virtual 3D city and landscape models. To this end, the techniques seamlessly integrate into hardware-accelerated rendering pipelines by utilizing shader and texturing technologies of GPUs for real-time image synthesis. To summarize, this dissertation makes the following contributions:

- An extended 3D semiotic model that complies with the general, interactive visualization pipeline is developed.
- An approach for LoA texturing of virtual 3D scenes by means of image filtering and deferred shading is proposed, where quantitative and qualitative evaluations confirm significant effects for the integration of thematic data and their perception in 3D GeoVEs.
- NPR techniques that employ principles of 3D semiotics and cartographic design to synthesize illustrative renditions of water surfaces, buildings, and infrastructure networks are provided.
- A generalized system is proposed that enables to integrate different graphic styles—photorealistic and non-photorealistic—and that provides their seamless transition according to user tasks and perspective view in 3D spaces. The system integrates an interaction framework for parameterization that supports touch and natural language inputs.
- Approaches for image abstraction and stylization of aerial and terrestrial images by bilateral and interactive oil paint filtering are presented.
- The potential benefit of the proposed techniques at several applications is discussed, including focus+context and illustrative visualization.

The proposed techniques are designed for generic application and seamlessly integrate into contemporary rendering systems. In terms of applications, this thesis focuses on textured polygonal meshes as the pre-dominant type of geometric representation for 3D geospatial models.

Structure of the Thesis

The remainder of this thesis is structured as follows:

II. Background. The next chapter outlines characteristics of 3DGeoVEs, summarizes graphic semiology principles and the construct of the *visualization pipeline* for effective data processing, and provides an extended 3D semiotic model. Further, the chapter gives an overview on the concepts of the real-time rendering pipeline, including core mechanisms such as texturing and shading.

III. Related Work. This chapter surveys related work in the field of NPR, focus+context visualization, and 3D geospatial information visualization. Approaches found in the cartography-oriented visualization of 3D geospatial data are drawn with respect to the extended 3D semiotic model.

IV. Level-of-Abstraction Texturing. This chapter proposes generalized rendering techniques for the abstraction and stylization of 3D scenes. It comprises a texture-based LoA technique that enables arbitrary 2D image filters to interactively process textured 3D models while preserving monocular depth cues.

V. Cartography-Oriented Visualization. This chapter provides real-time rendering techniques for a semantics-based and thematic visualization of 3D model contents, and specialized techniques for geospatial feature type entities of virtual 3D city and landscape models: building models, infrastructure networks, and water surfaces.

VI. Interactive Level-of-Abstraction Transitions. This chapter proposes a concept and implementation of a system that combines and seamlessly integrates various LoAs via image blending and geometric transformations. The system enables to emphasize important information and focus regions using photorealistic or detailed graphic styles, while abstracting less important information and context regions using non-photorealistic graphic styles to direct a viewer's gaze by salient stimuli attraction.

VII. Image-based Abstraction and Stylization. This chapter provides a generalized method for image-based abstraction of aerial images, and an artistic rendering technique that simulates oil paint effects by image filtering.

VIII. Conclusions and Future Research. Finally, this chapter concludes this thesis, reviews the achieved results and gives a prospect on future work.

The technical chapters identify and discuss potential impacts of the proposed techniques for geospatial applications such as map exploration, urban planning, and disaster management, including quantitative and qualitative evaluations.

II

Background

Geovirtual environments are typically composed of complex geospatial data that are not directly perceivable by humans (A. M. MacEachren & M.-J. Kraak 2001; Döllner *et al.* 2006). To this end, visualization concepts and interactive techniques are required that effectively communicate complex geospatial data by enabling human senses (e.g., vision) and cognitive qualities. Here, the continuous advancements in the programmability, computing power and availability of computer graphics hardware (Hughes *et al.* 2013) plays a major role to transform raw geospatial data into image-based representations, and to evolve geovirtual environments as ubiquitous, interactive, and versatile media. In particular, with the advent of standardized application programming interfaces (APIs) such as OpenGL, software engineers are able to leverage the high parallelism of computer graphics hardware to meet the performance requirements of interactive rendering systems (Akenine-Möller & Haines 2008).

This chapter outlines characteristics of 3DGeoVEs and the *visualization pipeline* as a generally accepted model for information processing by researchers and practitioners (Ware 2004). At this, design principles and core variables considered important for an effective design of 3D geospatial visualization are revisited and enhanced towards an extended 3D semiotic model. The chapter then provides an overview of mechanisms for the real-time rendering pipeline (Akenine-Möller & Haines 2008) that are implemented on GPUs and used by the proposed NPR techniques. This includes texturing and shading, data structures such as geometry buffers, and the concept of deferred shading.

1. 3D Geospatial Information Visualization

This thesis deals with virtual 3D city and landscape models as major components of 3DGeoVEs. Their characteristics and visualization are outlined in this section.

1. Characteristics of 3D Geovirtual Environments and Taxonomy

Applications of 3DGeoVEs typically comprise the presentation, exploration, analysis, and editing of space-oriented information. In contrast to static map-related representations, 3DGeoVEs provide an interactive and dynamic visualization of 3D geospatial information. A. M. MacEachren *et al.* (1999) propose characteristics of 3DGeoVEs that separate them from static representations and contribute to the sustainment of *virtuality*:

1. ***Immersion.*** Immersion refers to the sensation of *being in an environment* and influences the perception of information; visually, auditory or on a tactile basis. Dependent on the output device, immersion can ease orientation tasks, and improve the sensation of geospatial phenomena and depth.

2. ***Interactivity.*** Contrary to static media and representations of geospatial data, 3DGeoVEs facilitate the navigation within virtual worlds to ease exploration tasks. Common navigation techniques manipulate the virtual camera and viewpoints, and comprise the interaction with 3D scene entities for a user-defined adjustment of visual properties (Jankowski & Hachet 2014). These adjustments may either be triggered explicitly or indirectly, e.g., the latter to implement a view-dependent LoA.

3. ***Information intensity.*** Information intensity refers to adjusting directly (or indirectly) the LoD used for the visualization of 3D scene contents, e.g., dependent on view metrics such as the view distance or view angle. It is linked to the research question how concepts for LoA can be integrated into an interactive, context-aware visualization process.

4. ***Intelligence of entities***. Context-sensitive behaviors of 3D scene entities may actively ease navigation and interaction tasks. At this, coupling the intelligence of scene entities with interactive concepts may provide dynamic behaviors that adapt to a user's context. Prominent examples are computer games that implement state-based behaviors for a dynamic environment.

Virtual 3D city and landscape models are commonly composed from heterogeneous sources of geospatial information such as LiDAR data, cadastral data, or aerial images. Their correspondent data sets are either based on automatic, semi-automatic or manual data acquisition methods (Hu *et al.* 2003), most prominently remote sensing techniques for the acquisition of high-detail terrain models and high-resolution aerial images, as well as CAD tools for the modeling of high-detail 3D objects. The OGC standard CityGML[1] facilitates a persistent storage and exchange of virtual 3D city models including multiresolution entities (Kolbe 2009). The thematic model of CityGML classifies 3D model contents according to a feature type taxonomy—a *feature* is defined by the OGC as an abstraction of a real-world phenomenon and describes georeferenced geometry with auxiliary attributes[2]:

1. ***Digital terrain models:*** are specified as triangulated irregular networks, regular rasters, break and skeleton lines, or mass points.

2. ***Sites:*** buildings (parts, installations, and interiors), bridges, and tunnels.

3. ***Vegetation:*** solitary objects (e.g., trees), vegetation areas, and volumes.

4. ***Water bodies:*** thematic aspects and 3D geometry of rivers, canals, and lakes.

5. ***Transportation facilities:*** roads, tracks, railways, and squares modeled as graph structures or 3D surfaces.

6. ***Additional models:*** land use, city furniture, and generic models.

In particular, the thematic model is used in this thesis as a taxonomy to implement feature-specific visualization techniques that are based on principles of cartographic design.

[1] OpenGIS® City Geography Markup Language Encoding Standard (Open Geospatial Consortium) — http://www.opengeospatial.org/standards/citygml (Last followed on November 20, 2016)

[2] The OpenGIS Abstract Specification - Topic 5: Features (Open Geospatial Consortium) — http://www.opengeospatial.org/standards/as (Last followed on November 20, 2016)

2. Semiotics and Design Aspects

Semiotics deal with the study of symbols and how they communicate information in a meaningful way. In cartography, geospatial information is generally encoded using a semiotic model and transferred to a map representation, where the map producer and consumer typically *agree* on aspects of location or space (Brodersen 2007). To improve the communication process, the map should be readable, comprehensible, and visualized in a way that the information can be memorized easily, which may also be achieved by addressing emotional aspects (Kolacny 1969). In the following, all forms of map-related representations of mixed 2D and 3D geospatial information are referred to "3D maps" as a pragmatic notion.

2D Semiotics

The graphic-semiotic approach described in the seminal work by Bertin (1981) provides a theoretical foundation on how (geospatial) phenomena can be efficiently communicated using sign systems. Thereby, graphical elements and visual variables play a key role in the definition of symbols and their translation to information. The original notation by Bertin (1981) refers to *marks* as graphical elements that comprise five different types: *points*, *lines*, *areas*, *surfaces*, and *volumes*, which can be modified by seven visual variables: *location*, *size*, *shape*, *orientation*, *(color) hue*, *(color) value*, and *texture* (Figure II.1). The works by A. MacEachren (1995) and A. MacEachren *et al.* (2012) extended this list, adding the variables *(color) saturation*, *arrangement*, *crispness*, *resolution*, and *transparency*. The multifunctioning interplay of these variables is essential to the definition of cartographic design spaces (A. MacEachren 1995) and supports a pre-attentive visual processing (Garlandini & Fabrikant 2009).

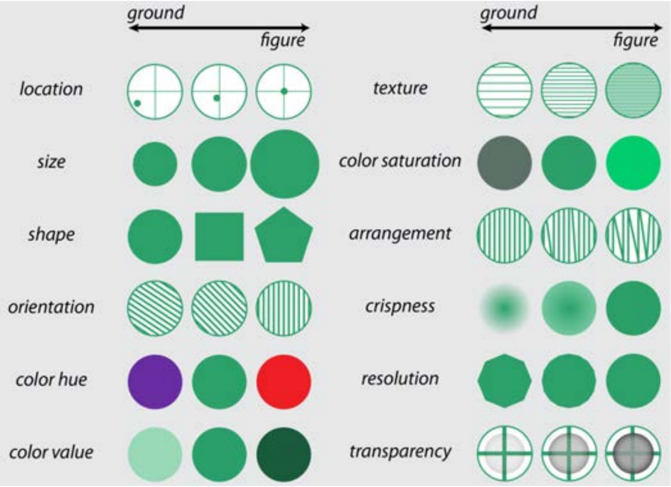


Figure II.1.: Visual variables defined by Bertin (1981), A. MacEachren (1995), and A. MacEachren (1995). Image courtesy by Robert E. Roth.

3D Semiotics

3D geospatial information visualization utilizes a central perspective view—similar to the human's 3D visual perception—and provides a natural access to geospatial information (Jobst 2008) and immersion (A. M. MacEachren *et al.* 1999) into the geospatial data. However, it also comes with additional challenges that arise within the production stages of "3D cartography" (Jobst & Döllner 2008b):

1. ***Occlusion.*** Compared to close objects, distant objects typically are not visible or are only partially visible due to occlusion.

2. ***Partial use of screen space.*** In low view angles, e.g., on a pedestrian level, a large amount of screen space is often visually cluttered such as in areas far away from the virtual camera.

3. ***Absence of map scales.*** A uniform scale enables a user to measure, estimate, and compare spatial relations, which however does not apply to spatial 3D

models that are rendered with a perspective view. The variation of scales complicates the estimation and comparison of spatial relationships.

4. ***Visual clutter.*** Because of perspective distortions, the size of on-screen objects decreases with the linear perspective. Hence, distant objects tend to be difficult to identify as they are mapped to increasingly fewer pixels.

During the last decade, cartographers developed and discussed semiotic models for 3D geospatial data that address these challenges. In particular, user involvement is considered to be a critical design aspect to maintain an iterative feedback loop between the visualization system—as design instance—and the user's requirements—as consumer (Buchroithner *et al.* 2000; Peterson 2005). This feedback loop changes the strict separation between the role of the map producer and map consumer (Jobst *et al.* 2008), because eventually the consumer becomes the producer. From the perspective of a task-oriented visualization, Reichenbacher & Swienty (2007) note that a good interactive design presents as much information as needed for relevant information and as little as required for context. In the following, some of the important design aspects are summarized.

Design aspects by Reichenbacher (2007). Depth cues are essential for perceiving model contents as three-dimensional and ease the mental modeling (Cutting & Vishton 1995; Pfautz 2000; Goldstein 2010). Reichenbacher distinguished perceptional design aspects that relate to depth cues between physiological and psychological cues. On the one hand, physiological cues refer to the *retinal parallax*, *convergence*, *accommodation* and *movement parallax*, and are "mainly affected by the structure of the human vision system and the relations between observer and object". On the other hand, psychological cues refer to the *retinal image size*, *linear perspective*, *aerial perspective*, *occlusion*, *shading* and *texture gradient*, and are more affected by the viewing experience and visualized model contents.

Design aspects by Häberling *et al.* (2008). Häberling *et al.* defined modeling, symbolization, and visualization as major design stages for the generalization in 3D spaces. Design aspects can be configured at each stage according to a user's context to ease the communication process. Modeling includes aspects for mapping raw geospatial data to *geometry*, *semantic attributes*, and *position attributes* suitable for rendering. The visual appearance of 3D geospatial objects is configured during symbolization and comprises traditional graphic appearance attributes, enhanced by attributes related to *text objects* (labeling) and *animation*. Finally, the visualization stage defines the mapping of 3D graphic primitives and attributes to the presentation medium and is configured by design aspects of the virtual camera, e.g., the *field-of-view* and type of *projection*, and scene-specific parameters related to *lighting*, *shading*, and *atmospheric rendering*.

Design aspects by Jobst *et al.* (2008). Jobst *et al.* proposed an extended semiotic structure for the visualization of 3D geospatial model contents, and discussed 3D mechanisms on graphical core variables and their mutual impact on each other. They note that an "extension for 3D cartography means that all added design mechanisms of 3D massively influence the coding with graphical variables and elements", and thus proposed a semiotic structure that includes variables of vision, composition and the psychological influences. In particular, Jobst *et al.* (2008) added *progressive* and *degressive* environmental projections as key types of multiperspective views to address the issue of occlusion in 3D spaces.

3. Visualization Pipeline and Extended 3D Semiotic Model

The transfer of existing, proven cartographic principles to modern media and imaging technologies, and the development of new cartographic methods, are key challenges for current and future research. In cartographic theory, a coupling of 3D semiotic models and structures to a well-established framework that supports user interaction and generalization has not been formulated. From a computer graphics perspective, however, this is mandatory to provide the conceptual means for an automated, hardware-accelerated visualization process, and to enable reference pipelines for visualization to be combined with established design aspects. To this end, the 3D semiotic model is extended to comply with the *visualization pipeline* of 3D computer graphics (Ware 2004).

Visualization Pipeline

The *visualization pipeline* is used as an abstract model to structure the process of visualizing geospatial data into three major stages: the *pre-processing* of raw data with filtering as predominant operation, the *mapping* of data onto computer graphical representations, and the *rendering* of mapped data for transforming these representations into visualization artifacts. An overview of the stages is given in Figure II.2.

Pre-processing Stage. The pre-processing of raw geospatial data is essential to reduce the level of complexity. Common operations include the filtering, selection, compression, and integration of geospatial data. All operations should be context-aware with respect to the user and task because often only a subset of the data is required for further processing.

Mapping Stage. The pre-processed data are mapped onto computer graphical representations as scene entities. Basic entities comprise textures and geometrical primitives, e.g., used to model triangular irregular networks, and appearance attributes such as color and material information. The data structure and encoding of the entities is specific to the rendering engine. Typical data structures represent scene graphs for a logical and spatial arrangement of the entities, and (linked) buffer objects suitable for shading.

Rendering Stage. The mapped data and scene description are synthesized to visualization artifacts that ease the perception by human sensory. The rendering stage is typically defined as a separate, hardware-accelerated rendering pipeline that is optimized with respect to the high parallelism of GPUs.

An important "interface" lies between human sensory and the visualization artifacts that enable "the scientist or engineer to perceive visually features which are hidden in the data but nevertheless are needed for data exploration and analysis" (Gershon 1994). Thereby, a common understanding is that a user has the capability to influence each processing stage, e.g., to adjust the data selection in the pre-processing stage, or parameters for the virtual camera via navigation techniques. To maintain immersion and interactivity, the feedback loop between user interaction and rendering should perform at interactive or real-time frame rates, i.e., a minimum of 6 or 15 frames per second (fps) respectively. This characteristic is a key component in the design of context-aware visualization systems, and thus considered by the NPR techniques proposed in this thesis.

■ *"The rate at which images are displayed are measured in frames per second (fps) [...]. At around 6 fps, a sense of interactivity starts to grow. An application displaying at 15 fps is certainly real-time." (Akenine-Möller & Haines 2008)*

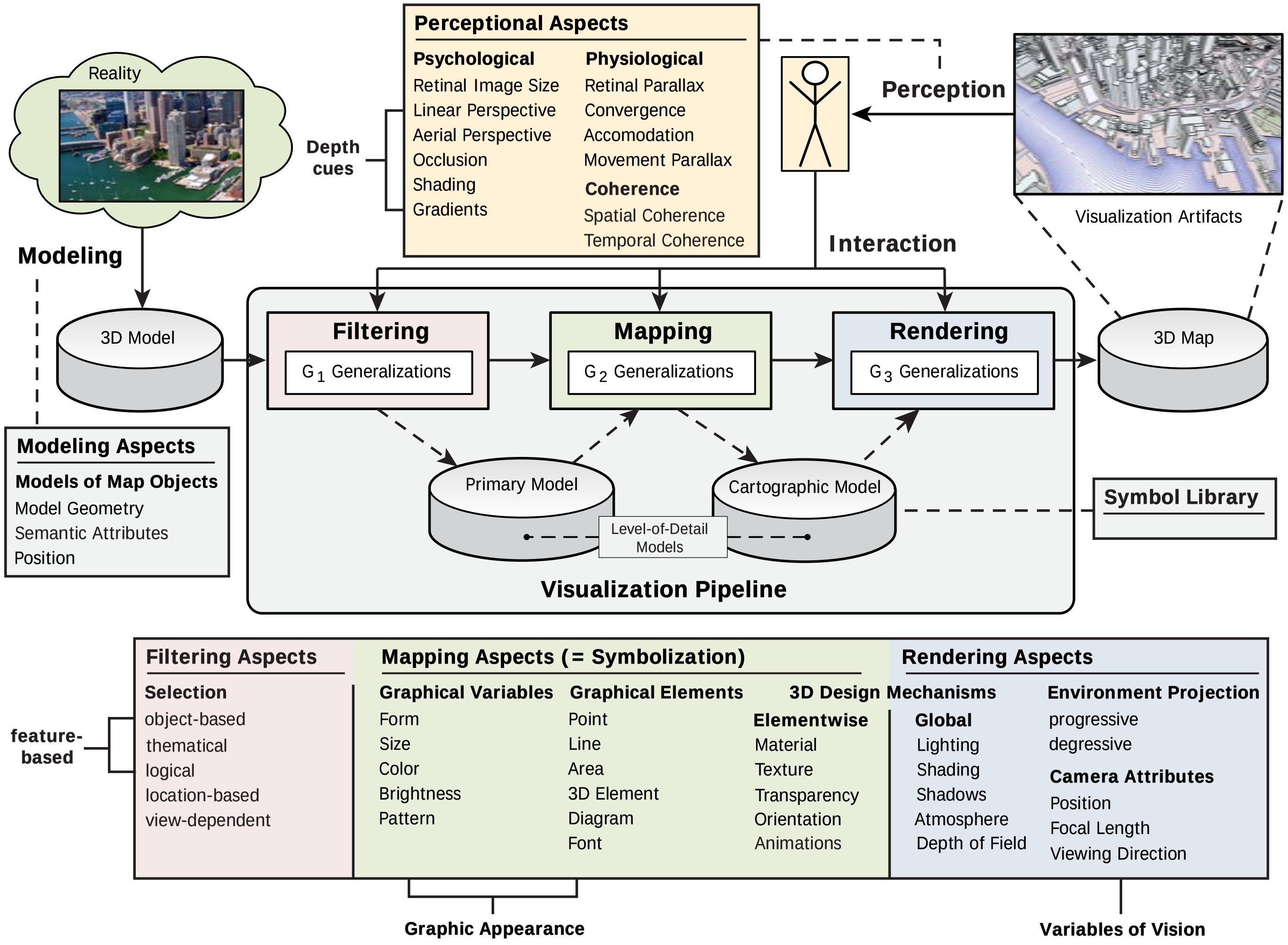


Figure II.2.: Extended 3D semiotic model that integrates the design aspects of Reichenbacher (2007), Jobst *et al.* (2008) and Häberling *et al.* (2008) into the general visualization pipeline, coupled with concepts for cartographic generalization (Foerster *et al.* 2007). The visualization process comprises five interfaces: 1) modeling of real-world phenomena (features), 2) filtering and pre-processing, 3) mapping of the primary model to a cartographic model via symbolization, 4) rendering, and 5) the perceptional interface.

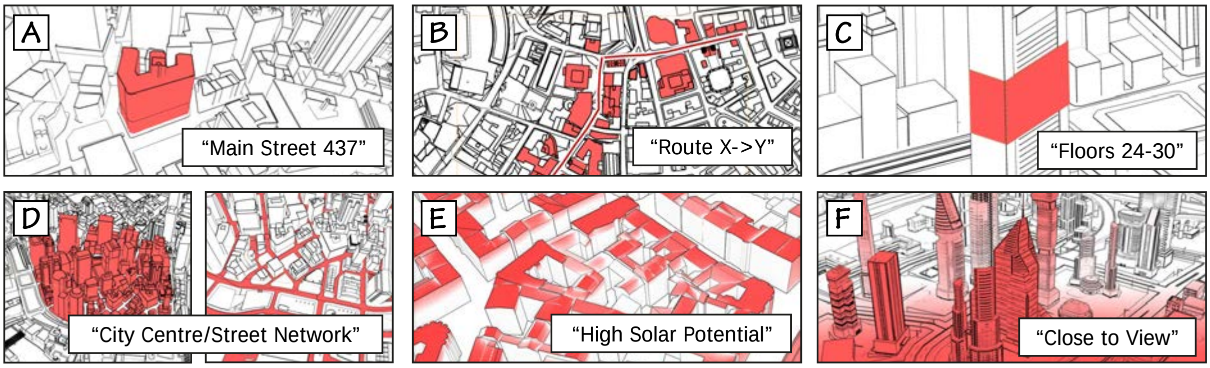


Figure II.3.: Exemplary filtering functions for focus and context definitions. (A) object selection, (B) 2D region of interest, (C) 3D region of interest, (D) logical selection, (E) thematic selection, (F) view-dependent selection.

Extended 3D Semiotic Model

Figure II.2 categorizes variables of the models by Reichenbacher & Swienty (2007), Jobst *et al.* (2008) and Häberling *et al.* (2008) in the processing stages of the visualization pipeline, where design aspects fall into five categories: modeling, filtering, mapping (symbolization), rendering, and perception. The proposed model embeds cartographic generalization operators to transform geospatial data into human-readable maps (McMaster & Shea 1992; A. MacEachren 1995; Sester 2007) such as class selection, simplification, enhancement, displacement, or typification (Foerster *et al.* 2007). Generalization operators can be applied (1) to the original geospatial data of a virtual model in the filtering stage, leading to a generalized primary landscape model, (2) to the mapped data in the mapping stage (symbolization) to yield a generalized cartographic model, and (3) at the rendering stage to obtain a generalized graphical representation. In practice, visualization techniques use a combination of generalization operators and can be configured by the user to provide adaptive, dynamic generalized models for interactive 3D systems that conform to graphic semiology principles. Here, additional design aspects related to filtering and perception are proposed:

Filtering Aspects. The filtering stage provides an effective interface to select only the information required for further processing. Dealing with 3D geovirtual environments, six categories for focus definition are distinguished (Figure II.3), which may be combined for multivariate filtering:

- ***Object selection***: The highlighting of single or groups of objects that serve as landmarks according to a user's context and interest.
- ***2D region of interest***: The highlighting of objects that are located close to, or within a 2D region of interest.
- ***3D region of interest***: The spatial highlighting of objects or components with additional constraints in height.
- ***Logical selection***: The selection of objects or components with respect to semantic constraints such as feature type, e.g., street networks.
- ***Thematic selection***: The selection of objects or components with respect to thematic data and according to a range of interest.

- *View-dependent selection*: The definition of regions of interest according to view-based metrics (e.g., viewing distance to the virtual camera).

Coherence Aspects. Image-based abstraction and NPR that is subject to artistic stylization often aims to reproduce a hand-drawn look. Here, a key challenge is to "minimize distracting flickering and sliding artifacts" (Bénard *et al.* 2011) and to preserve the cues that infer depth. In particular, the *shower-door effect* should be avoided at the rendering stage, i.e., the impression that a 3D scene is observed through a semi-transparent layer of (texture) marks. To this end, the extended semiotic model considers temporal and spatial coherence of graphical elements as perceptional design aspects. Spatial coherence applies when graphical elements coherently translate with the perspective view, whereas temporal coherence refers to the relative similarity between successively rendered images.

The extended 3D semiotic model is used in this thesis as a taxonomy for cartography-oriented visualization of 3D geospatial data. Moreover, selected design aspects are considered in the technical sections to address challenges that arise with 3D spaces such as filtering and environmental projections with respect to visual clutter and occlusion.

2. Hardware-Accelerated Computer Graphics

The NPR techniques presented in this thesis use the programmable rendering pipeline implemented on GPUs to rasterize 3D scenes as 2D images. The core mechanisms of the rendering pipeline are outlined in this section.

1. Programmable Rendering Pipeline

■ *"Real-time rendering is concerned with making images rapidly on the computer. [...] An image appears on the screen, the viewer acts or reacts, and this feedback affects what is generated next. This cycle of reaction and rendering happens at a rapid enough rate that the viewer does not see individual images, but rather becomes immersed in a dynamic process." (Akenine-Möller & Haines 2008)*

Hughes *et al.* (2013) denote four conditions of GPUs "to become and remain successful": performance differentiation, workload sufficiency, strong market demand, and the inertia of ubiquity. The performance differentiation to general-purpose CPUs is of particular interest, because GPUs employ up to thousand programmable processing units executed in parallel to accelerate the rendering process and enable real-time performance (Akenine-Möller & Haines 2008).

To obtain rendered images from 3D scene descriptions, "the input data undergoes a series of steps (or stages) called a *graphics pipeline*" (Boreskov & Shikin 2013). The work presented in this thesis is based on the OpenGL rendering pipeline that provides a standardized API for implementing rendering techniques. The main functionality of OpenGL is to convert vector-based primitives with pixel data (e.g., textures) into a rasterized image using the OpenGL state machine as graphics pipeline. This graphics pipeline is partitioned into three conceptual stages: an application stage, a geometry stage, and a rasterizer stage:

Application Stage. This stage is part of the graphics engine that is written in a high level language and is executed on the CPU to control and configure the basic stages of the pipeline. Scene graphs are traversed and scene entities are transmitted to the geometry stage—into video memory used by a GPU—as vertex and pixel data. The stage also includes optimization techniques to transmit only the data required for rendering to improve the performance, e.g., managing LoD models and performing culling with respect to the virtual camera settings.

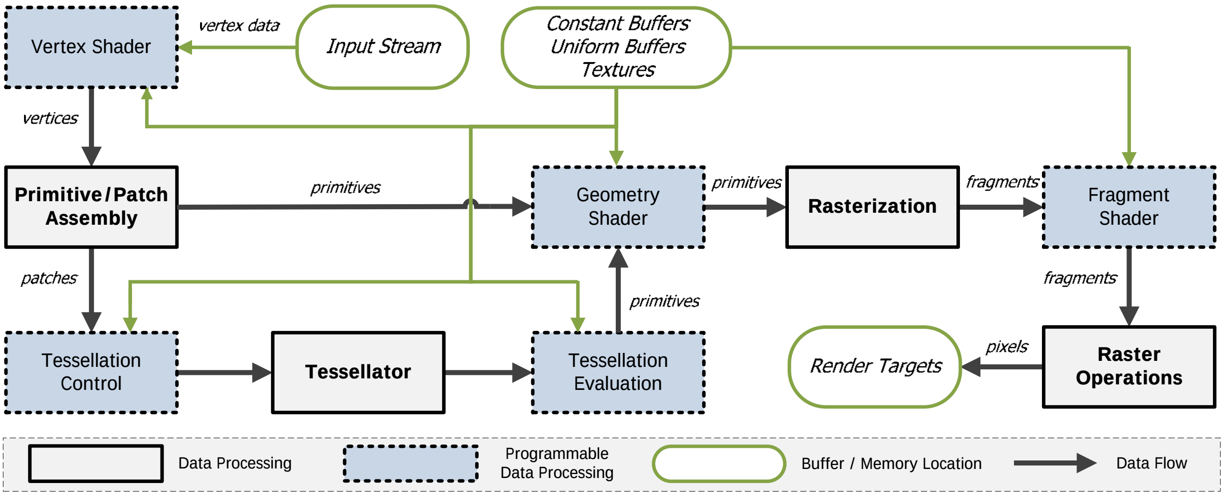


Figure II.4.: A simplified rendering pipeline according to the OpenGL 4 specification. All stages for data processing are parallelized. The render targets typically comprise a color buffer (framebuffer) used to display the final image.

Geometry Stage. Parallelized per-polygon and per-vertex operations apply at the geometry stage, e.g., to transform vertex data of primitives (e.g., triangles) into world space and view space as defined by the virtual camera. The transformed primitives are then projected into a unit cube with normalized clipping coordinates used for screen mapping, and are transmitted to the rasterizer stage.

Rasterizer Stage. This parallelized stage comprises functionalities to convert primitives to fragments that are stored in a framebuffer and displayed by the respective window system. The main functionalities comprise per-pixel operations that discard fragments outside a view frustum, and texturing that maps (color) values of images onto the fragments, e.g., according to the texture coordinates defined by the 3D scene entities.

OpenGL incorporates the OpenGL Shading Language (GLSL) that enables to program the rendering pipeline at the geometry and rasterizer stage via vertex, tessellation, geometry and fragment *shaders* that are written in a C-like high-level language (Shreiner *et al.* 2013). In particular, vertex shaders can be used to manipulate vertex properties such as position, texture coordinates, and color values, geometry shaders and tessellation shaders to integrate hardware-accelerated LoD rendering, and fragment shaders to manipulate pixel colors of rendered images, e.g., to implement custom shading and lighting effects. Figure II.4 outlines the simplified rendering pipeline of OpenGL 4 that includes the programmable stages for data processing. This *programmable rendering pipeline* lies at the core of NPR techniques, and thus is subject to the implementation of cartography-oriented design aspects as proposed by the 3D semiotic model. More detailed descriptions of the stages are found in the textbooks of Akenine-Möller & Haines (2008), Hughes *et al.* (2013), and Shreiner *et al.* (2013).

2. Core Mechanisms and Advanced Rendering Techniques

In the following, core mechanisms and advanced rendering techniques implemented with the programmable rendering pipeline—and used in this thesis—are outlined. This includes texturing and deferred shading using geometry buffers.

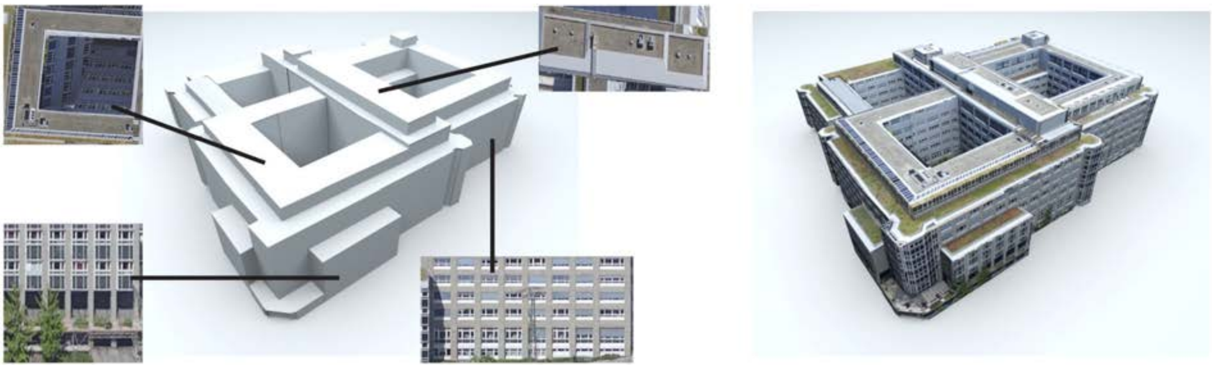


Figure II.5.: The visual appearance of 3D geospatial features is often modeled in texture space to reduce geometric complexity. Here, detail information is encoded in 2D color images (diffuse maps) used for texturing a virtual 3D building model. Model: © City of Nuremberg – Amt für Geoinformation und Bodenordnung, all rights reserved.

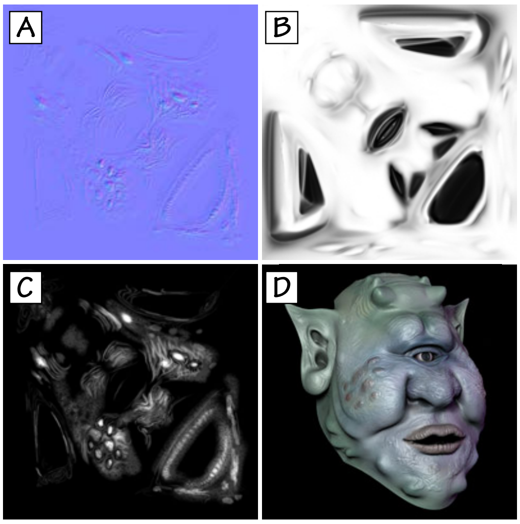


Figure II.6.: Examples of texture maps that encode geometry-related information. (A) normals, (B) ambient occlusion, (C) bump, (D) shaded model using A-C. Model: Jerry the Ogre © Keenan Crane.

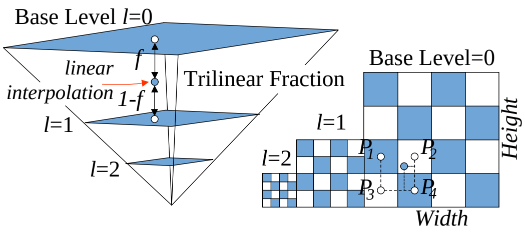


Figure II.7.: Mipmap image pyramid and trilinear sampling. Image courtesy by Ewins *et al.* (1998).

Texturing

Texture mapping is a key technology in today's computer graphics hardware for the visual design of virtual 3D scenes (Haeberli & Segal 1993). Texture maps (images) comprise arrays of data values that are stored in video memory and used to enrich 3D scene entities. Here, one approach is to reduce geometric complexity by representing shape with fine scale (color) variations encoded in diffuse maps (Figure II.5). Besides color information, this approach has been generalized to include arbitrary surface properties to enrich shading and lighting effects such as normal or displacement information (Figure II.6).

Texture mapping is based on a distinguished texturing pipeline that assigns texture data to an object-space location. This pipeline comprises four basic stages: (1) the object-space location is mapped to parameter-space coordinates using a *projector function* that interpolates vertex texture coordinates, (2) the parameter-space coordinates are transformed into texture space using a *corresponder function* that clamps values using addressing modes (e.g., wrap, mirror, clamp to edge), (3) the texture-space coordinates are used to obtain values from a texture by *texture sampling*, and (4) the values are potentially transformed by a *value transform function* (e.g., for normalization). For texture sampling, *mipmapping* is implemented as a hardware-accelerated approach for texture anti-aliasing, and is a popular LoD technique to reduce details with the linear perspective (Williams 1983). Thereby, "a mipmap is formed by taking the original image (level 0), at the base of the pyramid, and averaging each 2×2 area" (Akenine-Möller & Haines 2008) into a value on the next level up. The mipmap is then trilinearly filtered, i.e., using (1) bilinear filtering, where the four texels nearest to a texture coordinate are weighted based on their distances in texture space, and (2) a linear interpolation between bilinearly filtered mipmap levels (Figure II.7).

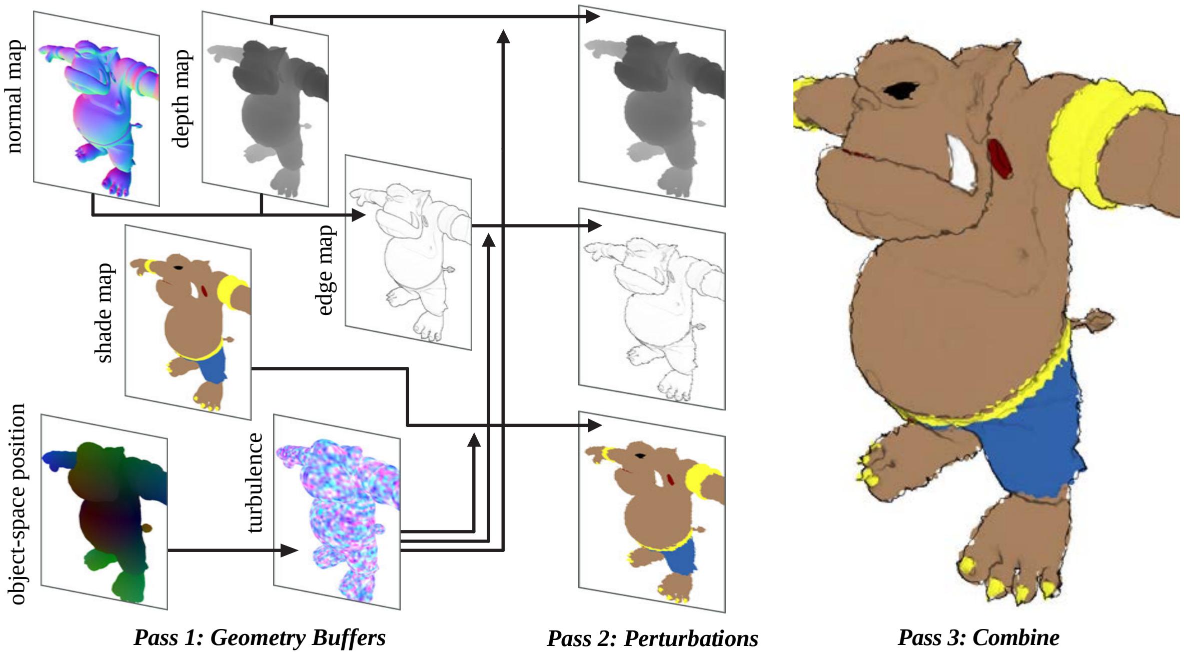
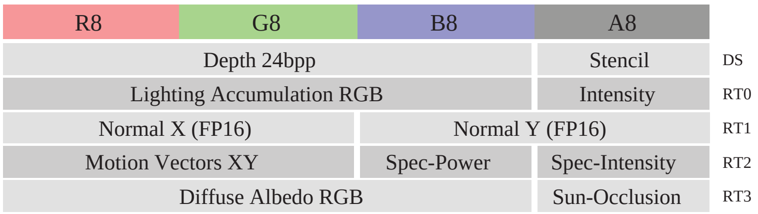


Figure II.8.: Deferred shading using geometry buffers and multi-pass rendering to synthesize sketchy drawings. The first geometry pass renders geometry-related information into textures—edge enhancement is typically performed in a sub-pass, the second pass combines the information to obtain perturbed texture maps, and the third pass combines the perturbed information to obtain the final output. Image courtesy by Nienhaus & Döllner (2004b).

Deferred Shading using Geometry Buffers

Forward rendering (or forward shading) describes the traditional method in which 3D scene entities are processed *once* by the stages defined by the (programmable) rendering pipeline (Figure II.4). Here, all geometries, material and light properties need to be processed or computed in a single pass, e.g., per-pixel lighting needs to repeated for as many light sources as needed—a process that often does not enable real-time performance. By contrast, *deferred shading* postpones the actual shading until a second (or subsequent) pass to separate its (complex) computations from the geometry stage (Shishkovtsov 2005), e.g., to use depth information for real-time shadow mapping (Eisemann *et al.* 2011) that performs independently of a 3D scene's geometric complexity.

The method of *geometry buffers* pioneered by Saito & T. Takahashi (1990) is central to deferred shading: in a first *geometry pass*, geometry-related information for each fragment (pixel) is synthesized in textures (the *G-buffer*) using multiple render targets without writing into the main framebuffer (*offscreen rendering*). The *G-buffer* information may then used in a subsequent pass to perform shading and lighting. For instance, normal, depth, color, and position information can be



R8	G8	B8	A8	
Depth 24bpp			Stencil	DS
Lighting Accumulation RGB			Intensity	RT0
Normal X (FP16)		Normal Y (FP16)		RT1
Motion Vectors XY		Spec-Power	Spec-Intensity	RT2
Diffuse Albedo RGB			Sun-Occlusion	RT3

Figure II.9.: Exemplary packed data structure used for a geometry buffer, comprising four 32bit RGBA buffers and a single depth buffer[3].

[3] The Rendering Technology of Killzone 2 — Michal Valient (Last followed on November 20, 2016) — http://www.guerrilla-games.com/presentations/GDC09_Valient_Rendering_Technology_Of_Killzone_2.pdf

■ *Image processing refers to using computer algorithms for signal processing of digital images, i.e., to analyze, compress, or segment them. The output may be an digital image or a set of parameters with characteristics related to the input image.*

synthesized in a *G-buffer*, and used to perform image-space algorithms such as edge enhancements to obtain sketchy drawings (Figure II.8).

Today, most graphics engines include deferred shading to implement advanced rendering techniques while maintaining real-time performance such as massive lighting, screen-space ambient occlusion, atmospheric scattering, and depth of field (Akenine-Möller & Haines 2008). In this respect, deferred shading, however, has some drawbacks: a *G-buffer* requires additional memory and bandwidth, hardware anti-aliasing can only be used for the first (geometry) pass, and order-independent transparency is more difficult to handle. Some of these issues can be alleviated using packed data structures to reduce the memory consumption of *G-buffers* (Figure II.9), post-process image processing for anti-aliasing and accumulation buffers to implement transparency effects.

General-purpose Computations

The highly parallel and multithreaded architecture of GPUs may be utilized to address problems that are subject to data-parallel computations such as *image processing*. Multiple platforms and programming models have been introduced for general-purpose computation on a GPU (GPGPU) such as OpenGL compute shaders, OpenCL, and CUDA as Nvidia's parallel computing architecture (Figure II.10).

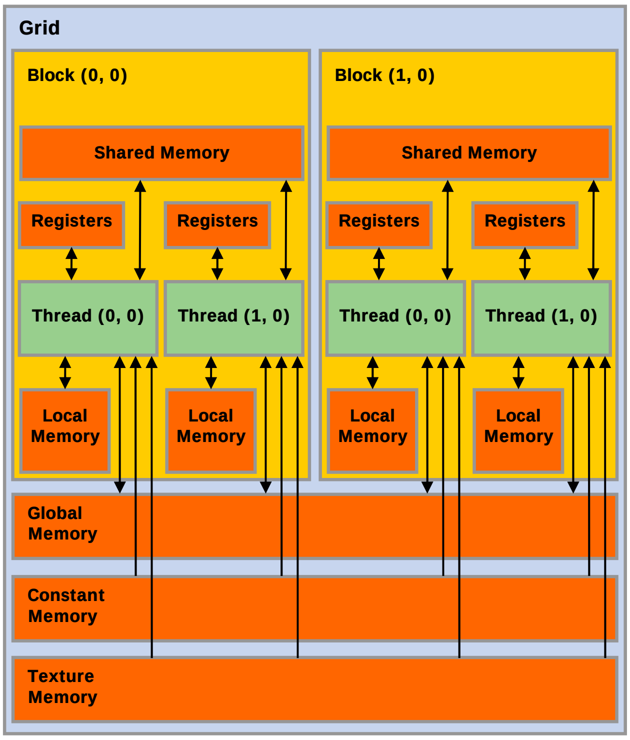


Figure II.10.: Thread hierarchy and data movement for the programming model of CUDA. Image source: CUDA C Programming Guide[4].

CUDA is used in this thesis to pre-process image data and post-process rendered images via image filtering. For post-processing render buffers and textures, CUDA introduces interoperability functions to directly access memory areas that are bound to an OpenGL context. Contrary to image processing using fragment-based shading, CUDA provides several advantages such as access to shared memory (high speed on-chip cache), arbitrary scatter writes, and mechanisms for thread synchronization.

CUDA defines *kernels* as C functions that are executed by parallel processing threads, each of which can be assigned to a single pixel of an image. The threads are organized in *blocks* that reside on specific processor cores, share (limited) memory resources, and can be synchronized in their execution to coordinate memory accesses. Thread blocks are executed independently so that they can be scheduled in any order across a number of processor cores and to scale the processed amount of data with the GPU resources. Thereby, "each thread has private local memory, each thread block has shared memory visible to all threads of the block and with the same lifetime as the block, and all threads have access to the same global memory"[4].

[4] CUDA C Programming Guide
— http://docs.nvidia.com/cuda/cuda-c-programming-guide (Last followed on November 20, 2016)

III

Related Work

The work presented in this chapter is partly based on

Semmo, A., Trapp, M., Jobst, M. & Döllner, J. "Cartography-Oriented Design of 3D Geospatial Information Visualization - Overview and Techniques". *The Cartographic Journal* **52,** 95–106 (2015b)

This thesis touches on research areas of computer graphics, cartography, and computational aesthetics. Accordingly, this chapter reviews the main contributions in NPR and illustrative visualization that share the following basic theme within these research areas: *abstraction as a key concept to support visual communication*. A review of more specialized topics appear in the later chapters where appropriate.

The remainder of this chapter is structured as follows: Section 1 gives an overview of NPR techniques that are dedicated to image-based and geometry-related abstraction. Section 2 outlines the concept of focus+context visualization and summarizes related techniques that provide LoA variations. Section 3 focuses on techniques that are related to illustrative 3D geospatial information visualization and cartographic design. Finally, Section 4 concludes the reviewed works with respect to the contributions of this thesis.

1. Non-Photorealistic Rendering

NPR techniques are diverse, and comprise—besides abstraction as primary concern—a plethora of algorithms that mimic the appeal of artistic media such as pen-and-ink (Winkenbach & Salesin 1994; Salisbury *et al.* 1997) and watercolor drawings (Curtis *et al.* 1997; Bousseau *et al.* 2006; Luft & Deussen 2006). For a comprehensive overview on artistic rendering, the interested reader is referred to the books by B. Gooch & A. Gooch (2001) and Strothotte & Schlechtweg (2002), and the state-of-the-art report by Kyprianidis *et al.* (2013). With respect to painterly rendering, an overview of relevant techniques is given in Chapter VII.

NPR techniques can be categorized according to the processed image or virtual scene contents and data domain:

- ***Highlighting of silhouettes and outlines vs. abstraction of contents.*** Feature silhouettes and outlines are primary design elements in line drawings and sketches, and play an important role for the perception and recognition of (3D) shape (Marr & Hildreth 1980; Todd 2004). Their explicit rendering or highlighting is distinguished from methods that deal with the abstraction of image features or 3D scene objects, e.g., that adjust the geometry, color palette, shading, or lighting (McGuire *et al.* 2010).

- ***Image-based processing vs. geometry-based processing.*** NPR algorithms may process information in image space or geometry-related information in object space. Geometry-related information may also be synthesized in a *G-buffer* to enable an accelerated processing in image space. NPR algorithms that process temporal data, e.g., video (Kyprianidis *et al.* 2013), are also widely explored, but are not a primary focus of this thesis.

The following sections review previous works that fall in these categories and are related to the techniques proposed in this thesis.

1. Abstraction and Stylization of Images

■ *Edge-aware image filters commonly smooth low-contrast regions while preserving high-contrast edges, and thus serve as edge-preserving smoothing or enhancing operators. Here, edges refer to significant intensity changes within a spatial neighborhood of an image.*

Image-based NPR techniques can be used to process scene textures and information stored in geometry buffers, or post-process rendering artifacts. Here, *edge-aware image processing* has emerged as a fundamental tool in NPR and computational photography for abstraction, artistic stylization, tone mapping, noise reduction, and colorization. Numerous filtering techniques have been proposed over the past years to approach these applications in an automated way, most of them deriving local image structures for feature-aware processing (Figure III.1).

Image Abstraction Filters

Typical approaches that operate in the spatial domain for abstraction use a kind of anisotropic diffusion (Perona & Malik 1990) and are designed for parallel execution. Thereby, diffusion tensors, e.g., based on an eigenanalysis of the structure tensor (Brox *et al.* 2006a; Weickert 1998), are used to obtain non-linear and space-variant transformations of an input image, with the goal to remove detail in high-contrast image regions without filtering across discontinuities to preserve (salient) edges.

Bilateral Filter. A popular choice to approximate an anisotropic diffusion in real-time is the bilateral filter, which works by weight averaging pixel colors in a local neighborhood based on their distances in space and range (Tomasi & Manduchi 1998). To this end, the bilateral filter weights pixels with a high difference in intensity less than a Gaussian filter to preserve image structures at a better scale. Several applications of the bilateral filter exist, for instance Winnemöller *et al.* (2006) used the bilateral filter—applied in a multi-stage process—for real-time artistic rendering with a cartoon look, J. Chen *et al.* (2007) and Fattal *et al.* (2007) used the filter to obtain multiscale image decompositions, and Cho *et al.* (2014) enhanced the range filter kernel to include a local patch-based analysis of texture features to filter texture information. Kyprianidis & Döllner (2008) and H. Kang *et al.* (2009) enhanced the bilateral filter by flow-based implementations adapted to the local image structure, which can be based on an eigenanalysis of the smoothed structure tensor (Brox *et al.* 2006a), to provide smooth outputs at curved boundaries and a more painterly look. Efficient implementations of the bilateral filter include a separable filter kernel (Pham & van Vliet 2005) and the bilateral grid (J. Chen *et al.* 2007) that employs uniform resampling to enable hardware-accelerated implementations on a GPU. Encountering the trend of accelerating the bilateral filter, however, the *guided filter* proposed by He *et al.* (2013) provides an alternative real-time approach with similar characteristics

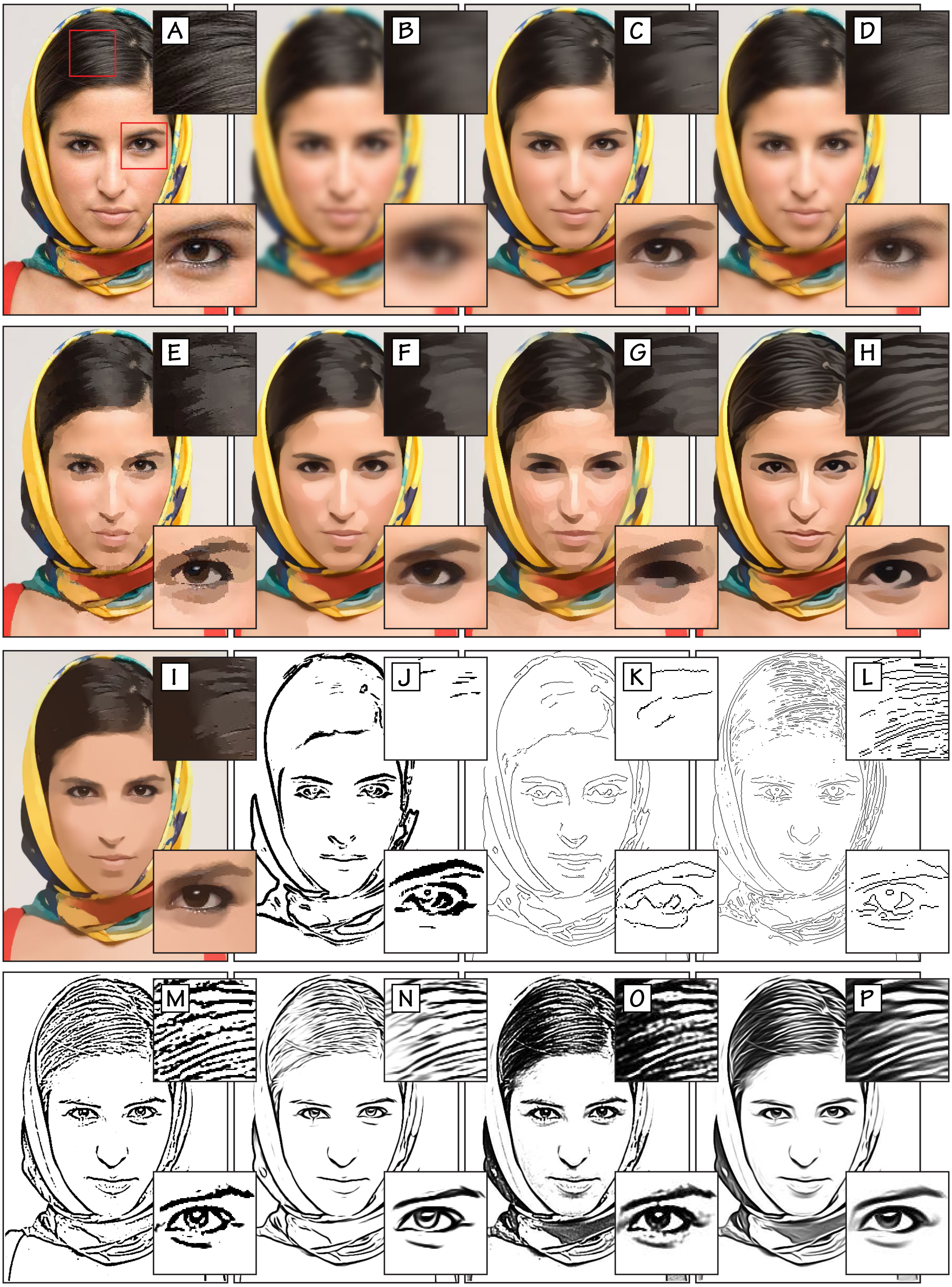


Figure III.1.: Comparison of selected image filters used for abstraction and stylization. (A) input, (B) Gaussian filter, (C) flow-based bilateral filter (Kyprianidis & Döllner 2008), (D) guided image filter (He *et al.* 2013), (E) mean-shift (Comaniciu *et al.* 2002), (F) multi-scale anisotropic Kuwahara filter (Kyprianidis 2011), (G) shape-simplifying abstraction (H. Kang & S. Lee 2008), (H) coherence-enhancing filter (Kyprianidis & H. Kang 2011), (I) L_0 gradient minimization (L. Xu *et al.* 2011), (J) Sobel filter, (K) Canny edge detector (Canny 1986), (L) LoG/DoG zero-crossings, (M) thresholded DoG, (N) thresholded flow-based DoG (Kyprianidis & Döllner 2008), (O) XDoG (Winnemöller *et al.* 2012), (P) flow-based XDoG (Winnemöller *et al.* 2012). Filter parameters are hand-tuned to achieve optimal quality. The images shown in (J-M) and (O-P) are obtained from Winnemöller *et al.* (2012). Input image © Maryse Casol.

as the bilateral filter, but reducing unwanted gradient *reversal artifacts*, i.e., the detection of false edges. Here, the filtered output is computed by considering the content of a guidance image, e.g., using the input image itself.

Mean-Shift. A mean-shift is a popular approach for discontinuity-preserving smoothing (Comaniciu *et al.* 2002) and saliency-guided image abstraction (DeCarlo & Santella 2002). It provides a non-parametric filter that estimates probability density functions by iteratively shifting color values to averaged color values of a local neighborhood. However, the approach is more suited to applications found in image segmentation, and would require elaborate post-processing to produce smooth outputs at curved boundaries.

Kuwahara Filter. Another popular approach that even works accurately for high-contrast images—contrary to the bilateral filter—and provides smoothed outputs at curved boundaries, is the Kuwahara filter (Kuwahara *et al.* 1976) and its generalized (Papari *et al.* 2007), anisotropic (Kyprianidis & H. Kang 2011) and multi-scale (Kyprianidis & H. Kang 2011) variants. The kernel of the Kuwahara filter is divided into overlapping subregions, where the response is defined as the mean of the subregion with minimal variance. To this end, the filter is able to maintain a uniform LoA due to local area flattening.

Shock, Morphological, and Geodesic Filters. Additional categories include shock filtering, e.g., in conjunction with constrained mean curvature flow (H. Kang *et al.* 2007) and diffusion tensors for coherence-enhancing abstraction (Kyprianidis 2011), morphological filtering based on *dilation* and *erosion*, and geodesic filtering using distance transforms to obtain cartoon-style abstractions (Criminisi *et al.* 2010; Mould 2012). These filters typically weight colors across feature boundaries, e.g., for shape-simplifying abstraction (H. Kang *et al.* 2007).

Filters using Global Optimizations. Many filters focus on image decompositions by solving optimization problems to separate detail from base information, e.g., based on weighted least squares (Farbman *et al.* 2008), local extrema for edge-preserving smoothing (Subr *et al.* 2009), locally weighted histograms (Kass & Solomon 2010), L_0 gradient minimizations (L. Xu *et al.* 2011), or region covariances (Karacan *et al.* 2013). Applications of these filters include HDR tone mapping and compression, detail exaggeration, JPEG artifact removal, edge adjustment, and colorization. Because of their global optimization scheme, however, these filters are typically not suited for real-time application.

Edge Enhancement Filters

Winnemöller *et al.* (2012) distinguish between gradient-based edge detection that thresholds the gradient magnitude of an image and Laplacian-based edge detection that identifies zero-crossings in the second derivative.

First Derivative Filters. First approaches that operate in the gradient domain focused on identifying image gradients with high magnitudes by using convolution filters such as the Prewitt and Sobel filter (Pratt 2001). Image regions with a gradient magnitude above a given threshold are then identified as edges and colorized for visualization. However, these approaches—although very efficient to implement—produce results that are sensitive to noise. The *Canny edge detector* (Canny 1986) as a multi-stage algorithm provides several enhancements by combining smoothing and differentiation operators for noise reduction. However, it may produce edge segments that become disconnected and generally does not provide real-time performance.

Second Derivative Filters. A fast approach that performs in real-time and is less sensitive to noise is to approximate the Laplacian of Gaussian (LoG) using difference of Gaussians (DoG) for *Marr-Hildreth* edge detection (Marr & Hildreth 1980). The DoG is described by the difference of two Gaussian-blurred images with different standard deviation, which is thresholded to provide smooth edges of delicate structures, e.g., used for the stylization of human faces (B. Gooch *et al.* 2004) and for cartoon rendering (Winnemöller *et al.* 2012). Enhanced versions are provided by H. Kang *et al.* (2007), Kyprianidis & Döllner (2008), and Winnemöller *et al.* (2012), who propose separable flow-based implementations of the DoG that are adapted to the local orientation of an input image to create smooth coherent outputs for line and curve segments.

The application of image filters in a generalized LoA texturing approach is provided in Chapter IV. Here, the separation between content abstraction and edge enhancement is maintained to conform with cartographic design aspects (Chapter II) such as lines as graphical elements to provide figure-ground relationships.

Image filters are also prominently used in the domain of (real-time) artistic rendering. For instance, bilateral filtering is used with color quantization to obtain cartoon-like effects (Winnemöller *et al.* 2006), morphological filtering using *closing* followed by *opening* operators to simulate characteristics of watercolor paintings (Bousseau *et al.* 2006), and the XDoG filter to obtain two tone black-and-white images that mimic woodcuts (Winnemöller *et al.* 2012). Additional categories for artistic rendering include example-based and stroke-based techniques such as halftoning and stippling (Deussen & Isenberg 2013). Here, a distinguished review of related work for aligning digitized hand-drawn strokes to the principal curvature directions of image features is given in Chapter VII.

2. Abstraction and Stylization of 3D Models

McGuire *et al.* (2010) classify the elements of visual style for 3D models into six categories: lighting, geometry, color palette, shading, object motion, and camera placement. This thesis draws upon NPR techniques that relate to geometry information, or adjust the color palette and shading for visualization purposes.

Contents

Related work that deal with 3D model contents use texture mapping and shading as core mechanisms to adjust the visual style.

Shading. Photorealistic shading does not necessarily achieve a comprehensive visualization when detail information (e.g., complex lighting) is involved (Hughes *et al.* 2013). A. Gooch *et al.* (1998) propose a non-photorealistic lighting model that simulates cool-to-warm shades found in technical illustrations, which convey visual prominence of edge lines and highlights (Figure III.2A). Toon shading (or cel shading) is an alternative approach for abstract visualization, where 3D object surfaces appear to be flat using sharp edges between colored regions instead of soft transitions, i.e., to simulate illustrations with reduced color

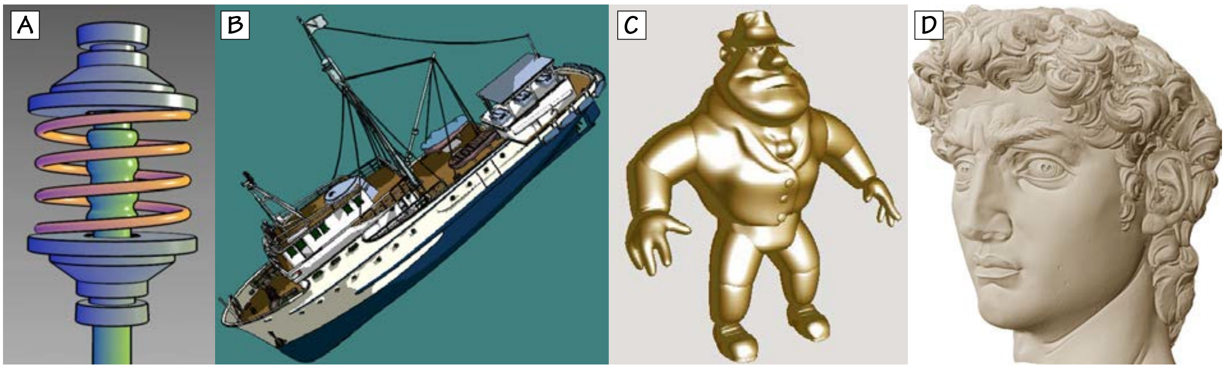


Figure III.2.: Shading models using principles of abstraction and stylization. (A) A. Gooch *et al.* (1998), (B) Lake *et al.* (2000), (C) Barla *et al.* (2006), (D) Rusinkiewicz *et al.* (2006). Image courtesy by the respective authors.

palettes. Decaudin (1996) introduced toon shading using the *G-buffer* concept of Saito & T. Takahashi (1990) to threshold the diffuse reflection according to surface normals. Lake *et al.* 2000 enhanced the method by thresholding the diffuse reflection on a per-vertex basis and using color binning to adjust the number of regions with a uniform color (Figure III.2B). Additional methods were presented by Anjyo *et al.* (2006) to include stylized highlights, Barla *et al.* (2006) to support view-dependent effects such as LoAs and depth-of-field, and D. Kang *et al.* (2009) by taking visual saliency into account to depict morphological features more precisely. Rusinkiewicz *et al.* (2006) propose non-photorealistic shading based on dynamically adjusted light positions for different areas of an object's surface (Figure III.2D). Their approach preserves details independent of the surface orientation and—by operating at multiple scales—is able to convey detail information at multiple frequencies.

Texture Mapping. Techniques that rely on texture mapping typically align tileable images unidirectionally or to the principal curvature directions of 3D object surfaces to improve shape recognition (Hertzmann & Zorin 2000; Girshick *et al.* 2000; Wei *et al.* 2009). Relevant object-space methods that are based on *art maps* reduce perspective distortion and scale variations via mipmapping (A. W. Klein *et al.* 2000). Praun *et al.* (2001) and Webb *et al.* (2002) extended this approach towards *tonal art maps* where patterns of mipmap levels are included into the subsequent levels (Figure III.4) to improve temporal coherence. The art maps approach may use a lapped texture parameterization (Praun *et al.* 2000) for feature-aligned shading and seamlessly integrates into the real-time rendering pipeline (Freudenberg 2001). Bénard *et al.* (2009) extended the approach by

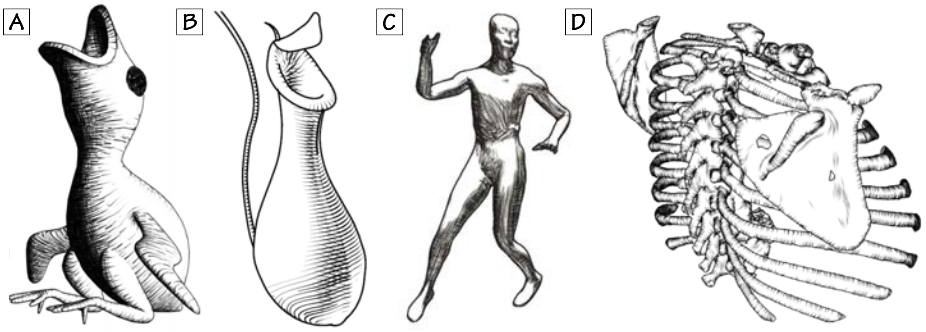
Figure III.3: Hatching aligned to principal curvature directions and lighting information. Image courtesy:

(A) Praun *et al.* (2001)

(B) Zander *et al.* (2004)

(C) Y. Kim *et al.* (2008)

(D) Lawonn *et al.* (2013)



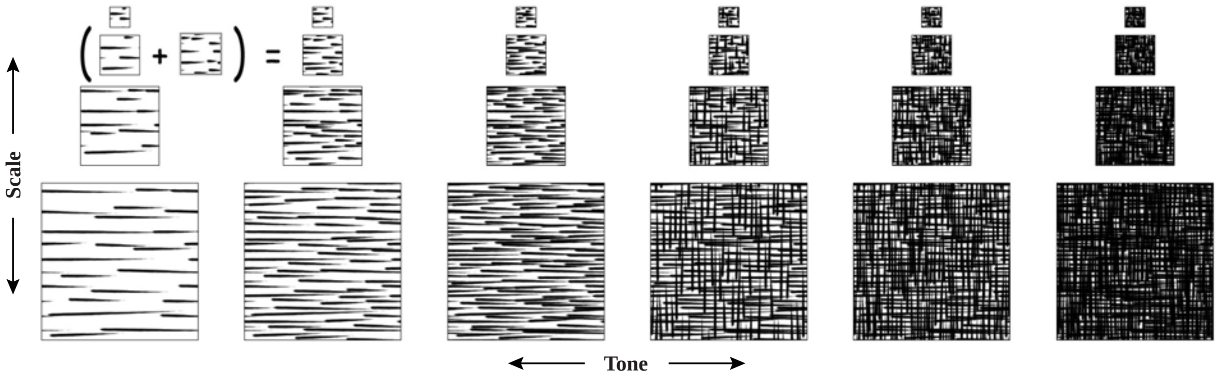


Figure III.4.: A tonal art map. The strokes in one texture level appear in all the subsequent texture levels of the pyramid (above and to the right). Image courtesy by Praun *et al.* (2001).

providing infinite zoom mechanisms to maintain a quasi-constant size of texture elements for arbitrary view distances. Art maps are particularly used for hatching and illustrative visualization (Figure III.3), i.e., "to emphasize regions with high curvature by drawing lines along principal curvature directions" (Lawonn *et al.* 2013). Additional categories for feature-aligned texturing, or 3D streamline parameterization (Zander *et al.* 2004), include normal-ray differential geometry (Y. Kim *et al.* 2008), diffusion techniques (K. Xu *et al.* 2009), and learning-based approaches (Kalogerakis *et al.* 2012; Gerl & Isenberg 2013). However, these methods either require significant (pre-)processing time or do not provide both spatial and temporal coherence.

Line Drawings

The enhancement of silhouettes and outlines of polygonal 3D models (Figure III.5) is a well-researched topic in NPR (Hertzmann 1999; Isenberg *et al.* 2003; Rusinkiewicz *et al.* 2008) to illustrate shape, aid comprehension, and support mental modeling. One approach for the edge enhancement of virtual 3D scenes is to perform image filtering on the rendered color images in a post-process stage. However, this approach generally lacks frame-to-frame coherence due to aliasing

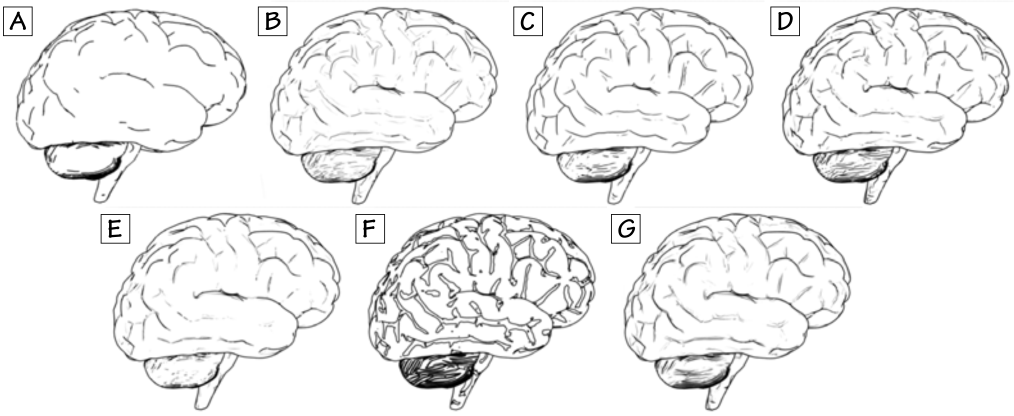


Figure III.5.: Overview of line types extracted in edge enhancement techniques. Image courtesy by Kai Lawonn. (A) contours (Isenberg *et al.* 2003), (B) ridges and valleys (Interrante *et al.* 1995), (C) suggestive contours (DeCarlo *et al.* 2003), (D) apparent ridges (Judd *et al.* 2007), (E) photic extremum lines (Xie *et al.* 2007), (F) demarcating curves (Kolomenkin *et al.* 2008), (G) Laplacian lines (L. Zhang *et al.* 2011).

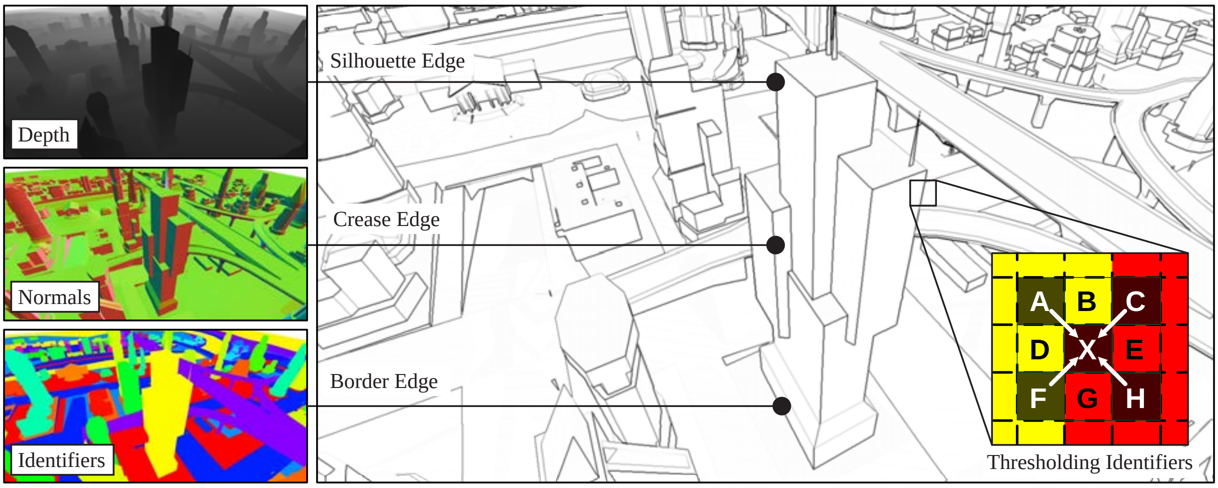


Figure III.7.: Edge enhancement in image space based on the approach of Nienhaus & Döllner (2003). Object identifiers, normal and depth information are thresholded to extract and combine silhouettes, crease and border edges.

and occlusion (Bénard *et al.* 2011), and only provides limited stylization capabilities. Isenberg *et al.* (2003) distinguish between image-space and object-space methods using geometry-related information that address these issues.

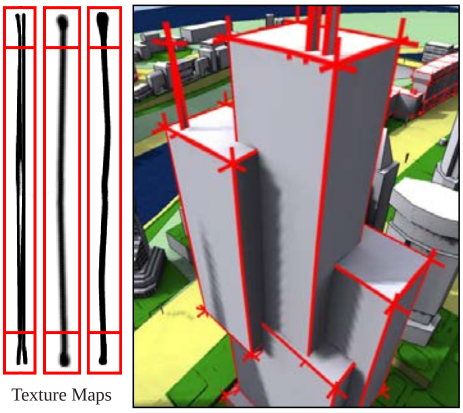


Figure III.6.: Object-space edge enhancement using impostors (Döllner & Walther 2003).

Image-space Techniques. A popular image-space approach is to use *G-buffer* information to detect discontinuities in depth to extract contour lines, and discontinuities in surface orientation to extract crease lines (Saito & T. Takahashi 1990; Raskar & Cohen 1999). The enhanced method of Nienhaus & Döllner (2003) additionally detects discontinuities at object boundaries. Here, intensity values are computed at each pixel of the *G-buffer* including depth values, normals, and (color-encoded) object identifiers by weighting the information of the 8-connected neighborhood (Figure III.7). Additional methods are provided by Y. Lee *et al.* (2007) and Vergne *et al.* (2011), where 2D local fitting is combined with spatially-varying convolutions to provide additional control over the stylization of the extracted lines (e.g., thickness), and Bénard *et al.* (2012) who approximate lines using *active contours* that advect with the motion of the scene for coherent stylization. Although these methods perform efficiently, however, they generally do not provide topological accuracy and sophisticated artistic control.

Object-space Techniques. Object-space approaches, by contrast, model lines as explicit geometry based on the connectivity information of surfaces meshes. To this end, they typically provide frame-to-frame coherence and additional capabilities for stylization such as via texturing of line segments. In addition to contour lines providing pre-dominant shape cues (Isenberg *et al.* 2003), different types of lines have been defined dependent on the used mathematical operations for extraction (Figure III.5): ridges and valleys at local extrema of the surface curvature (Interrante *et al.* 1995), suggestive contours based on zero crossings of the radial curvature to extend occluding contours (DeCarlo *et al.* 2003), apparent ridges using view-dependent curvature measures (Judd *et al.* 2007), photic extremum lines (Xie *et al.* 2007), demarcating curves as the zero crossing of the normal curvature (Kolomenkin *et al.* 2008), and Laplacian lines (L. Zhang *et al.* 2011). Techniques are required to resolve the visibility of these lines per render pass, e.g., by visibility culling (Isenberg *et al.* 2002) or using a *segment atlas* that contains per-sample visibility information (Cole & Finkelstein 2010). Rendering and stylization of the line segments may then leverage geometry shaders to dynamically synthesize and extrude textured edge geometries (Hermosilla & P. Vázquez 2009). For instance, Döllner & Walther (2003) communicate uncertainty of stylized 3D objects by using alpha-channel encoded strokes as impostors that are anchored at the line segments and aligned to the user's viewing direction (Figure III.6). In particular, the impostor-based approach can be parameterized in terms of color, width, length, displacement, and tilt offset.

In this thesis, the focus is less on developing novel computer graphics techniques for shading and edge enhancement than on their dedicated, interactive application to 3D geospatial data and their specialized entities, respectively. This includes feature-aligned texturing and edge enhancements as typical components of cartographic design to enhance and symbolize feature relationships. Here in particular, the image-space technique of Nienhaus & Döllner (2003) is used as it provides a robust method for complementary (depth) cue enhancement, performs in real-time, and provides reasonable frame-to-frame coherence when used with a multisampled *G-buffer*.

2. Focus+Context Visualization

Focus+context describes the concept to visually distinguish between important or relevant information from closely related information (Furnas 1986). Focus+context visualization conforms with the *visual information seeking mantra*:

> "Overview first, zoom and filter, then details-on-demand"
> — Shneiderman (1996)

by enabling users to interactively change the visual representation of data for points and regions of interest (*details-on-demand*), e.g., using highlighting techniques, while maintaining a context for orientation guidance (*overview*). The perceptional and cognitive qualities of humans to perceive features that correlate in their graphical representation or distance are utilized to enable importance-guided focus-of-attention effects. Prominent strategies are based on distortion-based viewing methods (Leung & Apperley 1994), methods that selectively display data dimensions on demand with filtering as pre-dominant operation (E. A. Bier *et al.* 1993), and cue techniques using graphic style variations for certain image regions or objects (Kosara *et al.* 2002). All strategies share the common goal to solve the problem of over-cluttered visual representations and direct the viewer's focus to image regions that encode important or prioritized information, and thus are of particular interest to this thesis. Because distortion-based methods typically involve layout adjustments and no explicit graphical abstraction or highlighting (Tominski *et al.* 2014), the remainder of this section focuses on strategies found in the adaptive parameterization of NPR techniques. Here, methods applied in image space and methods designed for virtual 3D scenes are distinguished.

1. Level-of-Abstraction Variations for Images

Image-based rendering and processing involves cue techniques that adjust the LoA according to regions of interest or saliency metrics, and lens-based techniques that parameterize the spatial selection of visualized image features.

Cue Techniques

■ *Visual cues are variations in the composition and graphical representation intended to help the viewer perceive information of interest more efficiently. Important visual cues comprise depth, motion, color, and contrast.*

Cue techniques relate to basic principles found in the creation of traditional art works, where artists are trained—beyond different painting styles and techniques—to enhance communication aspects of their emotions and ideas via highlighting and abstraction. This is often achieved by variances in graphic styles and feature compositions—e.g., luminance, color, and texture—to direct the viewer's gaze to salient image regions of high local contrast or high edge density (Mannan *et al.* 1996; Parkhurst & Niebur 2003). Bailey *et al.* (2009) studied this behavior for a subtle gaze directing technique that uses image-space modulations in luminance and color space for the peripheral regions of the field of view. McNamara *et al.* (2008) showed that this technique improves search task performance significantly, however, it does not involve explicit abstraction of image contents to intentionally restrict or reduce the information content. Here, image-based techniques may be parameterized according to input masks to compute and guide the different LoA representations produced by image abstraction, artistic rendering, and depth-of-field effects.

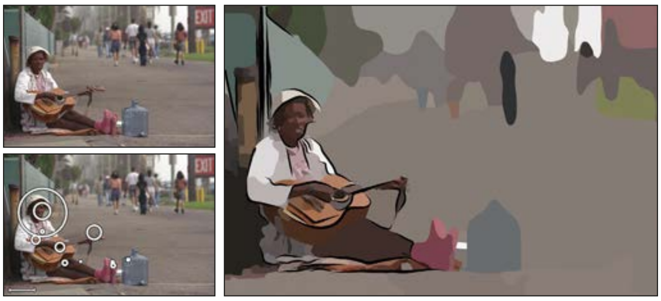


Figure III.8.: Image segmentation using eye fixations to direct the LoA. Image courtesy by DeCarlo & Santella (2002).

Image Abstraction. DeCarlo & Santella (2002) were the first to generate abstract renderings of photographs by image filtering with a locally varied LoA according to the viewer's eye movements and fixations. Thereby, they disassemble an image into hierarchical parts and boundaries, and use an eye tracker to gather the information required to predict "meaningful" image features and locations, and accordingly select the mean-shift-based segmentation scale (Figure III.8). Importance maps derived from the Gaussian scale space (Hertzmann 1998) and image saliency, however, identify regions that grab visual attention in the original image, and thus are less effective to modify the focus of attention. By contrast, Orzan *et al.* (2007) derived importance maps from edge-based hierarchies to intentionally shift the focus of attention (Figure III.9A). User-specified control maps can then be used to indicate the desired amount of abstraction, an approach also used in image filtering to protect image features from over-simplification (H. Kang & S. Lee 2008). Cong *et al.* (2011) used abstraction with mixed realities, where image features outside regions of interest are filtered, otherwise they are depicted photorealistically (Figure III.9C). This approach demonstrates the wide range of possible LoAs that can be used for focus+context visualization. More generalized approaches are provided by Brooks (2007) who locally mix different painting media for portraiture, and J. Chen *et al.* (2011) who integrate and blend multiple rendering styles using importance maps. The latter approach accommodates stylization techniques such as color desaturation, line drawings, and

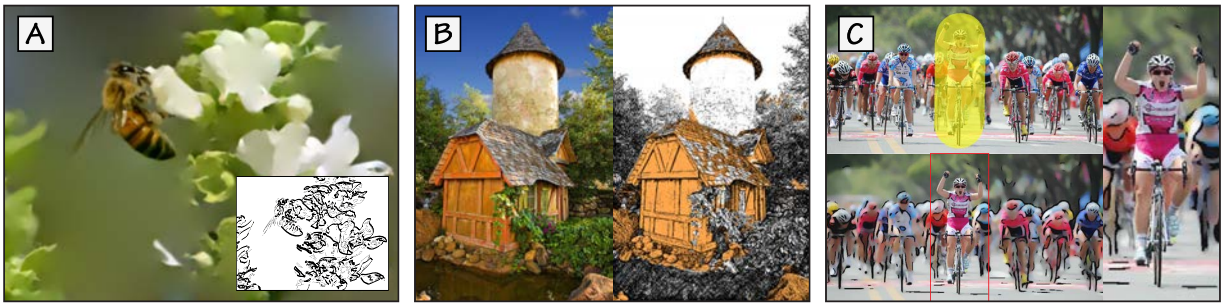


Figure III.9.: Cue-based focus+context techniques by image abstraction. (A) structure-preserving manipulation of a photograph using edge-based hierarchies (here: focus on the bee) (Orzan *et al.* 2007), (B) NPR with spot colors using a salience weighted spread criterion (Rosin & Lai 2013), (C) selective image abstraction using photorealistic graphics in a user-defined image region (Cong *et al.* 2011). Image courtesy by the respective authors.

edge-preserving smoothing. Additional techniques that adjust single graphic variables are possible, for instance the saturation in context regions can be reduced to render images with *spot colors* (Rosin & Lai 2013)—Figure III.9B.

Image-based Artistic Rendering. Adapting the LoA to focus interest to particular image regions plays a major role in image-based artistic rendering (Kyprianidis *et al.* 2013), i.e., to simulate the way artists might paint with different techniques and brushes (Hegde *et al.* 2013). A classical method for stroke-based stylization is to iteratively align brush strokes of varying color, size, and orientation according to the input image (Haeberli 1990; B. Gooch *et al.* 2002; Hays & Essa 2004; Zeng *et al.* 2009; Zhao & Zhu 2010; Lu *et al.* 2010). Here, relaxation-based painting algorithms (Hertzmann 1998) typically preserve detail information by locally minimizing the color differences of the intermediate results to the original image, e.g., using measures of variance (Shiraishi & Yamaguchi 2000). However, these heuristics rather preserve the LoA distribution of the original image than allow to draw emphasis to different image regions. By contrast, Collomosse & Hall (2005) proposed a salience adaptive approach based on differential levels of emphasis. Their brush-based painterly rendering technique is parameterized by a genetic algorithm to draw emphasis to individual regions of interest, while the stylized image coincides with the salience magnitude of the input image (Figure III.10). Here, further choices of emphasis are conceivable, e.g., based on a foreground/background separation of the image contents.

Depth-of-field Effects. Another conventional cue-based approach is to simulate the depth-of-field (DoF) effect known from photography and cinematography, i.e., where lenses and light transports are modeled to intentionally blur image

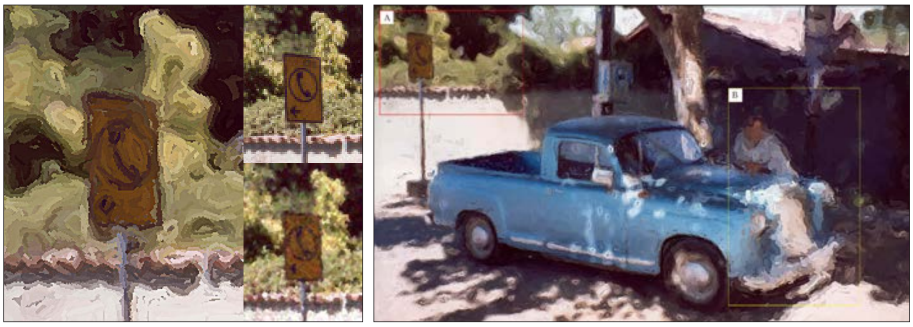


Figure III.10: Global optimization using a genetic algorithm—guided by a salience field—that produces differences in salient (e.g., the sign) and non-salient features. Image courtesy by Collomosse & Hall (2005).

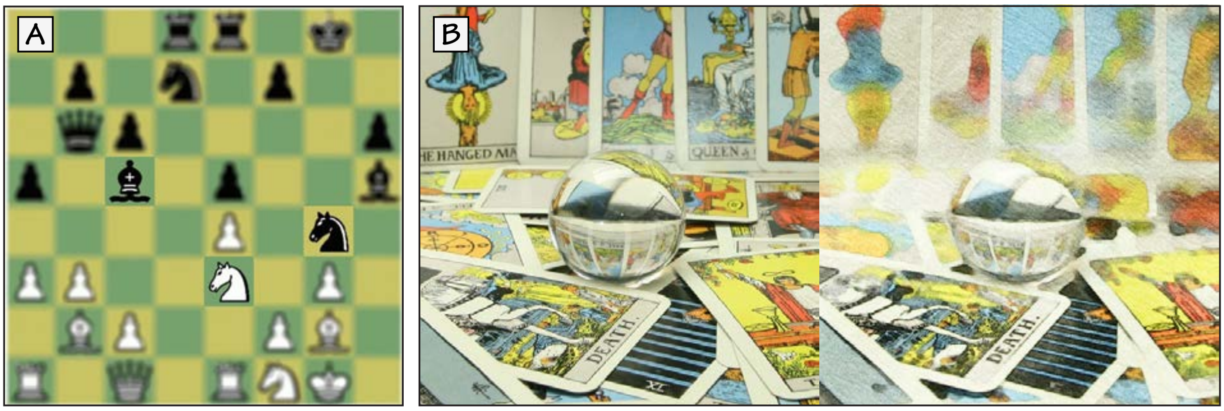


Figure III.11.: DoF effects for images. **(A)** semantic DoF based on box filtering (Kosara *et al.* 2001), **(B)** stylized DoF using lightfield data to guide a watercolor rendering (Bousseau 2009). Image courtesy by the respective authors.

features that are out-of-focus (H.-C. Lee 1990). The simulation of the DoF effect typically demands for depth information to create accurate results. Nonetheless, it has been applied in image abstraction to create lens-based effects where less important image regions are masked to guide human eye fixations. For instance, Kosara *et al.* (2001) adopted the DoF effect for visualization purposes, where image regions are blurred by box filtering, with the kernels adapted to importance maps to compose a semantic DoF (SDoF) effect (Figure III.11A). Bousseau (2009) enhanced the effect using a combination with filter parameterizations to obtain a stylized DoF effect. Here, light field data is used to replace traditional aperture effects (Figure III.11B). Although providing stylized outputs of varying LoA, however, the approach gives no local control for modifying the emphasis apart from using the light field information.

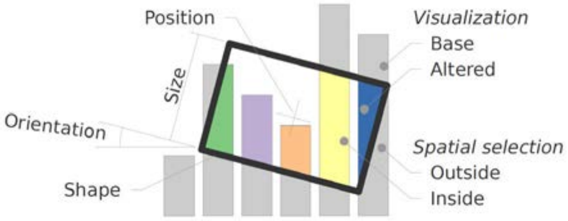


Figure III.12.: Schematic depiction of an interactive lens. Image courtesy by Tominski *et al.* (2014).

Lens-based Techniques

A lens refers to an "interactively parameterizable spatial selection according to which a base visualization is altered" (Tominski *et al.* 2014). Interactive lenses have become established means to facilitate the exploration of large data sets without requiring users to zoom into representations, and are quite versatile in their parameterization (e.g., shape, position, orientation)—Figure III.12. Besides their key properties to determine the spatial selection, Tominski *et al.* (2014) distinguish in their state-of-the-art report between general or data-specific applications of lenses and the task they are designed for, including *selection*, *exploration*, *reconfiguration*, *encoding*, *abstraction and elaboration*, *filtering*, and *connection* as basic operations. This thesis refers to lenses that adjust the graphical display to show more or less detail or change the visual mapping.

Tool glasses and magic lenses have been introduced by E. A. Bier *et al.* (1993). In their seminal work, they propose widgets as interface tools that can be interactively positioned and parameterized with visual filters to "modify the presentation of application objects to reveal hidden information, to enhance data of interest, or to suppress distracting information" (E. A. Bier *et al.* 1993). The approach has later been explored in design spaces of general 2D illustrations to

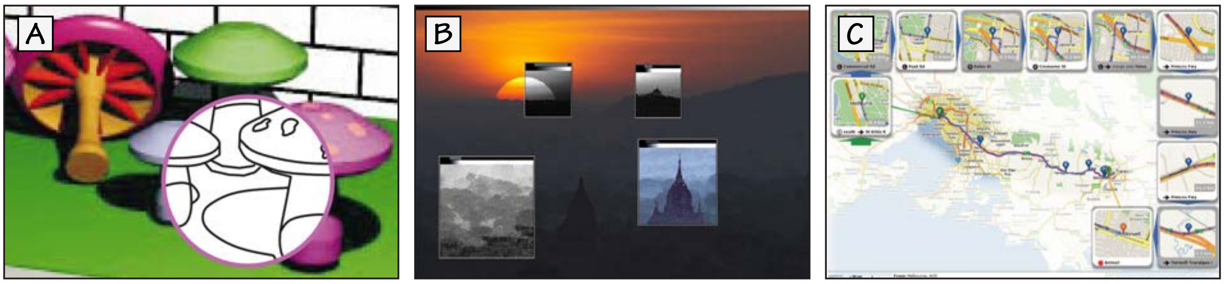


Figure III.13.: Magic-lens metaphors for image-based rendering. (A) *SelectLayer* lens to reveal a line drawing for a color image (E. Bier *et al.* 1997), (B) color lens technique with different parameters applied to a photograph (Elmqvist *et al.* 2011), (C) route visualization using detail lenses (Karnick *et al.* 2010). Image courtesy by the respective authors.

locally adjust graphical properties (E. Bier *et al.* 1997)—Figure III.13A. Pietriga & Appert (2008) proposed a general framework for viewport lenses that combines graphical representations of varying LoA using image blending and compositing (Porter & Duff 1984). Their approach is decoupled from explicit rendering techniques and seamlessly integrates into existing visualization systems.

Techniques that use magic lenses for visualization purposes primarily use sampling strategies to reduce visual clutter. For instance, G. Ellis *et al.* (2005) used filtering criteria to reduce the number of objects significantly and help users perceive potentially interesting data trends more easily. Their lens-based approach sub-samples single entities within a region of interest according to a user-defined sampling rate. Elmqvist *et al.* (2011) also used sampling lenses for exploration purposes, but followed an approach that dynamically optimizes color scales (Figure III.13B). Thereby, their technique "inspects the lens contents in data space, optimizes the initial color scale, and then renders the contents of the lens to the screen using the modified color scale" (Elmqvist *et al.* 2011).

The magic-lens metaphor has also been specialized in domain-specific applications, in particular in the geospatial domain to interactively select the degree of generalization. Karnick *et al.* (2010) proposed detail lenses for route map displays to provide overview and detail information in a single view. At this, the regions around multiple foci—defined by points-of-interest along a route—are buffered and displayed with high detail around the route, whereas the context regions of a map are intentionally displayed with lower contrast (Figure III.13C). Krüger *et al.* (2013) provide a technique that is similar to those using sampling strategies, but is designed for exploring trajectory data. Their approach enables analysts to reveal "movements that occur in a given time range, traverse a certain region, or end at a given area of interest" (Krüger *et al.* 2013).

2. Level-of-Abstraction Variations for Virtual 3D Scenes

In addition to color information, techniques for virtual 3D scenes typically have access to geometry-related information for LoA parameterization.

Cue Techniques

Cole *et al.* (2006) extended the idea of DeCarlo & Santella (2002) to virtual 3D scenes by emphasizing entities within area of interests "through local variations in shading effects such as color saturation and contrast as well as line qualities

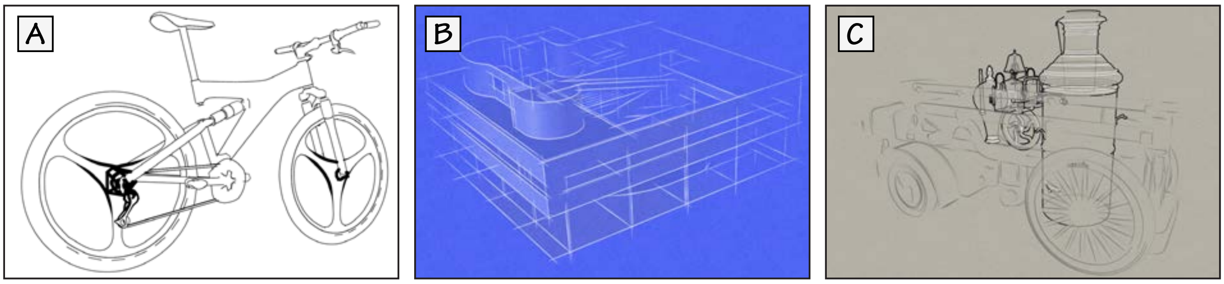


Figure III.14.: Adaptive stylization of line drawings. (A) 3D illustrative effect with varying line thickness (Isenberg *et al.* 2000), (B) use of line overshoots for blueprint stylization (Cole *et al.* 2006), (C) use of occlusion information to emphasize scene parts (Grabli *et al.* 2004). Image courtesy by the respective authors.

such as texture and density". They distinguished between a segmentation of the virtual 3D scene, e.g., for semantics-based stylization, and three additional focus models: 2D focal points to guide the stylization in image space, 3D focal points where the stylization falls off radially in 3D space, and focal planes inspired by real-world camera lenses (Figure III.15). Using these models, color effects such as desaturation, fade, and blur with adjusted line qualities such as texture, width, opacity, density, and overshoot are selected and combined via transfer functions.

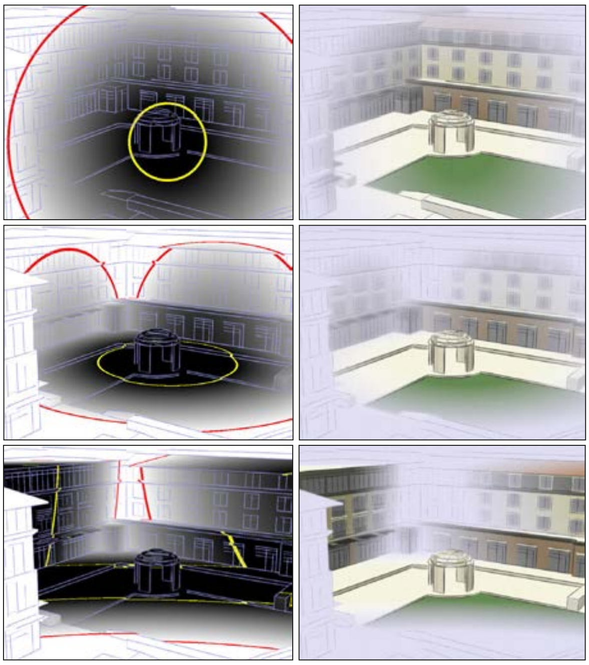


Figure III.15.: Focal models for directing gaze in virtual 3D scenes (from top to bottom): 2D focal point, 3D focal point, focal plane. Image courtesy by Cole *et al.* (2006).

Additional control in adjusting image-based rendering techniques is given by geometry-related information such as scene depth or object information for segmentation. For instance, Redmond & Dingliana (2007) used a Kuwahara filter (Kuwahara *et al.* 1976) and DoG filter with *Marr-Hildreth* edge detection (Marr & Hildreth 1980) to post-process rendered color images of a virtual 3D scene in image space, but using object identifiers and distance information to adjust the rendering style. In a follow-up work, they provide hybrid rendering styles, where image filtering is combined with cel shading and enhanced silhouette edges (Redmond & Dingliana 2009a). Thereby, an adaptive LoA is computed based on the distance between objects and regions of interest to direct and smoothly interpolate color contrasts and shader parameters (e.g., cel shades).

Besides tonal abstractions, previous work is found in adaptive lines stylization. Isenberg *et al.* (2000) proposed a general concept for rendering stylized objects with varying line thicknesses, patterns, and color attributes (Figure III.14A). They introduce *illustrative effects* that are aligned to spatial, possibly user-defined locations. A similar approach is followed by Cole *et al.* (2006), but with more control in rendering parameters and focus styles (Figure III.14B). Grabli *et al.* (2004), Shesh & B. Chen (2008) and Grabli *et al.* (2010) parameterized object-space lines with density control and adaptive simplifications according to scene or view properties to avoid visual clutter (Figure III.14C). At this, Barla *et al.* (2005) presented a generic approach to adjust the simplification of line drawings by a clustering strategy. Here, the main idea is to group "lines that capture the morphological structure of

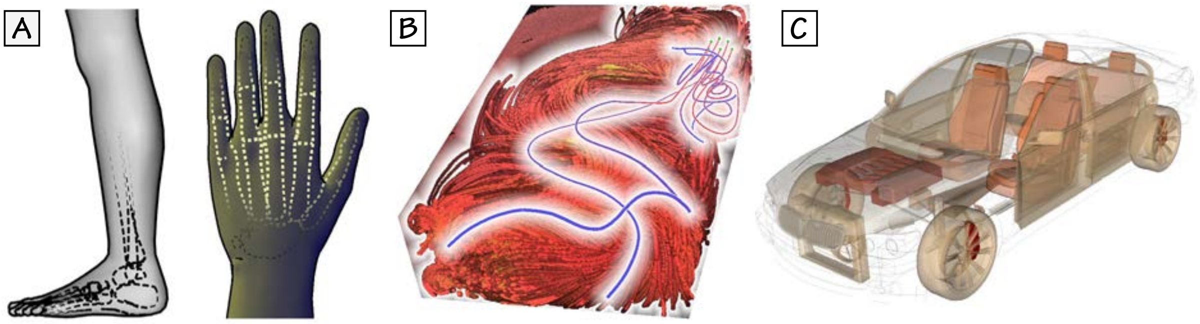


Figure III.16.: Magic-lens metaphors for virtual 3D scenes. (A) 3D medical models stylized using lens combinations (Neumann *et al.* 2007), (B) fluid flow visualization with sampling abstractions (Van Der Zwan *et al.* 2012), (C) descriptive NPR technique using model semantics for visualization (Chang & Collins 2013). Image courtesy by the respective authors.

the original drawing" (Barla *et al.* 2005) and synthesize lines in a geometry stage that represent this structure accurately. Finally, Goodwin *et al.* (2007) provided a shading approach to achieve artistic stroke thicknesses at interactive frame rates. Their work is based on the *isophote distance* as a quantity to represent the thickness of rim shadows at feature contours.

Lens-based Techniques

Cignoni *et al.* (1994) proposed the first three-dimensional extension to the magic lens metaphor. Their main idea originated from the need of LoD techniques to handle multiresolution data, but eventually lead to a generic visualization technique that defines a spherical volume of interest in 3D space that can be parameterized by filter operations to adjust rendering modes and shading styles. Viega *et al.* (1996) extended the metaphor by distinguishing between flat lenses where 3D scene representations are altered inside a lens frustum, and volumetric lenses with rigid shapes to limit the lens effect to a finite volume. In particular, they used volumetric lenses to implement x-ray effects by modulations of opacity to reveal hidden (occluded) scene parts. Neumann *et al.* (2007) provided an interactive framework that enables to parameterize NPR techniques to create expressive line renderings of varying emphasis (Figure III.16A). Technically, they employed spatially-explicit, viewer-aligned lenses that parameterize line entities in screen space, and can be specified by 2D lens controls that integrate common 2D input devices such as mouse or touchpad. Chang & Collins (2013) provided a semantics-based stylization of 3D scene entities—using opacity and color modulations—by aligning model semantics to physical entities from text documents, an approach they call *descriptive NPR* (Figure III.16C). Besides these general approaches, the magic lens metaphor has been particularly explored in medical and geospatial information visualization.

Medical Visualization. Flow visualization is of particular interest in medical visualization to interactively select detailed features for highlighting while reducing visual clutter in context regions (Fuhrmann & Gröller 1998). Viola *et al.* (2004) introduced importance-driven volume rendering, a generalized approach for cut-away views (W. Li *et al.* 2007), where the "level of sparseness" for individual scene objects is selected according to their importance. The approach was enhanced by L. Wang *et al.* (2005) to include non-linearly magnifications of scene objects for closer inspection in their spatial and semantic context. Combinations of volume rendering and focus+context visualization for 3D flow data were conducted, in particular, for fluid simulation using halo-like shading and distance-based blending of LoAs according to semantic lenses (Van Der Zwan *et al.* 2012)— Figure III.16B, and the exploration of blood flow (Gasteiger *et al.* 2011; Lawonn *et al.* 2015). Typically, the approaches include NPR techniques to indicate context objects (e.g., bones) and help distinguish between semantical classes of objects via explicit highlighting (Bruckner *et al.* 2010).

Geospatial Information Visualization. Focus+context visualization using lens-based strategies has been particularly explored in geospatial information visualization to reduce visual clutter and guide a viewer's gaze. For instance, Ropinski *et al.* (2005) applied the 3D magic lens metaphor to reveal information that is hidden in high-dimensional data sets and ease tasks in navigation, landscaping, and urban city planning (Figure III.17A). Trapp *et al.* (2008) presented focus+context visualization techniques on geometric level, using cell-based geometric generalizations (Glander & Döllner 2009) of varying LoA that are nested and clipped according to layered depth images defined by generalized 3D lenses (Figure III.17B). In addition, Trapp *et al.* (2010) proposed a framework for context-based style invariances, such as the SDoF effect, as a pre-attentive feature to improve the perception of 3D geospatial objects. Their implementation is based on distance transforms to provide smooth transitions between region-based style variances. S. Wang *et al.* (2011) provide stylized representations of 2.5D maps by representing virtual city entities in hierarchies according to user-defined regions of interest and using the structural information to visually emphasize or de-emphasize the entities. Pan *et al.* (2013) generalized and extended this approach to virtual 3D city models, where photorealistic rendering and NPR techniques are combined in a single view.

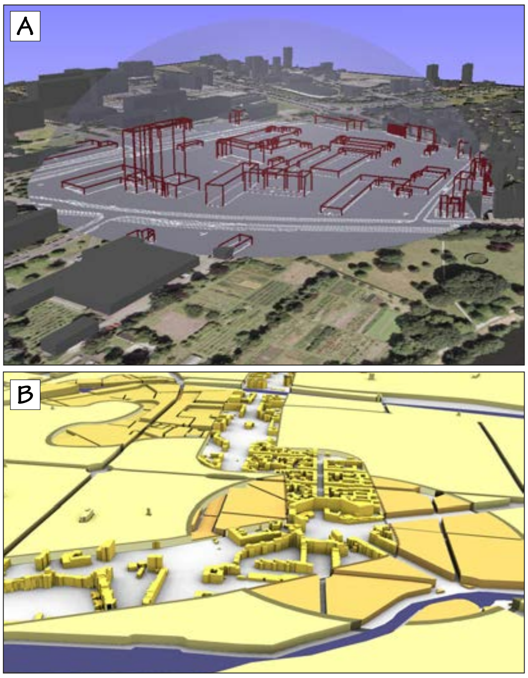


Figure III.17.: Magic-lens metaphors for 3D geovirtual environments. (A) application to a virtual 3D city model with wireframe rendering in the focus region (Ropinski *et al.* 2005), (B) route visualization using generalization lenses (Trapp *et al.* 2008). Image courtesy by the respective authors.

3. Evaluations and Summary

In their seminal work, DeCarlo & Santella (2002) used eye-tracking data to guide image abstraction. In a follow-up study, Santella & DeCarlo (2004) validated their abstraction method by eye tracking, where the locations of eye fixations showed to be directed to image regions of locally preserved detail. They note, that this behavior "accords with the idea that artists carefully manipulate detail to control interest and understanding" (Santella & DeCarlo 2004). Besides evaluating eye movements for predicting visual selection, they also used models of visual attention to measure image salience (Borji & Itti 2013), but note that they differ in their absolute capture of interest. Cole *et al.* (2006) applied the method of Santella & DeCarlo (2004) to 3D models and verified that their stylized focus pull effect significantly draws a viewer's gaze to the emphasized regions (Figure III.18). Redmond & Dingliana (2009b) compared photorealistic rendering techniques

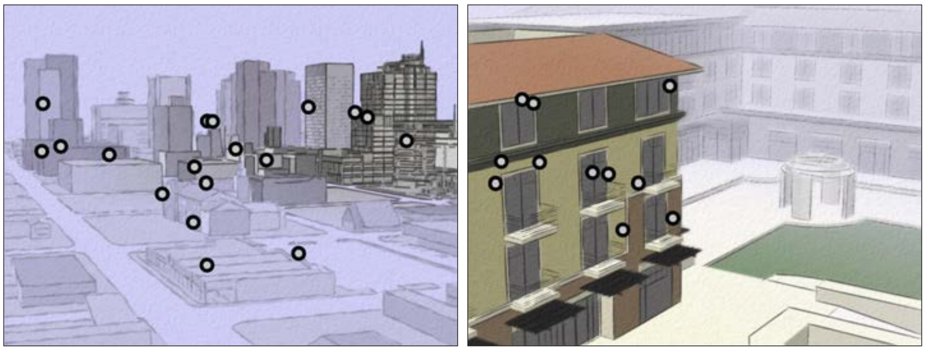


Figure III.18: Evaluation of the *stylized focus pull effect* with tracked eye-movements (overlays). Image courtesy by Cole *et al.* (2006).

with adaptive abstractions produced by NPR techniques in a visual search task. They also conclude that NPR can be useful to direct visual saliency, but the effect depends on the scene complexity, the user's context, and the used LoA. In particular, they argue that the evaluated techniques are not effective in 3D scenes with low detail and that semantics-based information should be encoded in the visual representations to significantly improve the users' task performance.

The pre-attentive nature of the SDoF effect has been verified in studies by Giller *et al.* (2001) and Kosara *et al.* (2002), who showed that it supports users to detect and locate objects more efficiently. Besides highlighting, however, the studies also showed that the SDoF effect is less useful to discriminate object classes by varying the sizes of the smoothing kernels. Eventually, this requires object properties to be encoded more effectively, e.g., by using principles found in cartographic design and graphic semiology principles (Chapter II).

To summarize, many works demonstrate how interactive focus+context visualization can be technically achieved to provide importance-guided representations of complex 2D and 3D data, e.g., by selecting and blending style variances according to regions of interest or view metrics. However, to serve the different interests of users in specific information, open challenges remain on how to adapt visualization to different contexts and contents and how to support "saliency-guided" visualization based on user interaction or dynamically changing thematic information. In Chapter VI, a concept and an implementation of a system is provided that enables different presentation styles, their seamless integration within a single view, and parameterized transitions between them, which are defined according to tasks and camera view. The system extends approaches found in focus+context visualization to enable smooth transitions between levels of structural abstraction with a context-dependent selection of LoAs using metrics defined per geospatial feature type, and their dynamic parameterization at run-time, e.g., using touch or natural language inputs.

3. Cartography-Oriented Visualization of 3D Geopatial Data

This section gives an overview on illustrative visualization techniques for 3D geospatial data. Table III.1 presents a non-exhaustive overview of techniques that consider cartographic design aspects in 3D spaces. This overview is drawn with respect to the extended 3D semiotic model (Figure II.2) of Chapter II, and is categorized by feature type semantics using CityGML's taxonomy, i.e., terrain, green spaces, buildings and sites, and transportation networks.

Table III.1: NPR techniques for 3D virtual environments based on cartographic design aspects. The techniques are sorted by year and primarily used feature type:

[T] terrain

[V] green spaces

[B] buildings and sites

[S] vector data/streets

[G] general

				Filtering			Mapping (Symbolization)								Rendering			Perception			
		Publication	**Primary Models**	feature-based	location-based	view-dependent	Graph. Variables	Graph. Elements	Color	Font	Material	Texture	Transparency	Orientation	Animation	Lighting	Shading & Shadows	Env. Projection	Psychological	Physiological	Coherence	**User Interaction**
Terrain Design	1985	Yoeli 1985	T				×	×				×		×			×					
	1994	Dowson 1994	T			×	×	×								×	×		×			×
	2000	P. J. Kennelly & Kimerling 2000	T				×	×						×			×					
	2001	Hurni *et al.* 2001	T				×	×				×				×	×					×
	2001	B. Jenny 2001	T	×			×									×	×					×
	2001	P. Kennelly & Kimerling 2001	T		×		×	×	×								×					
	2002	Lesage & Visvalingam 2002	T			×	×	×											×			×
	2003	Whelan & Visvalingam 2003	T			×	×	×											×			×
	2004	Buchin *et al.* 2004	T	×	×		×	×				×		×		×	×				×	×
	2009	Bratkova *et al.* 2009	T	×			×	×	×			×		×		×	×	×	×			
	2009	Degener & R. Klein 2009	T															×	×			
	2009	P. J. Kennelly 2009	T				×		×							×	×		×			
	2010	Glander *et al.* 2010	T		×		×	×	×						×		×					×
	2010	B. Jenny *et al.* 2010a	T				×	×									×		×			×
	2010	Leonowicz *et al.* 2010	T				×										×		×			×
	2011	Mower 2011	T	×	×		×	×				×		×		×	×				×	×
	2012	P. Kennelly 2012	T				×	×				×		×			×					
	2013	Deng *et al.* 2013	T			×												×	×		×	×
	2014	Samsonov 2014	T				×	×	×			×		×			×					×
	2015	Geisthövel & Hurni 2015	T	×			×	×				×		×			×					
	2015	Willett *et al.* 2015	T	×	×		×	×						×		×	×	×	×			×
Green	2000	Deussen & Strothotte 2000	V			×	×	×									×		×		×	×
	2006	Coconu *et al.* 2006	V			×	×	×									×		×		×	×
	2006	Luft & Deussen 2006	V	×		×	×	×	×			×	×			×	×		×		×	×
Building Design	2003	Döllner & Walther 2003	B	×		×	×	×	×		×	×		×			×					×
	2005	Döllner & Buchholz 2005a	B	×		×	×	×	×		×	×		×			×		×			×
	2005	Elias *et al.* 2005	B	×			×	×	×	×				×					×			
	2007	Glander *et al.* 2007	B	×		×	×	×	×		×						×					×
	2008	Grabler *et al.* 2008	B/S	×	×		×	×	×	×	×			×				×				×
	2008	Loya *et al.* 2008	B	×			×	×	×			×		×								
	2009	Qu *et al.* 2009	B/S	×	×		×	×		×									×			×
	2009	Zanola *et al.* 2009	B	×		×	×	×	×		×	×		×			×		×			×
	2013	Jahnke *et al.* 2011	B	×			×	×	×										×			
	2013	Jahnke *et al.* 2013	B	×			×	×	×			×		×								
	2013	Hirono *et al.* 2013	B/S	×	×	×	×	×											×			×
	2014	Kol *et al.* 2014	B											×				×	×			×
	2015	Deng *et al.* 2015	B/S	×	×	×	×	×											×			×
	2015	Kada *et al.* 2015	B	×				×														
Vector Data	2002	Kersting & Döllner 2002	S/T	×			×	×	×	×		×							×			×
	2003	Wartell *et al.* 2003	S/T																			
	2006	S. Takahashi *et al.* 2006	S/T	×		×												×	×		×	×
	2007	Schneider & R. Klein 2007	S																			
	2011	Vaaraniemi *et al.* 2011	S			×	×	×	×			×	×						×			×
Labeling	2001	Bell *et al.* 2001	G	×		×	×	×		×									×			×
	2005	Götzelmann *et al.* 2005	G	×		×	×	×		×				×					×		×	×
	2006	Maass & Döllner 2006	G	×		×				×				×				×	×			×
	2007	Ropinski *et al.* 2007	G	×		×	×	×		×				×			×		×			×
	2008	Cipriano & Gleicher 2008	G	×			×	×		×				×					×			×
	2008	Maass & Döllner 2008	G	×		×				×				×				×	×			×
	2013	Vaaraniemi *et al.* 2013	G	×		×	×	×	×	×				×					×			×
	2014	Vaaraniemi *et al.* 2014	G	×		×			×	×				×					×		×	×
MPVs	2008	Lorenz *et al.* 2008	G										×					×	×			×
	2008	Möser *et al.* 2008	G										×					×	×			×
	2010	H. Jenny *et al.* 2010	G															×	×			×
	2011	H. Jenny *et al.* 2011	G		×	×												×	×			×
	2011	Pasewaldt *et al.* 2011	G			×	×	×	×				×	×				×	×		×	×
	2013	H. Jenny & B. Jenny 2013	G	×			×					×		×		×	×	×				
Focus+Context	2008	Trapp *et al.* 2008	G	×	×		×	×	×		×						×		×			×
	2009	Glander & Döllner 2009	G	×	×	×	×												×			×
	2010	Trapp *et al.* 2010	G	×	×	×	×	×	×			×	×				×		×			×
	2011	S. Wang *et al.* 2011	G	×	×		×	×	×		×	×					×		×			×
	2013	Pan *et al.* 2013	G	×	×		×	×	×		×	×					×		×			×

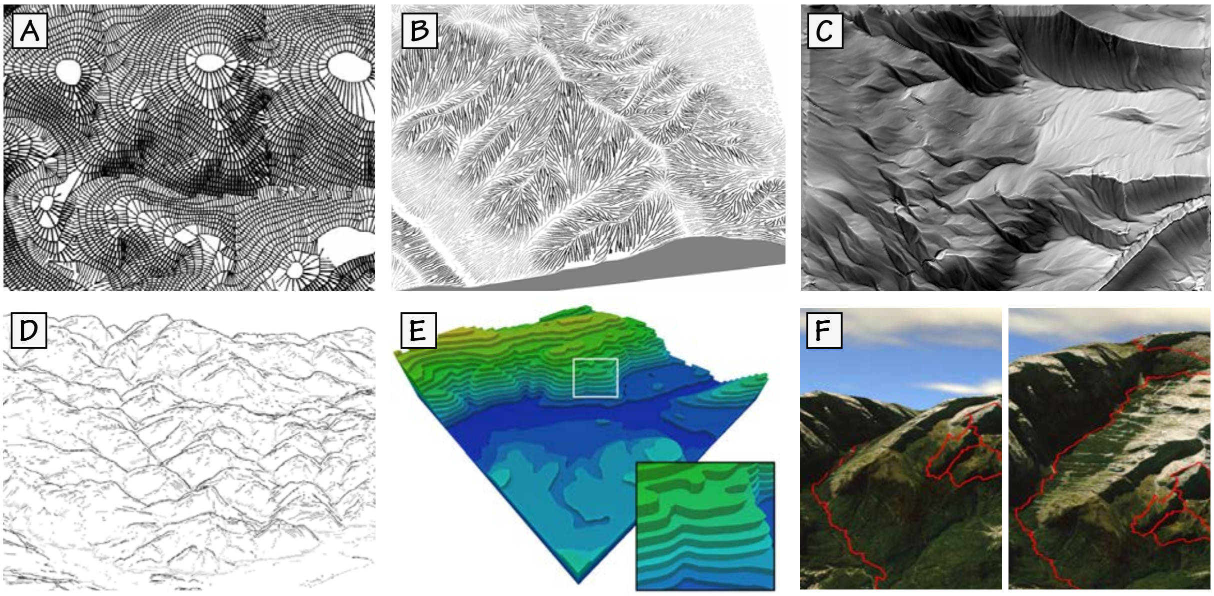


Figure III.19.: Illustrative visualization techniques for digital terrain models. (A) simulation of slope and shadow hachures (Yoeli 1985), (B) visualization of hachures with varying thickness (Buchin *et al.* 2004), (C) relief shading using Poisson image editing (Geisthövel & Hurni 2015), (D) rendering of profile strokes (Whelan & Visvalingam 2003), (E) stepped 3D terrain visualization via isolines (Glander *et al.* 2010), (F) panorama map synthesis using deformation techniques (Degener & R. Klein 2009). Image courtesy by the respective authors.

1. Terrain

The abstract representation of terrain has a long tradition in mountain cartography (Häberling & Hurni 2002). Visualization techniques are divided into four categories (P. J. Kennelly & Kimerling 2006): the simulation of traditional hand-drawn styles by means of slope lines and hachures, relief shading, contour-lining to indicate ridges and valleys, the communication of height information via isolines, and non-linear projections to minimize feature occlusions.

Example-based Texturing Techniques

Hachures are short lines that are drawn with respect to the steepest slope direction to create tonal variations for the representation of topography (Imhof 1982). Yoeli (1985) proposed the first algorithm to simulate slope and shadow hachures, where straight lines are placed between consecutive contour levels (Figure III.19A). P. J. Kennelly & Kimerling (2000) enhanced the method to produce small-scale hachures, and Hurni *et al.* (2001) extended the method to include ridge lines and scree representations, i.e., the usage of irregular dots that cover the slope (B. Jenny *et al.* 2010b), for digital cliff drawing. Buchin *et al.* (2004) simulated loose line drawings, slope lines and shadowed hachures by deriving morphometric variables (e.g., slope steepness, cross-section curvature) from digital terrain models for path tracing and adaptive line stylization (Figure III.19B). Bratkova *et al.* (2009) enhanced the method by using surface classification data to guide procedural stroke textures to feature semantics (e.g., forests, cliffs, and snow), whereas Samsonov (2014) proposed a technique that uses vector flowlines and a simultaneous application of slope and aspect hachuring. Cross-hatching is an alternative approach to hill-shading, where "lines of variable thickness and orientation approximate tonal variations associated with shading and shadowing" to represent shape (P. Kennelly 2012).

Shading Techniques

B. Jenny (2001) proposed a technique for cartographic relief shading that computes grey tones according to a light source and terrain morphology. The technique is locally adjusted to emphasize selected topographic features such as ridges, valleys, and watersheds. P. J. Kennelly (2009) presented an enhanced approach that includes color tones adapted to the local terrain aspect, multiple light sources with varying intensity to depict steep terrain more accurately, and curvature values. Leonowicz *et al.* (2010) locally altered the characteristic shape of terrain features using image filtering for generalization, whereas Geisthövel & Hurni (2015) locally enhanced the contrast in relief shading using Poisson image processing (Figure III.19C).

Edge-enhancement Techniques

The highlighting of contours is mainly driven by generic NPR techniques. Here, typical approaches are based on thresholding depth information, e.g., synthesized in a *G-buffer* (Saito & T. Takahashi 1990), to obtain contour lines for profile-based sketching (Dowson 1994). Further, techniques for image-space and object-space edge enhancement are distinguished, e.g., the first by thresholding luminance or depth information (Lesage & Visvalingam 2002; Whelan & Visvalingam 2003) to retrieve profile lines (Figure III.19D), and the latter to obtain pen-and-ink illustrations (Mower 2011).

Isoline Techniques

P. Kennelly & Kimerling (2001) proposed a first automated approach to simulate layered relief depictions, following the illuminated contour method of Tanaka (1950). Here, isocontours are depicted with stepped colors of varying thickness to communicate height information. Glander *et al.* (2010) provided an extended approach that synthesizes the geometry required for 3D isocontours in real-time (Figure III.19E), with application for time-variant data and optional smooth transitions to 3D (photo-)realistic terrain representations.

Projection-based Techniques

Perceptional design aspects are considered in progressive and degressive projections, where the goal is to reduce occlusions of geospatial features such as routes projected onto terrain surfaces in navigation scenarios. Degener & R. Klein (2009) computed panoramic maps in a variational approach by optimizing the visibility of selected features while minimizing the deformation of the terrain's shape (Figure III.19F). They used resemblance and visibility factors to constrain the deformation. By contrast, Deng *et al.* (2013) proposed a greedy approach based on ray-tracing that resolves occlusions once they are detected to maintain interactive frame rates. Finally, Willett *et al.* (2015) explored interactive deformations via local shearing to reveal depth information that can be "paired with existing interactions such as pan and zoom".

2. Water Surfaces

The illustrative rendering of water surfaces in 3D spaces with cartographic design aspects is not covered in current literature. Related works use colorization, edge enhancements, and texturing to achieve cartoon-like water effects (e.g., ripples) and to emphasize liquid movements (Eden *et al.* 2007; Yu *et al.* 2007). However, these approaches do not relate to traditional cartographic design aspects, where the figure-ground organization is typically highlighted to ease orientation, navigation, or analysis tasks.

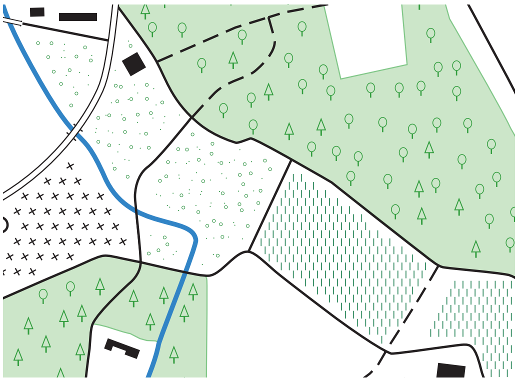


Figure III.20.: Irregular point pattern synthesis by B. Jenny *et al.* (2010b). Image courtesy by the respective authors.

3. Green Spaces

Only few works deal with the cartography-oriented visualization of green spaces in 3D virtual environments. In general, vegetation objects can be distinguished without and with individual characteristics (Döllner & Buchholz 2005b), the first via a synthesis of point patterns and element arrangements for symbolization, and the latter based on 3D object stylization.

Point pattern synthesis and element arrangements are discussed in the works by Hurtut *et al.* (2009) and B. Jenny *et al.* (2010b), where glyphs are regularly or irregularly distributed without overlapping and coalescing, and the ratio of the glyph sizes and graphical appearances is adjusted via example-based texturing (Figure III.20). The obtained texture patches may then be projected onto surfaces in 3D space to symbolize land cover information. Deussen & Strothotte (2000) and Coconu *et al.* (2006), by contrast, employed visualization techniques for 3D vegetation models such as individual tree objects. Here, tree models are stylized using view-dependent silhouette enhancements by thresholding depth differences (Deussen & Strothotte 2000), e.g., to create pen-and-ink illustrations of landscapes. Coconu *et al.* (2006) extended this approach to abstract rendering of plants, which includes mechanisms for contour-lining, leaf abstraction, hatching, and lighting (Figure III.21). Finally, Luft & Deussen (2006) provide a technique for real-time artistic stylization of 3D models by simulating characteristics of watercolor paintings, including cartography-oriented design aspects such as rendering with constrained color palettes and edge enhancements.

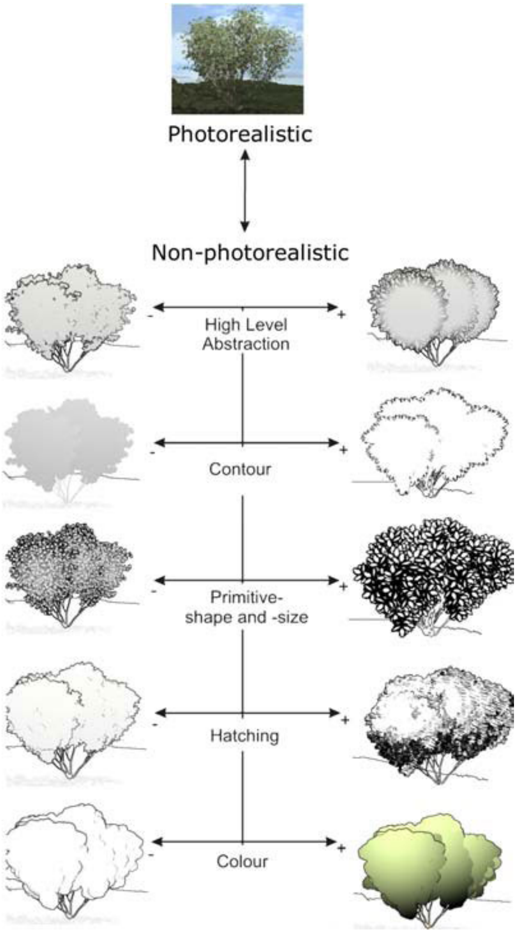


Figure III.21.: Design aspects used in the pen-and-ink illustration technique of Coconu *et al.* (2006). Image courtesy by the respective authors.

4. Buildings and Sites

Illustrative visualization techniques for 3D building and site models are often based on constrained color palettes with typified or stylized façade elements,

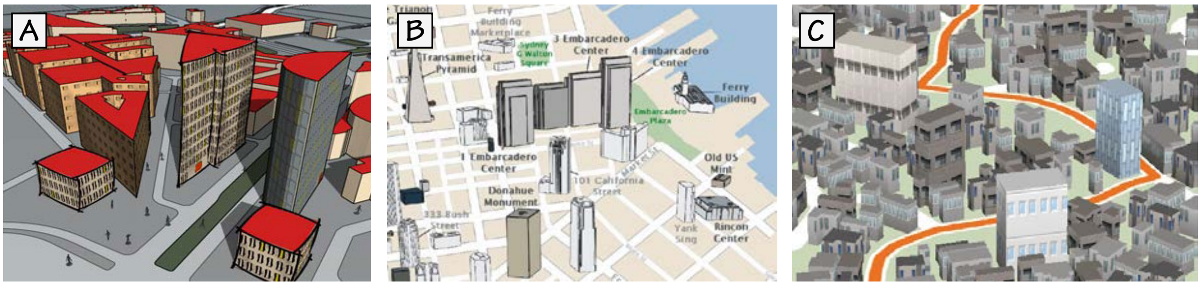


Figure III.22.: Illustrative visualization techniques for buildings and sites. (A) expressive rendering using object-space edge enhancements, tone shading and procedural texturing (Döllner & Buchholz 2005a), (B) visualization with a tourist map design (Grabler *et al.* 2008), (C) deformation of building instances for focus+context zooming and disocclusion of route segments (Hirono *et al.* 2013). Image courtesy by the respective authors.

and symbolized LoA representations. Döllner & Walther (2003) proposed texture-based techniques for sketchy rendering. In particular, they provided an approach for texture-based edge enhancement using view-aligned impostors, e.g., to communicate uncertainty in city planning and reconstruction scenarios. Döllner & Buchholz (2005a) enhanced the technique to include shadows, *n*-tone-shading, and procedural façade textures derived from semantic attributes such as building height (Figure III.22A). A more abstract visualization directed towards CAD-styles was proposed by Jahnke *et al.* (2008). Here, single colors with edge enhancements of varying stroke thicknesses are applied to encode semantic information. Zanola *et al.* (2009) evaluated these abstraction styles in their empirical study, which indicates that NPR is suitable for uncertainty visualization since the "participants' confidence ratings were significantly highest for photorealistic displays, followed by CAD-styles". Loya *et al.* (2008) also used procedural façade textures, but synthesized them from photorealistic textures by analyzing primary periodic features (e.g., windows) and dominant colors. Finally, Jahnke *et al.* (2013) surveyed typification operators to include façade elements of different distribution patterns, while Kada *et al.* (2015) proposed complementary, geometric abstractions of 3D building models using mesh simplifications, which is based on edge collapsing and vertex move operations for LoD representation.

The design of landmarks is of particular interest in cartography to ease orientation guidance. Elias *et al.* (2005) considered design concepts, where landmark categories are distinguished by component classes or attribute values (e.g., usage type), providing a LoA visualization that ranges from photorealistic depictions, over iconified styles to abstract symbols. Glander *et al.* (2007) provided a technique that emphasizes 3D landmarks by scaling the geometry according to an importance function and the view configuration. Grabler *et al.* (2008) presented a system for synthesizing tourist maps, where 3D building objects are ranked and filtered using semantic information retrieved from online resources (Figure III.22B). Here, landmarks are depicted with dominant colors and edge enhancements, and are aligned according to optimized views. Kol *et al.* (2014) also transformed the orientation of building façades in 3D spaces, but their methods are based on canonical views to improve the recognizability of buildings and ease the exploration of virtual 3D city models. Further relevant work is found in the disocclusion of transportation networks by using deformation techniques for focus+context zoooming (Qu *et al.* 2009; Hirono *et al.* 2013; Deng *et al.* 2015). Here, less important buildings and landmarks are scaled by solving an optimization problem to improve the visibility of route segments while minimizing distortions and maintaining feature proportions (Figure III.22C).

Transportation Networks

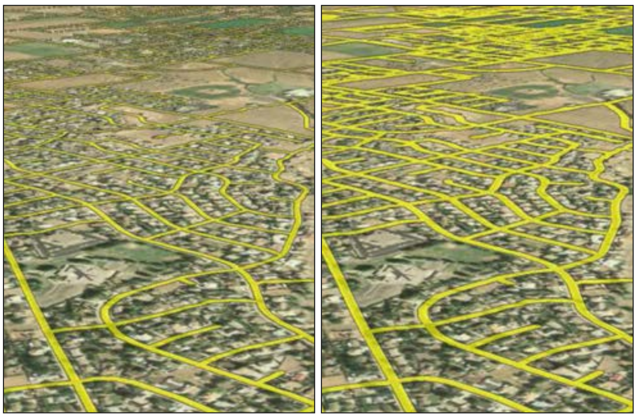


Figure III.23.: View-dependent design of transportation networks using (left) no scaling, and (right) a distance-based scaling function. Image courtesy by Vaaraniemi *et al.* (2011).

A common challenge for rendering transportation networks in 3D virtual environments is their seamless projection onto digital terrain models (*draping*). First approaches used geometry-based methods that explicitly model vector data as graphical elements and render them separately from a 3D terrain mesh (Polis *et al.* 1995; A. Weber & Benner 2001). However, these methods only provide pre-computations without the capability for dynamic design aspects, in particular at the rendering stage. A first approach for LoD rendering used texture-based mapping to project vector features on 3D terrain models (Kersting & Döllner 2002). Similar approaches used principles of shadow mapping for a perspective parameterization by taking the point of view into account (Schneider *et al.* 2005), continuous LoD methods (Wartell *et al.* 2003), and shading for dynamic feature editing (Bruneton & Neyret 2008). However, these approaches are limited in detail and sharpness once texture data is rendered in close view distances. Further, they require explicit LoD mechanisms to take effect prior to rendering which results in additional computational costs. Other methods utilized the hardware-accelerated stencil buffer with a type of shadow volumes (Schneider & R. Klein 2007; Vaaraniemi *et al.* 2011), or relied on screen-space rendering (Ohlarik & Cozzi 2011) to reduce computational costs, but typically do not deal with the problem of visual clutter in image regions of high perspective compression. Here, a promising approach is to adjust design aspects view-dependently such as the width of the contour lines and line segments (Vaaraniemi *et al.* 2011)—Figure III.23, or employ *contextual zooming*, where hierarchical route maps of varying resolution are selected and combined view-dependently to ease navigation tasks (F. Wang *et al.* 2014).

5. Complementary Data and Visualization Techniques

Besides focus+context visualization, techniques for the illustrative visualization of 3D geospatial data are found in labeling and multiperspective visualization.

Labeling

Labels are design elements used in cartography to enrich maps with meta-information. One may distinguish between placing labels within surfaces of the annotated features (*internal labeling*), and placing labels next to features or outside surfaces of interest—connected by an anchor line (*external labeling*). In both approaches, three major goals are identified: labels should be legible and associated with the correspondent feature (Imhof 1975), occlusion of visualized information should be avoided, and temporal coherence should be maintained (Vaaraniemi *et al.* 2013). Bell *et al.* (2001) introduced a view-management component for 3D virtual environments for "maintaining visual constraints on the projections of objects on the view plane such as locating related objects near each other or preventing objects from occluding each other". This approach selectively adapts design aspects of labels, such as position, size, and transparency, by approximating unoccupied view spaces via a set of rectangles. Maass & Döllner (2006) proposed a similar screen-space approach that uses vertical slots to stack labels, sorted by the view distance. However, these approaches may produce dense clusters and elongated anchor lines that may impair the visual association and depth perception (Maass *et al.* 2007).

Vaaraniemi *et al.* (2013) define concepts to improve the visibility of occluded labels in 3D spaces, including transparency and highlighting via glow effects; approaches that "preserve visual association, retain the label's readability, and run at interactive rates". In addition, Vaaraniemi *et al.* (2014) note that filtering is an important design aspect to avoid cluttered displays, and present a general approach using priority metrics, e.g., defined in a navigation scenario. For internal labeling, a typical approach is to derive medial axes of annotated feature surfaces based on a distance transform (Götzelmann *et al.* 2005; Ropinski *et al.* 2007; Cipriano & Gleicher 2008). Here, distance information is used to place labels within the annotated surfaces so that they follow the principal curvature directions (Figure III.24). A similar objective was pursued by Maass & Döllner (2008), but using geometric hulls for positioning.

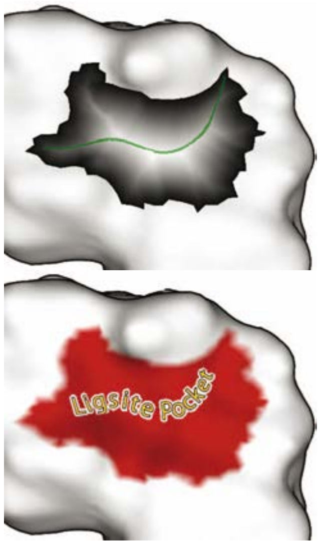


Figure III.24.: Internal labeling using distance fields for positioning. Image courtesy by Cipriano & Gleicher (2008).

Multiperspective Views

Landscape artists practiced the seamless combination of multiple perspective views (MPVs) in a single graphical representation for more than 400 years. Here, a major goal is to disocclude landmarks or regions that are of particular interest to the user. Prominent applications of MPVs can be found in the panorama maps of H.C. Berann, where orthographic projections are used in the foreground and upright projections are used in the background. Patterson (2000) discussed design aspects and variables of Berann's panorama maps that are partially implemented in previous techniques. For instance, the view in the foreground may use a high LoA to ease localization, while features in the background may be depicted with high detail for navigation assistance. The implementation of real-time degressive and progressive MPVs for virtual 3D city models were first proposed by Lorenz *et al.* (2008). Here, orthographic and upright projections are alternatively used for the foreground and background regions of a rendered image, between which a seamless interpolation is performed (Figure III.25). In addition, a view-distance-based LoA is used for focus+context visualization, e.g., using iconic depictions to ease the mental mapping to real-world phenomena, and a symbolized depiction for orientation guidance. Möser *et al.* (2008) extended the approach using isometric perspectives to improve the visibility of building façades and using Hermite curves to adjust the deformation, whereas Pasewaldt *et al.* (2011) use B-Spline curves to parameterize the deformation "without the need of combining several curves". H. Jenny *et al.* (2010) provided an interactive system for progressive projection to

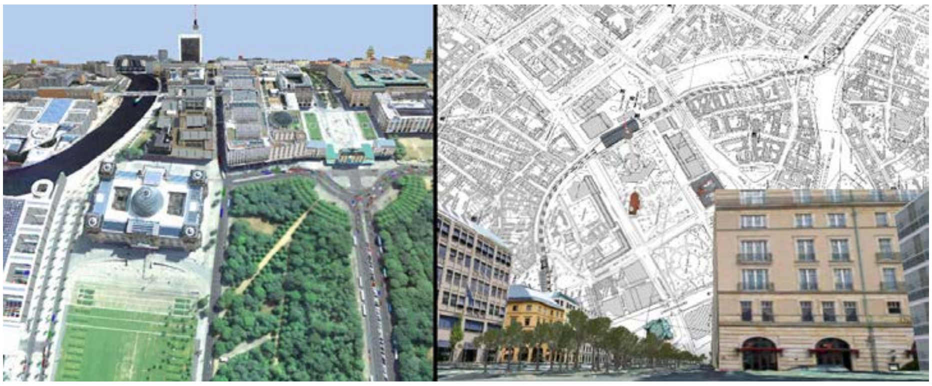


Figure III.25: Multiperspective visualization of a virtual 3D city model using two different configurations.

(left) progressive view

(right) degressive view

Image courtesy by Lorenz *et al.* (2008).

adjust projection parameters and create 360° strip panoramas. Later, H. Jenny *et al.* (2011) and H. Jenny & B. Jenny (2013) enhanced the system to integrate local terrain deformations and example-based texturing.

4. Concluding Remarks

Non-photorealistic rendering, focus+context visualization, and cartography-oriented visualization of 3D geosaptial data are identified as major related fields of research. In particular, edge-preserving filtering emerged as an essential building block to reduce image details without loss of salient structures. In this thesis, GPU versions of image filters are used to demonstrate how they can interactively serve perspective-coherent abstraction and highlighting of diffuse, normal, and ambient occlusion maps for illustrative and focus+context visualization, e.g., to enhance the stylized focus pull and semantic depth-of-field effects. Here, the proposed techniques generalize the art maps approach to provide a coherent stylization with respect to image filtering, i.e., to interactively filter high-detail texture information at run-time, e.g., photorealistic imagery, without using pre-designed mip-map levels, and enable a user-defined LoA texturing that is not limited to color maps but also copes with multiple layers of textures used for visualization, e.g., to integrate thematic information.

Considerable research has also been done in technology-driven directions for the cartography-oriented visual design of complex 3D geospatial information. Table III.1 summarizes the reviewed techniques with respect to the extended 3D semiotic model of Chapter II and their dedicated application to primary geospatial feature types. Here, it can be concluded that the visualization of digital terrain and building models have been the main focus, filtering is inherent to most techniques to reduce visual complexity, concepts such as multiperspective and focus+context visualization have been explored to direct a viewer's focus of attention to features or regions of interest, and that user interaction plays a major role to take a user's context into account. However, approaches for mapping cartographic design principles to the real-time rendering pipeline of 3D computer graphics, interactively and according to dynamically changing thematic information, remain to be explored. This thesis extends and complements the reviewed techniques by methods that synthesize cartographic renditions of basic feature types such as water surfaces—not addressed in current literature—, buildings, and transportation networks, which can be dynamically parameterized using core mechanisms of the real-time rendering pipeline such as texturing and shading. In particular, a system is proposed that enables the seamless integration of these provided graphic styles, and parameterized transitions between them, which includes a novel iconification concept to seamlessly interpolate between photorealistic and cartographic representations of 3D landmarks.

IV

Level-of-Abstraction Texturing

The work presented in this chapter is partly based on

Semmo, A. & Döllner, J. "Interactive Image Filtering for Level-of-Abstraction Texturing of Virtual 3D Scenes". *Computers & Graphics* **52,** 181–198 (2015)

Semmo, A. & Döllner, J. *Image Filtering for Interactive Level-of-Abstraction Visualization of 3D Scenes.* in *Proc. Computational Aesthetics* (2014b), 5–14

Semmo, A. & Döllner, J. *An Interaction Framework for Level-of-Abstraction Visualization of 3D Geovirtual Environments.* in *Proc. ACM SIGSPATIAL Workshop on MapInteraction* (2014a), 43–49

Engel, J., Semmo, A., Trapp, M. & Döllner, J. *Evaluating the Perceptual Impact of Rendering Techniques on Thematic Color Mappings in 3D Virtual Environments.* in *Proc. Vision, Modeling & Visualization* (2013), 25–32

Common 3D scenes are characterized by their visual complexity. In particular, the appearance of most objects can be defined by textures used to design the objects' appearance. Prominent examples of texture-based 3D scenes include virtual 3D building, city, and landscape models or environments used for gaming applications. As outlined in Chapter II, texture mapping is a key technology in today's graphics hardware. In particular, it provides important functionalities used in both photorealistic and non-photorealistic real-time image synthesis. Common texture maps encode color, diffuse, normal, or displacement information as *surface properties* to enrich shading and lighting effects (e.g., for the building façades in Figure IV.1).

To improve image quality of textured 3D scenes, important contents of textures should be emphasized while less significant details should be removed—a challenging task because feature contours and global salient structures must be preserved. Edge-preserving image filtering is a promising approach to address this problem. Popular image filters—reviewed in Chapter III—serve as smoothing or enhancing operators such as the bilateral filter (Tomasi & Manduchi 1998) and DoG filter (B. Gooch *et al.* 2004). Previous works have focused on applying these filters in a post-process stage on the rendered results to foster an artistically stylized rendering (Kyprianidis *et al.* 2013). These approaches are able to smooth low-contrast regions and preserve high-contrast edges; however, they are generally not able to preserve depth cues important for perceiving model contents as three-dimensional (e.g., occlusion, texture gradient). For instance, the fine granular patterns on the ground and rooftops of the 3D scene shown in Figure IV.1 are not preserved because spatial relationships and linear perspective are not considered, and fine objects may become indistinguishable from the

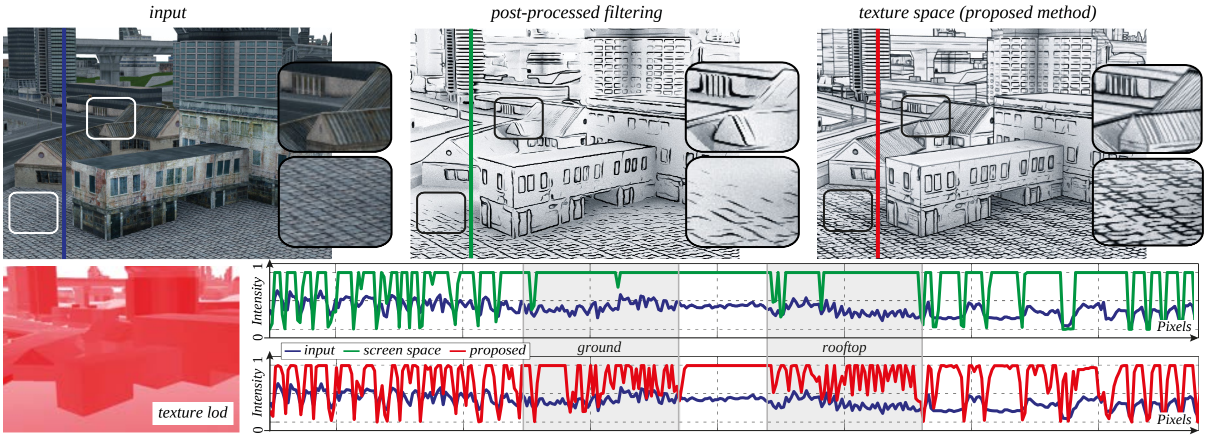


Figure IV.1.: Exemplary results of a flow-based difference-of-Gaussians filtering (FDoG) for a textured 3D scene, rendered in real-time using the proposed approach for LoA texturing. The closeups and scanline plots illustrate the high accuracy for texture gradients when using the proposed method (red line) instead of conventional filtering in a post-process stage on the rendered color image (green line).

background. For effective visual information encoding, "good image filtering" needs to preserve these cues to help perceive relative positions, sizes, distances, and shapes more clearly (Pfautz 2000; Goldstein 2010).

This chapter explores LoA texturing and its applications by means of image filtering. The contributions are based on the following aspects:

(1) Filtering should be perspectively coherent. Linear perspective, occlusion, and texture gradient are effective cues for humans to infer depth (Wanger *et al.* 1992; Surdick *et al.* 1994). Texture mapping considers these cues in perspective projections by foreshortening and scaling. Therefore, the proposed methods perform image filtering prior to texture mapping via decoupled deferred texturing. It is a simple yet effective approach of preserving these cues without requiring modifications of the original filter algorithms.

(2) Filtering should be interactively parameterized. To serve the different interests of users in prioritized information, the LoA should be dynamically adapted, which requires specialized interaction techniques for parameterization. To this end, the proposed approach is coupled with an extensible interaction framework to parameterize a context-aware image filtering according to spatial, semantic, and thematic data. The framework provides intuitive interfaces such as touch-based or natural language inputs via textual descriptions. In addition, interactive frame rates should be maintained to provide a responsive system during interaction and navigation in 3D space (e.g., dynamic viewing situations, regions of interest). For this reason, the proposed approach contributes per-fragment and progressive filtering together with caching strategies to enable real-time frame rates for local image filters without requiring to pre-process texture data.

In image-based artistic rendering it is often desired to give the impression that texture features have been applied (painted) on a flat canvas (Bénard *et al.* 2011). Instead, this chapter draws upon the potentials of image filtering for visualization purposes including cognitive principles—here: psychological design aspects; a contemporary field of research that poses challenges for the NPR community (A. A. Gooch *et al.* 2010). The goal is to preserve a perspective-coherent scale of texture features, an approach also practiced by artists to enhance depth sensation (Figure IV.2). Previous works have supported the assertion that perspective is commonly used as an important source of depth information (Wanger *et al.* 1992; Surdick *et al.* 1994; Pfautz 2000). In particular, surface textures are essential components of 3D scenes to judge distances, shapes, and spatial layouts (Gibson 1986; Ware 2004). A large body of research is dedicated to the way human sensory systems process these pictorial depth cues (Howard & Rogers 2012); however, there are only a few works that reflect the importance of preserving them when filtering information.

The benefits of the proposed methods are shown for various applications by incorporating texture information coherently with respect to linear perspective. In particular, they enhance focus+context applications, including the *stylized focus pull* (Cole *et al.* 2006) and *semantic depth of field* (Kosara *et al.* 2001) effects for information highlighting and abstraction (Trapp *et al.* 2010). The proposed framework also implements GPU versions of global filters to demonstrate how they can serve perspective-coherent filtering of diffuse, normal, and ambient occlusion maps for geometric detail removal and illustrative visualization. Because filtering is performed in texture space, however, no explicit geometric abstraction is applied, for which specialized rendering techniques are required.

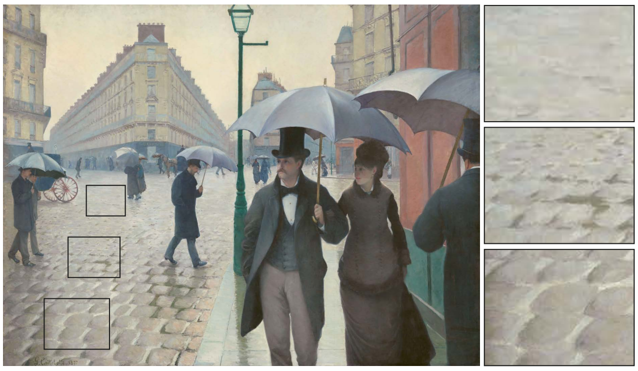


Figure IV.2.: The painting "Paris Street, Rainy Day" (1877) by Gustave Caillebotte (Google Art Project). The artist carefully uses texture gradient with linear perspective to enhance depth sensation. For instance, the cobblestones get progressively smaller in depth until they are completely smooth.

The remainder of this chapter is structured as follows. Section 1 motivates the relevance of perspective coherence for image filtering in 3D perspective views. Section 2 presents the technical approach to deferred texture filtering. Evaluations and applications for the proposed methods are presented and discussed in Section 3. Finally, Section 4 summarizes the achieved results.

1. Motivation

This section motivates *perspective coherence* for LoA texturing. The example 3D scene shown in Figure IV.3 is based on a flow-based bilateral filter (Kyprianidis & Döllner 2008). The input textures were converted to a low-pass filtered mipmap pyramid (Williams 1983) used for anti-aliasing. The filtering was performed (1) in a post-process stage on the rendered results and (2) in texture space prior to perspective projection. Given a texture $T: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ within the RGB color space as input, the first approach is described by an image filtering F performed *after* sampling the convolution of $T * P$ via mipmapping, with P being a projector function that maps texture information into the domain of the rendered image $I: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ and K being the mipmap kernel:

$$I(x,y) = F[K \cdot (T * P)](x,y). \tag{IV.1}$$

While this approach works in a similar manner as pre-filtering a texture in regions of high texture LoD, it is not able to preserve structures in regions of low

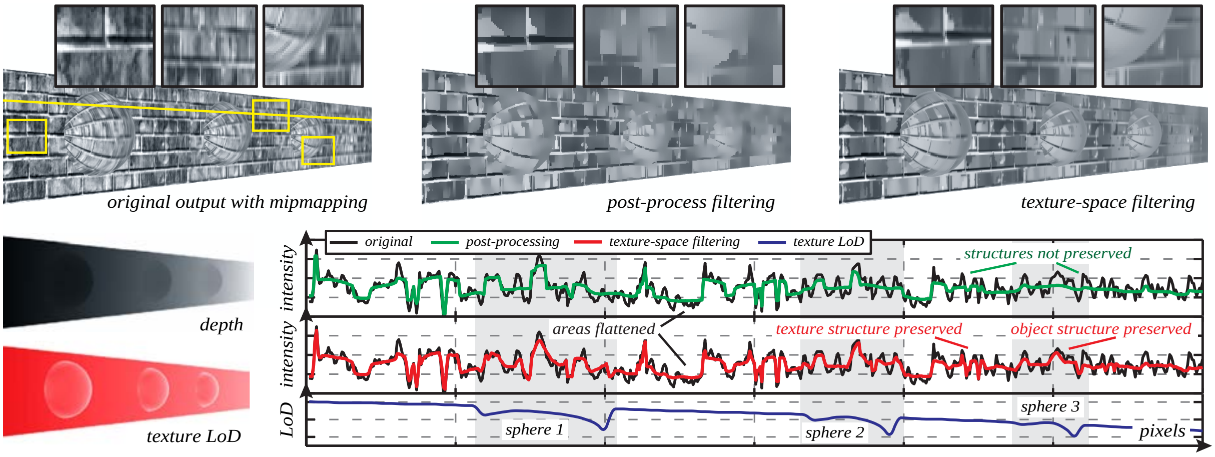
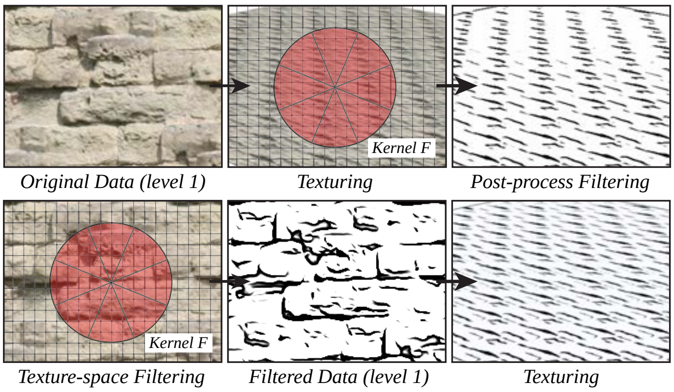


Figure IV.3.: Exemplary textured 3D scene for which the flow-based bilateral filter ($\rho = 2.0, n_e = 1, n_a = 4, \sigma_d = 4.0, \sigma_r = 5.0$) is applied: (1) on the original output, (2) on each mipmap level prior to texture mapping. The scanline plots illustrate the differences in regions of high texture LoD and occlusion, where the second approach clearly preserves texture gradients and object boundaries.

texture LoD because only compressed information serves as input for the image filtering kernel F. This effect can be observed in the intensity plots along the yellow scanline shown in Figure IV.3 and in the overview given in Figure IV.4.

In the second approach, mipmap levels are filtered separately prior to texture mapping to approximate the filtering effect of F on $K(T * P)$, with K_F being the mipmap kernel that uses the pre-filtered mipmap levels:

$$I'(x, y) = K_F \cdot (T * P)(x, y). \tag{IV.2}$$



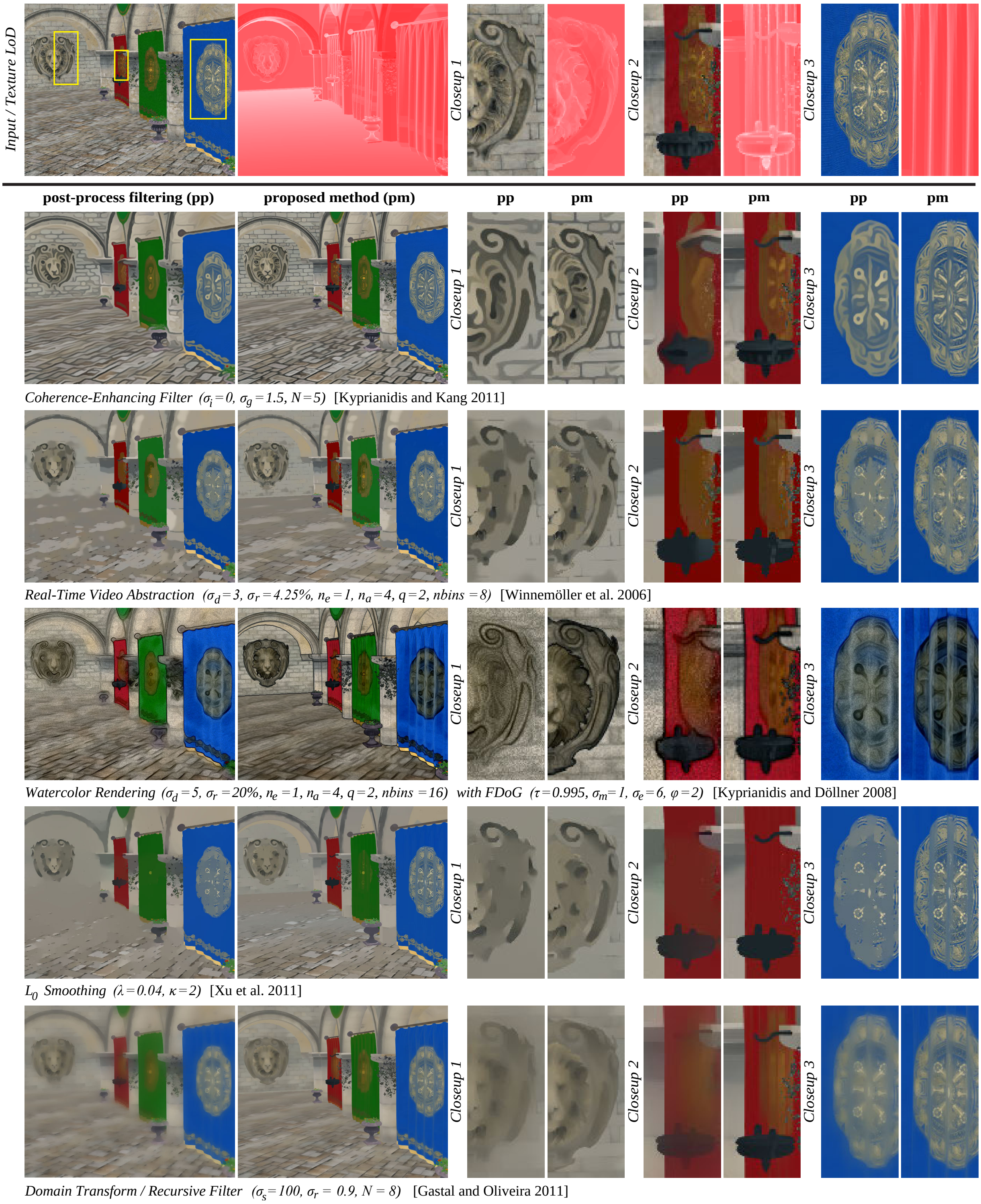
Original Data (level 1) Texturing Post-process Filtering

Texture-space Filtering Filtered Data (level 1) Texturing

Figure IV.4.: Comparison between post-process filtering (top row) and texture-space filtering (bottom row) using the FDoG filter (Kyprianidis & Döllner 2008), with mipmapping enabled for both approaches.

Figure IV.3 (scanline plots) and Figure IV.4 (bottom row) demonstrate that this approach naturally preserves texture gradients and object boundaries when trilinearly filtering the respective mipmap levels. While this is not a new approach in 3D computer graphics, e.g., used for pre-filtering shadow maps (Donnelly & Lauritzen 2006), artists use similar principles in their work to enhance the sensation of depth by adapting the detail and size of texture features with the linear perspective (Figure IV.2).

The framework that is contributed in this chapter implements Equation IV.2 with optimization techniques to enable real-time performance for local image filters. The framework uses conventional mipmapping and projective texturing capabilities of the hardware-accelerated rendering pipeline to inherently take the linear perspective, occlusions, and texture gradients into account. Using the approach with state-of-the-art edge-preserving image filters, a comparison between the filtering approaches (1) and (2) is performed. The results are presented in Figure IV.5 and Figure IV.6. The comparison was carried out qualitatively on the basis that texture gradients and occlusions induced by linear perspective are preserved. The results demonstrate how



Coherence-Enhancing Filter ($\sigma_i = 0$, $\sigma_g = 1.5$, $N = 5$) [Kyprianidis and Kang 2011]

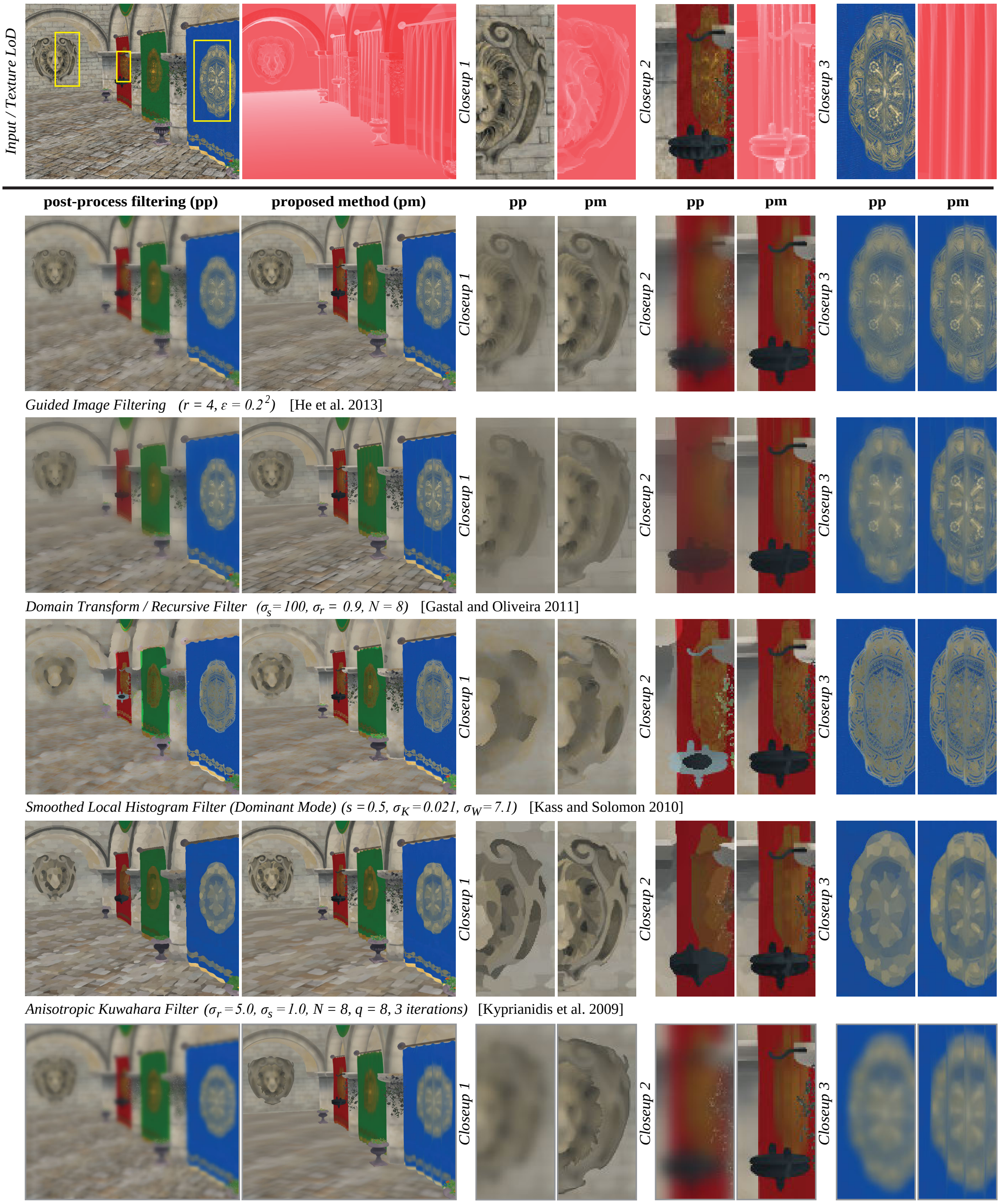
Real-Time Video Abstraction ($\sigma_d = 3$, $\sigma_r = 4.25\%$, $n_e = 1$, $n_a = 4$, $q = 2$, $nbins = 8$) [Winnemöller et al. 2006]

Watercolor Rendering ($\sigma_d = 5$, $\sigma_r = 20\%$, $n_e = 1$, $n_a = 4$, $q = 2$, $nbins = 16$) *with FDoG* ($\tau = 0.995$, $\sigma_m = 1$, $\sigma_e = 6$, $\varphi = 2$) [Kyprianidis and Döllner 2008]

L_0 Smoothing ($\lambda = 0.04$, $\kappa = 2$) [Xu et al. 2011]

Domain Transform / Recursive Filter ($\sigma_s = 100$, $\sigma_r = 0.9$, $N = 8$) [Gastal and Oliveira 2011]

Figure IV.5.: Image filters applied to a 3D scene using the proposed framework. The top row shows the original rendering output and texture LoD, the first column the results of post-process filtering (pp), and the second column the proposed method for LoA texturing (pm). The texture gradients on the floor, the textured banners, and the lion figure in the back are aggressively smoothed when filtered in a post-process stage, while the proposed approach preserves their structures and overall object borders without compromising the filters' qualities. Model: Sponza Atrium scene © Marko Dabrovic and Frank Meinl from Crytek, all rights reserved.



Guided Image Filtering ($r = 4, \varepsilon = 0.2^2$) [He et al. 2013]

Domain Transform / Recursive Filter ($\sigma_s = 100, \sigma_r = 0.9, N = 8$) [Gastal and Oliveira 2011]

Smoothed Local Histogram Filter (Dominant Mode) ($s = 0.5, \sigma_K = 0.021, \sigma_W = 7.1$) [Kass and Solomon 2010]

Anisotropic Kuwahara Filter ($\sigma_r = 5.0, \sigma_s = 1.0, N = 8, q = 8$, *3 iterations*) [Kyprianidis et al. 2009]

Gaussian Smoothing ($\sigma = 5.0$)

Figure IV.6.: Additional image filters applied to a 3D scene using the proposed framework (continued from Figure IV.5). Model: Sponza Atrium scene © Marko Dabrovic and Frank Meinl from Crytek, all rights reserved.

each filter's capability to preserve texture gradients and occlusions is improved, while the filters' qualities with respect to image smoothing or enhancing remain unaffected. For instance, the ringing artifacts of the red banner in Figure IV.5 are reduced with the coherence-enhancing filter (Kyprianidis 2011), the texture gradient on the floor and its transition to the back wall are preserved with bilateral filtering (Winnemöller *et al.* 2006) and L_0 gradient minimization (L. Xu *et al.* 2011), and the overall structures are emphasized with much higher precision.

Using texture-based methods for coherent stylization is not a new approach. For an overview on this topic, the interested reader is referred to the survey by Bénard *et al.* (2011). As reviewed in Chapter III, object-space methods reduced perspective distortions and scale variations via mipmapping, i.e., art maps (A. W. Klein *et al.* 2000; Praun *et al.* 2001), and infinite zoom mechanisms (Bénard *et al.* 2009) to maintain a quasi-constant size of texture elements for arbitrary view distances. Conceptually, the proposed approach behaves similar to the art maps approach but generalizes the idea of using mipmapping for coherent stylization with respect to image filtering in two ways: (1) instead of using pre-designed mip-map levels or relying on procedural modeling, the approach interactively filters high-detail texture information at run-time (e.g., photorealistic imagery), (2) the approach enables a user-defined LoA texturing (e.g., for focus+context visualization) that is not limited to color maps but also copes with multiple layers of textures used for visualization (e.g., normal maps, thematic maps).

Previous works used bilateral, DoG, and Kuwahara filtering with *G-buffer* information (Saito & T. Takahashi 1990) in a post-process stage to express uncertainty (Döllner & Kyprianidis 2010), direct a viewer's gaze to certain image regions (Redmond & Dingliana 2007; Redmond & Dingliana 2009b), and convey different emotional and experiential representations (Magdics *et al.* 2013). The proposed method also uses *G-buffer* information for deferred rendering, but decouples filtering from shading to preserve texture gradients and object boundaries when mapping the filtered results. Thereby, the framework implements a range of local filters to demonstrate how they can be used for real-time filtering of 3D scene contents for progressive LoA texturing. Because the framework is designed for generic application without requiring modifications of the original algorithms, future extensions are easy to implement. In particular, the DoG filter is used to enhance important edges based on both geometry and texture information. Thereby, the LoA texturing method produces accurate filtering results with respect to linear perspective to preserve texture gradients (Figure IV.1), performs in real-time, and provides frame-to-frame coherence.

2. Level-of-Abstraction Texturing

An overview of the framework is shown in Figure IV.7. It is designed for generic application, can be seamlessly integrated into existing 3D rendering systems and pipelines, is extensible for arbitrary 2D image filters, and has no particular requirements regarding the consistency of the input geometry—e.g., triangulated 3D meshes vs. 3D point clouds. Additional attributes—e.g., those encoding semantics—and textures for appearance and thematic information can be processed for content-based filtering and multitexturing, for instance using CityGML (Kolbe 2009) for 3D geospatial models.

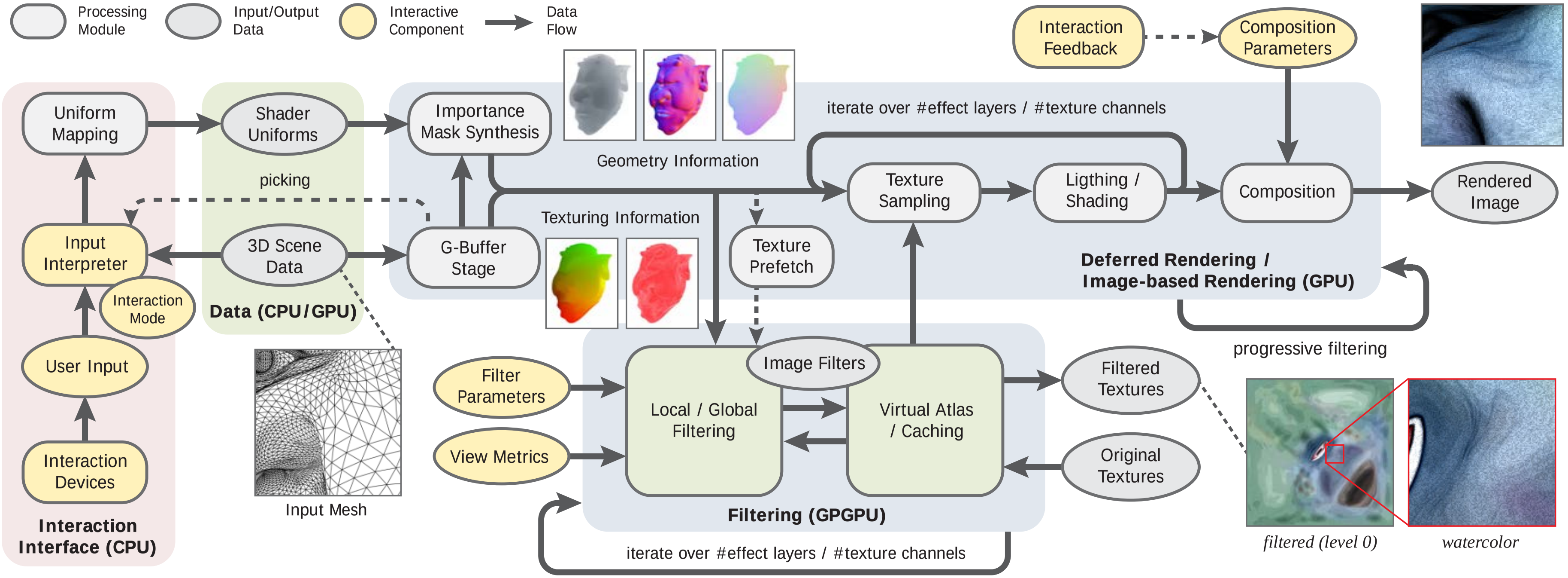


Figure IV.7.: Schematic overview of the framework designed for interactive LoA texturing. The approach decouples the interaction interface and image filtering from rendering to provide an extensible architecture. Further, it performs filtering prior to texture mapping to process texture data coherently with respect to the linear perspective, and to preserve spatial relationships. Model: Jerry the Ogre © Keenan Crane, all rights reserved.

The implementation performs filtering prior to texture mapping, for which decoupled deferred texturing is proposed. Filtering is performed using GPGPU separately on each mipmap level for LoA texturing. The implementation enables real-time performance for local image filters via per-fragment and progressive filtering using visibility information, together with a caching mechanism to ensure that image parts are only filtered once for a given configuration. Progressive filtering is based on a computational budget that is applied per rendered frame to ensure a responsible system during navigation or interaction. Parameter sets of image filters can be defined per texture channel—e.g., diffuse vs. normal maps—and may be used to define layers with different LoAs. These are blended according to view metrics or regions of interest for focus+context visualization. All geometry and texture buffers are represented as stencil-routed A-buffers (Myers & Bavoil 2007) to support an order-independent image blending of filtering effects, e.g., to obtain blueprint rendering styles (Nienhaus & Döllner 2004a). Optional post-processing may be performed in screen space and combined with the filtering results, e.g., to provide depth of field effects.

The approach can be parameterized at multiple stages to give explicit control over the filtering process, as well as implicitly by view metrics such as the viewing distance. In addition, it integrates an interaction framework to parameterize the filtering effects more directly and concisely, e.g., using interaction techniques such as (multi-)touch gestures or natural language interface to automatically parameterize magic lenses or highlighting techniques. Here, the main idea is to decouple the interaction interface from rendering, and parameterize uniform buffers or projected distance maps as an abstraction layer in-between. Interaction devices and techniques are interpreted according to a pre-configured interaction mode, and mapped to a functional description of focus+context definitions. These definitions are evaluated in the deferred rendering stage to dynamically compute importance masks and perform image filtering according to a user-defined stylization preset.

1. Decoupled Deferred Texturing

The goal is to provide high-quality, interactive LoA visualization of general, textured 3D scenes by image filtering that complies with the key aspects named in the beginning of Chapter IV. To this end, texture mapping is decoupled from the geometry pass and postponed by deferred texturing to transfer the filtering to texture space.

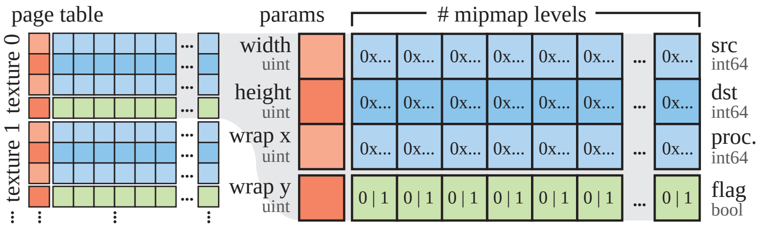
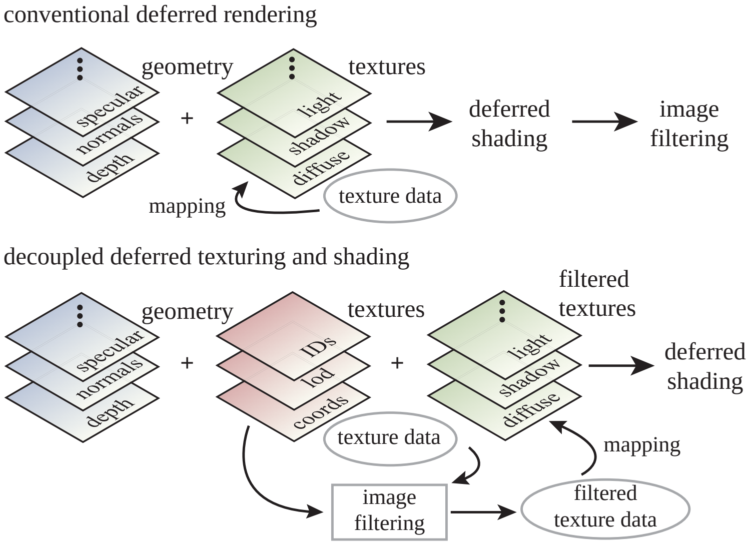


Figure IV.8.: Layout of the virtual page table used for texturing of original and filtered texture data on the GPU.

First, in a pre-process stage, textures are assigned unique identifiers $T_{\text{ID}} \in \mathbb{N}$. After computing the mipmap pyramids, a set of levels $(T_{m_0}, T_{m_1}, ..., T_{m_n})$ is defined per scene texture. The mipmap levels are transferred to GPU texture memory and referenced by their virtual address to enable *bindless texturing* with random read/write access. In addition, these virtual addresses are used for *virtual texturing* to dynamically resolve texture identifiers during processing.

A page table with a memory layout shown in Figure IV.8 is used. For each registered texture, this table references its dimensions, wrap modes required

Figure IV.9: Conventional deferred rendering performs texture mapping in the geometry stage (top). By contrast, the proposed decoupled deferred texturing is shown for perspective-coherent filtering (bottom).



to handle filtering across texture borders (e.g., clamped vs. mirrored), and the virtual address of each mipmap level. The total number of mipmaps is adapted to the maximum texture dimension that can be processed by the GPU. Afterwards, rendering is performed in a series of three stages:

R1 ***Geometry Stage:*** Conventional deferred rendering stores processed surface information in a *G-buffer* (Saito & T. Takahashi 1990), including texture sampling to synthesize diffuse, normal and thematic maps used for visualization. The proposed method, by contrast, computes additional buffers related to texture information: identifiers, coordinates, and mipmap LoD (Figure IV.9). Texture identifiers are assigned once in a pre-process stage and uniquely reference texture objects linked to virtual address space.

R2 ***Filtering Stage:*** Texture parts required for rendering are sampled and filtered using the *G-buffer* information of the first stage. The results are written to separate texture buffers and used together with the original texture data as input for shading.

R3 ***Shading Stage:*** The results of the geometry and filtering stages are used for *deferred texture mapping*. At this stage, the texture information stored in the *G-buffer* is used to easily toggle between original and filtered texture variants. Optional filtering in screen space may be performed in a post-processing stage, e.g., to provide edge detections for contour enhancement or Gaussian filtering for depth of field.

The following sections describe the filtering and shading stages in more detail.

2. Image Filtering

Once the *G-buffer* has been computed, the following three basic stages are performed for each render pass:

F1 ***G-buffer mapping:*** The *G-buffer* is mapped as an arrayed 2D graphics resource, where each pixel (x, y) within the dimension of the rendered image is mapped to the texturing information $(T_{\text{ID}}, T_{\text{coord}}, T_{\text{lod}})$.

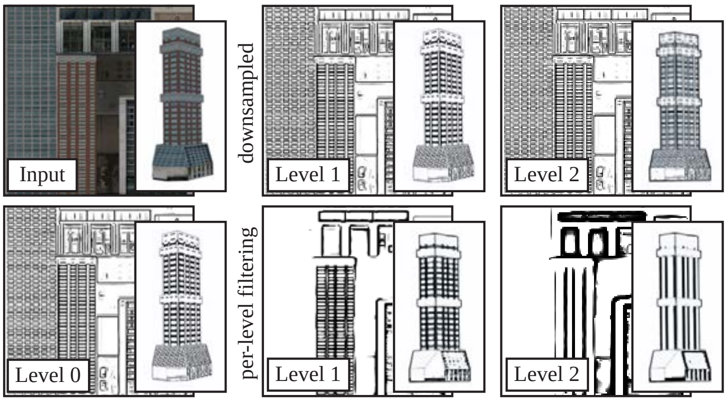


Figure IV.10: Example of LoA texturing by means of trilinear and DoG filtering.

(top) downsampled filtered input (using level 0 / bottom left)

(bottom) input downsampled and then filtered per mipmap level

F2 ***Texture prefetch:*** Relevant textures required for rendering are determined. The mipmap levels $(T_{m_i}, T_{m_{i+1}}) \leftarrow (T_{\mathrm{ID}}, T_{\mathrm{lod}})$ that correspond to each fragment in the *G-buffer* are computed in parallel and stored in a global structure using atomic operations. To avoid redundant filtering, a processing flag is set for each mipmap level in the virtual page table (Figure IV.8).

F3 ***Image filtering:*** Each image of the unique mipmap level T_{m_i} is filtered with the correspondent filter and configuration. Additional borders are introduced according to the wrap modes $(T_{\mathrm{wrapx}}, T_{\mathrm{wrapy}})$ to avoid visible seams when texturing. The results are written to the destination buffer and the processing flag is set.

Image filtering (F3) is performed separately on each mipmap level. Afterwards, the results are blended in the shading stage by trilinear filtering. Compared to filtering the highest mipmap level and downsampling the intermediate results, the approach enables a progressive LoA texturing (Figure IV.10). In this manner, visual clutter can be reduced significantly in areas of high perspective compression without requiring the adaption of filter parameters such as kernel sizes. Special care is required when mipmapped texture atlases are used for filtering to avoid bleeding across image tiles. Typically, this problem is addressed by introducing additional spaces between tiles, but it does not ultimately solve the problem for wide filter kernels or global operations. Virtual and bindless texturing alleviates this problem but requires using non-packed textures as input. An alternative approach may define tile masks per mipmap level for thresholding; however, this method increases the memory footprint.

3. Optimization Techniques and Enhancements

Dependent on the computational complexity of an image filter, the filtering stage (F3) could take significant processing time and stall the rendering stage. To this end, per-fragment filtering, caching, and progressive filtering are introduced for optimization.

Per-fragment Filtering. Because in most cases only small parts of textures are used for rendering and a lower texture LoD is selected in background regions of 3D perspective views, visibility-driven filtering allows for significant performance improvements, i.e., amortizing the computational overhead over a screen-space filtering. Therefore, per-fragment filtering is introduced by using the *G-buffer* to naturally process only the information required for rendering (Figure IV.11). This is achieved by integrating the *texture prefetch* in the filtering process, for which the texels required for trilinear filtering are determined for each synthesized fragment after rasterization. For trilinear filtering, this involves the four related texels used to perform bilinear filtering on each mipmap level (i.e., up to eight texels in total). The correspondent filtered values are then computed in parallel using a top-down filtering approach: information required for filtering is recursively resolved on-demand during processing.

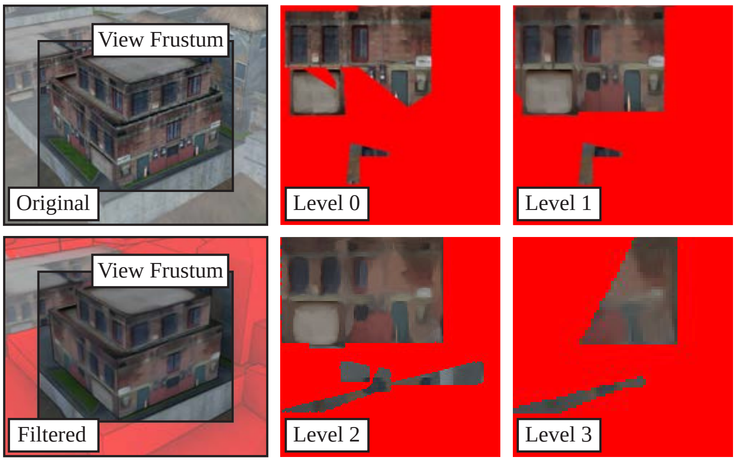


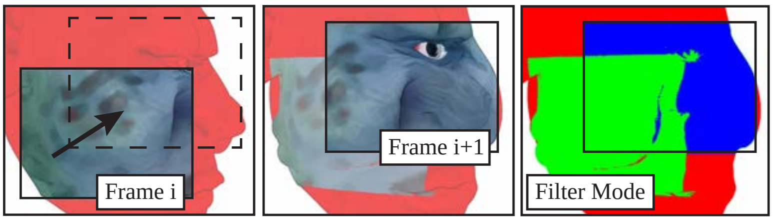
Figure IV.11.: Local filtering is performed only for visible texture parts as performance optimization, whereas unused texture parts are uninitialized (red). Geometry outside the view frustum (bottom left) is only textured for the reused parts.

Caching. A caching mechanism is used so that texture parts are only filtered once for a given configuration and reused for subsequent render frames (Figure IV.12). The virtual page table (Figure IV.8) references additional image masks $(T_{p_0}, T_{p_1}, ..., T_{p_n})$ per mipmap level that indicate whether a pixel has already been processed or needs further processing.

Progressive Filtering. Some local and many global filters that solve an optimization problem do not perform at interactive frame rates. Filtering the highest mipmap levels first and using them as fallback is used for progressive filtering. The proposed system uses a computational budget that is applied per rendered frame and can be interactively configured. Detail information is then progressively blended in subsequent render frames (Figure IV.13). This procedure also ensures a responsive system during interaction, e.g., when adapting filter parameters at run-time. In addition, it prepares an easy deployment on multi-GPU systems to decouple computationally expensive global filters from rendering.

The filtering kernel for both optimization techniques is summarized in Algorithm 1. Using these techniques, the computational cost of processing each texel over post-processing can be amortized, on the one hand, because pre-filtered texture information serves as input via mipmapping and, on the other hand, because the caching strategies avoid reprocessing. The performance evaluation—discussed in the following section—demonstrates that this behavior enables local image filters to process textured 3D scenes at real-time frame rates.

Figure IV.12: Filtered texture parts are cached (green), reused for subsequent frames, and combined with new filtered parts (blue) as performance optimization.



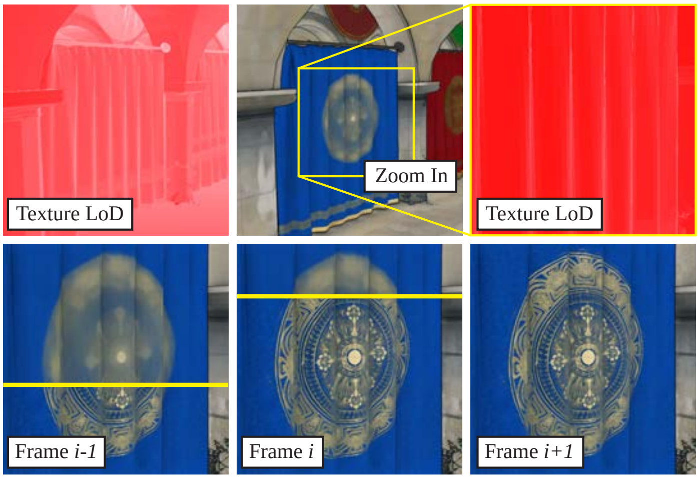


Figure IV.13: Example of a textured 3D scene that is progressively filtered using a computational filtering budget per render pass to maintain interactive frame rates. Model: Sponza Atrium scene © Marko Dabrovic and Frank Meinl from Crytek, all rights reserved.

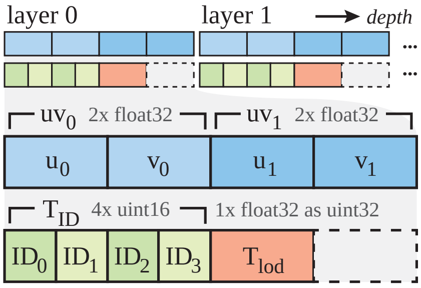


Figure IV.14.: Extended *G-buffer* used for depth buffering and multitexturing.

Multitexturing and Context-based Filtering. To enable filtering of multitextured 3D scenes, the *G-buffer* is enhanced as follows (Figure IV.14). First, each fragment synthesized by the rasterization is assigned two sets of texture coordinates together with multiple texture identifiers defined per texture channel. Filter configurations can then be defined per texture channel to enable a content-based filtering that is adapted to 3D model semantics. Second, fragments are buffered in depth and a sorting is performed in a post-process stage to enable order-independent transparency effects. For this, the G-buffer is extended to a stencil-routed A-buffer (Myers & Bavoil 2007) encoded as 3D texture for GPGPU processing. Depth sorting and image blending of the A-buffer is performed during shading. Third, different filters and configurations can be defined per input texture, for which the virtual texture table is extended by correspondent destination buffers. The filter results are then blended according to view metrics (e.g., viewing distance) or pre-defined regions of interest using image masks for focus+context visualization, e.g., to simulate the stylized focus pull effect of Cole *et al.* (2006). The number of texture coordinates and channels is not ultimately defined but should be limited to bound memory consumption. Because all processes are designed for generic application, the *G-buffer* can be extended by additional attributes.

4. Deferred Shading and Composition

Once the *G-buffer* is computed and filtering is performed, the results are used as input for texture sampling, which is independently performed from shading and lighting. Optionally, screen-space ambient occlusion (Shanmugam & Arikan 2007) and unsharp masking the depth buffer (Luft *et al.* 2006) can be performed using normal and depth information of the *G-buffer* to improve depth perception further. These effect layers can be individually amplified, colorized, and blended by regular image compositing (Porter & Duff 1984). Results that use DoG filtering are combined with the edge enhancement technique of Nienhaus & Döllner (2003) to include silhouette, border, and crease edges according to depth, normal, and object identifier information. In contrast to traditional DoG filtering in screen space, the proposed method is able to decouple edges based on texture and

Algorithm 1: *Per-fragment filter kernel for texture data*

```
function local_image_filtering: begin
    Input: G-buffer G, texture page table P with color lookup $P_\mathcal{C}$ and process flag lookup
           $P_\mathcal{F}$, filtering budget B
    k ← 0                                              /* global number of texels filtered */
    for pixels p ∈ G do in parallel
        (ID, lod, u, v) ← G (p)                                         /* sample G-buffer */
        if ID = 0 then                                                         /* early out */
            return
        $(T_0, T_1)$ ← P (ID, ⌊ lod ⌋ and ⌈ lod ⌉)                              /* mipmap LoDs */
        forall $(T, u_\mathcal{S}, v_\mathcal{S})$ of textureGather $(T_0)$
        and     $(T, u_\mathcal{S}, v_\mathcal{S})$ of textureGather $(T_1)$ do          /* process max. 8 samples */
            if P $(T, u_\mathcal{S}, v_\mathcal{S})$ not marked as processed then       /* start of critical
             section */
                if k < B then                                            /* threshold budget */
                    $P_\mathcal{F}$ $(T, u_\mathcal{S}, v_\mathcal{S})$ ← mark as processed
                    $P_\mathcal{C}$ $(T, u_\mathcal{S}, v_\mathcal{S})$ ← filtering $(T, u_\mathcal{S}, v_\mathcal{S})$     /* assign new color */
                    k ← k + 1                          /* increase number of filtered texels */
                else   /* progressive filtering: lookup in other mipmap level */
                    $P_\mathcal{C}$ $(T, u_\mathcal{S}, v_\mathcal{S})$ ← lookup (ID, lod, u, v)
                end
            /* end of critical section                                                    */
        end
    end
end
```

geometry information. Figure IV.15 demonstrates that this approach produces more accurate filtering results with respect to linear perspective. In addition, it provides real-time frame rates and frame-to-frame coherence without requiring specialized methods such as *texture advection* (Bénard *et al.* 2011).

5. Interaction Interface

The proposed framework gives control over the different parameters defined per image filter, including kernel size, quantization intensity, sensitivity of edge detection, and weights of flow fields (Figure IV.16). In addition, users are able to define parameters to adjust the filter composition:

- texture channel semantics with filters and configurations defined per channel for content-based LoA texturing;

- view metrics and region masks together with transition parameters used for blending layered filtering effects;

- colors for enhanced geometry edges, screen-space ambient occlusion, and unsharp masking effects;

- multi-sampling and order-independent transparency parameters used for image composition.

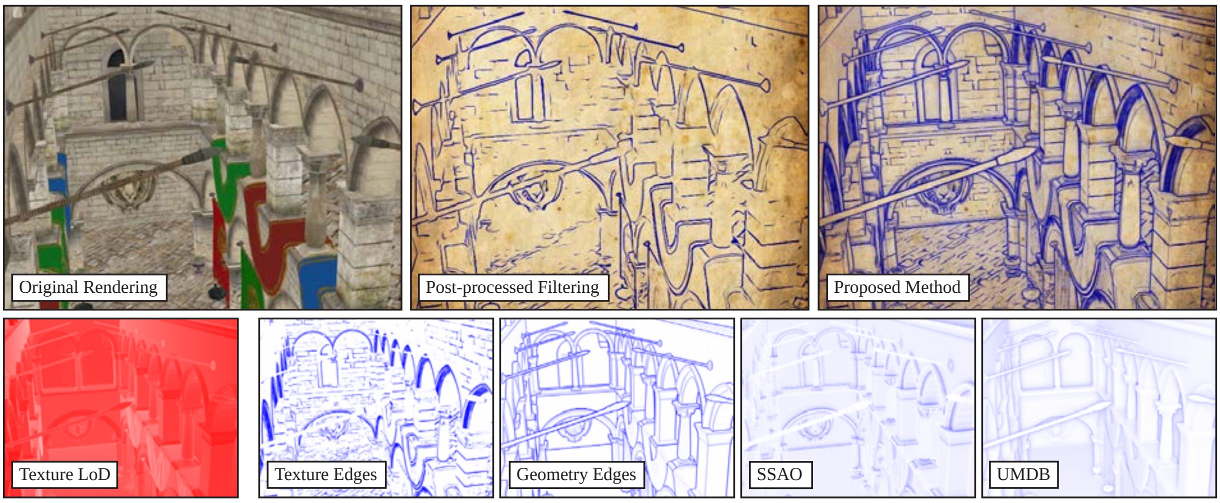
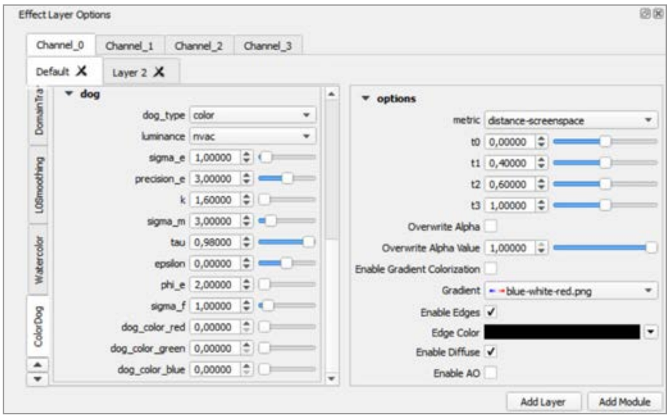


Figure IV.15.: Example of post-processing effects used in the framework. Edge detection is decoupled into DoG filtering of texture and geometry information. The results are combined with screen-space ambient occlusion (SSAO), unsharp masking the depth buffer (UMDB), and a background texture to compose the final images.



dog

dog_type color
luminance nvac
sigma_e 1,00000
precision_e 3,00000
k 1,60000
sigma_m 3,00000
tau 0,98000
epsilon 0,00000
phi_e 2,00000
sigma_f 1,00000
dog_color_red 0,00000
dog_color_green 0,00000
dog_color_blue 0,00000

options

metric distance-screenspace
t0 0,00000
t1 0,40000
t2 0,60000
t3 1,00000
Overwrite Alpha □
Overwrite Alpha Value 1,00000
Enable Gradient Colorization □
Gradient blue-white-red.png
Enable Edges ✓
Edge Color ■
Enable Diffuse ✓
Enable AO □

Figure IV.16.: Excerpt of the graphical user interface for the parameterization of image filters (left) and effect layers (right), here exemplified for the FDoG filter (Kyprianidis & Döllner 2008).

For the practical usage of the framework in real-world scenarios, in particular for non-expert users, however, adjusting these parameters can be cumbersome and such cases thus call for more high-level and direct interaction interfaces. For effective visualization of 3D scenes, direct user interaction is a critical design aspect to visualize as much information as needed for focus and as little as required for context. Four major interface schemes have been identified for focused and contextual views (Cockburn *et al.* 2009): zooming, focus+context, overview+detail, and cue techniques. Using these schemes to parameterize LoA texturing in 3D geovirtual environments, however, remains a challenging task, because these environments are often inherently complex with respect to appearance and thematic information. Here, a major goal is to provide an interaction framework that seamlessly integrates the proposed method for LoA texturing, is extensible for custom interaction devices and techniques, and considers the following challenges:

1. The filtering and rendering stages should be decoupled from concrete interaction interfaces, and be parameterized via high-level operations to facilitate an easy deployment of new interaction devices and techniques.

2. Interactive frame rates should be maintained to provide a responsive system to the user.

3. Mapping of direct interaction from 2D into 3D space should be handled, e.g., with respect to occlusions (Isenberg 2013).

In the following, a generic workflow is defined on how user interaction can be mapped to definitions of focus and context, parameterized by LoA texturing.

Focus+Context Definition and Interaction Techniques

One challenge for the design of an interaction framework is to strive for consistency (Shneiderman *et al.* 2009) while having the user in control of parameterizing the LoA texturing. Here, a key observation is that no constraints regarding the input device or technique should be made to allow users to choose an adequate interaction method when parameterizing the framework for a given task and environment (e.g., desktop vs. mobile devices). The main idea is to decouple the functional focus and context descriptions from the interaction device and filter configurations, e.g., so that mouse/keyboard, touch-based, natural language, or implicit gaze-based interfaces can be equally used to define regions of interest.

Technically, the interaction interpretation is formulated as mapping the user-defined input to a high-level functional description for focus and context definition. Interaction modes are required for disambiguation and to avoid redundant mappings, but should be made as concise as possible, e.g., using quasi-modes (Raskin 2000). Exemplary mappings include object selections by textual lookup using natural language inputs or *point-and-click* metaphors. Similarly, circular regions of interest may be defined via *pinch-to-zoom* metaphors or sketching, e.g., to parameterize illustrative cutaway renderings (Knödel *et al.* 2009). Technically, high-level descriptions can be mapped from parameter space into model space using logical collections of 3D scene data as input, enriched with descriptive information encoded as attributes, e.g., with CityGML for geospatial models (Kolbe 2009).

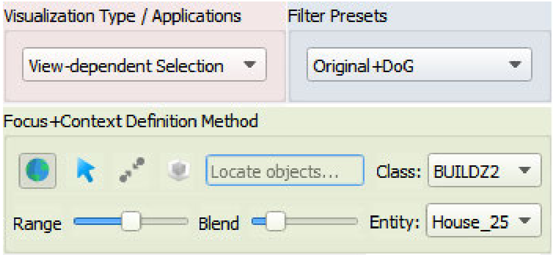

Visualization Type / Applications: View-dependent Selection
Filter Presets: Original+DoG
Focus+Context Definition Method
Locate objects…
Class: BUILDZ2
Entity: House_25

Figure IV.17.: Interaction interface and tools used in the prototypical implementation of the framework to parameterize a focus+context visualization.

Chapter II distinguishes between six types of focus definition for 3D geovirtual environments (Figure II.3). Direct interaction for these types should trigger immediate visual feedback to symbolize a correspondent mode, for which specialized shading effects are provided. For instance, the boundaries of a circular region of interest could be visualized using projected lines as visual cues. Figure IV.17 gives an overview of the interaction interface implemented in the framework that resembles some of the focus+context definition types shown in Figure II.3. Besides techniques for orbital, zoom and pan navigation, this interface is (1) context-aware according to a user-defined application (e.g., highlighting regions of interest), (2) allows users to select filter configurations from pre-defined presets, and (3) provides techniques for interactive focus+context definition. The latter includes regional definitions via the *pinch-to-zoom metaphor*, highlighting of objects or regions of interest via a search bar, value-range sliders for numerical constraints, and on-screen pointing techniques for direct object selection. In addition, optional interaction techniques may be used for parameterization, e.g., using on-screen pointing or textual input to locate and highlight 3D scene objects.

Shader Uniform Mapping

In the framework, focus and context definitions are either stored using GPU uniform buffers or mapped to distance maps (Figure IV.18), whose parameters are evaluated during shading. In the first case, these attributes are compared with the information synthesized in the *G-buffer* (e.g., object identifiers, 3D world

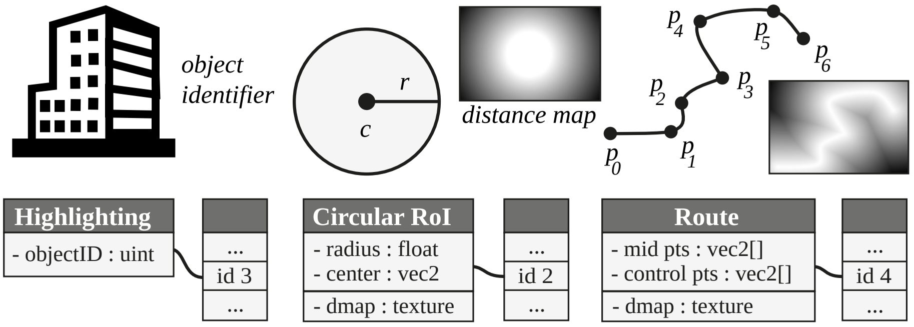


Figure IV.18: Focus definitions are mapped to uniform variables or are buffered as projected distance maps, which are evaluated during shading.

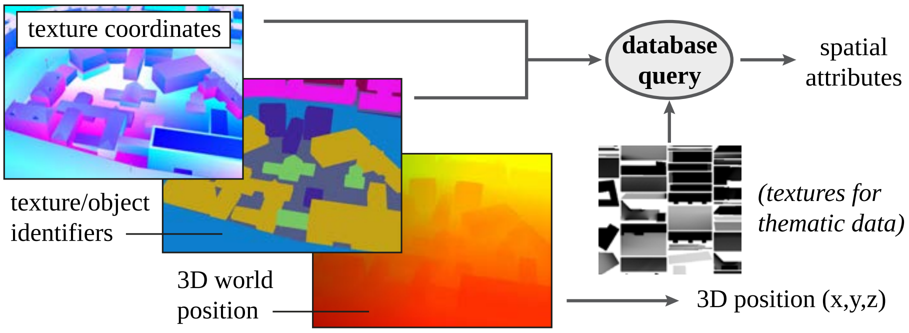


Figure IV.19: Texture coordinates, identifiers, and positions synthesized in a *G-buffer*. The buffer information can be used to query additional attributes, which may be stored in a database in main memory.

position) for focus detection. In the second case, a Euclidean distance transform is performed to buffer line segments or 2D regions of interest (Danielsson 1980). The synthesized distance maps are then projected onto the scene geometry and evaluated during shading. This approach enables image-based operations to be efficiently implemented such as fragment-based thresholding of the Euclidean distance between objects, e.g., the buffered distance to a route (Figure IV.18).

For many interaction modes, a basic function is to map 2D inputs to 3D attributes via raycasting, e.g., using touch interfaces for object selection (Jankowski & Hachet 2014). It is typically performed using intersection tests with the 3D scene geometry, an approach that does note perform at interactive frame rates. Instead, an image-based approach based on the synthesized *G-buffer* information is used to query 3D scene attributes—also referred to as *picking*. Here, it is observed that the synthesized world position, texture coordinates, and identifier information (i.e., objects, textures) are sufficient to query arbitrary attributes stored in a database (Figure IV.19). For instance, texture identifiers and coordinates directly map into texture space for a fragment-based information lookup, whereas object identifiers are mapped to object-specific attributes.

Importance Mask Synthesis and Shading

The shader uniforms are evaluated using fragment shaders. For each definition type, a normalized importance mask is synthesized that indicates whether a fragment should be shaded for focus or context (Figure IV.20). Blend functions are utilized for image composition (Porter & Duff 1984) and to enable smooth transitions between focus and context regions, but may also be configured for hard transitions, e.g., to avoid distorted color tones when using heterogeneous image filters. Finally, the importance masks are blended to enable multivariate effects, e.g., a route with a circular region of interest at the destination.

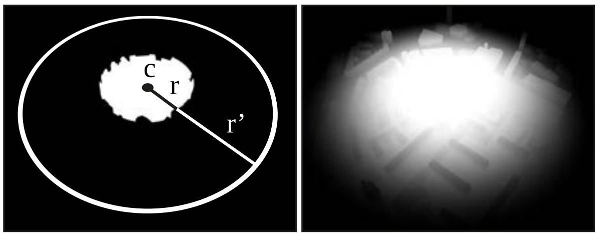


Figure IV.20.: Example of a synthesized importance mask for a circular region of interest.

Presets of image filter configurations are supplied to enable an automated LoA texturing setup. For instance, a blueprint rendering may be automatically selected to represent construction sites in an urban planning scenario. Post-processing the synthesized importance masks with a screen-space distance transform may also be used for view-dependent highlighting, e.g., to obtain glow effects (Trapp *et al.* 2010). Results using these interactive techniques for LoA texturing are presented in the next section.

3. Results and Discussion

This section presents and discusses applications, evaluations, and limitations of the proposed methods.

1. Implementation

The framework is implemented using C++, OpenGL, and GLSL, and the image filters on the GPU with CUDA. OpenSceneGraph[1] is used as the rendering engine to handle 3D data sets. For global image processing, CUDA's *fast Fourier transform (cuFFT)* is used to implement the L_0 gradient minimization (L. Xu *et al.* 2011), the Deriche method (Deriche 1993) to perform Gaussian smoothing in constant time per input pixel, and recursive methods proposed by Nehab *et al.* (2011) for efficient box filtering. The image filters operate in RGB or the perceptual CIE-Lab color space. The implementation uses texture and surface objects introduced in CUDA 5 for virtual texturing along with the *OpenGL Interop* functionality for random read and write access. It is assumed that similar results are achievable using OpenCL or compute shaders.

For bindless texturing, the OpenGL 4.4 extension `GL_ARB_bindless_texture` is used. The *G-buffer* is packed to RGBA or RGB texture channels and is encoded as a stencil-routed A-buffer (Myers & Bavoil 2007) for order-independent image blending. The parallel-banding algorithm (Cao *et al.* 2010) is used to perform work-load efficient distance transforms and to compute importance masks for regions of interest. All results were rendered on an Intel® Xeon™ 4× 3.06 GHz with 6 GByte RAM and NVidia® GTX 760 GPU with 4 GByte VRAM. In addition, user interaction was tested with a 23.6" Lenovo® L2461xwa multitouch monitor.

2. Applications

First of all, the individual applications of the respective image filters, e.g., HDR tone mapping, detail exaggeration, edge adjustment, or colorization, can be maintained because they are merely transferred to the texture domain. For instance, Figure IV.21 shows a result of the proposed LoA texturing method using L_0 gradient minimization (L. Xu *et al.* 2011) for perspective-coherent denoising of texture-encoded thematic data and overall contrast enhancement. Here, solar potential was computed by a radiation summed up over a whole year and encoded in thematic 3D scene textures.

[1] OpenSceneGraph — http://www.openscenegraph.org (Last followed on November 20, 2016)

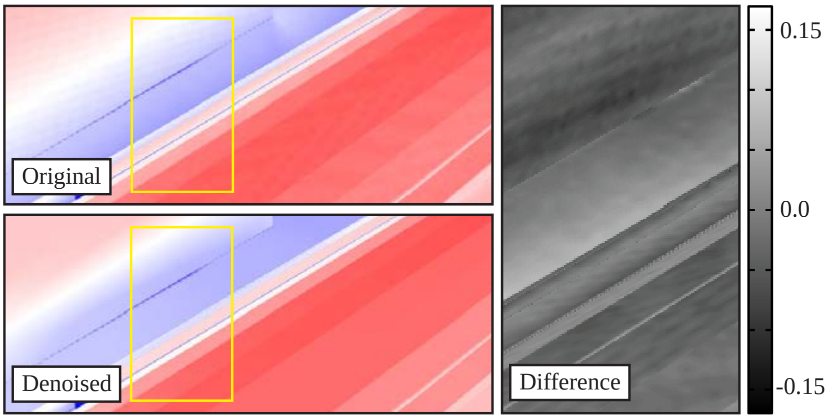


Figure IV.21: Edge-preserving denoising of texture-encoded thematic data via L_0 gradient minimization (L. Xu *et al.* 2011) with parameters $\lambda = 0.02, \kappa = 2$. The (filtered) texture data is then projected onto the surface geometries. The difference for the visualized outputs is normalized and computed in RGB color space.

Focus+Context Visualization

Highlighting regions of interest while removing detail in context regions is a major goal in the effective information transfer and for directing a viewer's gaze to important or prioritized information (DeCarlo & Santella 2002; Santella & DeCarlo 2004). Figure IV.23 shows a result of the framework to implement a *stylized focus pull* effect (Cole *et al.* 2006) based on the view distance. Emphasis is drawn to the respective image regions using the FDoG filter coupled with image-based enhancements of geometry edges for context regions. Thereby, the proposed method is able to preserve the overall structure of objects in the background and in the context regions—e.g., windows on the building façades to emphasize geometry-related information that is encoded in textures. In addition, the scale of the outlined geometric features is drawn with respect to perspective foreshortening, and thus is able to maintain geospatial relationships.

Because the image composition is performed in the deferred shading stage and multiple layer effects may be defined, the proposed LoA texturing is generic with respect to filter combinations and extensible for further view metrics (e.g., region masks or view angles). For demonstration, the *pinch-to-zoom* metaphor is coupled with the interaction framework to directly change the graphical representation in regions of interest. Based on a multitouch interface, the user starts pointing with two fingers and spans—via the zoom metaphor—the range of interest in the projected world space. A user-defined preset for image filtering then automatically adjusts the LoA texturing in the focus region, providing optional smooth transitions controlled via a blend slider. Alternatively, a natural language interface may be used to locate and highlight regions of interest via a search bar. Figure IV.22 exemplifies a semantic lens that automatically blends a detailed 3D model for focus with a map for context. Here, the *apparent greyscale* algorithm of Smith *et al.* (2008) is used for filtering the context regions.

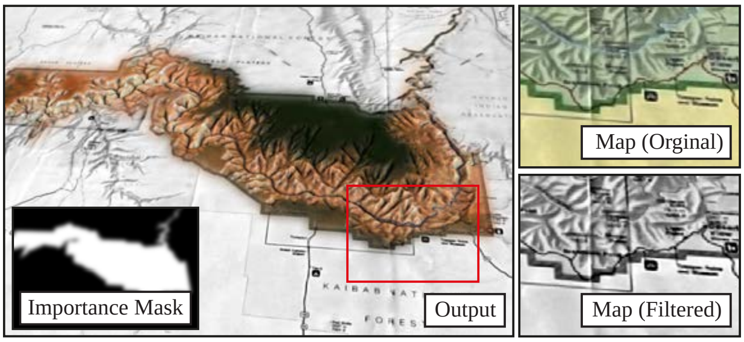


Figure IV.22.: Semantic highlighting using a 3D terrain model for focus and a 2D map for context (*Grand Canyon National Park*). The map texture is filtered using the *apparent greyscale* algorithm of Smith *et al.* (2008) with parameters $p = 0.5, k = \{0.5, 0.5, 0.5, 0.5\}$.

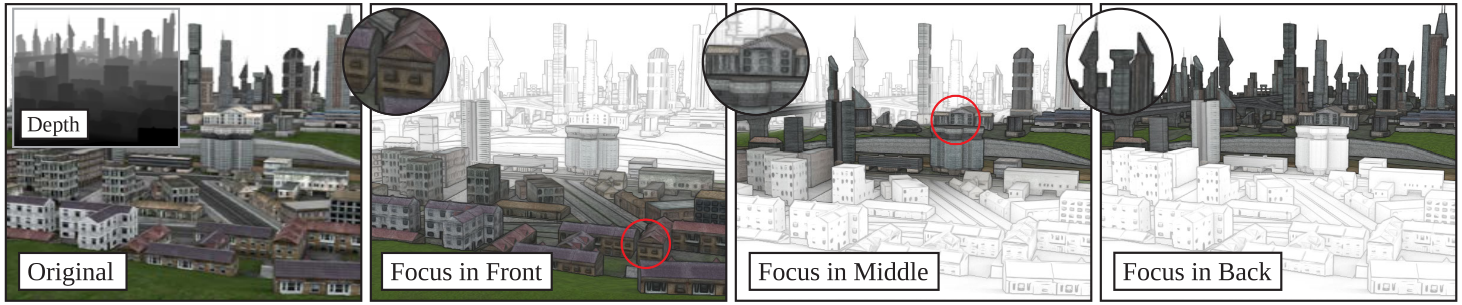


Figure IV.23.: Example of a stylized focus pull where the distance-based focal plane is gradually directed from the foreground to the background. A watercolor and DoG filter are used to draw emphasis to regions of interest and preserve depth cues in context regions (e.g., building windows).

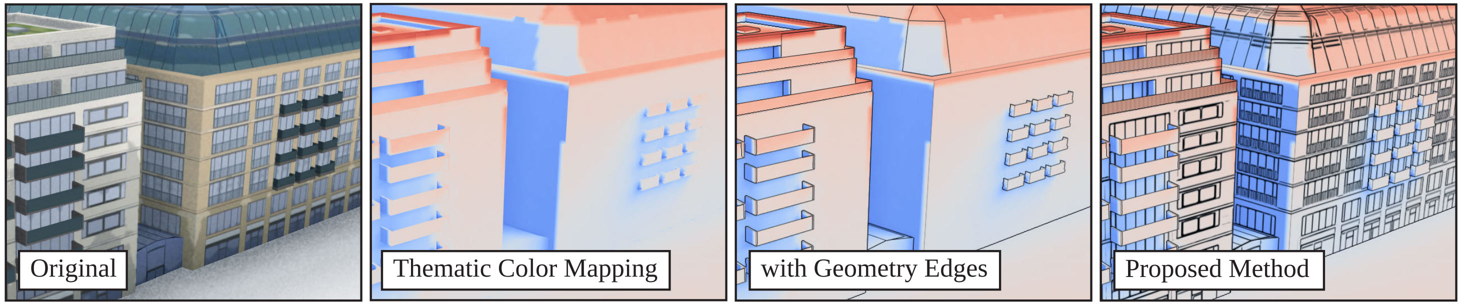


Figure IV.24.: An example for thematic data visualization, where the framework is used to combine geometry edges and DoG filtered color textures and, hence, make the correlation between thematic information (here: solar potential) and geospatial features plausible.

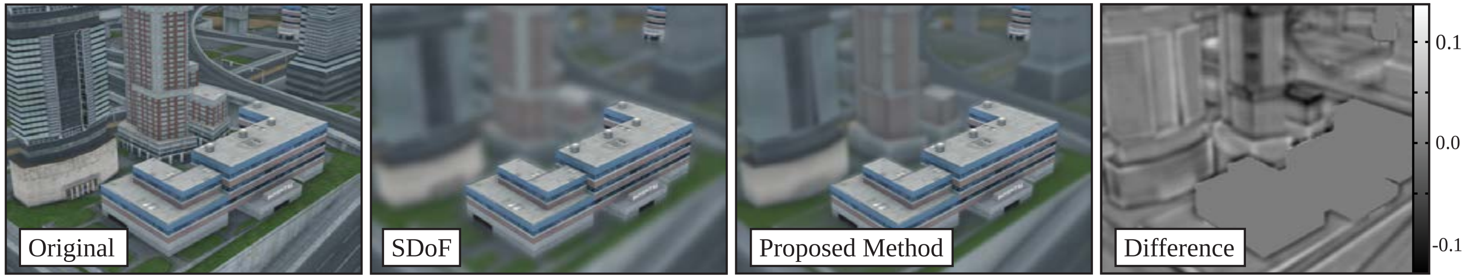


Figure IV.25.: Extended semantic depth of field effect. LoA texturing is used for perspective-coherent Gaussian smoothing of color maps ($\sigma_t = 3.5$) prior to additional smoothing in screen space ($\sigma_s = 3.5$). Compared to the traditional approach, which only filters in screen space ($\sigma_s = 7.5$), the proposed method preserves scene structures in context regions better at a similar LoA.

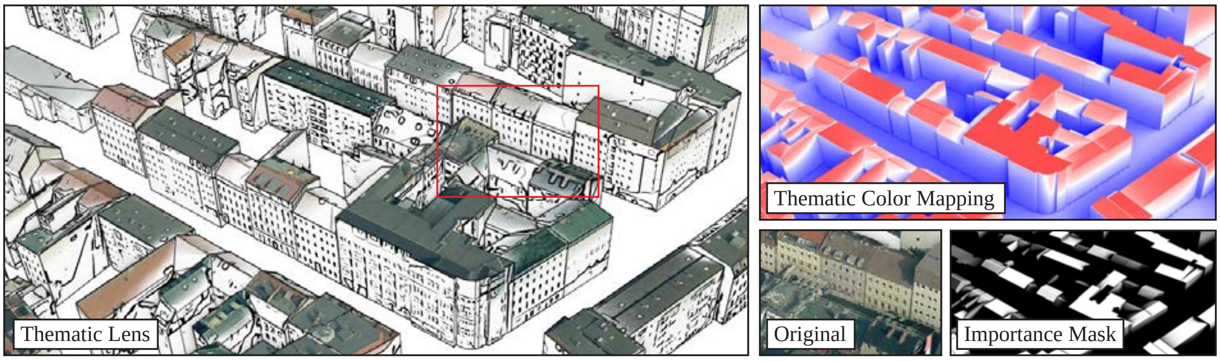


Figure IV.26.: Interactive selection of "high solar potential data" using the framework to direct the filtering process to context regions. The user is able to control the focus and context definition via a value-range slider together with a blend slider to provide smooth transitions between both definitions.

Visualization of Thematic Data

The multitexturing support is used to enhance depth cues important for visualizing color-encoded thematic data in virtual 3D scenes. Figure IV.24 shows a parameterization in which the FDoG filter is applied to detect edges in the color textures and blend the result with the thematic information. Compared to an edge detection that only processes geometry information, the proposed approach also enhances structural information that is not explicitly modeled as geometry, but is captured by aerial or terrestrial imaging. This way, the correlation between thematic data and surface structures is much more plausible. Further approaches deal with directly parameterizing a thematic visualization for information highlighting. Here, the implemented slider interface for value-range thresholding is used to parameterize the range of interest for thematic data. Figure IV.26 shows a result, where areas with a high solar potential are visualized with detailed graphics, and image regions with a low solar potential are automatically filtered, providing smooth transitions in-between. At this example, a user is able to interactively adjust the threshold for the normalized thematic value that defines when photorealistic or abstract textures are used for visualization.

Semantic Depth of Field

Depth of field is able to direct the pre-attentive cognition of prioritized information. Using LoA texturing for visualization, a variant of the *semantic depth of field* (SDoF) effect (Kosara *et al.* 2001) is implemented, in which scene objects are selectively highlighted via image masks to direct a Gaussian smoothing in screen space. First, diffuse maps are filtered in texture space, i.e., to control the LoA of textured surfaces. Afterwards, a regular Gaussian smoothing is performed in screen space to blur geometry edges.

As shown in Figure IV.25, using this "dual" filtering approach enables a clearer visualization of structures induced by geometry edges. By contrast, the regular screen-space approach requires wider filter kernels to achieve a similar LoA effect but at the cost of a considerably high degradation of scene structures. The implemented search interface can be used to automatically parameterize the SDoF effect for object highlighting.

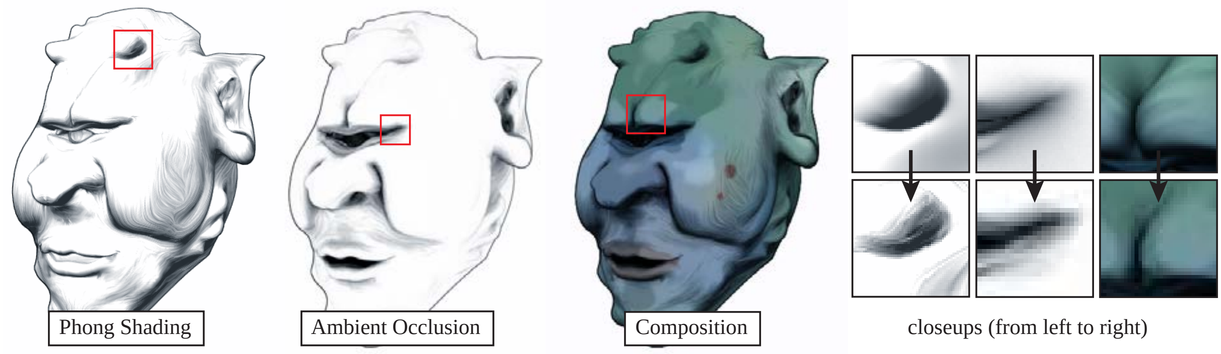


Figure IV.28.: Stylization of Phong shading, ambient occlusion, and color maps. An oil paint filter is used for the normal and ambient occlusion maps, together with the abstraction filter of Winnemöller *et al.* (2006) for color maps.

Geometric Detail Removal

Bump or displacement mapping is an essential process for enriching shading and lighting effects by geometric detail. The proposed methods were used to perform edge-preserving smoothing of normal maps to coarse bump mapping with respect to the linear perspective. Figure IV.27 shows results of a domain transform (Gastal & Oliveira 2011) applied to normal maps, where mipmapping enables a smooth transition between the different levels of structural abstraction.

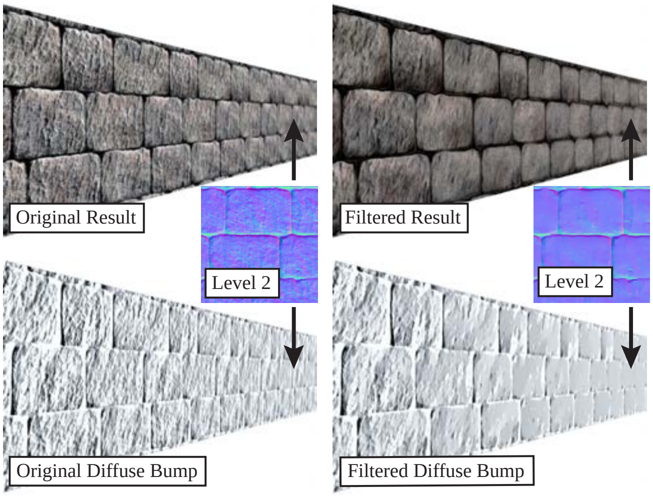


Figure IV.27.: Smoothed bump mapping using a watercolor filter and domain transform (recursive mode, $\sigma_s = 30.0, \sigma_r = 0.25, N = 3$) for color and normal maps.

Stylized Shading

Further results relate to filtering normal and ambient occlusion maps to obtain stylized shading and lighting effects. The rich parameterization options of the framework give users creative control over this process. For instance, Figure IV.28 shows how an oil paint filter—based on a smoothed structure tensor—was used to apply Phong shading and ambient occlusion with a sketchy style. Similar directions were proposed by DeCoro *et al.* (2007) for stylized shadows and toon shading by Barla *et al.* (2006) for general LoA; however, without the capability for flow-based image abstraction. By contrast, the proposed approach provides stylized variants of texture maps that includes salient structures, which are blended by conventional shading. Hence, it is especially useful to interactively stylize photorealistic texture maps (e.g., captured by terrestrial photography).

Blueprint Rendering

The capability to layer filtering in depth was used to provide a novel blueprint rendering approach. In contrast to image-based techniques that solely operate on *G-buffer* information (Nienhaus & Döllner 2004a), a DoG filter is used for perspective-coherent abstraction of diffuse maps that preserves texture gradients (Figure IV.29 left). The filtering results are colorized and blended

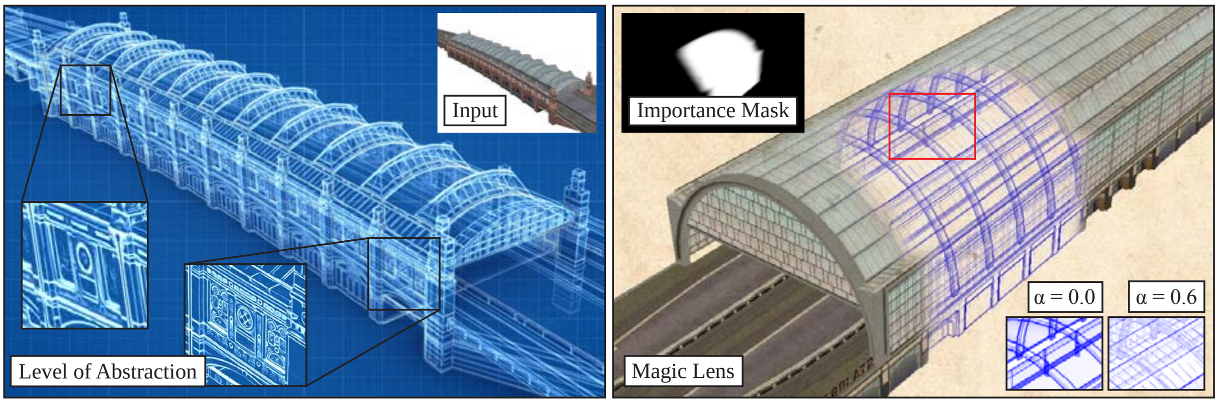


Figure IV.29.: Blueprint rendering based on geometry edges and FDoG filtered color maps. The filtering is performed on each depth layer, the results are then sorted in depth and blended in a post-process stage. **(Left)** : LoA capabilities of the proposed method, where the close-ups represent the first depth layer. **(Right)** : A magic lens parameterized via a touch interface to direct a blueprint rendering to a region of interest and to highlight interior structures.

by order-independent transparency. Using the interaction framework, interiors of 3D models model can be made interactively explorable via the magic lens metaphor (E. A. Bier *et al.* 1993; Viega *et al.* 1996). A user is able to span regions of interest via direct touch interaction and shift the lens to a desired location. The focus area may then be visualized with the blueprint rendering approach (Figure IV.29 right). The proposed method provides frame-to-frame coherence—also for the previous applications—primarily because the respective mipmap levels are processed with respect to perspective projection, *prior* to trilinearly filtering the results on the GPU for smooth interpolation (Equation IV.2).

3. Evaluation

This section provides evaluations of the proposed LoA texturing methods with respect to thematic color mappings, visual saliency, and rendering performance.

Visual Saliency Analysis

Saliency maps of the outputs, obtained with the methods for focus+context visualization, are compared with a high-detail rendering in Figure IV.30. Here, the *graph-based visual saliency* model of Harel *et al.* (2007) is used to indicate the benefits of the proposed LoA texturing for directing a viewer's gaze. The saliency model predicts human fixations more reliably than the classical approach of Itti *et al.* (1998)—"achieving 98% of the ROC area of a human-based control" (Harel *et al.* 2007)—using an organic approach based on Markov chains. Figure IV.30 shows how the visual saliency of the stylized focus pull effect shown in Figure IV.23

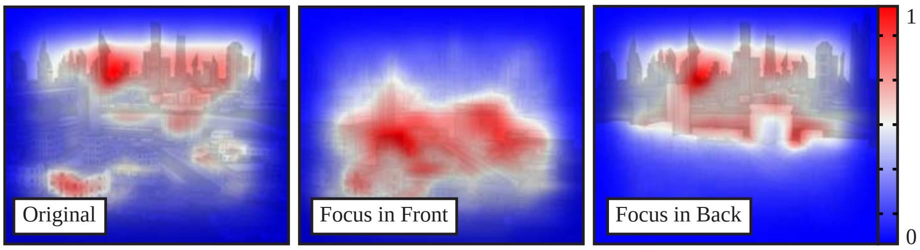
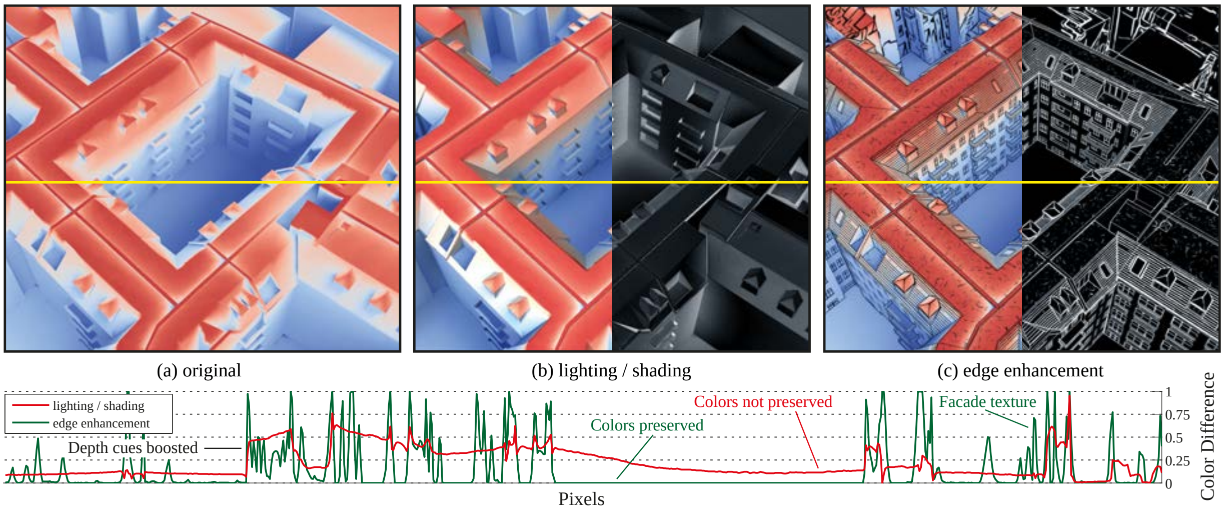


Figure IV.30: Visual saliency analysis for Figure IV.23, compared to the respective photorealistic version, using the algorithm of Harel *et al.* (2007).



(a) original

(b) lighting / shading

(c) edge enhancement

Figure IV.31.: Visualization of thematic data in a 3D geovirtual environment using a diverging blue-red color scheme: (a) original output, (b) visualization enhanced by Blinn-Phong shading and screen-space ambient occlusion to include depth cues, (c) visualization enhanced by the proposed LoA texturing approach for edge enhancement in image space and texture space. Euclidean RGB color differences of (b) and (c) to (a) are illustrated respectively; the scanline plots illustrate the effect of boosted depth cues but altered color information along the yellow lines. Compared to shading and lighting, the proposed method alters information only at feature boundaries.

shifts to the respective focus regions, contrary to the original rendering with detailed textures. Similar results have been qualitatively verified by Santella & DeCarlo (2004) and Cole *et al.* (2006) but using an eye tracker for evaluation.

User Study on the Visualization of Thematic Data

Thematic data, encoded as surface data in textures, represents an integral part of geospatial data along geometry, topology, semantics, and appearance. Using colors for thematic mapping is a fundamental approach in visualization, and is essential for 3D geovirtual environments to communicate multidimensional information. For instance, the results of a solar potential (Engel & Döllner 2010), crime (Wolff & Asche 2009), or heat transmission analysis may be mapped to color themes found in cartography (A. M. MacEachren *et al.* 2004).

Human eyes project 3D virtual environments onto two-dimensional retinas, where depth sensation is generated by subconscious interpretation of depth cues. A variety of rendering techniques exist to enhance pictorial (monocular) depth cues such as shading and global illumination (Akenine-Möller & Haines 2008). Combined with color-encoded thematic information, however, these techniques alter color information (Figure IV.31), thus, colors may not (or ambiguously) be mapped to thematic information, which typically causes additional mental effort. Moreover, applying thematic textures may omit structural information that is not explicitly modeled as geometry but captured by aerial or terrestrial imaging (e.g., building façades). Yet, this information is often essential for landmark identification and mental modeling (Lynch 1960), but neither can be enhanced by illumination techniques as no geometry is involved, nor by image blending as it would distort the thematic and appearance information.

Several questions arise when using rendering techniques with thematic color mappings to enhance depth cues in 3D geovirtual environments, in particular if they alter the perception of the visualized thematic data and to which degree

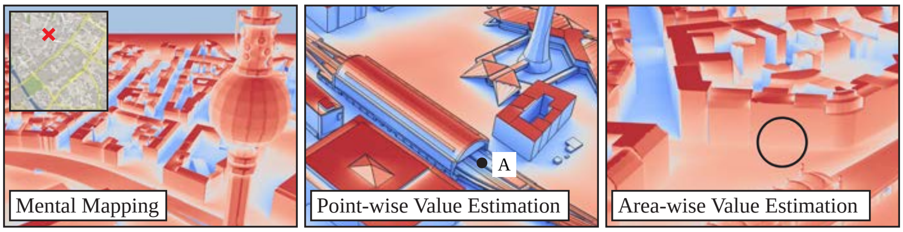


Figure IV.32: Exemplary trials for the tasks that the participants had to carry out in the study.

edge enhancement techniques, including the proposed LoA texturing methods, improve the orientation in 3D and creation of mental maps. The joint user study of Engel *et al.* (2013), which utilizes the proposed LoA texturing techniques for visualization, addresses these questions. Participants performed perception-related tasks (Figure IV.32) in a thematic visualization scenario of a virtual 3D city model with different rendering techniques and color mappings.

Task 1 – "Point-wise value estimation": The participants had to estimate the thematic data value at a point of interest. A scale from 0 to 100 percent was shown at the bottom of the 3D view as orientation guidance.

Task 2 – "Area-wise value estimation": Similar to task 1, the participants had to estimate the average value within a region of interest that was highlighted in the 3D view by a circle with a radius of 50 pixels.

Task 3 – "Mental Mapping": The participants had to locate a point of interest in the 3D view that was shown as a red cross on a separate 2D map next to the visualized virtual 3D city model. Participants were asked to click the target position in the 3D view and were allowed to correct this position.

Participants and Experimental Design. 21 volunteers (17 male) between the ages 21 and 49 were recruited. All of them had normal or corrected-to-normal vision and no known visual impairments. The study design was within subjects $4 \times 3 \times 8$ (task × color mapping × rendering technique), which resulted in a total of 96 trials per participant. The rendering techniques comprised Blinn-Phong shading, SSAO, image-space edge enhancement, and abstract façade textures produced with the proposed LoA texturing approach (Figure IV.33); the other four were combinations and a plain color mapping as reference comparison. The number of color mappings was limited to a warm, desaturated black-body radiation, a diverging blue-red color scheme, and a continuous white-red color scheme with a single hue. For all trials, each combination of color mapping

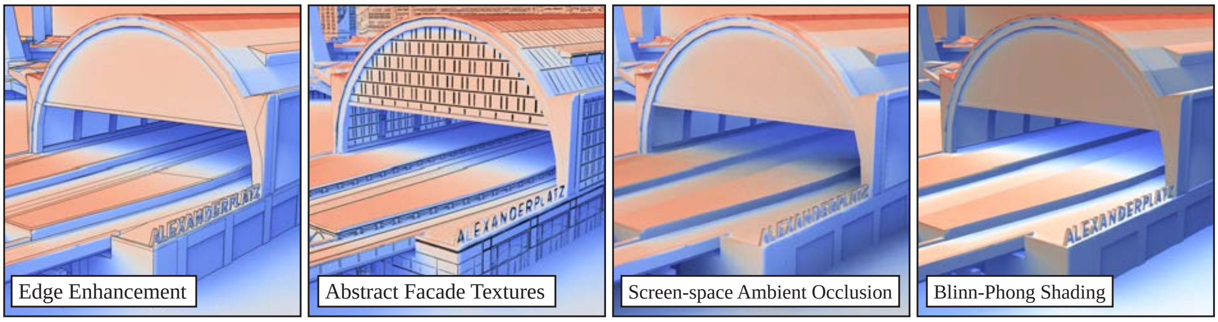


Figure IV.33.: Comparison of different enhancement rendering techniques examined in the user study. The techniques that perform edge enhancements in image space and texture space are based on the proposed LoA methods.

and rendering technique was setup for a distinct random camera setting. An additional trial at the beginning of each task was counted as practicing. The task and trial order were counterbalanced between subjects to avoid sequence effects. Participants received verbal instructions prior to each task, and upon their completion they rated the usefulness of each rendering technique for all color mappings, rated on a scale from 0–"not useful at all" to 7–"very useful". In addition to the independent variables (i.e., task, rendering technique, color mapping), the completion time and quantitative error measures for the trials were recorded, i.e., the distance error in image space for the mental mapping task, and the absolute error in percentage for the value estimation tasks. The study took about 45 minutes per participant.

Apparatus. All participants used the same two-monitor setup (24 inch, TFT, 1920×1200 pixels) under standard room lighting conditions. The 3D views of the model were rendered in real-time with a fixed camera configuration and 1920×1080 pixel resolution. Additional pixels were used to present a normalized legend (values between 0 and 100) of the active color mapping with marks visualized every 10 units. The second monitor displayed the interface elements for the answers, and a 2D map (900×900 pixels) for the mental mapping task.

Hypotheses. The following hypotheses were proposed for these user study (refer to Engel *et al.* (2013) for a complete list): (H1) all rendering techniques would improve the spatial perception and, thus, would reduce task time and error rate for mental mapping and distance estimation, (H2) the estimation of thematic values would be more difficult with any rendering technique and would result in higher error rates and task completion times, and (H3) there would be a distinct order of rendering techniques for each task.

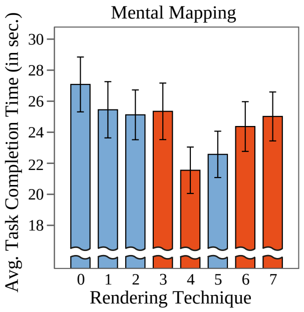


Figure IV.34.: Task completion time for the mental mapping task (see Figure IV.35 for disambiguation).

Results. Among others, the effect of rendering techniques on the error rate was different among the value estimation tasks (Figure IV.35). For a point-wise estimation, all evaluated rendering techniques were at least 18.5% less accurate than without using any enhancement, which confirms hypothesis H2. Shading and lighting techniques showed the greatest decline in accuracy and increased the task completion time, which may be explained by the constant changes in color brightness and altered color information. The opposite effect is observed for the area-wise estimation: all rendering techniques reduced the estimation error. The highest accuracy was performed with the image-space edge enhancement technique, in average, 44.6% better than with a plain color mapping, whereas the other techniques performed between 13.1% and 30.1% better. Using a repeated measures ANOVA, a significant main effect was found on rendering technique ($p < 0.05$, $F_{7,15} = 3.25$). Post-hoc pairwise comparisons using Bonferroni corrected p-values showed significance in the effect of image-space edge enhancement. In both estimation tasks the large-scale changes of the illumination techniques required higher mental effort and longer completion times. For the mental mapping task, each rendering technique reduced the mean completion time compared to a plain rendering (Figure IV.34), which indicates that they ease mental mapping and orientation guidance. An error of over 50 pixels in screen space was classified as missing the target position, which resulted in 6.6% failure trials among the participants. All rendering techniques improved the completion time by at least

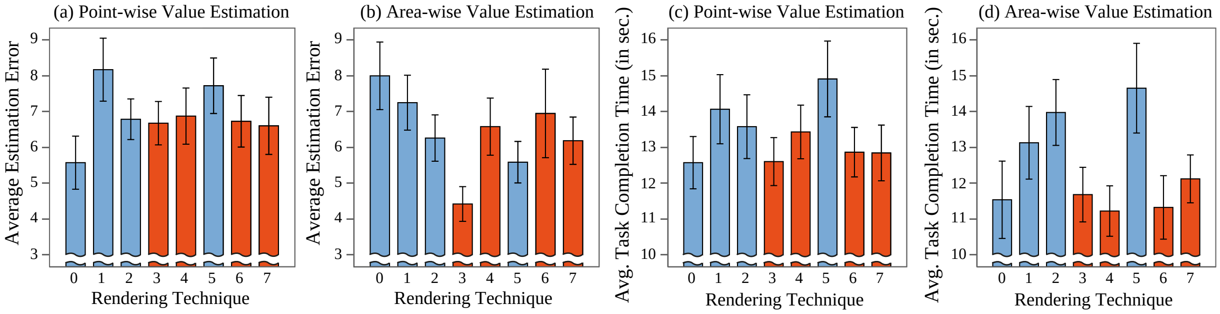


Figure IV.35.: Error and task completion times for value estimation with (0) no enhancement, (1) Blinn-Phong shading, (2) SSAO, (3) edge enhancement (EE), (4) abstract façade textures (AFT), (5) Blinn-Phong shading+SSAO, (6) SSAO+AFT, (7) AFT+EE. The results for the techniques that include edge enhancements are highlighted in red.

6.4% over using a plain color mapping. The proposed LoA texturing technique improved the completion time by 25% in average when locating positions in 3D space, primarily because they provide additional information over the geometric appearance and semantics of buildings to ease the identification of landmarks. This also relates to the questionnaire, where the edge enhancement techniques were rated the most helpful for the mental mapping task. While the user study confirms that the LoA texturing approach improves the users' performance, the image-space edge enhancement, however, was overrated when estimating thematic values, which showed to be a useful technique for an area-wise but not for a point-wise estimation. By contrast, abstract façade textures were underrated for the area-wise estimation than they actually performed.

To summarize, while hypotheses H1 and H3 did not hold, the proposed LoA texturing techniques revealed to be a good choice for orientation and mental mapping tasks. In addition, the study indicates that an edge enhancement in image space is able to improve the estimation of thematic data.

Performance

The performance of the framework depends on which image filters are used—which is identified as bottleneck—and how many effect layers are defined. The virtual 3D scene depicted in Figure IV.23 was rendered with the anisotropic Kuwahara filter ($N = 4$) (Kyprianidis *et al.* 2010a) and the proposed SDoF variant with the setup described in Section 3. The scene is composed of 540 unique 3D objects with 15 texture atlases (each 1024×1024 pixels). A fly-through sequence was used that lasts 15 seconds with filter caches cleared prior to each iteration. The results provided in Table IV.1 show that the performance scales with the screen resolution, reaching interactive frame rates in HD resolution and real-time performance when using the proposed caching concept. Per-fragment filtering without caching is almost on par with conventional post-process filtering in HD screen resolution, thus it could also be used for dynamic textures in a spatio-temporal visualization scenario. The timings include all filtering and rendering cycles with potential memory transfers. The memory consumption is proportional to the screen resolution with respect to the *G-buffer* and proportional to the number of filter layers with respect to cached textures targets. It is assumed that the consumption can be significantly decreased when using sparse textures (OpenGL 4.3 feature) in the implementation of the framework.

Table IV.1.: Performance evaluation of the framework (average frames per second) using the anisotropic Kuwahara filter with polynomial weighting functions (AKF) (Kyprianidis *et al.* 2010a) and the SDoF configuration used to produce Figure IV.25. Minimum frames per second are indicated in brackets. Setups: (1) filtering entire mipmap levels with caching enabled, (2) per-fragment filtering with caching disabled and (3) enabled, (4) filtering in a post-process stage.

	Screen Resolution	Setup 1	Setup 2	Setup 3	Setup 4
AKF	800 × 600	75.6 (12)	12.9 (9)	87.9 (44)	17.2 (16)
	1280 × 720	51.9 (12)	11.5 (10)	67.3 (26)	14.7 (14)
	1920 × 1080	31.5 (11)	11.3 (10)	37.3 (14)	11.8 (11)
SDoF	800 × 600	53.8 (23)	12.3 (8)	66.3 (54)	29.4 (28)
	1280 × 720	44.1 (20)	11.1 (8)	52.9 (41)	19.6 (18)
	1920 × 1080	30.2 (13)	8.6 (7)	33.5 (24)	10.0 (9)

4. Limitations

In contrast to image-based artistic stylization, the proposed methods are not able to inherently control the LoA of the scene geometry. However, this enables a more controlled geometric abstraction by specialized techniques. Alternatively, the methods may be combined with filtering in a post-process stage to selectively filter across object boundaries. In addition, UV mapped textures are required as input, otherwise a spatial filtering (e.g., for the SDoF variant) is not practical. There is also room for improvement of the system's performance. For instance, interactive filtering cannot be maintained when solving sparse linear systems (e.g., for image decomposition) because local image filtering cannot be performed. This also applies to iterative approaches that are resistant to parallelization. Outsourcing computationally expensive filters to multi-GPU systems could alleviate this problem, which is supported by progressive and decoupled filtering.

4. Summary

In this chapter, an interactive image filtering approach for LoA texturing of virtual 3D scenes is proposed. The decoupled deferred texturing concept enables to process texture maps coherently with respect to linear perspective by arbitrary image filters to preserves depth cues. A caching concept with optimization schemes is fashioned by per-fragment and progressive filtering that enable interactive or real-time frame rates, and thus user-defined adjustments of the LoA at run-time. In addition, a framework is proposed that is extensible by custom image filters and interaction techniques to parameterize the filtering process via mouse, touch, or natural language. Several results demonstrate its manifold application to the fields of visualization. Finally, evaluations demonstrate effects for adjusting the visual saliency in focus+context scenarios and the potential impact of edge enhancement techniques performed in image space and texture space for depth-cue enhancement and thematic color mappings. In particular, using abstract façade textures revealed to be a good choice to ease orientation, navigation, and mental mapping tasks, whereas a general image-space edge enhancement improves the estimation of thematic data that are visualized using color mappings.

V

Cartography-Oriented Visualization

The work presented in this chapter is partly based on

Semmo, A., Trapp, M., Jobst, M. & Döllner, J. "Cartography-Oriented Design of 3D Geospatial Information Visualization - Overview and Techniques". *The Cartographic Journal* **52,** 95–106 (2015b)

Semmo, A., Kyprianidis, J. E., Trapp, M. & Döllner, J. *Real-Time Rendering of Water Surfaces with Cartography-Oriented Design.* in *Proc. Computational Aesthetics* (2013), 5–14

Semmo, A., Hildebrandt, D., Trapp, M. & Döllner, J. "Concepts for Cartography-Oriented Visualization of Virtual 3D City Models". *Photogrammetrie - Fernerkundung - Geoinformation,* 455–465 (2012a)

Trapp, M., Semmo, A. & Döllner, J. *Interactive Rendering and Stylization of Transportation Networks Using Distance Fields.* in *Proc. International Conference on Computer Graphics Theory and Applications* (2015), 207–219

Virtual 3D city and landscape models play an important role for the communication of and interaction with complex, 2D and 3D geospatial information (Chapter I). As soon as a compact, yet optimized presentation of these models with respect to the information transfer to the viewer is required, perceptional and graphical design issues need to be considered. Chapter III reviewed the fields of NPR, illustrative visualization, and focus+context visualization that address these issues. For an efficient communication, however, it requires having an adequate application of NPR techniques on feature level, otherwise the results are limited to a "monotonic" visualization that lacks cartographic design aspects.

In this chapter, concepts for cartography-oriented visualization of virtual 3D city and landscape models are proposed. These are based on coupling generalized NPR techniques, including the LoA texturing technique provided in Chapter IV, with semantics-based information for a user, context, and media-dependent representation of 3D geospatial model entities. In particular, NPR techniques to interactively synthesize cartographic renditions of basic feature types are proposed in correspondence with the extended 3D semiotic model of Chapter II. This includes an iconification concept to seamlessly interpolate between photorealistic and cartographic representations of 3D landmarks, a real-time rendering technique for the view-dependent visualization of transportation networks, and an interactive system to render water surfaces with cartography-oriented design. All presented techniques enable real-time image synthesis to facilitate a visualization of dynamically changing thematic information.

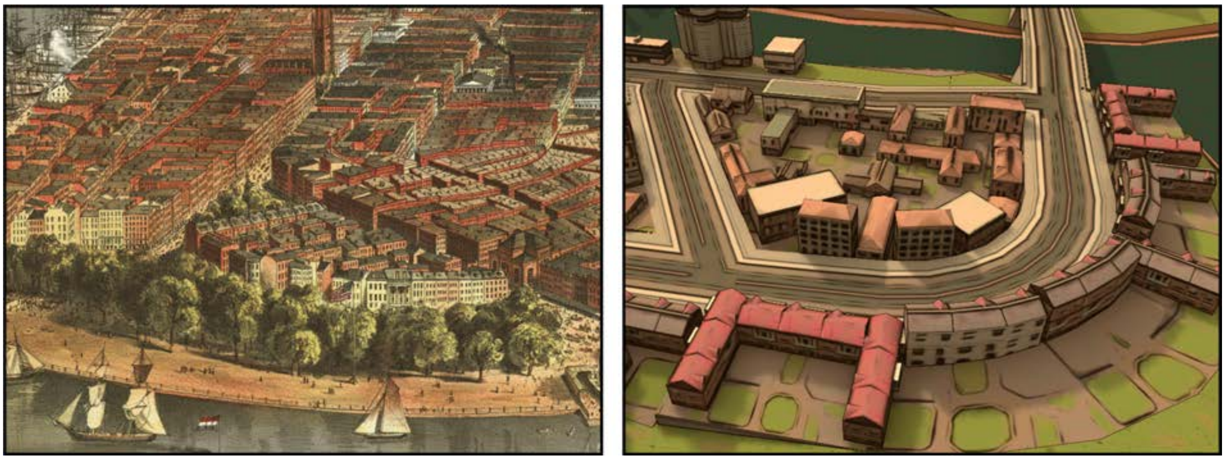


Figure V.1.: Comparison between (left) a hand-drawn map (New York City Panoramic – 1873 Wall Map Mural) and (right) a computer-generated visualization of a virtual 3D city model based on abstract textures and color grading.

1. Cartography-Oriented Visualization of Virtual 3D City Models

In this section, NPR techniques for the cartography-oriented visualization of virtual 3D city model entities are provided. First, generalized NPR techniques and application examples are outlined. Afterwards, specialized NPR techniques with respect to building and site models, transportation networks, and water surfaces are proposed.

1. Challenges

In contrast to 2D maps and illustrations by famous cartographers such as Merian (2005), virtual 3D city models have special characteristics in terms of their data complexity and interactive presentation. The following challenges for computer-generated, cartography-oriented illustrations are identified:

1. **Automated and generic abstraction techniques.** A non-photorealistic rendering of virtual 3D city models uses abstraction techniques that should work independently from the model contents. Figure V.1 exemplifies a hand-drawn map in comparison to a computer-generated visualization that is based on the LoA texturing approach of Chapter IV and color grading.

2. **Interaction with 3D geovirtual environments.** A user should be able to adapt the abstraction of a 3D scene to the performed task and according to prioritized information. One example is an adaptive configuration of color schemes for prioritized feature classes.

3. **Real-time rendering.** Rendering techniques should perform in real-time, or at least interactive frame rates, to maintain an immersive experience, i.e., to maintain the sensation of "being in" a 3D geovirtual environment (A. M. MacEachren *et al.* 1999).

4. **Frame-to-frame coherence.** While navigating through a 3D geovirtual environment, the viewing direction can be continuously changed. Correspondingly, visualization techniques need to maintain spatial and temporal coherence (Bénard *et al.* 2013) to avoid visualization artifacts by changes in viewing parameters, depth values, and occlusion.

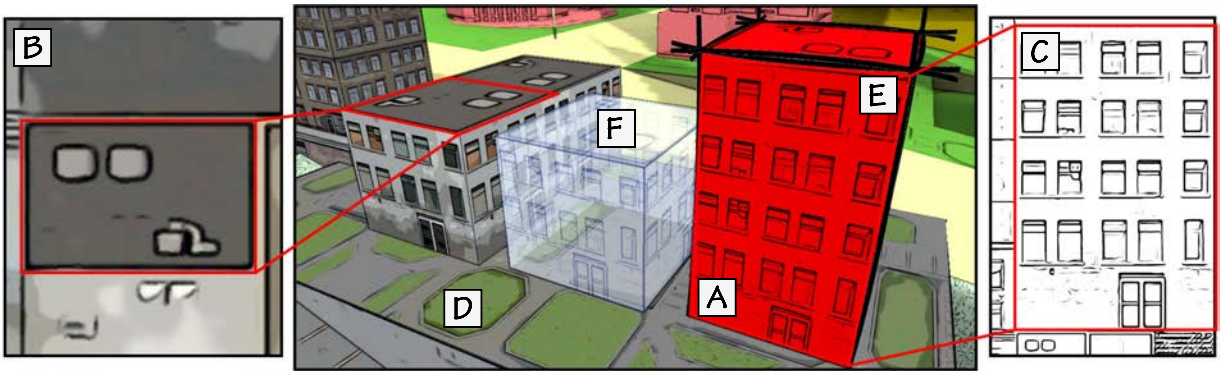


Figure V.2.: Overview of generalized rendering techniques and configurations: (A) colorization, (B/C) abstraction in texture space, (D/E) edge enhancement in image space and object space, and (F) transparency.

2. Generalized Visualization Techniques

This section outlines techniques for the visualization of thematic information that consider semiotic aspects (Figure V.2) and the aforementioned challenges.

Colorization. The colorization of 3D model entities adopts design principles proposed by Brewer (1994). For instance, qualitative color schemes can be used to communicate feature classes efficiently and ease comprehension of a city's structure. By contrast, sequential color schemes can be used to communicate thematic information for analysis purposes, for instance to illustrate the distribution of green spaces in a virtual city model (Figure V.3D).

Edge Enhancement of 3D Model Entities. The LoA texturing technique of Chapter IV operates in the color domain and is content-independent, thus it can also be applied to aerial images, terrestrial photography and synthesized textures. The stylization of textures is performed according to thematic or semantics-based information. For example, the outputs of an edge enhancement and bilateral filter (Tomasi & Manduchi 1998) can be blended at rendering time to stylize features of specific classes differently (Figure V.3B/C).

Transparency of 3D Objects. As reviewed in Chapter III, transparency effects are a well-known method to improve the visibility of occluded model entities in 3D space. For example, alpha blending can be used to enable blue print illustrations of 3D scenes (Nienhaus & Döllner 2004a), e.g., to highlight construction states of building models (Figure V.3C) and aid the perception of complex structures.

Application Examples. The visual quality of an abstraction or stylization technique is a well known issue in aesthetics of computer graphics. It primarily builds on configurations that adapt to a viewer's needs, technical background, and cognitive qualities. To this end, 3D model entities are mapped to corresponding feature classes using shader uniforms to provide a rendering that is adapted to the image contents (Figure V.3). Thereby, new possibilities emerge for the communication of thematic information. For example, structures of a *city's image* can be explicitly highlighted such as routes and districts (Lynch 1960) to ease orientation and navigation tasks. Further, thematic information can be visualized using qualitative and quantitative color schemes, e.g., for the exploration of crime maps (Wolff & Asche 2009). Finally, parameterized abstractions can visually enhance architectural aspects for city planning.

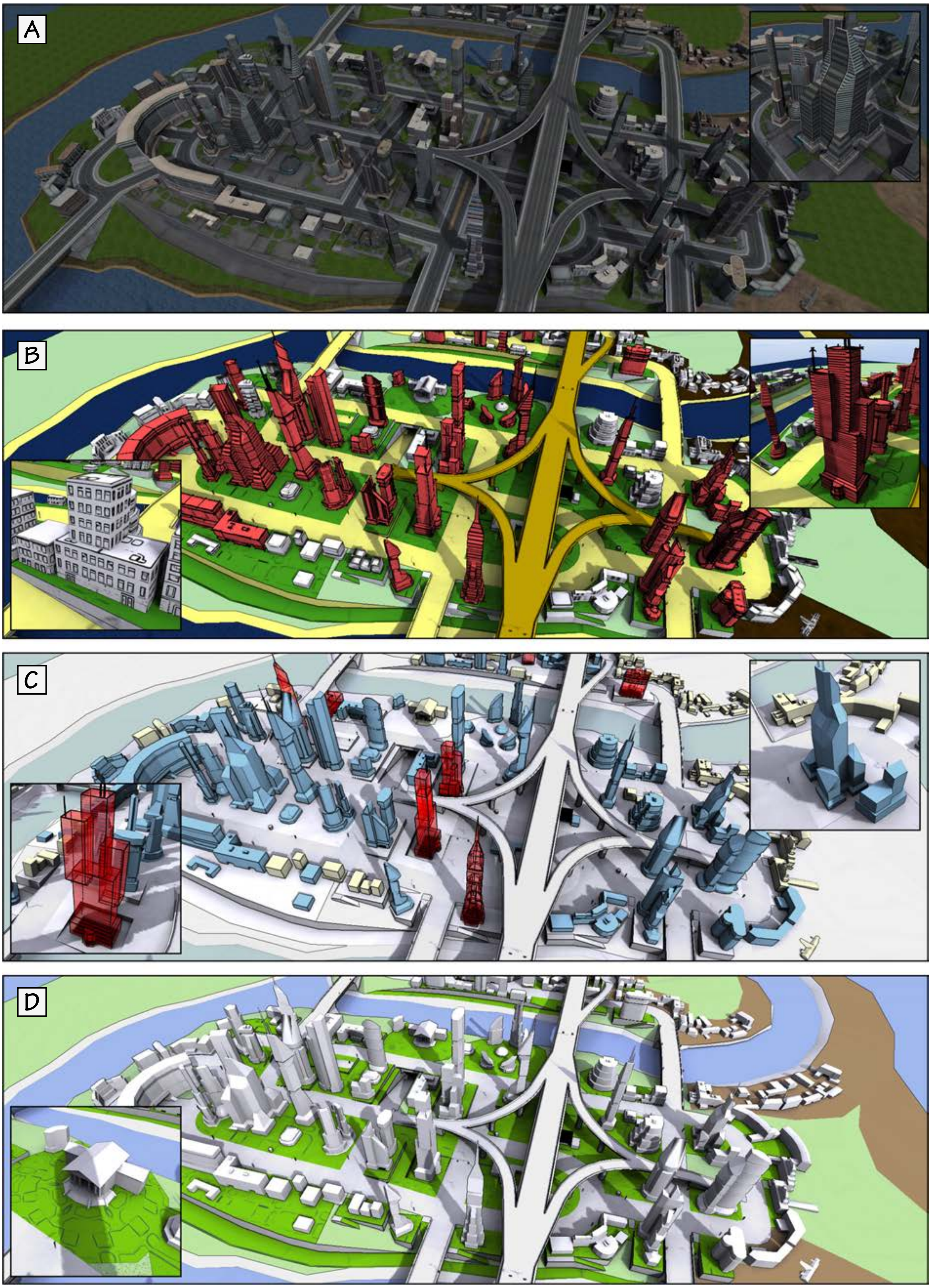


Figure V.3.: Exemplary scenarios for a cartography-oriented visualization of a virtual 3D city model, in comparison to (A) a detailed representation: (B) highlighting of feature classes that is based on a diverging color scheme and edge enhancements to improve the perception of a city's structure, (C) visualization of construction dates and building types by using a diverging color scheme with transparency effects to indicate planned constructions, (D) highlighting the distribution of green spaces, e.g. to help analyze life quality within an urban environment.

3. Visualization of Buildings and Sites

Textbooks on map design and thematic cartography (Imhof 1975; A. MacEachren 1995; M. Kraak & Ormeling 2003; Tyner 2010) were studied to extract design principles with respect to semiotic aspects found for the depiction of buildings and sites. The following principles were identified from this empirical analysis (Figure V.4):

1. Constrained color palettes are often used for visualization, e.g., unique colors to distinguish between roofs and façades or between sacred and secular buildings.
2. Silhouettes and contour lines are often depicted to enhance building entities.
3. Individual features may be depicted using strokes, e.g., to indicate windows.
4. Important buildings may be shown from a "best view", i.e., a landmark is rotated to face its most characteristic façade.

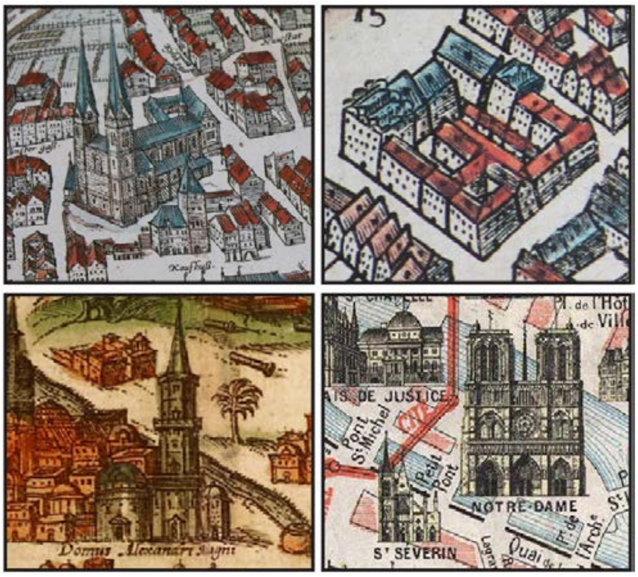


Figure V.4.: Hand-drawn examples for the cartographic design of buildings and sites.

Design Aspects 1-3: An algorithm is implemented that automatically extracts dominant colors from 3D scene textures. Thereby, the extraction of single colors is preferred in image regions with constant color tone using entropy-based metrics. Afterwards, the algorithm of Levin *et al.* (2004) is used to obtain recolorized façade textures, e.g., in the style of natural color maps (Patterson & Kelso 2004). Alternatively, only the most dominant color may be encoded per feature component (e.g., roofs vs. façades) and used as material color to further reduce the color spectrum. For a detailed description of the algorithm, the reader is referred to Chapter VII. As a complementary effect, DoG filtering is performed in texture space for edge enhancement, e.g., to locate variations in color values and intensities as defined by window contours, coupled with the image-based approach of Nienhaus & Döllner (2003) to enhance geometry-related contour lines. Together with the provided recolorization algorithm, this cartography-oriented visualization approach improves feature contrasts, provides a decluttered representation with inherent shadow removal, and expresses uncertainty (Figure V.5). Using the feature congestion measure of Rosenholtz *et al.* (2007), Figure V.7 shows how this approach reduces visual clutter, thus being able to potentially optimize the transfer of information, e.g., in routing applications, where often only a few selected information is required for orientation. In addition, the approach enables to implement compression algorithms that reduce the memory consumption up to a factor of 10 in case pre-dominant colors are stored as attributes and edge information is stored with 8 bit precision.

Design Aspect 4: Certain buildings and sites within a virtual environment serve as landmarks, i.e., reference points with a characteristic appearance or location, or interest to a viewer. The visualization of landmarks is essential for

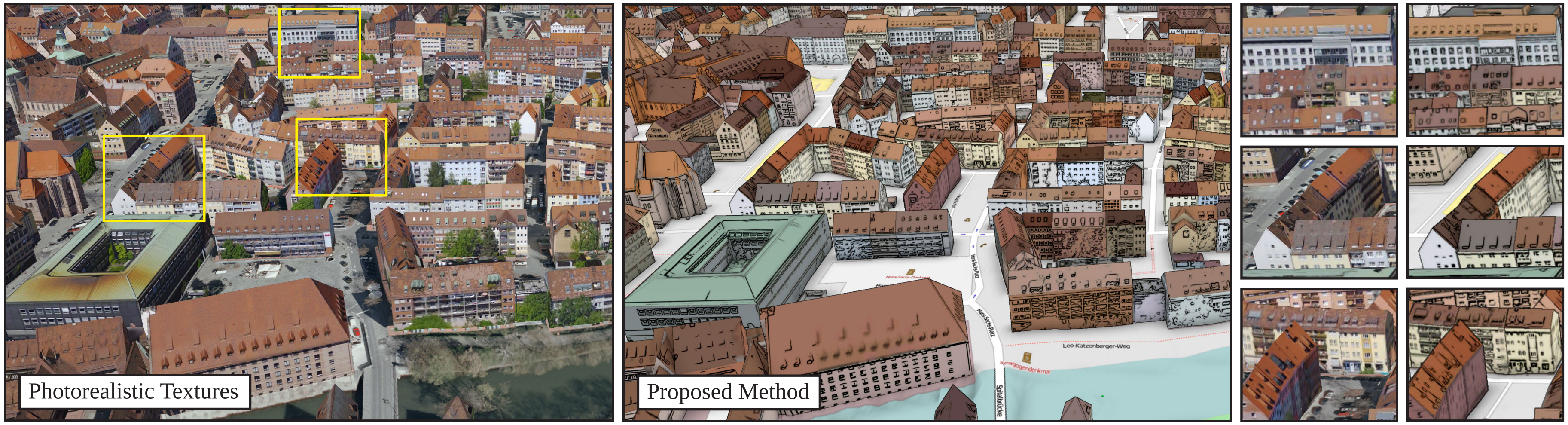


Figure V.5.: Cartography-oriented design of 3D geospatial information visualization for a virtual 3D city model. Building façades encoded as diffuse color maps are filtered and colorized by their dominant colors per component (exterior walls vs. roofs), and blended with feature contours. Model: © City of Nuremberg – Amt für Geoinformation und Bodenordnung, all rights reserved.

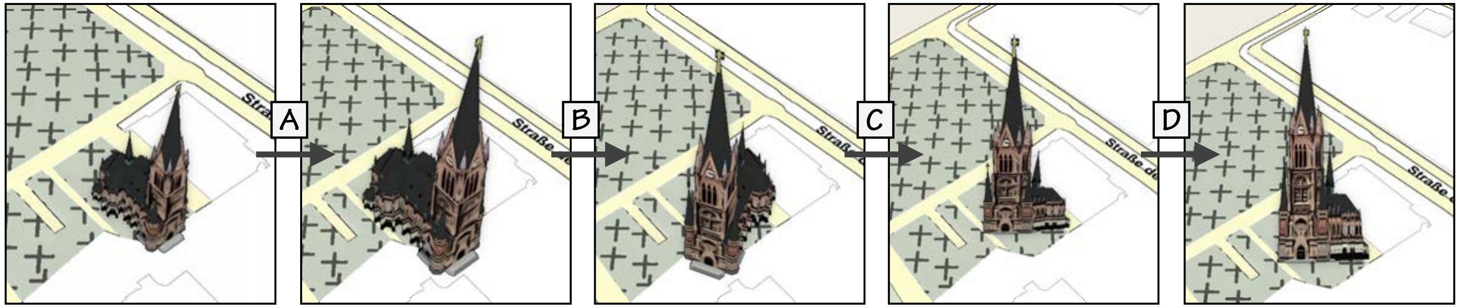


Figure V.6.: The proposed technique for landmark iconification includes the following geometric transformations: (A) scaling, (B) rotation to "best view", (C) flattening, (D) billboard transformation.

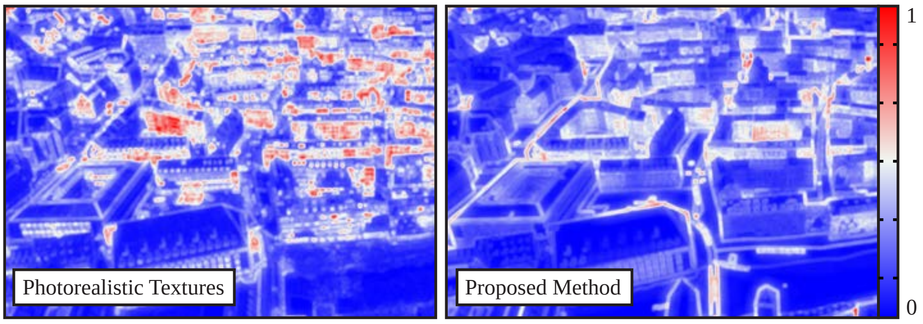


Figure V.7: Visual clutter analysis for Figure V.5, compared to the respective photorealistic version, using the algorithm of Rosenholtz *et al.* (2007) for evaluation.

localization, orientation, and navigation, and often practiced in the design of tourist maps (Grabler *et al.* 2008). A visualization technique is implemented that uses global deformations applied prior to rasterization to automatically transform 3D buildings and sites to iconified variants. The iconified landmarks are rotated according to their "best views" to match the direction of the virtual camera's view. For buildings, "best views" often face the street or main entrance; the information can be manually modeled or automatically approximated using viewpoint entropy (P.-P. Vázquez *et al.* 2004). To obtain a deformed landmark in world-space coordinates, the following four steps are performed on a per-vertex basis during rendering:

1. ***Landmark scaling***. Landmarks are scaled non-linearly to improve their visibility (Glander *et al.* 2007) in far view distances (Figure V.6A). Here, a weighted *smoothstep* function is used to compute the scale factor.

2. ***Rotation to best view***. Landmarks are pitched to horizontally align their "best view" direction with the virtual camera's view direction (Figure V.6B).

3. ***Object flattening***. Landmarks are flattened in depth; i.e., their vertices are projected to the plane facing the horizontal view direction (Figure V.6C).

4. ***Cylindrical billboard transformation***. The flattened landmark is yawed by the camera elevation to vertically face the view direction (Figure V.6D).

This rendering technique is able to linearly blend original and transformed vertices using shader technology to provide smooth interpolations. In addition, complementary transformation techniques can be seamlessly integrated such as global deformation to reduce occlusions in regions of interest (Möser *et al.* 2008; Degener & R. Klein 2009; Qu *et al.* 2009) or terrain geomorphing (Wagner 2003).

4. Visualization of Transportation Networks

Transportation networks, such as streets, railroads, and metro systems, constitute primary elements in cartography for reckoning and navigation. In recent years, they have become an increasingly important part of 3D geovirtual environments for the interactive analysis and communication of complex hierarchical information, for example in routing, logistics optimization, and disaster management. Various rendering techniques were proposed that deal with integrating transportation networks within 3D geovirtual environments, including the geometry-based, texture-based, and stencil-based approaches reviewed in Chapter III. However, these approaches neglect the many challenges of an interactive,

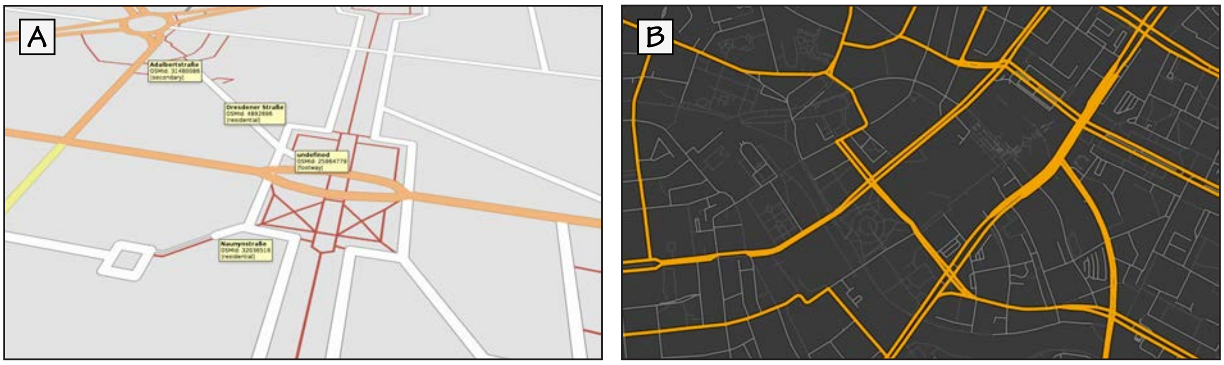


Figure V.8.: Rendering of an OpenStreetMap data set using different map stylizations. (A) perspective projection at a high zoom level, (B) orthographic projection using a high-contrast color configuration.

cartography-oriented design process to adapt the LoA according to a user's context and dynamically changing network data.

This section presents a real-time rendering technique for the view-dependent visualization of geometrically complex transportation networks within 3D geovirtual environments. The proposed technique is based on distance maps (Frisken *et al.* 2000) and deferred texturing that shifts the design process to the shading stage for real-time stylization. The approach is demonstrated and discussed by means of street networks using cartographic design principles for context-aware stylization, including view-dependent scaling for clutter reduction, contour-lining to provide figure-ground relationships, handling of street crossings via shading-based blending, and task-dependent colorization.

Design Principles from Cartography

The following two groups of design principles are extracted from an empirical analysis of the works by famous cartographers and map designers (e.g., Harry Beck, Figure V.9B) and according to the semiotic model of Chapter II: graphical elements (e.g., lines, points) and their position, and graphical variables (e.g., color, line thickness, decoration elements such as labels).

Graphical Elements. In general, two approaches for the depiction of transportation networks exist: (1) non-explicit elements by the principle of surroundedness (A. MacEachren 1995) for effective figure-ground distinction on maps, for instance where streets are formed as a unit via enclosing features (Figure V.9A), and (2) the explicit graphical depiction via connected lines and points (nodes) (Figure V.9B-D) on which the proposed technique focuses on. In modern maps, *(P1) contour lines often surround fine-textured fills or solid colors to add visual contrast and improve the figure-ground perception* (A. MacEachren 1995). Hierarchical representations of street networks often have *(P2) primary streets overlap secondary or tertiary streets*. In sketch maps, these lines or contours are often drawn excessively wavy or fuzzy to express uncertainty. Following a LoA concept, *(P3) dynamic filtering and scaling of these geometric features improves the perception of roads at high view distances and avoids overcluttering*. The choice of shape often varies in thematic cartography, ranging from solid lines to dotted representations (e.g., to distinguish between car driving and biking directions). Finally, labels are primary design elements to enrich networks with meta-information. By convention, *(P4) names follow principal line directions and are placed within streets, or outside line* *segments and oriented with links*, e.g., the latter in schematized maps (Figure V.9B), to ensure legibility (Imhof 1975).

Graphical Variables. In many maps, *(P5) a hierarchy of emphasis is drawn among reference elements such as different line weights and colors* to portray different grades of roads (Figure V.9B/D). In modern maps, *(P6) streets are tinted using qualitative color schemes to represent street classes* and to distinguish them from the underlying terrain. This association may enable cognitive grouping of each network type (M. Kraak & Ormeling 2003). To date, standardized color schemes for transportation networks have not been established; they vary from country to country. But it can be observed that *(P7) yellow established as a conventional color tone for main streets, with a discrete gradation towards grey and white shading for tertiary roads* (Figure V.9D). Lately, principles for color blindness have also been examined using OpenStreetMap data (Kröger *et al.* 2013). All of these graphical variables are subject to change according to the zoom level to avoid overcluttered displays.

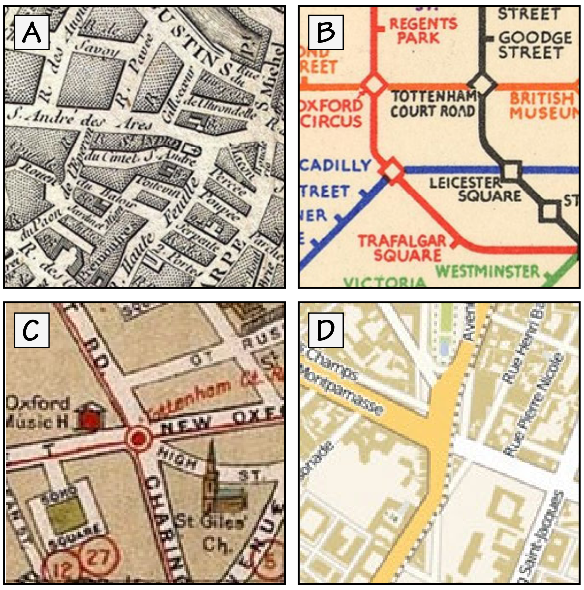


Figure V.9.: Maps depicting transportation networks. (A) Paris in 1787 designed by Louis Brion de la Tour, (B) the *Pocket Underground* map from 1933 by Harry Beck, (C) London map, (D) contemporary map of Paris.

Method

Figure V.10 provides a schematic overview of the rendering pipeline comprising control and data flow. It consists of the following three stages:

Preprocessing: This stage loads and transforms a given transportation network with its associated meta data into a compact representation—an attributed point cloud—to provide a low memory footprint. This operation is required to be performed only once per data set.

Distance transform: Starting from the pre-processed input, this stage synthesizes textured polygons of respective widths that are subsequently rasterized into distance-field buffers (distance maps) defined for each network category.

Stylization and compositing: In this stage, deferred texturing based on the distance maps is performed in screen space using a single post-processing pass. It enables application-wise procedural and raster-based texturing for colorization with LoD support. The resulting colors of each network category are subsequently composited in a bottom-up approach with respect to their ranking.

Representation of Networks. OpenStreetMap data is used as input for road data and 2D terrain information (Haklay & P. Weber 2008). According to the identified design principles, cartographic route depictions are synthesized by a hardware-accelerated rasterization of line segments—represented by rectangular geometry. Using geometry shaders, the geometry synthesis and tessellation is performed on graphics hardware, which however requires the geometry of input lines and additional route data to be represented as per-vertex attributes. To this end, the attributed topological data of a street network is processed in a pre-processing

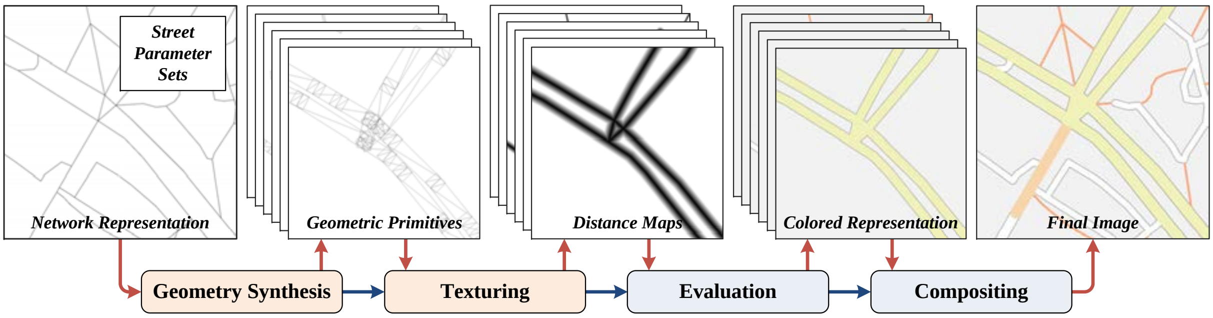


Figure V.10.: Schematic overview of the rendering pipeline for transportation networks. Given a compact network representation and stylization parameters, textured geometry and distance maps are synthesized per network category within a single rendering pass. The results are then used for stylization and image composition in a post-processing pass.

stage to obtain GPU-based representations. The attributed graph is encoded by a node buffer and a segment buffer, both suitable for hardware-accelerated rendering. At this, the nodes comprise position and grade information, i.e., the number of segments adjacent to the respective node, while the segments comprise indices to the nodes and the rank of a route segment. In addition to the rank information provided by OpenStreetMap, a specific rank for highlighted routes is introduced that overrides all default ranks. Further, the segments' lengths in world-space units are encoded for optional length-parameterization.

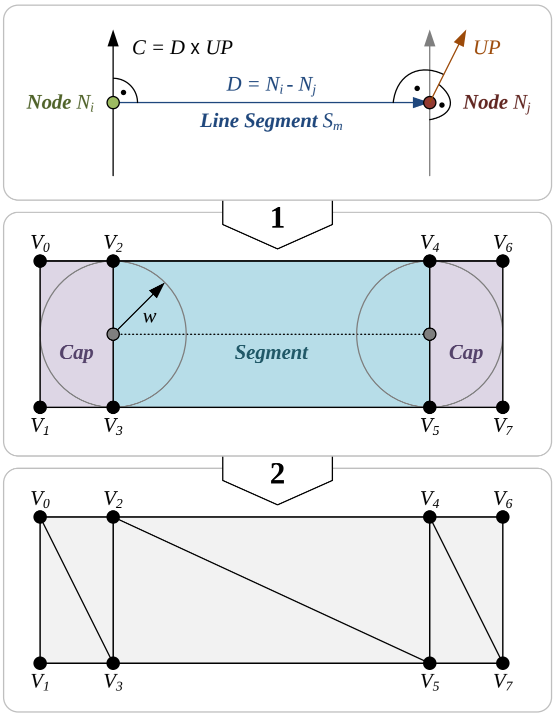


Figure V.11.: Schematic overview of the geometry synthesis for transportation networks.

Distance Map Computation. Figure V.12 shows the process of computing distance maps for a given network configuration. First, geometry is synthesized by converting point primitives—issued for rendering—to triangle strips. This can be efficiently implemented using geometry shaders to minimize the memory footprint for representing a street network. Based on this compact network representation, polygons are created for each line segment (Figure V.12A) and its respective line ends, i.e., the *caps* (Figure V.12B). Figure V.11 schematizes this process in more detail. Further, vertex texture coordinates are computed for each polygon, which are encoded using triangle strips, and yields distance drop-offs during rasterization (Figure V.12C). One advantage of this concept is the respective computation of distance maps per street category. This enables a flexible stylization and image compositing at the rendering stage based on the distance maps. To this end, the respective texture coordinates are interpolated during rasterization, and the distance values are computed using bilinear sampling. The required distance maps are represented using a single 2D texture array, whose layers are indexed according to the type of geometry (cap or segment) and the grade of a node. The individual 2D texture layer then encodes the distance drop-off, which enables a flexible parameterization of design aspects and provides a simplified implementation. In particular, the approach bypasses

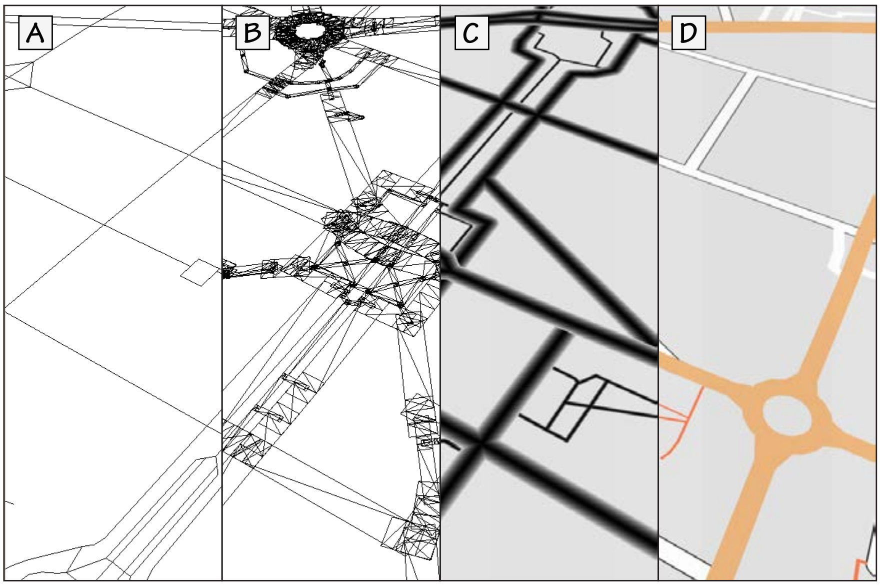


Figure V.12: Overview of the processing stages for the visualization of transportation networks.

(A) input data

(B) synthesized polygons

(C) distance maps

(D) stylized transportation network segments

the geometric computation problem identified by Vaaraniemi *et al.* (2011) that yields visual gaps between adjacent and consecutive segments, i.e., by using the *minimal* operator as a logical connective for line segments.

Stylization of Street Networks

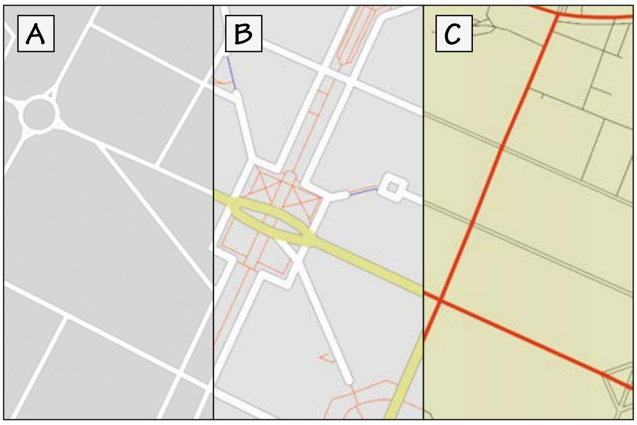


Figure V.13.: Examples of different configurations for the stylization of transportation networks based on distance maps.

The stylization is performed in a single post-processing stage via deferred texturing using individual style parameters defined per network category. The provided approach enables the stylization of segments based on their category. A *style parameter set* comprises the following parameters: the width of a geometric segment in world-space coordinates, distance values for differentiation between main and border segment colors *(P1)*, as well as the color of main and border segments. In addition, a style parameter set may comprise raster-based or procedural 2D texture maps for example-based rendering, e.g., to obtain sketch maps (Kopf *et al.* 2010). The proposed style parameterization can be further extended to define LoD variants for each style. This is especially useful for a number of applications such as (1) counterbalancing perspective foreshortening of segments at high view distances, (2) enable interactive filtering using lens-based interaction metaphors (Tominski *et al.* 2014), and (3) the reduction of visual clutter (Jobst & Döllner 2008a). To this end, the definition of a style parameter set is extended with an additional parameter that yields a list of *LoD tupels*. During run-time, LoD values are computed per vertex and fragment with potential style blendings defined per LoD. Using these style definitions, the conversion of the synthesized distance maps into respective colors is performed on a per-fragment basis using an additional post-processing step. Figure V.13 outlines potential use cases computed from a single set of distance maps, Figure V.13A shows a constant line width for a single stylization parameter set and all street categories, while Figure V.13B shows different configurations for each street category. Figure V.13C depicts the highlighting of a single street category that contrasts the remaining scene.

Results

The proposed rendering technique was implemented using C++, OpenGL and GLSL. The real-time image synthesis is separated into two rendering passes: the first pass creates the distance maps using off-screen rendering, and the second pass applies stylization using deferred texturing based on these distance maps. The distance maps are represented using single color channels with 8-bit value precision and a resolution of 512 × 512 pixels. The evaluated per-category colors are composited using a bottom-up image blending (Porter & Duff 1984). Here, the colors are blended starting from the lowest rank to the highest rank respectively *(P5/P6)*. This enables to overlap different category ranks *(P2)* and to visualize tunnels or similar constellations.

Table V.2 shows the results of a performance evaluation for different screen resolutions and different data sets of different geometric complexity (Table V.1). All tests were conducted on an Intel® i3-3110M 4× 2.4 GHz and Intel® HD 4000 GPU, with view-frustum disabled and backface culling enabled. The average run-time latencies indicate that the approach is fill-limited due to the heavy per-fragment operations, and limited by the number of applied parameter sets. Overall, the achieved frame rates satisfy a real-time rendering performance.

Table V.1.: Geometric complexity of the test data sets used for the performance evaluation.

ID	Data Set	#Nodes	#Ways
A	Berlin 1	5571	1028
B	Istanbul	2004	263
C	Berlin 2	9502	1766

Table V.2.: Performance evaluation for the test data sets of Table V.1, using different screen resolutions and a varying number of total style configurations (numbers in milliseconds).

Resolution	ID	Parameter Sets			
		1	2	4	8
390 × 260	A	3.0	3.2	4.1	5.5
	B	2.9	3.3	4.1	5.4
	C	3.0	3.4	4.2	5.5
670 × 450	A	3.0	3.2	4.1	5.7
	B	2.9	3.3	4.2	5.6
	C	3.0	3.4	4.2	5.8
1280 × 800	A	25.5	29.0	36.1	50.1
	B	25.4	29.0	36.2	50.1
	C	25.3	29.2	36.2	50.2

Figure V.14 gives an overview of application examples. The support of different style configurations facilitates the synthesis of localized maps, e.g., in terms of color *(P7)*, without requiring to change the geometric representation. Figure V.14A shows three different view-dependent stylizations that can be interchanged during rendering. In Figure V.14B, network segments with lower rank are faded in the rear part of the scene, and the foreshortening of the segments' widths is counterbalanced to reduce visual clutter. This is achieved using the proposed LoD approach by enabling: (1) the fading of low-ranked route segments and (2) increasing the width of high-ranked route segments with increasing view distance *(P3/P5)*. Thus, the depiction of low-ranked or unimportant routes can be omitted while important routes are emphasized. The focus+context functionality depicted in Figure V.14C is common in visualization frameworks for interactive filtering that adapts to a user's context, which includes region interest *(P3)*. To support this feature, two distinct style configurations are defined for the focus and context regions. Here, the focus shows a detailed view on a route network comprising all route categories, while the context only depicts the three major categories. A screen-space lens is used to control the transition between the respective LoD levels. Similar to lens-based filtering, Figure V.14D is based on a LoA approach to explicitly highlight a certain navigation route, while omitting

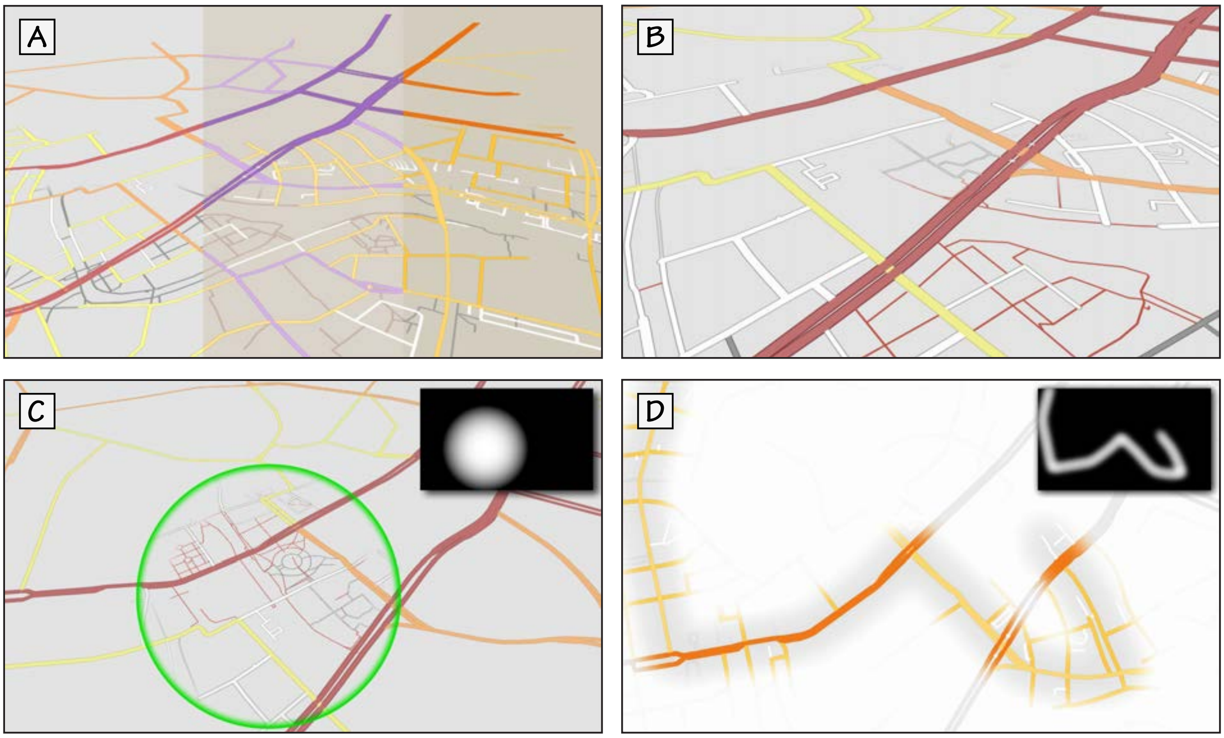


Figure V.14.: Application examples rendered with the proposed technique. (A) f.l.t.r OpenStreetMap, Bing, and Google Maps stylization, (B) distance-based stylization, (C) lens-based filtering, (D) region-of-interest rendering.

the rendering of the remaining network areas. This can be used for highlighting segments relevant for navigation. Here, a transition based on a drop-off function conveys parts of the context required for navigation, e.g., junctions or routes with a high (important) rank.

With respect to conceptual limitations, the proposed rendering technique causes two problems when using distance maps for visualization: *intrusion* and *protrusion*. Intrusion is caused if the distance maps of two routes of the same category intrude each other because of a large line width parameterization. Thus, the rendering results may appear as being connected to the viewer. Further, protrusion appears where two street categories with different width parameter intersect, especially at T-junctions. Here, the distance map of a high ranked street protrudes the distance map of a lower ranked street.

2. Visualization of Water Surfaces

Water surfaces represent key elements in mapmaking and surveying because they shape our world and convey important information to a number of domains, including hydrology, urban planning and environmental sciences. Hand-drawn illustrations of such surfaces are often carefully designed to help a viewer explore the geospatial environment. Usually, these designs effectively establish land-water distinction to facilitate orientation, navigation, or analysis tasks.

Yet the many different shapes of water surfaces pose a number of challenges that have required contemporary craftsmanship and design skills. Typical challenges include *(1) the symbolization of the land-water interface* using *(2) design elements (e.g., strokes) that exactly align with the shorelines* to effectively provide

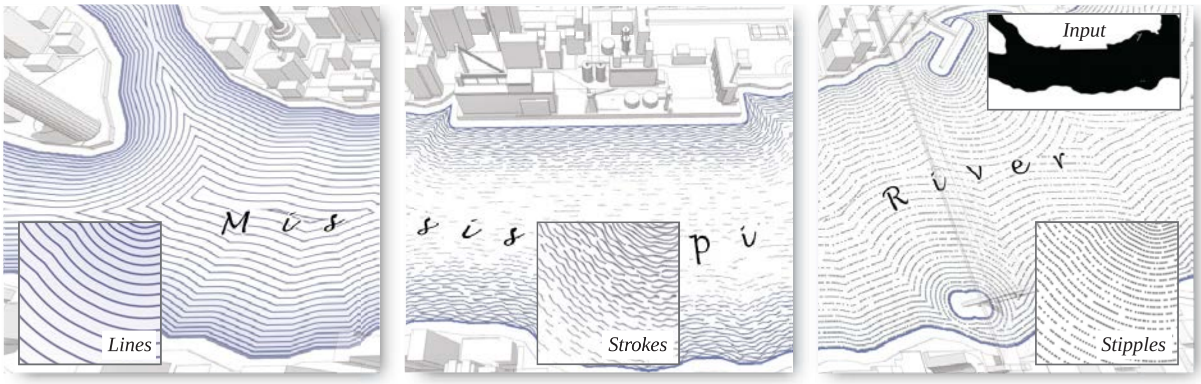


Figure V.15.: Illustrative rendering techniques for water surfaces implemented in the proposed system: (left) waterlining, (middle) contour hatching, (right) water stippling, and complementary labeling.

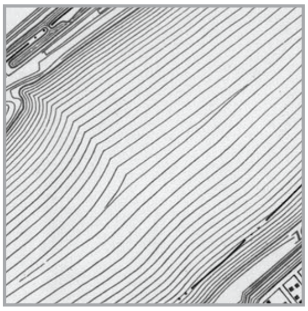
figure-ground and express motion. Over centuries, cartographers have developed illustration techniques and design principles that address these challenges (Imhof 1972; Robinson *et al.* 1995; Merian 2005). Certain techniques have become abundantly used in modern cartography, such as waterlining, hatching, or water stippling; and most of them express aesthetical appeal, provide excellent figure-ground balance, and establish a sense of motion (Christensen 2008; Huffman 2010). For instance, fine solid lines are placed parallel to shorelines to effectively communicate shoreline distances (Figure V.15 left).

To date, computer-generated illustrations of water surfaces are mainly based on photorealistic rendering techniques (Darles *et al.* 2011). Approaches in illustrative rendering have been subject to cartoon-like water effects (Eden *et al.* 2007; Yu *et al.* 2007), but have neglected the many challenges water exhibits for map design. By contrast, this section presents an interactive system for rendering water surfaces with cartography-oriented design that addresses the aforementioned challenges. For this, the section makes the following contributions. First, design principles from traditional and modern cartography are identified to create more effective illustrations of water surfaces. Based on these findings, Euclidean distance maps are computed using a novel feature-aligned distance transform to derive principal curvature directions of complex water shapes. Using this information, real-time rendering techniques for waterlining, contour hatching, water stippling and labeling are contributed (Figure V.15) that facilitate a view-dependent LoA. Finally, these rendering techniques are tested with various real-world virtual 3D city and landscape models. The results reveal potential applications within 3D geovirtual environments for map exploration, landscaping, urban planning, and flooding simulation.

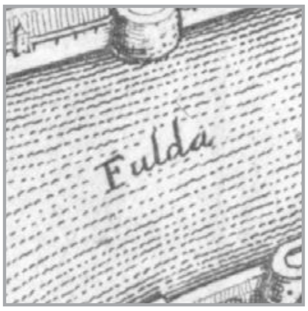
1. Design Principles from Cartography

Well-designed illustrations of water surfaces provide figure-ground, establish a sense of motion, and communicate meta-information (e.g., water names). A variety of illustration techniques have been developed by cartographers whose design principles address feature-aligned texturing and symbolization. However, most cartographers develop and vary their own illustration styles. Therefore, the work by famous cartographers such as Matthäus Merian (Merian 2005) and Jouvin de Rochefort printed in map collections are analyzed, in addition to textbooks on map design and thematic cartography (French 1918; Imhof 1975; A. MacEachren 1995; M. Kraak & Ormeling 2003; Tyner 2010). From this analysis, the following design principles are extracted:

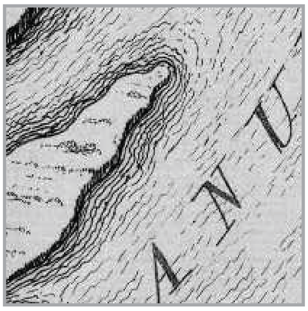
Waterlining. Waterlining became popular in the first half of the 20th century for lithographed maps, because areas of solid color tones could not be produced at that time. With this technique, *(P1) fine solid lines are drawn parallel to shorelines, and the spacing between succeeding lines gradually increase* (French 1918). A common mistake is to "make the lines excessively wavy or rippled" (French 1918) or the distance between lines with insufficient continuity. If drawn with care, waterlining provides dynamism and effectively propagates distance information (Christensen 2008; Huffman 2010).



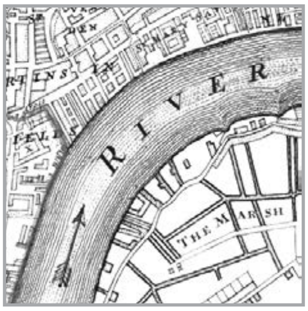
Water Stippling. Another conventional technique for hand-drawn black-and-white maps is water stippling (Tyner 2010). Similar to waterlining, distance information is propagated by aligning small dots with non-linear distances to shorelines. Compared to stippling in traditional artwork, water surfaces are depicted by *(P2) stipples with varying density that irregularly overlap along streamlines* to establish a sense of motion. In perspective views, some cartographers draw *(P3) stipples with higher density at occluded areas to improve depth perception, or with varying density to symbolize flow velocity*.



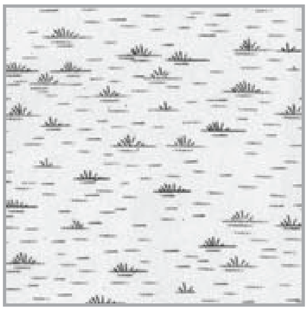
Contour Hatching and Vignetting. Contour hatching has been widely used to balance the accurate propagation of distance information and establishment of motion: by using *(P4) individual strokes that are placed with high density near shorelines and complemented by loose lines placed with increasing irregularity towards the middle stream*. In contrast to waterlines, *(P5) excessively wavy strokes are drawn to express motion*. Alternative illustrations use non-feature-aligned cross hatches for land-water distinction. These methods have been replaced in the second half of the 20th century by color tones and drop shadows (Tyner 2010). Today, coastal vignettes with solid color tones are used to establish figure-ground but fail to express water movements.



Labeling. Labels are design elements used in cartography to enrich maps with meta-information. By convention, *(P6) cartographers depict names of water features with italic (slanted) letters* to distinguish them from land features for which upright letters are used (Tyner 2010). Typically, *(P7) names follow principal curvature directions and are placed within water surfaces* (Imhof 1975) to ensure legibility.



Symbolization. Symbolization is a common practice in cartography to reflect data and phenomena (Imhof 1972). To date, standardized symbolization for water surfaces has not been established. However, certain conventions have been used over the years, including *(P8) the irregular placement of signatures with area-wide coverage* to communicate water features (e.g., wetland, saltwater vs. freshwater), or the placement of glyphs along streamlines to symbolize the flow direction of rivers.



2. Related Work

The proposed techniques are related to previous works on texture synthesis and illustrative rendering.

Feature-guided Texture Synthesis

From the analysis of the design principles, it is observed that texture features are aligned with shorelines to symbolize the land-water interface. Geometric properties of complex shapes can be reconstructed quite effectively using distance fields (Frisken *et al.* 2000). The proposed methods use distance maps to synthesize waterlines and align stipples with shorelines in real-time. Most algorithms use vector propagation to compute these maps by an approximate Euclidean distance transform (Danielsson 1980) such as jump-flooding (Rong & Tan 2006). For the proposed methods, the work-load efficient *parallel banding algorithm* (PBA) is used instead to compute an *exact* distance transform with a GPU (Cao *et al.* 2010).

Feature-guided texturing based on principal curvature directions can significantly improve shape recognition (Girshick *et al.* 2000). For an overview on this topic, the interested reader is referred to the survey by Wei *et al.* (2009). The relevant approaches identified in Chapter III use normal-ray differential geometry (Y. Kim *et al.* 2008) or diffusion techniques (K. Xu *et al.* 2009) to derive principal curvature directions, or use learning-based approaches (Kalogerakis *et al.* 2012; Gerl & Isenberg 2013) to align textures to salient feature curves. However, these methods either require significant (pre-)processing time or do not provide spatial and temporal coherence. By contrast, the concept of a *feature-aligned distance transform* is proposed that provides continuous Euclidean distance values in a tangential direction to the shorelines. Here, a GPU-based flooding algorithm is used to compute a feature-aligned distance map in real-time. Together with bilinear texture interpolation (Green 2007), this map can be used to parameterize and place texture features along shorelines with frame-to-frame coherence.

Non-Photorealistic Rendering

Coastal vignettes and waterlines are used for cartographic line generalization (Christensen 1999) and in geoinformation systems to improve figure-ground perception. Stippling is a well-studied field in illustrative rendering for digital half-toning. Conventional approaches represent local tone by a well-spaced placement of small dots (Kyprianidis *et al.* 2013). Previous work proposed feature-guided image stippling (D. Kim *et al.* 2008; S. Kim *et al.* 2010) that is adapted to the gradient direction of distance maps. Water stippling works similarly, but does not match density distributions to local tones (e.g., by blue noise, Kopf *et al.* 2006) because dots irregularly overlap with varying density *(P2)*. For this, an enhancement to the texture bombing algorithm of Glanville (2004) is proposed that aligns water stipples with waterlines and renders them in real-time.

Feature-guided hatching has received significant attention in previous works and comprises user-defined (Salisbury *et al.* 1997), patch-based (Praun *et al.* 2000; Webb *et al.* 2002), shading-based (Praun *et al.* 2001), and learning-based (Kalogerakis *et al.* 2012; Gerl & Isenberg 2013) algorithms. In contrast to these approaches, the proposed method obtains texture coordinates by Euclidean and feature-aligned distance maps, which gives more artistic control over parameterizing individual strokes in real-time. For cross hatching, tonal art maps (Praun *et al.* 2001) are aligned to Euclidean distance maps according to the view distance. This way, a continuous LoA is achieved that reduces visual clutter at high view distances. A LoA configuration may also be combined with shape simplifications to produce aesthetic renditions of digital maps (Isenberg 2013).

As reviewed in Chapter III, internal labeling is prominently addressed in the visualization of virtual 3D scenes. Previous works compute geometric hulls (Maass & Döllner 2008) or derive medial axes based on a distance transform (Götzelmann *et al.* 2005; Ropinski *et al.* 2007; Cipriano & Gleicher 2008) for shape-aligned labeling. The method presented here also uses distance maps to align font glyphs with the shoreline distance and orientation.

3. Method

An overview of the system is shown in Figure V.16. The input data consists of a set of 2D or 3D water surfaces that are typically defined as triangular irregular networks. Using orthographic projections, the models' shapes are captured in 2D binary masks to facilitate quantitative surface analysis that works in image space. This analysis includes the computation of Euclidean and feature-aligned distance maps by iteratively propagating distance information in the normal and tangent directions of shorelines. This information is used for cartography-oriented shading, including waterlining, contour hatching, water stippling, and labeling. To enable a continuous LoA, the shading results are parameterized, blended, and mapped onto the surfaces according to the view distance. Because water surfaces are processed separately, the system can be seamlessly embedded into existing rendering systems, or combined with rendering techniques that pre-process the 3D geovirtual environment (e.g., terrain, Buchin *et al.* 2004).

Quantitative Surface Analysis

The goal is to provide quality, interactive illustrations of water surfaces that comply with the identified design principles. This section provides a background on how geometric properties of water surfaces can be derived using distance transforms. It includes a novel feature-aligned distance transform that is used to align individual strokes with the orientation of shorelines.

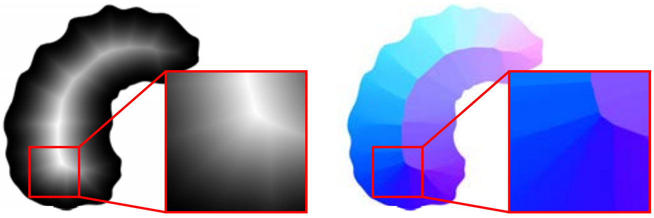


Figure V.17.: Visualization of **(left)** normalized shoreline distances and **(right)** shoreline directions for a given water surface.

Euclidean Distance Transform. Distance information is computed to determine parts of a water surface with equal shoreline distance. Let $I : \mathbb{R}^2 \rightarrow \{0, 1\}$ denote a water surface captured as a binary image, with $I(p) = 0$ marking water areas, and $I(p) = 1$ marking land areas by pixels $p \in I_D$ (Figure V.16 top left). A distance transform of I defined as

$$\omega_I(p) = \min_{q \in I_D}\left(||p - q|| + \chi(q)\right) \quad \text{with} \quad \chi(q) = \begin{cases} 0 & \text{if } I(q) = 1 \\ \infty & \text{otherwise} \end{cases}$$

obtains the minimum Euclidean distance of each pixel to a shoreline. A fast, parallel implementation to compute this information as a *distance map* D (Fi-

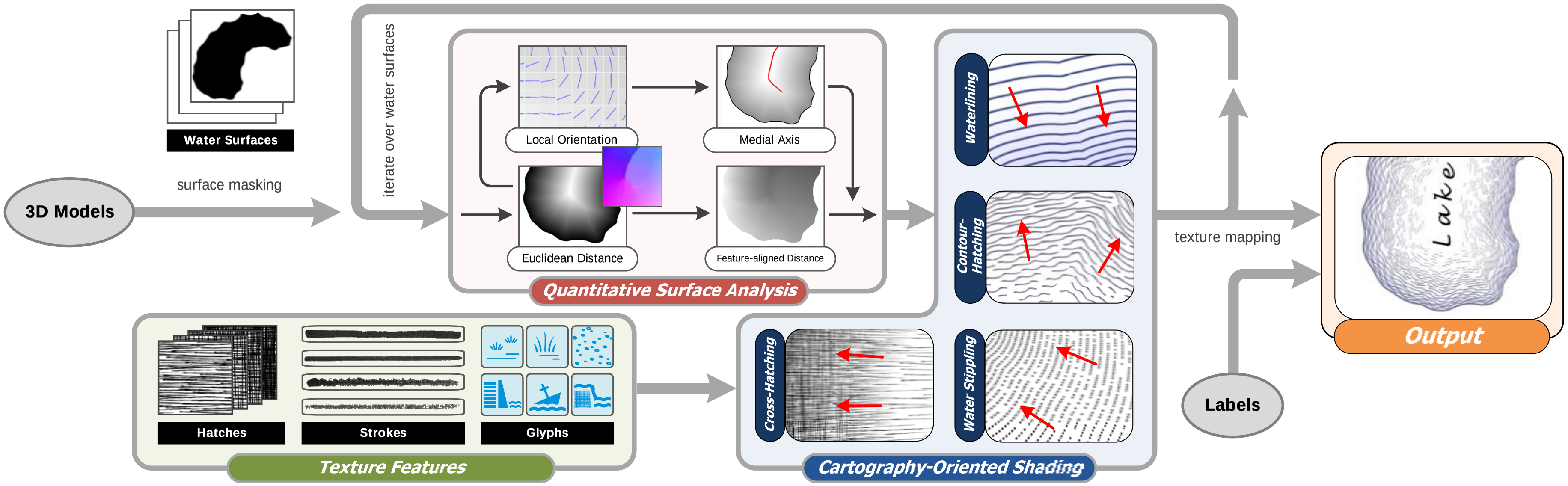


Figure V.16.: Schematic overview of the system, which implements cartography-oriented shading using the results of quantitative surface analysis.

gure V.17) is based on the PBA (Cao *et al.* 2010). Iteratively propagating distance information with the PBA is also performed to obtain the nearest shoreline position as directional information (Figure V.17). Subsequently, $D(p)$ is used as a lookup function for shoreline distances $d \in \mathbb{R}^+$, and $D_b(p)$ to lookup the nearest shoreline position $b \in I_D$.

Local Orientation Estimation. The estimation of local orientation is based on the image gradients of the distance map D. A popular choice to approximate the directional derivatives in x- and y-direction is the Sobel filter which, however, yields non-smooth tangent information on the medial axes because opposite gradients cancel out (Figure V.18). A simple alternative is to use the smoothed structure tensor (Brox *et al.* 2006b) and perform an eigenanalysis to obtain gradient and tangent information. This leads to more stable estimates of the local orientation (Figure V.18).

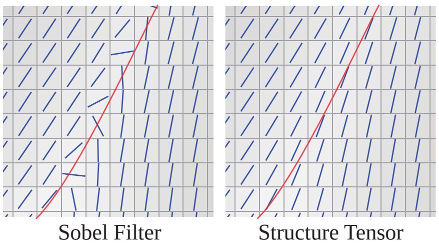


Figure V.18.: Visualization of tangential fields derived from a distance field.

Medial Axes Computation. The medial axes are derived from the distance map D (Cao *et al.* 2010) to align design elements (e.g., labels) along the middle stream of water surfaces. The medial axes are obtained by comparing and thresholding the directions to the nearest shorelines in the local neighborhood for each $p \in I_D$. For this, the unsigned gradient orientation $n \in \mathbb{R}^2$ of the smoothed structure tensor is used:

$$b^+ = ||p - D_b(p+n)||\,, \quad b^- = ||p - D_b(p-n)||\,.$$

Thresholding the angle between b^+ and b^- then yields an approximate of the medial axes:

$$\arccos\left(b^+ \cdot b^-\right) > \theta \in [0, \pi].$$

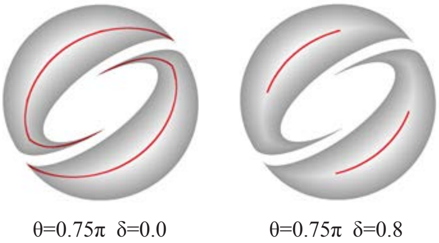


Figure V.19.: Thresholds used to adjust the proposed algorithm for medial axes computation.

In addition, the shoreline distance is thresholded by $\delta \in \mathbb{R}^+$ to avoid placing design elements too close to shorelines. For all the examples, values $\theta = 0.75\pi$ and $\delta = 0.8$ are used (Figure V.19).

Feature-aligned Distance Transform. Contour hatching for water surfaces is a complex problem, since the properties of individual strokes (e.g., length, spacing) vary with the shoreline distance *(P4)*. A typical approach is to define orientation fields on a surface to guide an example-based texture synthesis to salient feature curves (Wei *et al.* 2009). Yet animating individual strokes on water surfaces requires fine control over the parameterization and placement per rendering pass, in particular to simulate water movements. The proposed approach parameterizes the level-set curves of distance map D to obtain Euclidean distance values along its tangential field (Figure V.20). By parameterizing these *feature-aligned distances* (v-coordinate) and the shoreline distances (u-coordinate), they can be directly used as texture coordinates for real-time shading (Figure V.21).

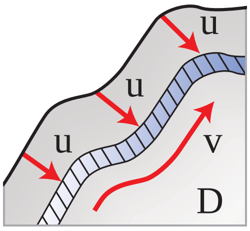


Figure V.20.: Schematic overview of a feature-aligned level-set curve.

Algorithm. For computing a *feature-aligned distance map T*, an approach similar to vector propagation (Danielsson 1980) is used. Using the non-normalized distance map D, level sets correspond to the integral part of the shoreline distances, e.g., $\lfloor d \rfloor = 0$ for the zero level sets. Starting with the shorelines, random pixels are selected as seed points from which (1) Euclidean distances are

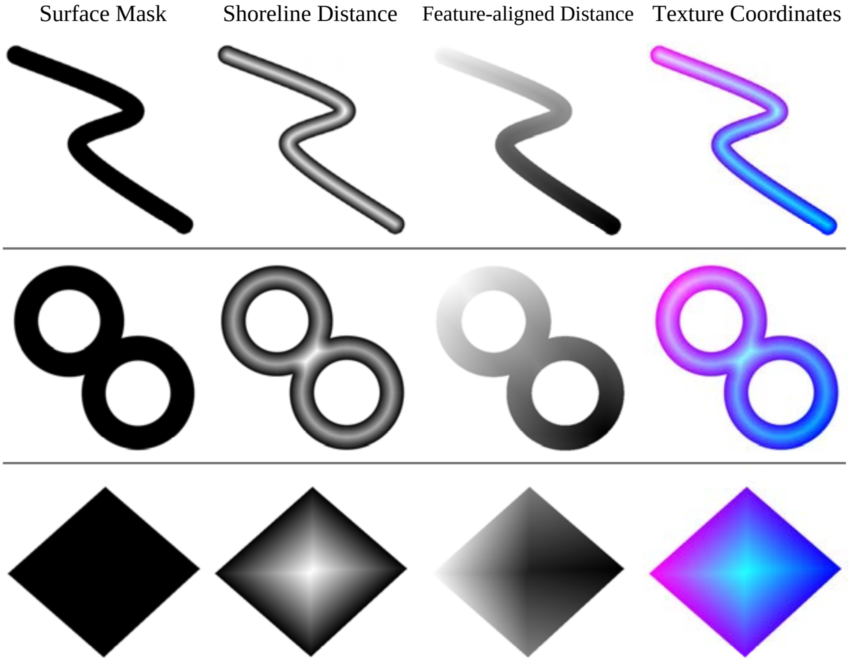


Figure V.21: Exemplary distance maps computed by the proposed system.

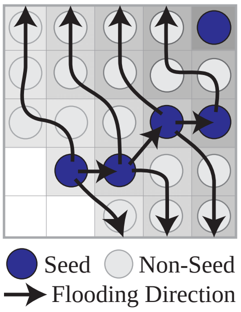


Figure V.22.: Schematic overview of the proposed feature-aligned flooding.

propagated along the level sets, and (2) seed point information is propagated to the inner level sets (Figure V.22). These two steps are repeated for each level set until no more pixels are available for processing. A parallel algorithm was implemented by iteratively flooding distance information within the local neighborhood (e.g., 3 × 3) of a pixel, which dynamically propagates seed points during distance map construction (Figure V.23). An efficient parallel algorithm for normalization of this map is based on a reduction (Nehab *et al.* 2011).

Discussion. Exemplary results show that the approach provides continuous feature-aligned distance values (Figure V.21 and Figure V.23). Contrary to texture synthesis based on energy minimization (e.g., K. Xu *et al.* 2009), the proposed method does not provide continuity across the level sets of a surface. But since individual texture features (e.g., hatches) are aligned with the level-set curves, no such constraint is required. This allows to perform an exact distance transform since no compression or stretching is required to meet continuity in all major directions of a surface. Because texture features can only be placed on the level sets, choosing an adequate distance map resolution is important to balance rendering quality and performance. On the one hand, vector propagation along the level sets performs non-linearly with the map resolution, and optimization techniques known from jump flooding (Rong & Tan 2006) are less helpful since highly curved sections require small step widths. On the other hand, too low map resolutions insufficiently approximate the shorelines' directions. As a

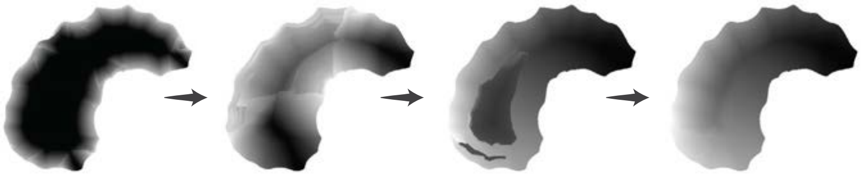


Figure V.23: Intermediate results of the proposed algorithm that iteratively computes a feature-aligned distance map (256 × 256 pixels, 503 iterations).

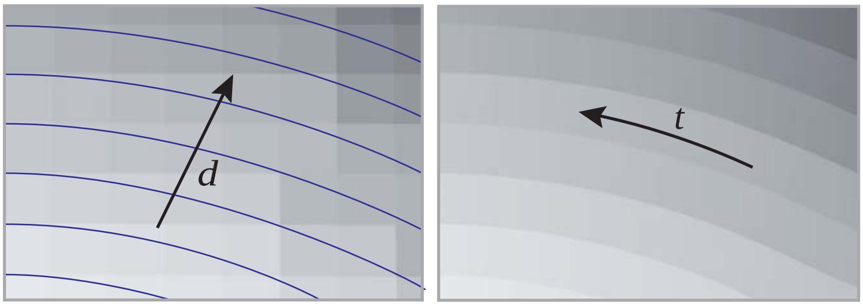


Figure V.24: Bilinear sampling of a feature-aligned distance map.

(left) nearest neighbor sampling

(right) proposed bilinear sampling

compromise, bilinear sampling is used for a piecewise-linear approximation of shoreline and feature-aligned distances. This approach has been proven effective for the magnification of glyph contours, even with low-resolution distance maps (Green 2007). Because bilinear sampling accurately reconstructs distance information (Frisken *et al.* 2000), feature-aligned distance maps of up to 256 × 256 pixels can be used for rendering and computed in real-time using the proposed GPU-based flooding algorithm—refer to the evaluation at the end of this section.

Bilinear Sampling. In contrast to signed distance maps, the accurate reconstruction of feature-aligned distance values requires a modified version of bilinear sampling to avoid filtering across the level sets of D. For this, the shoreline distance d for a point $p \in I_D$ is determined and compared to the distance information of the four samples p_0 to p_3 (Figure V.25):

$$T(p) = A + B - ((1-\delta_y)Bg(A) + \delta_y Ag(B))$$
$$A = (1-u_1)p_0 + u_1 p_1 \quad u_1 = (g(p_0)(\delta_x - 1) + 1)g(p_1)$$
$$B = (1-u_2)p_2 + u_2 p_3 \quad u_2 = (g(p_2)(\delta_x - 1) + 1)g(p_3)$$

where $g(q) = 0$ with $q \in \{p_0, p_1, p_2, p_3\}$ if p, q correspond to different level sets to omit a pixel from interpolation, otherwise $g(q) = 1$. This modified version provides continuous feature-aligned distance values on the level sets (Figure V.24).

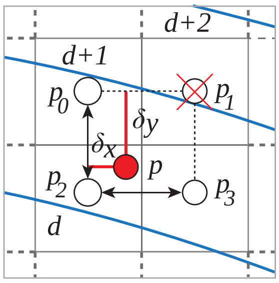


Figure V.25.: Schematic overview of the modified bilinear sampling.

Shading Techniques

The results of the quantitative surface analysis are used for cartography-oriented visualization. The following techniques are shading-based and bilinearly sample distance maps to accurately reconstruct distance information. They can be parameterized in terms of tone and density to provide a view-dependent LoA.

Waterlining and Water Stippling In the following, a non-normalized version of distance map D is used to independently apply waterlining and stippling from a water surface's scale. To comply with non-equidistant interspaces *(P1)*, target distance values $\varphi(d)$ are computed using a non-linear step function:

$$\varphi(d) = \frac{(\lfloor (s \cdot d)^e + h \rfloor - h)^{1.0/e}}{s}.$$

The spacing of φ is parameterized by $e, s \in \mathbb{R}^+$ to define a corresponding number of steps in the interval of D (Figure V.26). In addition, these steps can be shifted along d using $h \in [0, 1]$.

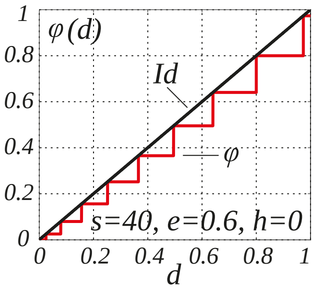
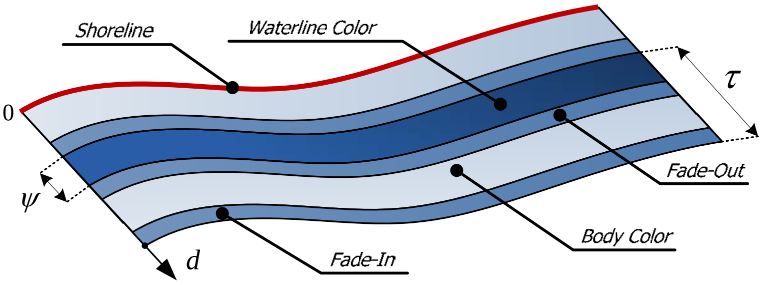


Figure V.26.: Non-linear step function $\varphi(d)$.

Figure V.27: Parameters used for the design of waterlines, adapted to the shoreline distance.



Waterlining. Waterlines correspond to shaded areas of a water surface with equal shoreline distance. To render waterlines, the distances $\tau \in \mathbb{R}^+$ between the positions obtained by $\varphi(d)$ are thresholded by a corresponding width $\psi \in \mathbb{R}^+$, and padded by fade-in and fade-out intervals (Figure V.27) for anti-aliasing, and to provide smooth transitions. At high view distances, this reduces the number of rendered waterlines and provides a smooth transition while zooming.

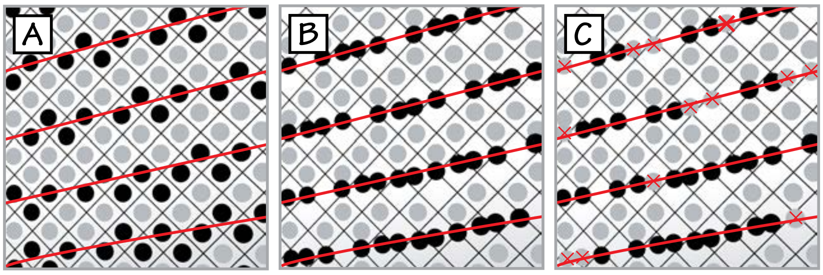
Water Stippling. Water stippling refers to placing small dots with irregular distribution along waterlines to convey shape and motion. The proposed algorithm uses an enhanced variant for *texture bombing* (Glanville 2004) to place water stipples with feature-aligned distribution and irregular density. The basic idea of texture bombing is to randomly place glyphs in regularly distributed grid cells. This algorithm is extended by three phases; *stipple selection*, *stipple displacement*, and *stipple filtering* (Figure V.28). Instead of using a random placement of stipples, offsets are computed that align them with the waterlines of D. The algorithm starts with stipples that are centered in regularly distributed grid cells and mapped onto a water surface (Figure V.28A).

1. ***Stipple Selection:*** Stipples within grid cells that cross a waterline were selected for further processing (Figure V.28A). For this, the distance to the next waterline $(d - \varphi(d))$ is thresholded.

2. ***Stipple Displacement:*** The gradient direction of D is used to compute the approximate target position to the nearest waterline. If origin and target positions correspond to the same grid cell (first phase), a stipple is displaced towards the target position. This results in stipples lined up with the waterlines (Figure V.28B).

3. ***Stipple Filtering:*** The displacement of stipples in the gradient direction of D increases the irregular distribution along waterlines. To render stipples with non-regular intervals, noise or pseudo-random numbers can be used for additional filtering (Figure V.28C and Figure V.29).

Figure V.28: Schematic overview of the water stippling phases: (A) selection, (B) displacement and (C) filtering. Waterlines are marked as red lines, rendered stipples as black dots.



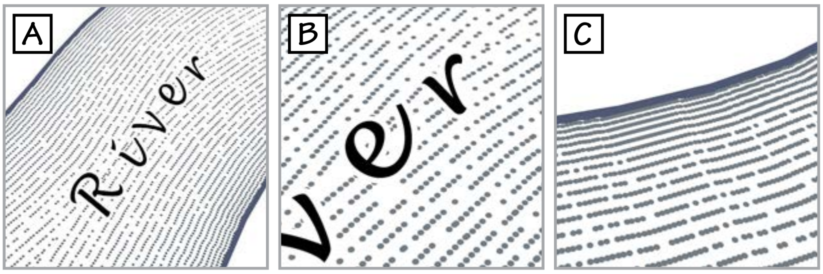


Figure V.29: A result of the water stippling technique showing a non-linear, feature-aligned, and irregular distribution of stipples.

Up to this point, stipples might be rendered with low density because waterlines force them to split into multiple directions. To regularize the density near coastal areas, the phases 1-3 are repeated to place additional layers of stipples within slightly shifted grid cells. It is observed that two layers are sufficient to meet this requirement (Figure V.29C). Experimentation with different parameters may also yield locally varied stipple densities and tones, for example using tonal art maps to symbolize flow velocity *(P3)*. In addition, iterative application of the algorithm with shifted step functions can be used to indicate highlights or shadowed areas. For instance, a second pass with $\varphi(d)$ shifted halfway by $h = 0.5$ can be used to place additional stipples at occluded areas to improve depth perception of a virtual 3D scene *(P3)*.

Contour Hatching. To symbolize water movements, a novel contour hatching technique is developed. Once a feature-aligned distance map is computed, individual stroke maps are irregularly placed with non-linear distance to shorelines to express motion (Figure V.30). Similar to the hatching technique of Kalogerakis *et al.* (2012), parameters are defined to provide artistic control over this placement:

- *Length* ($l \in \mathbb{R}^+$) defines the length of a stroke.
- *Thickness* ($t \in [0, 1]$) defines the width of a stroke.
- *Spacing* ($s \in [0, 1]$) controls the stroke density.
- *Randomness* ($r \in [0, 1]$) controls the stroke irregularity.

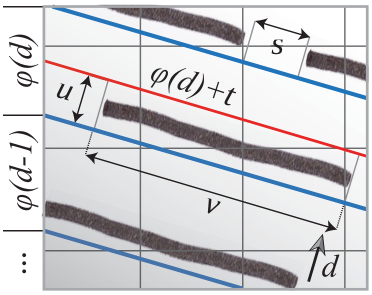


Figure V.30.: Schematic overview of contour hatching that is aligned to the distance maps D and T.

The main idea is to derive texture coordinates u, v for each rendering fragment by bilinearly sampling the distance maps D and T. To obtain the u-coordinate, waterline positions and the stroke width t are used to compute the fraction $u = (d - \varphi(d))/t$. To obtain the corresponding v-coordinate, feature-aligned distance values of the sampled distance map T are scaled by l to match the desired stroke length. In addition, noise is used to clamp the v-coordinate for an irregular placement of individual strokes, and the s and r parameters for density

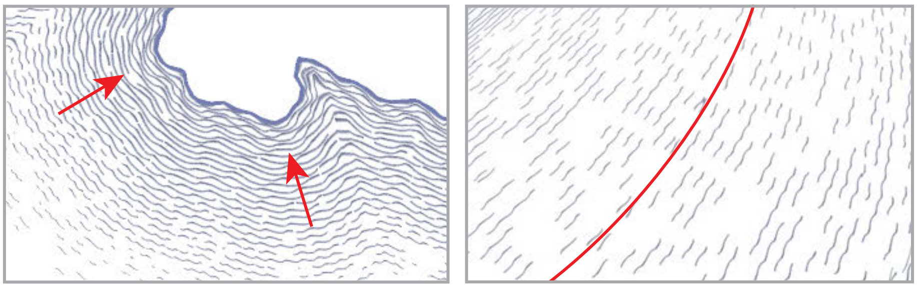


Figure V.31: Contour hatching for water surfaces: (left) dense, wavy strokes near shorelines, (right) loose strokes near the medial axes.

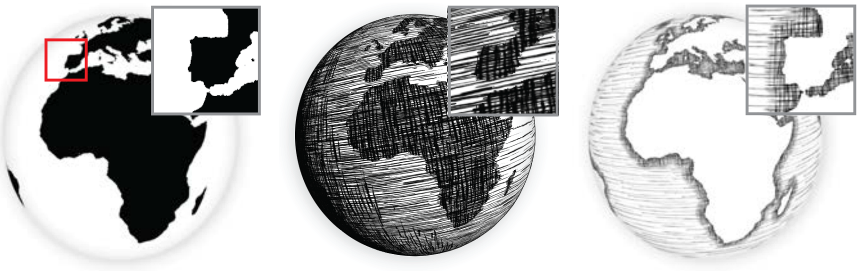


Figure V.32: A 3D virtual globe shaded by cross-hatched strokes: (left) binary mask, (middle) result of Webb *et al.* (2002), (right) proposed method.

control and filtering. To render contour hatches excessively wavy *(P5)*, texture maps are used that had been digitized from hand-drawn strokes. Because the stroke placement works in object space, it provides frame-to-frame coherence, and avoids the *shower door effect* known from techniques that work in image space, e.g., the technique of Y. Kim *et al.* (2008).

From the analysis of design principles, it is observed that stroke layers of varying tone and density are used based on the shoreline distance *(P4)*. This observation can be modeled by the algorithm using 3 layers with different parameter sets: dense, solid strokes near shorelines, loose strokes with irregular density, and strokes with shorter lengths near the medial axes (Figure V.31). Because the technique is texture-based, water movements can be modeled quite easily by shifting individual strokes along the major directions of the distance maps D or T, for instance by a temporal displacement of the v-coordinate using a sine function to animate rivers.

Water Vignetting and Cross Hatching. In modern cartography, water vignettes are based on color gradients. A simple approach is to threshold the shoreline distance and interpolate between a shoreline and water body's color. This effect is used to complement the waterlining technique. A similar effect can be achieved by cross hatching the shoreline areas by a tonal art map (Praun *et al.* 2001). A map of five varying levels of stroke size and density is used, which are blended and mapped on water surfaces according to the shoreline distance, and parameterized according to the view distance to create a continuous LoA. In contrast to tone-based shading, this approach does not affect shading of landmass (Figure V.32), which allows to visualize additional geospatial features such as terrain more easily.

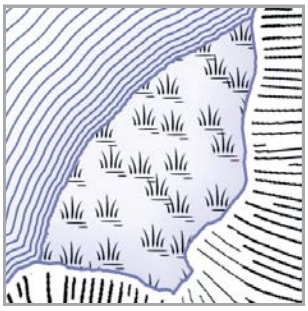


Figure V.33.: Visualization of thematic information using glyphs, here: to indicate flooded areas.

Thematic Visualization. Distance maps are used for internal labeling that complies with the design principles of cartographers *(P6/P7)*. The main idea is to derive piecewise cubic Bézier curves from the medial axes of the distance map D and warp text to these curves accordingly (Ropinski *et al.* 2007; Cipriano & Gleicher 2008). To obtain the control points, pixels of the medial axes are traced and iteratively downsampled in image space by nearest-neighbor interpolation. Together with the tangent information of the structure tensor, arc-length parameterization is used to warp text with the flow direction of water surfaces, and align it with the viewing direction (Figure V.34). For symbolization, *texture bombing* is used and parameterized so that signatures always face the view direction when viewed in 3D spaces (Figure V.33) and to comply with *(P8)*. Alternatively, an example-based approach may be used to arrange signatures with more artistic control, for which the reader is referred to the work of Hurtut *et al.* (2009).

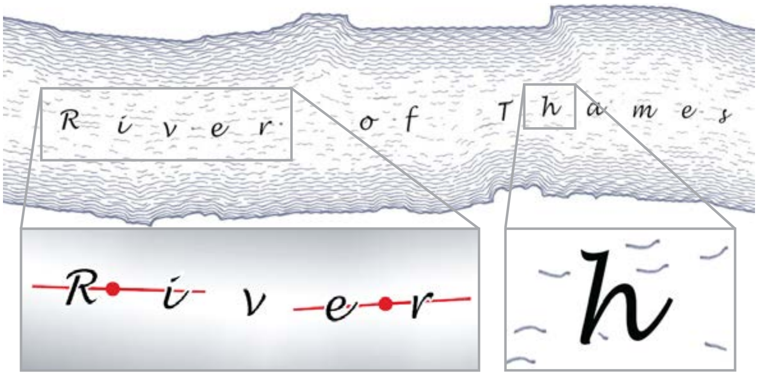


Figure V.34: Exemplary result of the proposed labeling algorithm that aligns font glyphs according to the shoreline distance and orientation.

4. Results

The system was implemented using C++ and OpenGL/GLSL. OpenSceneGraph was used as the rendering engine to handle 3D data sets. The operations of the quantitative surface analysis are designed for parallel execution, and were implemented in CUDA to significantly improve the rendering performance. In particular, the PBA algorithm is used for the computation of Euclidean distance maps (Cao *et al.* 2010), together with a reduction for normalization (Nehab *et al.* 2011). NVidia's OpenGL extension `NV_path_rendering` is used to enable the rendering and transforming of high quality, instance-based text in a single pass.

Applications

Figure V.35 shows a comparison of the rendering techniques using an example of *Spirit Lake* (at *Mount St. Helens*, USA). It is observed that waterlining is a functional illustration technique that is able to communicate distance information quite effectively. The effects of water stippling and contour hatching are similar but add a sense of motion and uncertainty. Coastal vignetting, by contrast, primarily focuses on the land-water interface itself to improve figure-ground perception. These techniques complement other cartography-oriented shading techniques quite well. This is demonstrated by shading the terrain in the environment with hachures of varying thickness according to the slope steepness (Buchin *et al.* 2004). Moreover, the rendering techniques provide a LoA, which is shown in Figure V.38 and the right image of Figure V.35, where more or less features are depicted according to the view distance to reduce visual clutter. These examples also indicate the capability of the techniques to handle 3D scenes. Because the rendering techniques are texture-based, they are independent from a model's geometric complexity, and provide spatial and temporal coherence. Finally, evaluations were made to use the rendering techniques concurrently. For instance, water stipples can be seamlessly blended with waterlines according to the view distance; or coastal vignetting to complement waterlining (Figure V.15).

The waterlining technique may be useful in flooding simulations to assess distances to safety zones. When performed over time, waterlines dynamically shift with the flood distribution to convey motion and enhance the depiction of land cover. This effect is exemplified in Figure V.36 for the city of Boston. Here, a virtual plane is temporally shifted upwards to represent the change in the mean sea level. Using this plane as a clipping mask with an orthographic projection, the corresponding flooded areas are obtained and shaded in real-time.

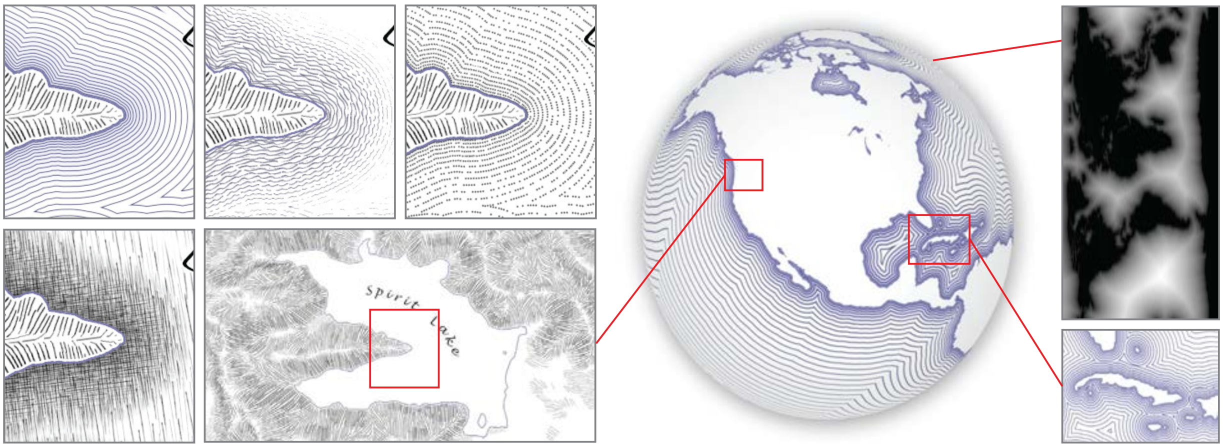


Figure V.35.: Exemplary results of the rendering techniques for 3D mapping, (left) compared with each other within the environment of Mount St. Helens, (right) waterlining applied to a globe that provides a continuous LoA when zooming in and out.

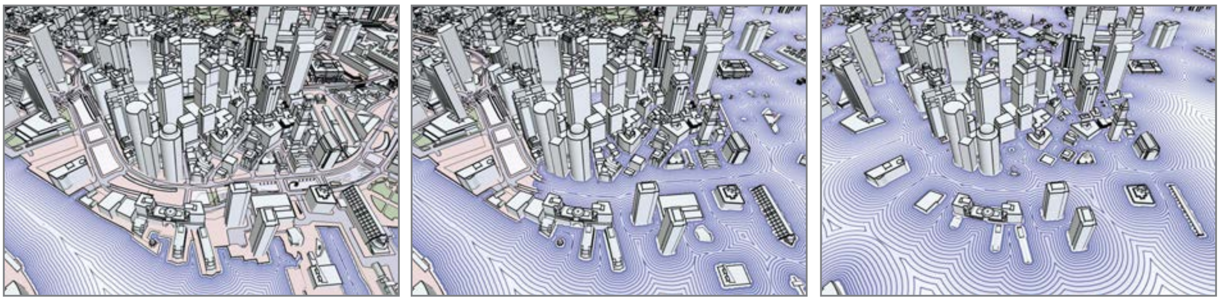


Figure V.36.: Flooding simulation for the city of Boston enhanced by the waterlining technique and illustrative rendering to express uncertainty.

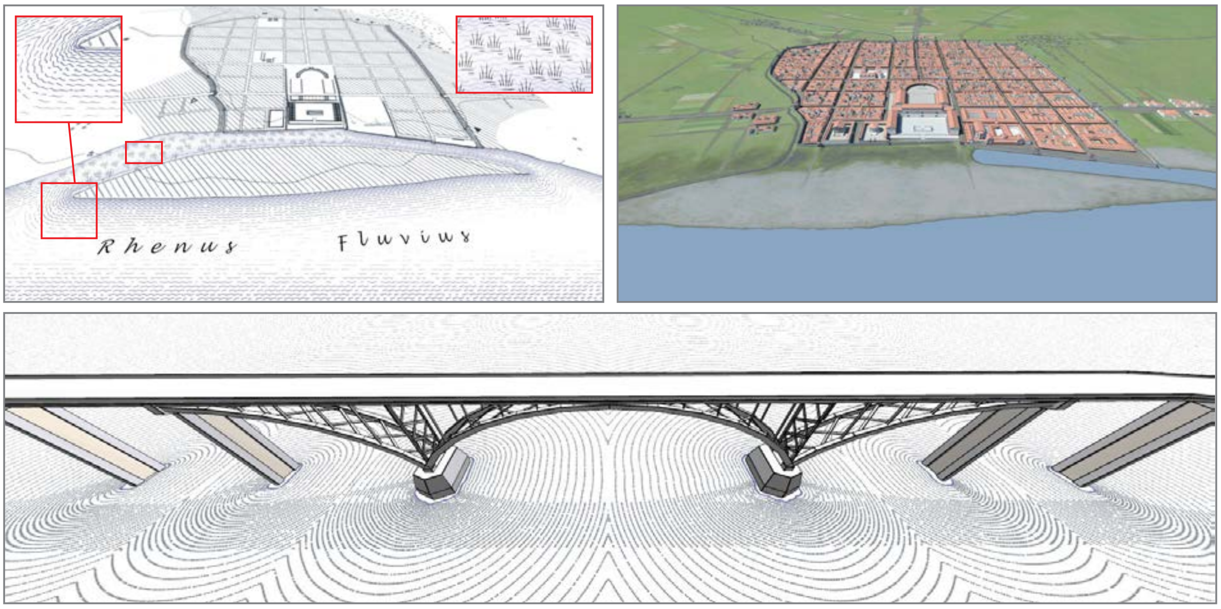


Figure V.37.: Further results of the proposed rendering techniques for urban planning and landscaping: (top) contour hatching used to express uncertainty in a reconstructed topographical model of ancient Cologne (Germany), which served as the basis for the 3D reconstruction shown to the right, (bottom) water stippling used to enhance the rendering of a bridge construction.

Table V.3.: Performance evaluation (in *milliseconds*): distance (D) and feature-aligned distance transform (T), orientation and medial axis computation.

Image Res.	D	T	Orient.	M. Axes	Total
128 × 128	1.6	26.6	0.1	0.7	29.0
256 × 256	2.2	96.3	0.2	0.8	99.5
512 × 512	4.5	371.6	0.6	0.9	377.6

Figure V.37 demonstrates the application of the rendering techniques to urban planning and cultural heritage. Within these domains, it is often desired to avoid authentic impressions, in particular because of missing evidence in the (re)construction or because construction plans may be altered in the future. The top image shows a topographical reconstruction of ancient Cologne (Germany), in which contour hatching is used to express uncertainty. Here, the individual strokes are animated to express water movements. In addition, symbolization is used to highlight those river areas that were flooded in ancient times. The bottom image shows a bridge construction, where water stippling is used to add expressiveness. Here, the stipple density is increased in the shadowed areas to integrate cues that ease depth perception.

Performance Evaluation

Table V.4.: Performance evaluation of the illustrative rendering techniques for different screen resolutions (in frames per second).

Screen Res.	waterlining	stippling	contour hatching
800 × 600	534	162	159
1280 × 720	523	88	84
1600 × 900	521	59	55
1920 × 1080	514	42	41

The performance tests of the system were conducted on an Intel® Xeon™ 4× 3.06 GHz with 6 GByte RAM and NVidia® GTX 660 Ti GPU with 2 GByte VRAM. *Spirit Lake* (Figure V.35) was used as a test model. The results in Table V.3 show that the run-time of the quantitative surface analysis scales with the resolution of a distance map. The feature-aligned distance transform is shown to be a limiting factor; however, its implementation is not heavily optimized and there is potential to increase the performance. Different sizes of distance maps are compared for the illustrative rendering techniques. Similar to signed distance maps (Green 2007), stable results are achieved when bilinearly sampling a low resolution of a feature-aligned distance map (128 × 128 pixels). Table V.4 shows that the illustrative rendering techniques perform at real-time frame-rates in HD resolution. During rendering, it is observed that the shading techniques are fill-limited and achieve, in SD resolution, twice the performance of an HD resolution. The timings indicate that the system for feature-aligned waterlining, stippling, and hatching performs in real-time, and therefore is applicable to render animated 3D scenes.

Limitations

The shading techniques work in object space and texture space respectively, and the computation of distance maps requires closed polygons for processing. For large-scale water surfaces of complex shape, distance maps of high resolution are required to achieve quality shading results. Here, memory resources limit the distance map sizes that can be processed by a GPU; for the evaluated system ≈ 16 Megapixels (MP) with 2 GByte VRAM. Moreover, it is observed that the feature-aligned distance transform does not perform in real-time when computing distance maps with > 0.25 MP. Nonetheless, empirical evaluations showed that distance maps of 256 × 256 pixels are typically sufficient when using bilinear texture sampling.

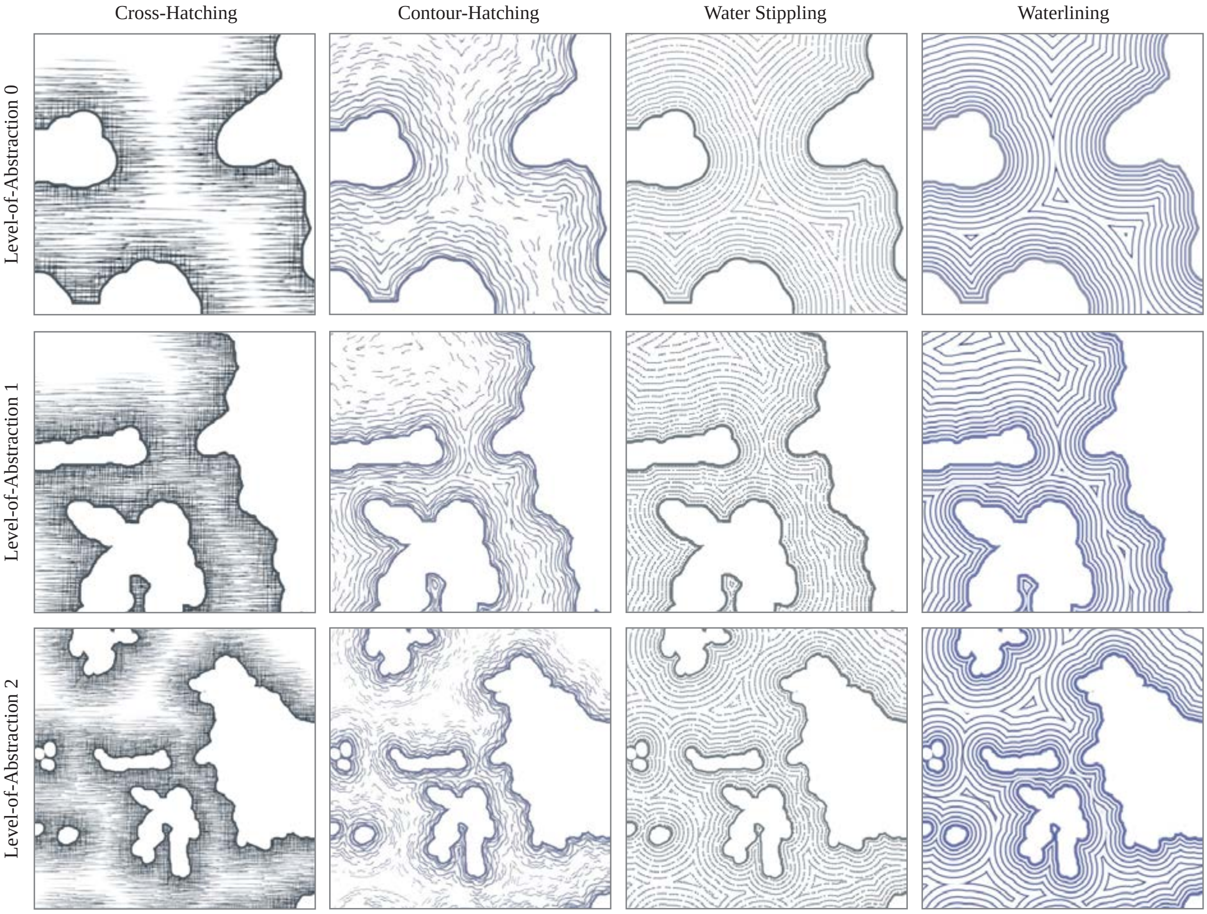


Figure V.38.: Exemplary parameterization results of the rendering techniques for a view-dependent LoA. In order to reduce visual clutter at high view distances (bottom row), the number of rendered waterlines, stipples and hatches are reduced significantly. Once these parameterizations have been authored by the system, blending between them is performed in real-time using OpenGL fragment shaders.

3. Summary

This chapter proposes rendering techniques for cartography-oriented visualization of geospatial feature types in 3D geovirtual environments. The generalized and feature-specific techniques implement style variations that follow design principles found in cartography. They demonstrate the capability to shift the design process from pre-processing to the interactive stages of the visualization pipeline, which is promising for digital mapping services to provide an accomplished context-aware visualization, e.g., sensitive to the environment and purpose of visualization (user's task). In particular, the view-dependent visualization techniques for building models and transportation networks show how to express uncertainty and how to implement view-adaptive rendering that facilitates interactive filtering and highlighting. In addition, the real-time rendering techniques for water surfaces demonstrate how to adopt design principles from traditional cartography to improve figure-ground perception and express a sense of motion. Thereby, a novel feature-aligned distance transform is proposed to align individual strokes with the shorelines of water surfaces. Because all techniques work in object space or include mechanisms of texture mapping, temporal and spatial coherence is achieved.

VI

Interactive Level-of-Abstraction Transitions

The work presented in this chapter is partly based on

Semmo, A., Trapp, M., Kyprianidis, J. E. & Döllner, J. "Interactive Visualization of Generalized Virtual 3D City Models using Level-of-Abstraction Transitions". *Computer Graphics Forum* **31,** 885–894 (2012b)

Semmo, A. & Döllner, J. *An Interaction Framework for Level-of-Abstraction Visualization of 3D Geovirtual Environments.* in *Proc. ACM SIGSPATIAL Workshop on MapInteraction* (2014a), 43–49

Pasewaldt, S., Semmo, A., Trapp, M. & Döllner, J. "Multi-Perspective 3D Panoramas". *International Journal of Geographical Information Science* **28,** 2030–2051 (2014)

Visualization systems typically apply a homogeneous graphic style to depict 3D geospatial models in virtual environments: photorealistic rendering is usually used for detailed presentations, or illustrative, abstract rendering to draw attention to prioritized information (Santella & DeCarlo 2004). Using a suitable graphic style can be beneficial for making a visualization meaningful in its corresponding context and usage scenario (A. MacEachren 1995). For instance, detailed presentations can aid the exploration of local environments, whereas 2D maps can be an effective medium for navigational purposes.

Systems like Google Maps or Bing Maps integrate different graphic styles to serve users with a presentation suitable for viewing maps or getting driving directions. Because these systems provide high interactivity, a user's task and context, such as viewing situations and regions of interest (RoIs), can be dynamically changed. Typically, a user is able to switch the graphic style to display more or less detail in RoIs or context regions to avoid cluttered information. However, concurrent visualization leads to constant reorientations and additional cognitive load (Jobst & Döllner 2008a) because of hard transitions between the graphic style, LoD, and view perspective. Therefore, a great potential lies in the seamless combination of graphic styles into a single view to communicate only relevant information and direct a viewer's gaze by salient stimuli attraction.

A seamless combination of generic 2D and 3D graphic styles in a visualization pipeline by means of computer graphics is yet to be achieved. One approach is to select a LoA in a context-dependent way. The relevant focus+context techniques reviewed in Chapter III use image blending or deformation to highlight RoIs, but do not provide different LoAs for selected entities and prioritized information, blend only dual graphic styles, or are domain-specific, e.g., with respect to routing (Qu *et al.* 2009). This motivates a system approach that is designed to integrate customized graphic styles in a context-dependent way.

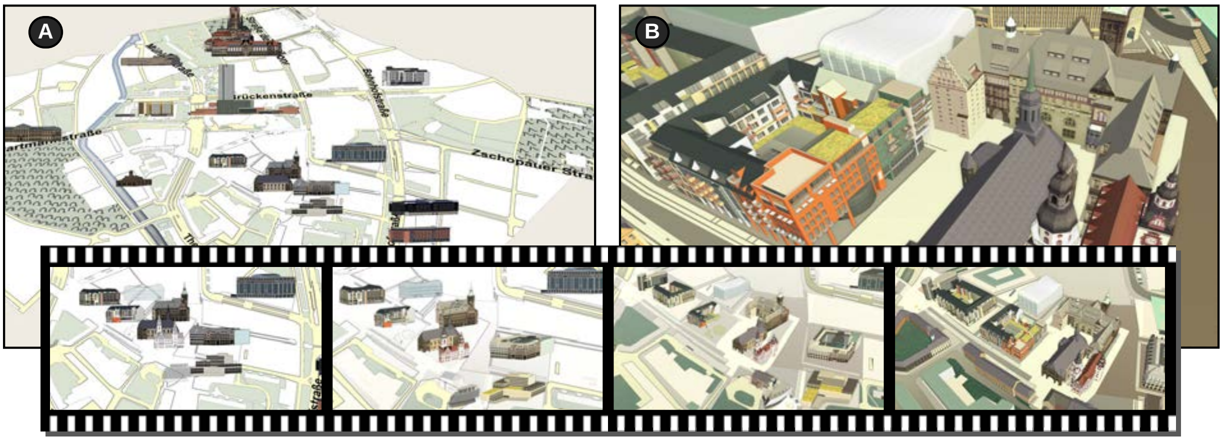


Figure VI.1.: Exemplary result of the proposed visualization system that enables the seamless transition between abstract graphics (A) and a photorealistic version (B) view-dependently. The sequence below shows single frames of this transition.

This chapter presents a concept and an implementation for a system that enables different graphic styles, their seamless integration, and parameterized transitions. The system selects the LoA used to represent 3D model entities in a task-dependent, view-dependent, and resolution-dependent way (Figure VI.1). Being based on shader technology and multi-pass rendering, the system seamlessly integrates into common visualization pipelines, providing context-dependent visualization for novel visualization techniques and geoinformation systems. The system can be further used to author and visualize smooth LoA transitions to improve important applications in geovirtual environments: in particular, map viewing and wayfinding. To summarize, this chapter makes the following contributions:

1. A concept and an implementation for a system that enables seamless combinations of various 2D and 3D graphic styles.
2. A model for the parameterization of transitions of graphic styles in a visualization pipeline (Figure VI.2).
3. Usage scenarios based on the cartography-oriented graphic styles proposed in Chapter V to demonstrate the benefits of the system.

1. Background

Visualizing virtual 3D city and landscape models by LoA transitions is related to context-aware abstraction, animated transitions and geomorphing.

Context-aware Visualization

The presented system extends approaches found in focus+context visualization (Chapter III) to (1) enable smooth transitions between levels of structural abstraction with (2) a context-dependent selection of LoAs using metrics defined per feature type, and (3) their dynamic parameterization at run-time. In addition, it provides cartography-oriented, thematic visualization using different LoAs for selected model entities or RoIs. Further, the system maintains cartographic relations in stylized foci and visualizes with cartography-oriented design to reduce visual clutter in context regions and support a saliency-guided visualization approach. Because the system seamlessly integrates into a visualization pipeline and generalizes context regions, it can be used to enhance focus+context zooming (Qu *et al.* 2009) and multi-perspective rendering techniques (Lorenz *et al.* 2008; Möser *et al.* 2008; Pasewaldt *et al.* 2011).

Level-of-Abstraction Transitions

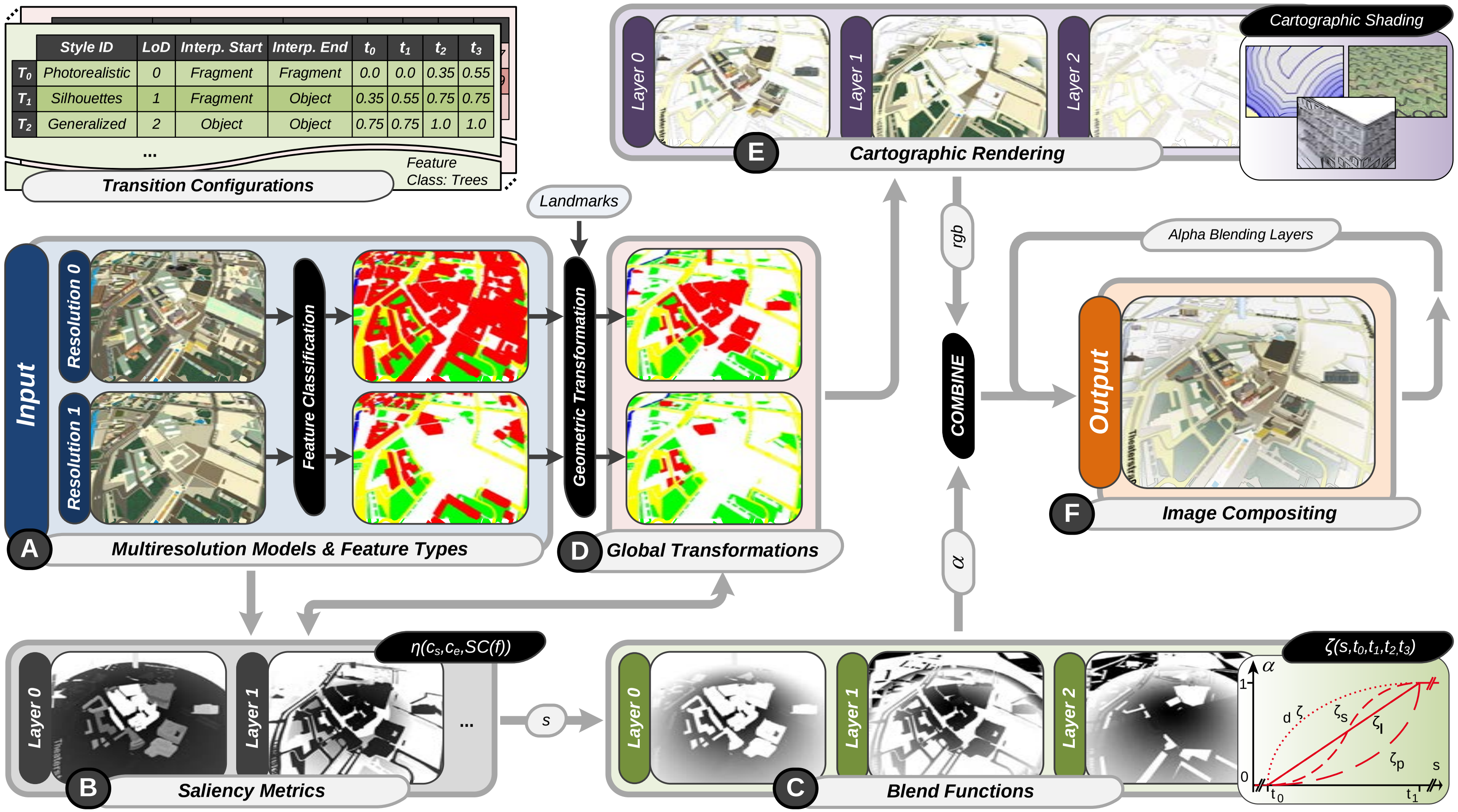
A well-known method for image compositing is alpha blending (Porter & Duff 1984), which is used in multi-perspective rendering to enable a "quasi"-continuous transition between focus and context regions (Lorenz *et al.* 2008; Möser *et al.* 2008; Pasewaldt *et al.* 2011). The proposed system uses cumulative alpha blending to blend multiple RoIs with varying LoA. An alternative approach for smooth transitions is to animate visual and structural changes. Previous work in information visualization showed that animated transitions ease orientation and guidance (G. G. Robertson *et al.* 1993; Tversky *et al.* 2002; Heer & G. Robertson 2007), and aid the reconstruction of information spaces (Bederson & Boltman 1999). Moreover, animated transitions "improve graphical perception of changes between statistical data graphics" (Heer & G. Robertson 2007), facilitate understanding and increase engagement. In the proposed system, global deformations of 3D building models are animated using the approach described in Chapter V to enable predictable transitions between a detailed and cartographic visualization of landmarks (Elias *et al.* 2005).

Morphing is a visual effect that enhances animations by smooth transitions between models with varying resolution (A. W. F. Lee *et al.* 1999; Seitz & Dyer 1996). For example, geomorphing is used to provide smooth transitions and temporal coherence for LoD rendering of terrain models (Hoppe 1998; Wagner 2003). However, morphing is based on assumptions about geometric representations and can only be applied to models with a "suitable" geometry. Virtual 3D city and landscape models, in general, cannot fulfill such assumptions. Moreover, morphing of the models has to take cartographic generalization (A. MacEachren 1995) into account. Previous work on continuous LoD exemplified how smooth, view-dependent transitions can be achieved using collapsing as the pre-dominant generalization operator (Lindstrom *et al.* 1996; Hoppe 1998; Kada *et al.* 2015).

2. Method

An overview of the system presented in this chapter is shown in Figure VI.2. The input data consists of textured multiresolution 3D models (Figure VI.3A) and task-dependent transition configurations. These models are typically defined as triangular irregular networks, e.g., acquired by remote sensing, procedural generation, or manual modeling. A 3D scene is composed of *3D objects* that are categorized using a taxonomy, and grouped according to their appearance. Based on this information, the system performs visual abstraction:

- by geometric transformation of objects (LoD),
- and by cartographic shading (LoA, such as waterlining, signatures for green spaces, and abstract building façades (Chapter V).



	Style ID	LoD	Interp. Start	Interp. End	t_0	t_1	t_2	t_3
T_0	Photorealistic	0	Fragment	Fragment	0.0	0.0	0.35	0.55
T_1	Silhouettes	1	Fragment	Object	0.35	0.55	0.75	0.75
T_2	Generalized	2	Object	Object	0.75	0.75	1.0	1.0
	...							

Figure VI.2.: Overview of the system's approach of LoA transitions. (A) Classification using semantic information, (B/C) blend value computation based on view-related metrics, (D) global transformation of landmarks, (E) shading, (F) order-independent image blending and compositing. Transition configurations are used by components (B-F).

To perform context-dependent visualization, objects are rendered with different graphic styles that are continuously blended. To each graphic style of a certain object used for visualization, *interest values* are assigned that are computed at rendering time based on user-related metrics such as view distance, view angle, or region interest. The interest values are computed for all visible and non-visible (i.e., occluded) fragments of a 3D scene object. After a normalization, these are used as *blend values* to compose the final image by order-independent image blending (Porter & Duff 1984). The remainder of this section describes the stages of the transition pipeline and its architecture.

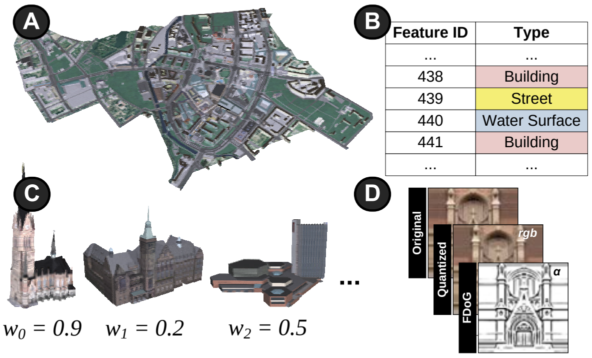
1. Pre-processing

To enable LoA transitions for complex scenes, such as virtual 3D city models, global information about a model's contents is required: object types, their location in the 3D scene, global interests, and how visual attributes adapt to user interaction—referring to the *intelligence of objects* (A. M. MacEachren *et al.* 1999). The pre-processing of this information is explained in the following.

Scenario Definition

The system is based on *usage scenarios* that define how a 3D scene is visualized for a given task and how graphics are dynamically adapted to a user's context. A scenario consists of a set of objects with unique interest values and a set of *transition configurations*. To enable a parameterization of graphic styles for each object, information about an object's type (i.e., building, green space, street, water, or terrain) and sub-type (e.g., coniferous forest, deciduous forest) is stored (Figure VI.3B). Thereby, the system enables cartography-oriented design, leading to improved perception of context information (Jobst & Döllner 2008a). The required semantic information can be derived automatically from texture and material information by grouping objects with a similar appearance. Alternatively, semantic information can be provided manually at run-time or as part of the model data, e.g., using CityGML (Kolbe 2009).

Parts of the input data are best-view directions of buildings and sites to enable a cartographic visualization of landmarks (Elias *et al.* 2005; Grabler *et al.* 2008)—as demonstrated in Chapter V. The definition of landmarks is context-dependent (Grabler *et al.* 2008), using interest values defined per object (Figure VI.3C). The computation of these records is not limited to pre-processing, but can be updated at run-time if models are added or removed from a 3D scene, or if a user's interest in a specific feature type changes. Thereby, the system maintains interactive frame rates and a context-dependent visualization.



Feature ID	Type
...	...
438	Building
439	Street
440	Water Surface
441	Building
...	...

Figure VI.3.: Geospatial data used as input: (A) multiresolution models, (B) semantic information, (C) landmarks with interest values w_i and best views, (D) stylized textures.

Transition Configuration

Transitions between graphic styles are implemented by rendering 3D scene objects multiple times and compositing the intermediate results using image blending (Porter & Duff 1984). This simplifies the extension of the system with new graphic styles. The sequence of graphic styles can be configured as follows:

- A *scope* defines if a graphic style applies to a certain interest in an object.
- The transition is parameterized with a fragment-, object- or group-based interpolation. For this purpose, the axis-aligned bounding box of each object is computed and stored as an attribute.
- The parameterization of the LoD and LoA, e.g., with respect to color, texture abstraction, and edge enhancements.

Thereby, the system enables a user-defined visual abstraction of 3D scene objects. Figure VI.2 (top left corner) exemplifies a transition configuration for tree models.

Image Abstraction

A flow-based bilateral and DoG filter (Kyprianidis & Döllner 2008) is utilized to automatically stylize textures in a pre-processing stage. The input textures are first converted to mip maps (Williams 1983), and then processed for each level separately. This provides a continuous LoA of textured surfaces, while using standard capabilities of graphics hardware (Williams 1983). In contrast to Kyprianidis & Döllner (2008), the output of the edge enhancement is not combined with the quantized color output (Figure VI.3D). Instead, colors and outlines are blended at rendering time to enable individual parameterizations. The image abstraction is performed once per model. For a 3D city model (CityGML LoD3) with 1520 unique texture maps, each with an average resolution of 128 × 128 pixels, this process takes ~20 minutes (using the first hardware configuration and city model of *Chemnitz* described in the evaluation section).

2. Rendering

The rendering comprises the following steps: (1) computing the interest of objects for a user's task and context such as defined by the viewing perspective and region interest, (2) visual abstraction depending on the objects' interest, and (3) image compositing.

Algorithm 2: *Extended scene graph traversal (CPU)*

```
Input: A usage scenario $S$ with 3D scene objects $\{O_i\}$ and
       transition configurations $\{T_j\}$
forall $t \in \{T_j\}$ do
    forall $o \in \{O_i\}$ do
        if feature type of $o$ and $t$ match and LoD of $o$ and $t$ match
          and $f$ is inside validity range of $t$ (=not culled) then
            Render $o$ and apply graphic style of $t$
        end
    end
end
```

Computing Interest using Saliency Metrics

The thematic categorization is used to stylize model contents with high interest differently from contents with low interest. The system interprets a high interest by selecting detailed graphics, and a low interest by selecting abstract graphics for rendering respectively, where interest values in-between yield a mix of graphic styles. To identify areas to be visualized with high detail, the interest values for the visible object fragments are computed based on user-related metrics such as view distance, vertical view-angle, and region interest (Figure VI.4). Other metrics can be added as long as their value range is normalized. For instance, view metrics can be defined by normalized Euclidean distances and angles as

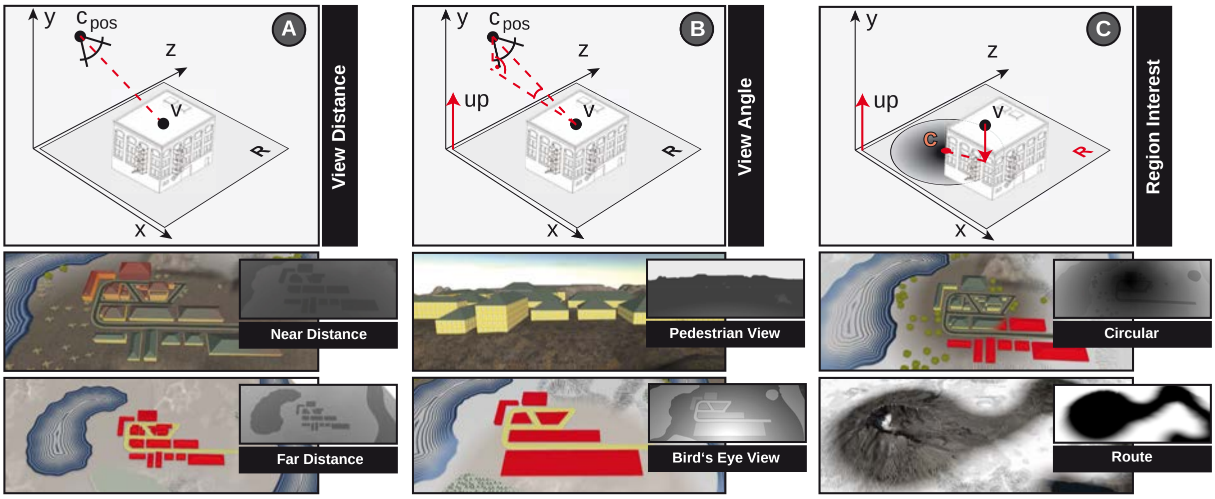


Figure VI.4.: User-related metrics defined by the system: view distance (A), view angle (B) and region interest (C). The debug outputs show areas of a 3D scene to be visualized with high detail (black) and low detail (white), respectively.

is shown in Figure VI.4. The region interest is represented by a distance map that is computed using the *parallel banding algorithm* (Cao *et al.* 2010), and is used to visualize RoIs or routes through a virtual 3D city model (Trapp *et al.* 2008). The computation of distance maps is based on the assumption that the terrain in the locality of the camera can be approximated by a plane (Figure VI.4). In contrast to previous techniques (Cole *et al.* 2006; Möser *et al.* 2008; Lorenz *et al.* 2008), the system enables multivariable transitions based on interest values and view metrics, resulting in increased flexibility. For instance, a weighted blending between view distance and view angle can be defined to prevent high detail presentations in bird's eye views with a near view distance.

A transition between graphic styles is based on image blending (Porter & Duff 1984). Blend values are computed for each transition configuration, and 3D scene objects with matching feature types and LoD. This procedure is performed during scene graph traversal on the CPU (Algorithm 2), and by using linear or smooth blend functions on the GPU (Figure VI.2C). The validity range of each transition configuration determines the blend value of a graphic style for an object of certain interest. Depending on how the threshold values for two successive transition configurations are defined, two general cases can be identified (Figure VI.5):

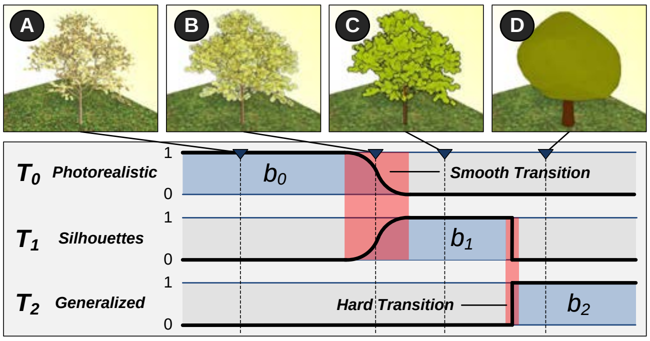


Figure VI.5.: Exemplary transition states for tree models.

1. ***Smooth transition.*** A smooth transition from one graphic style to another is defined by the *fade-out interval* and *fade-in interval* of two successive transition configurations. A smooth transition is enabled using the *smoothstep* function within these two intervals (Figure VI.5 A-C).

2. ***Hard transition.*** For certain configurations, discrete LoA transitions are appropriate—for instance, if two graphic styles lead to distorted color tones (Giegl & Wimmer 2007). *Hard transitions* are enabled if the overlap of a fade-in and fade-out interval is set to zero (Figure VI.5 C-D).

Algorithm 3: *Image compositing using shaders (GPU)*

```
   Input: Buffered color, alpha and depth values per pixel
   Output: Blended color values per pixel
1  begin
2      $color \leftarrow$ background color;
3      $count \leftarrow$ #SAMPLES;
4      for $n \leftarrow 0$, #SAMPLES $- 1$ do
5          $depths[n] \leftarrow$fetchABufferDepth$(n)$;
6          $count \leftarrow count - \lfloor depths[n] \rfloor$;
7      end
8      for $n \leftarrow 0, count$ do
9          $colors[n] \leftarrow$fetchABufferColor$(n)$;
10     end
11     blendPingPongFBO$(colors[count], depths[count])$;
12     $count \leftarrow count + 1$;
13     SortColorByDepth$(colors, depths, count)$;
14     for $n \leftarrow count - 1, 0$ do
15         $color \leftarrow mix(color.rgb, colors[n].rgb, colors[n].a)$;
16     end
17 end
```

For the computation of blend values, three interpolation modes are distinguished: (1) a fragment-based interpolation for smooth transitions within 3D scene objects, (2) an object-based interpolation with blend values applied uniformly by using the center of a object's axis-aligned bounding box as focus point, and (3) a group-based interpolation with a shared focus point among objects—for instance, to replace tree instances by a coarse geometry symbolizing woodland.

Image Compositing

The final image is composed using alpha blending (Porter & Duff 1984). Similar to LoA texturing (Chapter IV), a *stencil routed A-buffer* (Myers & Bavoil 2007) is used to buffer fragments in depth at real-time frame rates. Because certain 3D scene objects have a high complexity in depth (e.g., foliage of trees), the system provides the capability to render objects off-screen into a *Ping-Pong* buffer. It comprises two render textures, uses a depth test, and switches its render texture for successive graphic styles. For image compositing, the output of both textures is blended and combined with the information of the A-buffer.

To improve the rendering performance, the A-buffer sorting is enhanced using a dynamic image composition (Algorithm 3). Fragments with maximum depth value are excluded so that only routed samples are blended (Algorithm 3, lines 4-7). Further, the system improves shape perception by *unsharp masking the depth buffer* (Luft *et al.* 2006), which is enhanced by locally weighting depth differences according to the alpha values to reduce ringing artifacts.

3. Results

The system was implemented using C++, OpenGL and GLSL. Two platforms were used for performance evaluation: (*setup 1*) an Intel® Xeon™ 4× 3.06 GHz with 6 GByte RAM and NVidia® GTX 560 Ti GPU with 2 GByte VRAM, and

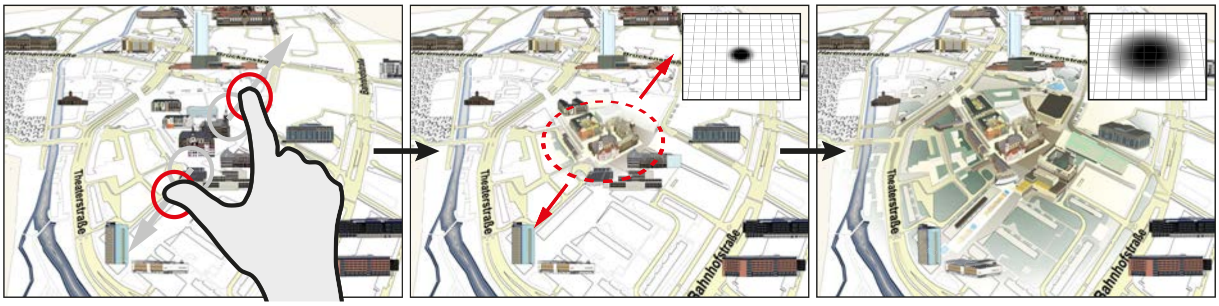


Figure VI.6.: Touch-based interaction using the framework proposed in Chapter IV with the pinch-to-zoom metaphor to parameterize the LoA of 3D geospatial objects in a region of interest for focus+context visualization.

(*setup 2*) an Intel® Core2Duo™ 2× 3.0 GHz with 4 GByte RAM and NVidia® GTX 460 GPU with 1 GByte VRAM. To show the effectiveness of the system, usage scenarios were authored for a virtual 3D city model of Chemnitz (Germany) with 458 objects, 223,743 vertices, 176,601 faces, and 1,520 texture maps (Figure VI.1 and Figure VI.7 A-D); another 3D city model with 532 objects, 63,630 vertices, and 40,993 faces (Figure VI.7 E); and a virtual landscape model of Mount St. Helens (Figure VI.7 F). To improve the rendering performance, view-frustum and back-face culling were enabled, and geometry instancing applied for the vegetation objects. For order-independent blending, an A-buffer with 8 samples was used. To enable an intuitive parameterization of the system, it is coupled with the interaction framework proposed in Chapter IV for LoA texturing. For instance, the *pinch-to-zoom* metaphor can be used to define and parameterize a user-defined region of interest (Figure VI.6).

1. Applications

Despite a manual selection of the LoA (Figure VI.7 A), view-distance-based transitions were authored to visualize 3D scene objects near the virtual camera at high detail, and distant objects in an abstracted way (Figure VI.7 B). This approach can be of interest in dynamic and ubiquitous information systems. For instance, navigation systems could use this visualization to present places close to a viewer's position with high detail for local orientation assistance, and places far away with less detail and emphasized landmarks for navigational assistance. Thereby, the exploration of complex virtual environments can be improved, in general, since the viewer is not required to switch between a 3D perspective view and a map view, as common in map view services like Google Maps. Because of emphasized objects in the foreground, it further facilitates direction guidance of a viewer's gaze while preserving context information in the background.

A detailed and a map-like visualization may also be seamlessly blended based on the vertical view angle. The system is able to provide a map-like visualization to highlight thematic information (e.g., landmarks, labeled roads, green spaces, and water surfaces), leveraging the capability to implement cartographic color schemes (Brewer 1994) and NPR techniques (Chapter V). This may be useful to communicate only relevant information in image regions of high information compression, in particular at high view distances. For instance, the driving speed could be used in a car navigation scenario to adjust the LoA, while orientation assistance is provided by integrating landmark illustrations.

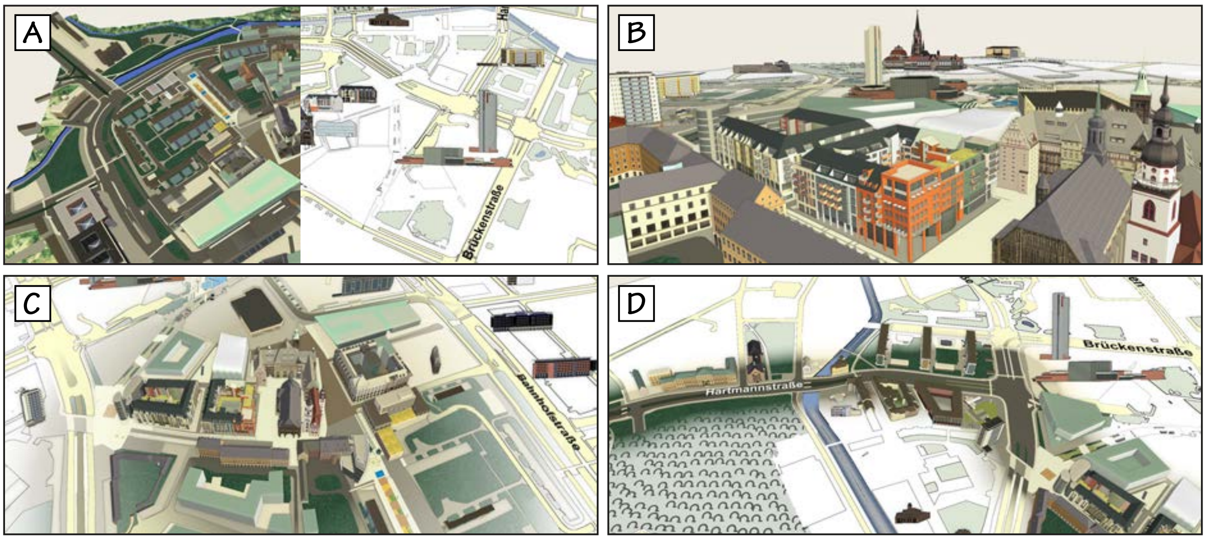


Figure VI.7.: Exemplary applications authored with the visualization system: (A) manual LoA selection (left: detailed representation, right: abstract representation), (B) distance-based transitions, (C) a circular RoI, (D) route highlighting.

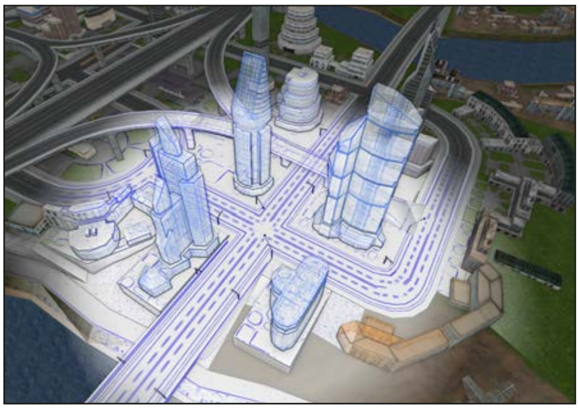


Figure VI.8.: Blueprint stylization for urban planning.

Additional applications of the proposed system are found in wayfinding, i.e., when using view-based metrics for LoA transitions. For example, 3D scene objects can be highlighted along routes to attract and direct a viewer's focus—e.g., as a navigational aid to guide a user to a destination or RoI (Figure VI.7 C). Within this application domain, the system is able to provide significant improvements over previous techniques designed for occlusion-free route visualization (Qu *et al.* 2009; Möser *et al.* 2008) because these neglect information abstraction in context regions. For this purpose, the multiperspective rendering technique of Pasewaldt *et al.* (2011) has been extended using the proposed cartography-oriented visualization techniques and LoA transitions for focus+context visualization (Figure VI.9). Technically, the multiperspective rendering technique computes a deformation matrix in a vertex shader, which is applied for each vertex independently. The acquired *bending* is performed per frame and facilitates real-time image synthesis. Figure VI.9 and Figure VI.10 show a combination of the approach with the proposed LoA transitions, i.e., to provide more control over the parameterization of resulting panorama maps by enabling the implementation of design principles from cartography. Moreover, the combined approach enables a seamless combination of 3D and 2D depictions while reducing occlusion and visual clutter of distant 3D scene objects (Figure VI.10).

Finally, the system is feasible to visualize RoIs as blueprints and seamlessly combine these with high-detail graphic styles in the context area (Figure VI.8). Applications designed for urban planning could use this approach to highlight complex structures or architectural features of 3D building models. This visualization approach may also be used to communicate uncertainty for construction plans that are part of (time-based) model variants, and to highlight selected information obtained from database queries for analysis purposes.

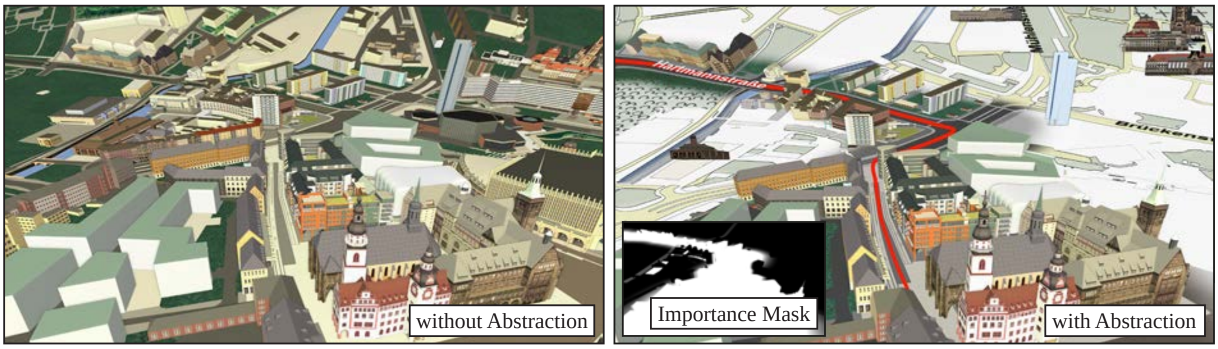


Figure VI.9.: (left) The degressive 3D panorama bends the 3D virtual environment toward the user, thus, replacing the horizon and reducing visual noise. (right) The proposed system is used to complement the multiperspective view for highlighting task-relevant information in a routing scenario.

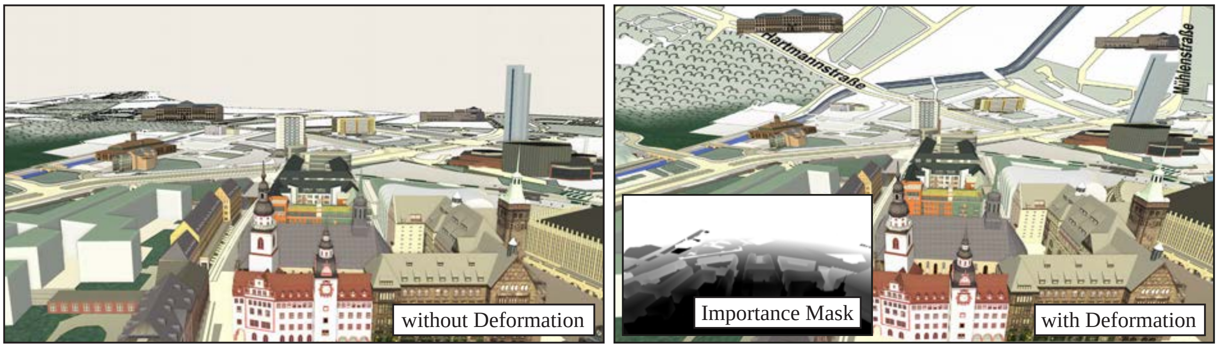


Figure VI.10.: A distance-based LoA transition configured in two ways. (left) without using a multi-perspective visualization approach. (right) using a degressive perspective to increase screen-space utilization and to reduce visual clutter in the background region.

2. Evaluation

This section provides evaluation results for the proposed system with respect to image saliency and performance.

Image Saliency

To demonstrate the advantage of the provided visualization approaches, saliency maps of the system's outputs are compared with a homogeneous high-detail visualization typical for mass-market systems (e.g., Google Earth). As is observed in Figure VI.11, visual saliency of homogeneous graphic styles is distributed across focus and context regions. By contrast, visualization of the proposed system yields directed high saliency within a circular RoI and for single landmarks in the context area. In case of saliency-guided route visualization, the saliency aligns with the route due to high frequencies in color, orientation, and depth.

Performance

The performance tests were conducted for the aforementioned platforms, virtual environments, and usage scenarios. The test results in Table VI.1 show that the system provides interactive frame rates in HD resolution. It was observed that the performance depends on the total number of transition configurations defined for a usage scenario. For instance, a view-distance-based transition performs, in mean, 27.5% slower than a visualization with a homogeneous

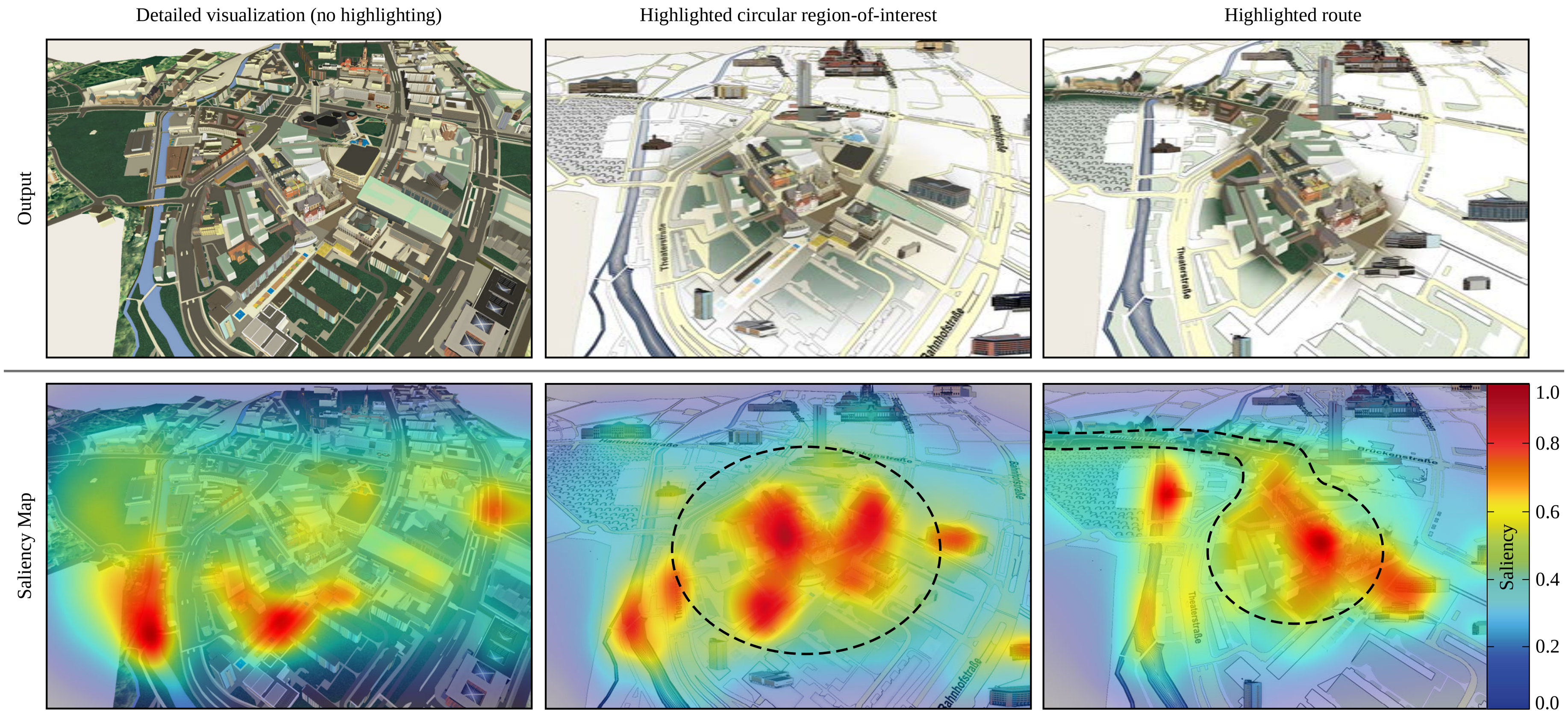


Figure VI.11.: Examples showing a circular RoI and a highlighted route within the city of Chemnitz, compared to a detailed version. The bottom row shows the respective saliency maps using the algorithm of graph-based visual saliency of Harel *et al.* (2007) for evaluation.

Table VI.1.: Performance evaluation measured in frames per second (mean) for three virtual environments, three screen resolutions, and two platforms (see Section 3 for the specifications).

Screen Resolution	1920 × 1080		1280 × 720		800 × 600	
Model	Setup 1	Setup 2	Setup 1	Setup 2	Setup 1	Setup 2
Chemnitz	5.7	5.0	5.8	5.2	6.1	5.3
MegaCity	14.2	13.1	17.2	16.6	20.4	20.1
Mt. St. Helens	46.5	36.8	72.2	57.6	87.8	78.2

graphic style. Further, it was observed that the system is fill-limited, with a performance increase of 75% (in mean) when using a resolution of 800 × 600 pixels over 1920 × 1080 pixels. For the virtual 3D city model of *Chemnitz*, it was observed that the system is CPU-limited because of the rendering engine being limited to a single-threaded traversal of the scene graph. Moreover, the results for the virtual model of *Mount St. Helens* show that the system can handle 3D scenes with high visual complexity in full HD resolution at real-time frame rates.

The memory consumption (VRAM) of the system mainly depends on the A-buffer (32bit color, 24bit depth, and 8bit stencil values per sample), the Ping-Pong buffer (32bit color and 32bit depth values per pixel), and the geometry buffer (32bit edge map, 32bit ID map, and 32bit normal map):

$$M = \frac{W \cdot H \cdot (2N+7)}{262,144}\text{MB},$$

where W and H refer to screen resolutions in pixels, and N to the number of samples used by the A-buffer.

3. Limitations

During development and authoring of usage scenarios, it was observed that the parameterization of the system can be cumbersome. Motivated by this, functionality to serialize transition configurations, feature classifications, and parameterizations of graphic styles was added to the system, to be able to maintain designs in libraries and easily deploy usage scenarios. It was further observed that development of new graphic styles can become time consuming. Therefore, a shader editor was integrated into the system, which facilitates modification of vertex, geometry, and fragment shaders at run-time. Finally, the approach to buffer fragments in depth is memory consuming and requires sufficient samples, or additional rendering passes using occlusion queries (Myers & Bavoil 2007) to avoid visual artifacts. The proposed Ping-Pong buffer only resolves this issue if 3D scene objects are rendered opaquely.

4. Summary

This chapter presents a concept and an implementation of a system that visualizes virtual 3D city and landscape models with parameterized LoA transitions for a seamless combination of graphic styles in a single view. The system provides interactive, view-dependent visualization by coupling metrics—related to the view perspective or region interest—with rendering techniques subject to cartography-oriented design. It is extensible by custom 2D and 3D graphic styles, integrates into a visualization pipeline, and can be used to improve existing visualization techniques based on focus+context zooming and multi-perspective views. Usage scenarios based on the system's capability for thematic visualization demonstrate the system's benefits for typical applications of geovirtual environments, in particular map viewing, navigation, and wayfinding. Since the system operates in 3D space, it also may be applied to enhance the x-ray volumetric lens effect (Viega *et al.* 1996) for indoor visualization. The visualization of complex virtual 3D city and landscape models on mobile devices may also benefit from the proposed system because the provided LoA techniques reduce the information compression on displays with limited size. Here in particular, saliency maps of the presented results show that the system is feasible to draw attention to important or prioritized information.

VII

Image-based Abstraction and Stylization

The work presented in this chapter is partly based on

Semmo, A., Limberger, D., Kyprianidis, J. E. & Döllner, J. "Image Stylization by Interactive Oil Paint Filtering". *Computers & Graphics* **55,** 157–171 (2016)

Semmo, A., Limberger, D., Kyprianidis, J. E. & Döllner, J. *Image Stylization by Oil Paint Filtering using Color Palettes.* in *Proc. Computational Aesthetics* (2015a), 149–158

Semmo, A., Kyprianidis, J. E. & Döllner, J. *Automated Image-Based Abstraction of Aerial Images.* in *Proc. AGILE* (2010), 359–378

Images are an integral part of 3D geospatial information visualization. They show the richness of visual details and the natural appearance of real-world phenomena. The following sections provide techniques for the abstraction of aerial images and terrestrial photography as major image types. First, a generic method for tile-based image-based abstraction of aerial images is proposed in conjunction with the LoA texturing proposed in Chapter IV. Afterwards, an artistic rendering technique is proposed that simulates oil paint effects by image filtering and texture synthesis.

1. Image-based Abstraction of Aerial Images

Aerial photography has evolved as a fundamental tool for geospatial data capturing, processing, and visualization. With today's usage of aerial photography in photorealistic visualization, aerial images have become a viable medium in applications such as in cartography, photogrammetry, construction, planning, and marketing. They are also used as popular source of geospatial information in today's geographic information systems and by the general public, in particular since the widespread application of aerial photography in web-based mapping services. Furthermore, the technical advances in the field of oblique imagery have opened new avenues for efficient 3D geoinformation acquisition (Lillesand *et al.* 2014). Simultaneously, with today's technical advances in remote sensing for aerial photographs, aerial images have been established as a fundamental data source in 3D geospatial information visualization for constructing detailed models of reality.

1. Motivation

The perception and cognitive processing of aerial images by the human is still faced with the specific limitations of photorealistic depictions such as low contrast

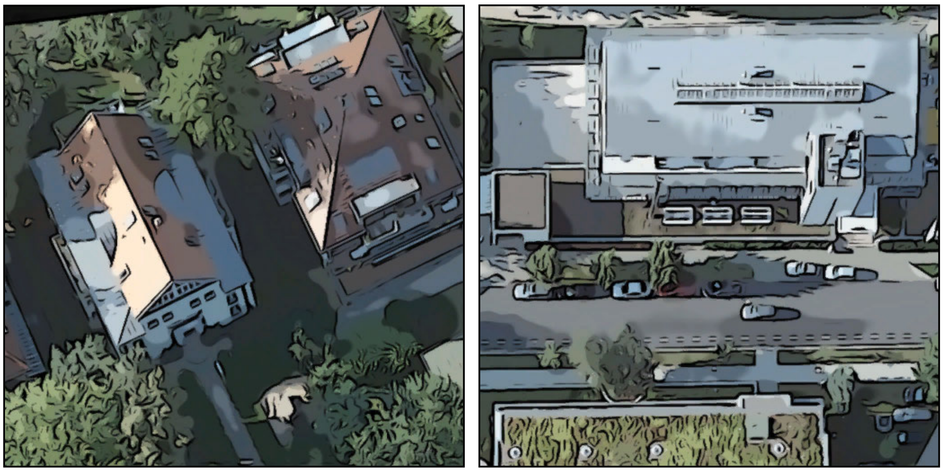


Figure VII.1: Abstracted aerial images created using the proposed method for tile-based image filtering. Here, the flow-based bilateral filter of Kyprianidis & Döllner (2008) was used for processing.

areas, unsharp object borders as well as visual noise. Instead, a non-photorealistic visualization can be a better choice for spatial perception, analysis, understanding, and knowledge discovery (Meng 2002) in many areas of application. Research in visual perception argues that NPR should take into consideration the specific nature and type of objects displayed, since "meaningful abstraction clearly affected viewers in a way that supports an interpretation of enhanced understanding" (Santella 2005). To this date, research in the fields of NPR has not developed specific solutions for a proper stylistic representation of the terrain with projected aerial images.

In this section, a generalized approach for automatically abstracting aerial images in a non-photorealistic way is provided (Figure VII.1); the results supply a concise presentation of the terrain in a NPR of 3D geovirtual environments. Common image abstraction techniques focus on image filtering that can be processed as a whole by the GPU. The proposed method, by contrast, contributes an approach to filter massive image data in a seamless way. To this end, an image tiling procedure is introduced that is optimized for post-processing images on GPUs and avoids visible artifacts across junctions. This is technically achieved by filtering additional connection tiles that overlap the main tiles of the input image. Thereby, a pre-processing of stylistic image representations is facilitated such that rendering is feasible on low-cost computing devices or devices with limited computing power, e.g., smartphones. The technique also enables to provide different LoAs for aerial images by computing a mipmap pyramid, where each of the mipmap levels is filtered with adapted abstraction parameters (refer to Chapter IV). These mipmaps can then be used to perform LoA rendering of abstract aerial images. Here, a high abstraction potential is identified in landscape images and a potential benefit from edge enhancements in urban environments.

For filtering, primarily those aerial photographs are processed that correspond to two classes: oblique photos and orthophotos. Oblique photos are images that are captured at an angle of about 40 degrees from four principle directions. By contrast, orthophotos are most valuable to cartography and city planning, and are the primary source for geospatial information systems to create maps that have a uniform scale and provide undistorted geometry.

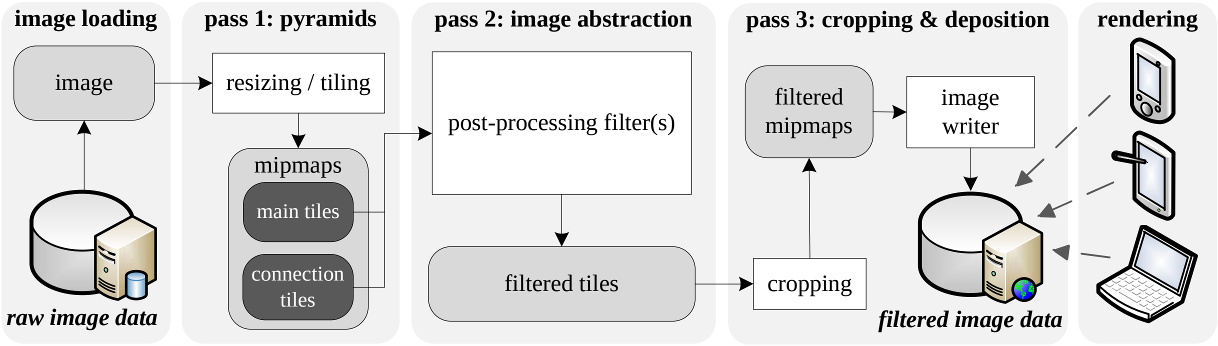


Figure VII.2.: Overview of the abstraction pipeline for aerial images. Raw image data (e.g. GeoTIFFs) is loaded and pre-processed by image resizing and tiling. Filtering is performed on each of the image tiles, which are cropped and written back into main memory. Finally, the filtered mipmaps are stored onto hard disk drives, from where they can be requested for rendering onto different devices.

2. Method

The technique is designed in such a way that GPU limitations with respect to video memory and image sizes can be by-passed by means of image tiling. Figure VII.2 illustrates the proposed pipeline used for tile-based image filtering. The method starts to incrementally load an aerial image into main memory. Next, the image is tiled in evenly sized main parts such that the GPU can process them as a whole. Mipmap pyramids are computed for each of these tiles—or the input image respectively—with each level being processed separately, i.e., to obtain the LoA effects described in Chapter IV. In addition, *connecting tiles* that overlap the main tiles are taken into account to avoid visible artifacts at the junctions of the *main tiles* during rendering. This procedure includes cropping each of the tiles in the third pass of the pipeline.

The storage of the filtered mipmaps can be handled differently. The filtered mipmaps may be compressed by the *S3 Texture Compression* algorithm (McCabe & Brothers 1998) and stored as *Direct Draw Surface* (DDS). The DDS scheme is an optimized data format for storing mipmaps and DXT compressed textures, and can be rapidly loaded into texture memory. An alternative method could base the storage on a streaming procedure that makes use of the TIFF file standard, which also supports storage of mipmaps. Computing devices that do not have the power to compute these images in an adequate time span, or do not provide the required hardware resources, can then request these images and use them for rendering.

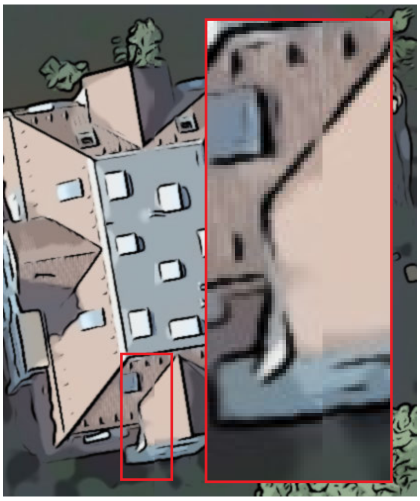


Figure VII.3.: Filtering image tiles separately yields visible junctions because edges and colors become disconnected.

Image Resizing and Tiling

The proposed method starts to downsample an aerial image to obtain the mipmap levels, and divides the mipmap levels in smaller parts. Web services typically provide aerial images divided in evenly sized tiles; for example preferring a tile dimension of 256 or 512 pixels. By contrast, the proposed method does not assume to have a pre-tiled image, and instead divides the scaled mipmap levels with respect to the GPU architecture, e.g., using dimensions between $1,024 \times 1,024$ and $8,192 \times 8,192$ pixels.

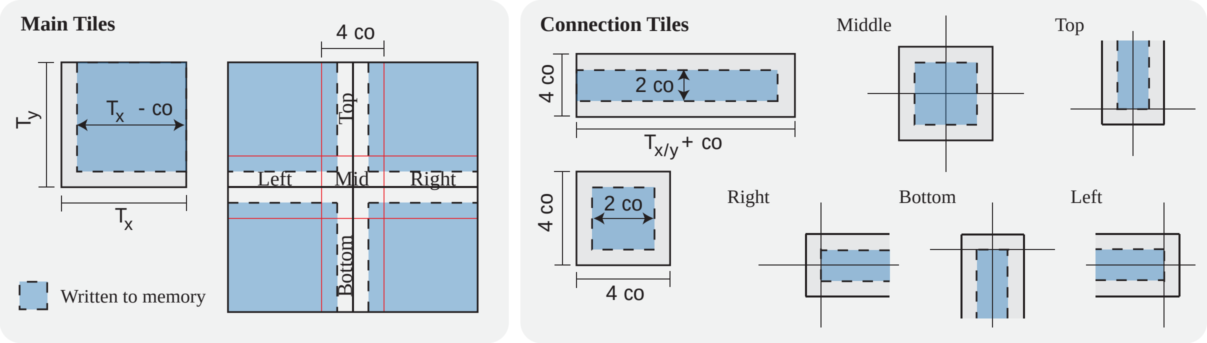


Figure VII.4.: Overview of the tiling procedure with respect to main tiles and connection tiles, used to encounter the problem of visible artifacts at junctions. (T_x, T_y) denotes the size of image tiles.

A plain, "naïve filtering" in local image domains will cause visible artifacts at the junctions as soon as the scene is being reconstructed and rendered (Figure VII.3). To this end, additional tiles are taken into account that overlap these junctions. Conceptually, these connection tiles overlap two adjacent main tiles in each direction by a cut-off value that is adapted to the domain range of the local image filter. For instance, the bilateral filter is defined by its standard deviation σ_d and is iterated by a number n_a, which yields a minimal *cut-off*:

$$\mathrm{co} = \lceil 2\sigma_d \rceil n_a.$$

The connection tiles may occur in each main direction of the defined area and in the center of four adjacent main tiles (Figure VII.4).

Image Filtering

The filtering is performed on each mipmap level separately. Using trilinear texture mapping at the rendering stage (Williams 1983), a view-dependent LoA is achieved. Figure VII.5 shows an example, where subtle details for the roof of a building are filtered, outlines of shadows are increasingly coarsened and complex features such as treetops are generalized. This is explained by the bilateral filter having less image information of a distinct feature left for weighting the color values into the local neighborhood.

The abstraction technique of Kyprianidis & Döllner (2008) was used for image processing, which includes a flow-based bilateral and DoG filter, and local color quantization. Four iterations are used for bilateral filtering, using a standard deviation of 3.0 for the domain and 4.25 for the range weights. The

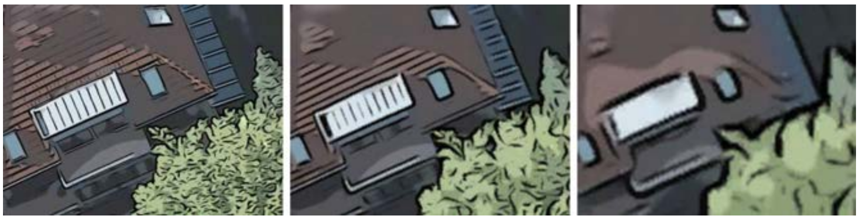


Figure VII.5: Level of abstraction by means of trilinear filtering. The viewpoint distance decreases linearly from left to right. The greater the viewpoint distance, the fewer details are depicted at the roof of the building and the tree.

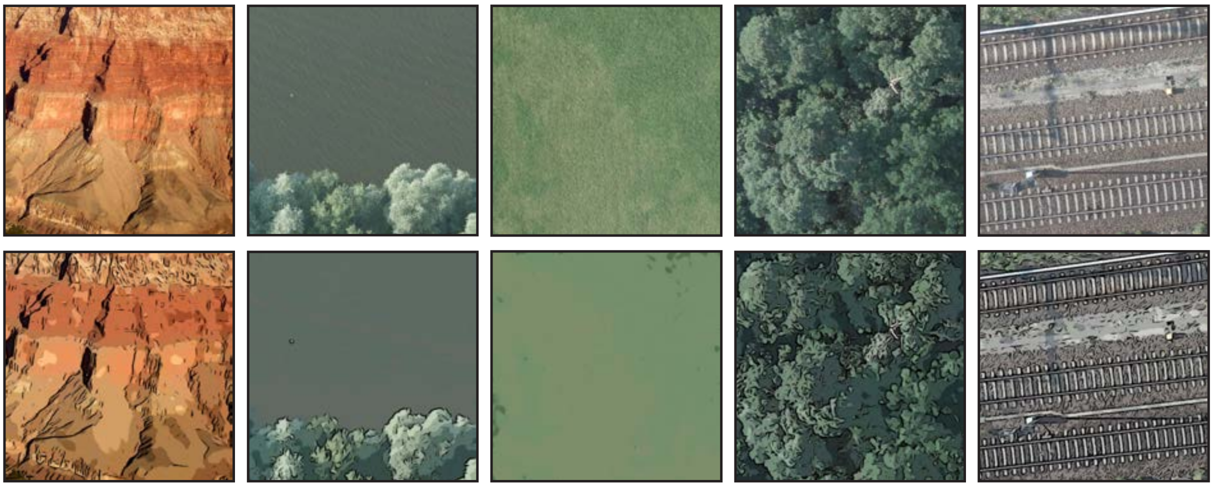


Figure VII.6.: Common features types that occur in aerial images. From left to right: terrain, water surfaces, green spaces, trees, and transportation networks. The original input is shown at the top and the filtered output at the bottom.

color quantization produces at most 8 distinct colors for a local image region. Figure VII.6 depicts outputs for common feature types using this configuration.

3. Performance Evaluation

The percentage of connection tiles in relation to the input size of an image is calculated as follows. Let $I_x \times I_y$ be the dimension of the input image and let $T_x \times T_y$ be the dimension of main tiles. Then the number of main tiles in x- and y-direction are given by:

$$n_{T_x} = \frac{I_x}{T_x} - 1 \quad \text{and} \quad n_{T_y} = \frac{I_y}{T_y} - 1$$

The total size in pixels for the connection tiles is:

$$p_t(I_x, I_y) = 4 n_{T_x} n_{T_y} \text{co}^2 + (n_{T_x} + n_{T_y} + 2 n_{T_x} * n_{T_y})(4\text{co}(I_x + I_y + \text{co}))$$

The number of these connection tiles increase non-linearly with the dimension of the input image $I_x \times I_y$. In case of a tiling size of $2,048 \times 2,048$ pixels and cut-off of $\text{co} = 24$ pixels, the percentage levels off at a limit of approximately 18.9%. GPUs that are able to post-process image tiles of $8,192 \times 8,192$ pixels end up with an off-cut of 4.7% (2.3% with $\text{co} = 12$).

An empirical analysis of the run-time also shows the decreased overhead if larger main tiles are used. The default configuration previously described was used on a test system with an Intel® Core 2 Duo E8400 2× 3.0 GHz and NVidia® GeForce 9600 GT with 1 GByte VRAM. Images with a size of $8,192 \times 8,192$ pixels took in mean 21.1 seconds to process for a tiling of $1,024 \times 1,024$ pixels (16.5 seconds with $\text{co} = 12$), whereas the same image took 17.8 seconds to process for image tiles of $2,048 \times 2,048$ pixels (15.4 seconds with $\text{co} = 12$).

To summarize, the approach contributes to the feasibility of non-photorealistic, automated aerial image abstraction as a pre-processing stage, e.g., required by applications used on mobile devices with a low computing power and limited display size. Here in particular, the provided techniques may enhance visual

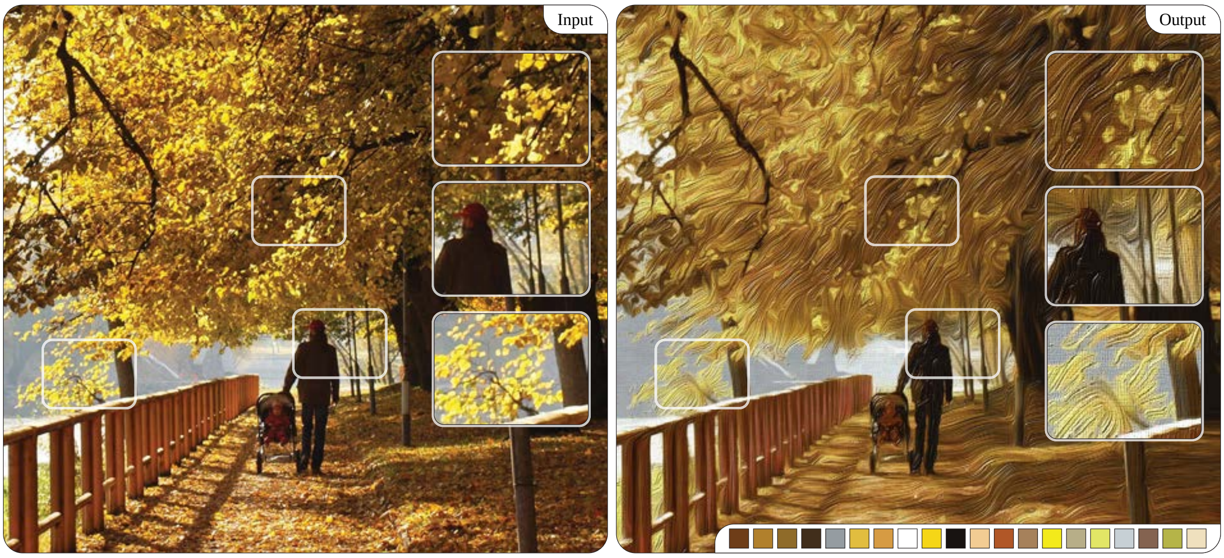


Figure VII.7.: Exemplary application of the proposed technique to automatically transform a color image (left) to a filtered variant with oil paint characteristics (right).

clarity and generalize the contents of aerial images to improve their perception and recognition.

2. Image Stylization by Interactive Oil Paint Filtering

Image-based artistic rendering received significant attention in the past decades for visual communication, covering a broad range of techniques to mimic the appeal of artistic media (Kyprianidis *et al.* 2013). Oil paint is considered to be among the most popular of the elementary media because of its qualities for subtle color blending and texturing (Scott 2005). Starting with the beginning of semi-automatic painting systems in 1990 (Haeberli 1990), stroke-based techniques that align and blend primitives on a virtual canvas have been the pre-dominant category to simulate oil paint (Hertzmann 2003). While their example-based texturing approach is able to provide high-quality outputs of expressive nature and great opportunities for layering, however, stroke-based techniques are usually hard to parameterize to simulate paint with soft color blendings or no visible borders—e.g., as practiced in the Renaissance era (such as *sfumato*, Earls 1987) and prevalent in many figurative art works (Figure VII.8). To this end, image filtering is a promising alternative approach to produce painterly looks with more subtle color blendings—in particular with the recent advancements in shape-adaptive smoothing (Kyprianidis *et al.* 2013) such as anisotropic diffusion (Weickert 1998) and shock filtering (H. Kang & S. Lee 2008). Simulating the visual characteristics of oil paint via image filtering, however, is a difficult task with respect to three main issues:

I1 The color distribution should be optimized to conform to a global color palette while preserving contrasts of important or prioritized features.

I2 The paint texture should be oriented to the main feature curves to mimic the way an artist might paint with a brush.

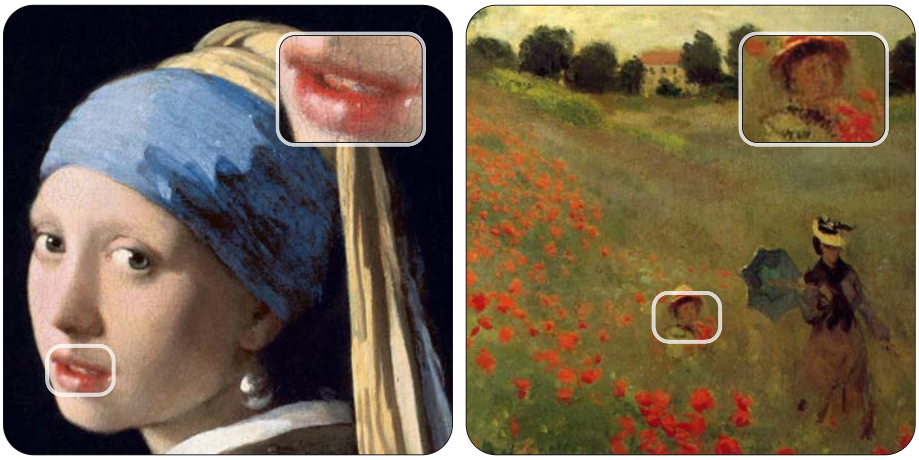


Figure VII.8: Oil paintings by **(left)** J. Vermeer (1665) and **(right)** C. Monet (1873). The artists use constrained color palettes with soft color blendings, two characteristics that are simulated by the proposed filtering technique.

I3 The stylization process should be locally adjustable to enable creative control over the visual output.

This section presents a technique for image stylization that employs (re-)colorization and non-linear image filtering to devise artistic renditions of 2D images with oil paint characteristics. Rather than attempting to simulate oil paint via aligned strokes (Haeberli 1990; Hertzmann 1998; Hays & Essa 2004; Zeng *et al.* 2009) or through physically-based techniques (W. Baxter *et al.* 2004; Lu *et al.* 2013), the proposed method formulates I1 to I3 as sub-problems of image filtering (Figure VII.7). The first problem is solved by performing a recolorization, using the optimization-based approach of Levin *et al.* (2004), with the dominant colors of the input image for quantization. This approach produces more homogeneous color distributions than local image filtering techniques and gives users more control in refining global color tones. The second problem is solved using the smoothed structure tensor (Brox *et al.* 2006b), which is adapted to the feature contours of the quantized output, together with principles of line integral convolution (Cabral & Leedom 1993) and Phong shading (Phong 1975) to obtain a flow-based paint texture in real-time. Finally, the third problem is addressed by an interactive painting interface that implements GPU-based per-pixel parameterizations via virtual brush models to give users local control for adjusting paint directions, shading effects, and the LoA. The provided approach provides versatile parameterization capabilities to resemble paint modes that range from high detail to abstract styles.

The remainder is structured as follows. Section 1 reviews related work on image stylization, color quantization, and paint texture synthesis. Section 2 presents the methods used for oil paint filtering, including methods to adjust the LoA according to importance masks. Section 5 proposes an interactive painting interface with brush tools to locally adjust paint configurations and the LoA. Finally, Section 6 presents further results and implementation details, including comparisons to previous stroke-based techniques.

1. Related Work

Related work is found in the fields of image stylization and filtering, color quantization, and paint texture synthesis.

Image Stylization and Filtering

For artistic image stylization, three approaches can be distinguished: (1) stroke-based and example-based methods, (2) region-based techniques, and (3) image filtering (Kyprianidis *et al.* 2013). A classical method for stroke-based stylization is to iteratively align brush strokes of varying color, size, and orientation according to the input image (Haeberli 1990; Hertzmann 1998; B. Gooch *et al.* 2002; Hays & Essa 2004; Zeng *et al.* 2009; Zhao & Zhu 2010; Lu *et al.* 2010). For an overview on this topic, the interest reader is referred to the survey by Hegde *et al.* (2013).. Example-based rendering typically involves texture transfers by image analogies (Hertzmann *et al.* 2001), a method previously used to create portraits with a painterly look (Zhao & Zhu 2011b; T. Wang *et al.* 2013), and in neural networks to mimic painting styles (Gatys *et al.* 2015) but which typically requires training data as input. An essential building block for region-based stylization is segmentation. Several methods based on a mean shift have been proposed for image abstraction (DeCarlo & Santella 2002; Wen *et al.* 2006) and the simulation of artforms and fabrics such as stained glass (Mould 2003) and felt (O'Donovan & Mould 2006). However, the rough boundaries of the segmented regions created by these methods would require elaborate post-processing to achieve color blending characteristics of oil paint.

To substantially modify areas or image regions, the proposed method is based on local image filtering using flow fields. Flow fields have been used to adapt bilateral filtering (Kyprianidis & Döllner 2008; H. Kang *et al.* 2009) and particle-based techniques (Yoon *et al.* 2012) to local image structures. For the proposed method, quantized color outputs are smoothed by flow-based Gaussian filtering to provide smooth interpolations at curved boundaries, however, it restrains from weighting in the color domain to achieve firmer color blendings. The additional filter categories reviewed in Chapter III include morphological operations based on dilation and erosion, e.g., for watercolor rendering (Bousseau *et al.* 2006), and global optimization schemes for image decomposition. However, the latter category is less suited for rendering with a constrained color palette, which instead requires filtering schemes found in color quantization.

Color Image Quantization and Abstraction

The typical goal of color quantization is to approximate an image with a relatively small number of colors while minimizing color abbreviations. Popular approaches are based on the median-cut algorithm (Heckbert 1982), clustering using octrees (Gervautz & Purgathofer 1988), k-means (Kanungo *et al.* 2002), and adaptive segmentation via perceptual models (J. Chen *et al.* 2005) or roughness measures (Yue *et al.* 2014). However, these algorithms may absorb colors because of their global optimization scheme, or only operate in the color space. Other approaches also consider the spatial space via local luminance mapping and thresholding (Winnemöller *et al.* 2006), and in its optimization scheme to preserve image details (Yu *et al.* 2014), but are mainly self-organizing with respect to the derived color palette. By contrast, a technique is proposed that derives colors from local image regions using a scoring system for optimal distribution, and uses the derived color palette for image quantization. At this, the optimization framework of Levin *et al.* (2004) is parameterized to propagate seed pixels—using colors of the derived palette—to the remaining pixels at the premise that pixels in space with similar intensities should have similar colors, with an additional pass for luminance quantization. A related optimization scheme was proposed by T.-h. Kim *et al.* (2007) to seamlessly stitch uniformly-colored regions, but for the application of dequantization. The output produced by the proposed method is then post-processed using a flow-based Gaussian filter to provide firm color blendings.

Principles of abstraction and highlighting are commonly used by artists to enhance communication aspects of their emotions and ideas. At this, the concept of LoA plays a major role for guiding a viewer's focus to certain image regions and improve the perception of information that are meant to be of particular interest (DeCarlo & Santella 2002; Santella & DeCarlo 2004). The proposed method follows the approach of image masking to provide different LoA representations. Here, the main focus lies on the definition of parameter sets to locally control the filtering effects and their LoA, as well as the development of a modular interface to inject image-based metrics such as feature semantics, image saliency, and the view distance derived from images with depth information, e.g., regarding a foreground/background separation. Rosin & Lai (2013) also use image salience and edge-based criteria to direct a color (de-)saturation and render images with spot colors, e.g., to make foreground objects stand out. Their technique also uses *hue* values for color quantization, however, they merely focus on three classes to guide the abstraction: dark (black), intermediate (gray) and light (white). By contrast, an approach is seeked that guides the colorization via global and possibly refined color palettes that is subject to an optimization scheme to minimize color differences.

Interactive Paint Texture Synthesis

Contrary to physically-based paint modeling (Chu *et al.* 2010; Lu *et al.* 2013), the paint texture synthesis is separated from color abstraction. The proposed computation is based on the smoothed structure tensor (Brox *et al.* 2006b) and an eigenanalysis to obtain gradient and tangent information for directed smoothing similar to the work by Kyprianidis & Döllner (2008). The smoothing pass is adapted to the main feature contours retrieved via the flow-based Laplacian of Gaussian (FLoG) (Kyprianidis 2011) to avoid ringing artifacts and provide outputs with adaptive detail. Similar to the work by Hertzmann (2002), bump mapping via Phong-based shading (Phong 1975) is used to synthesize a normal-mapped height texture that is aligned to the local feature orientations of the quantized image. The synthesis involves noise textures that are blurred in gradient flow direction to create a painting-like effect, an approach that is similar to line integral convolution (Cabral & Leedom 1993) and is also followed in the texture-based design of tensor fields (E. Zhang *et al.* 2007; Kagaya *et al.* 2011). By contrast, the proposed computation is defined as a composition of ordered, parameterizable image processing steps and performs in real-time, thus it may also be injected by user-specified motions (Hays & Essa 2004; Olsen *et al.* 2005) via painting (Hanrahan & Haeberli 1990).

The often tedious, error-prone process of manually tweaking image filtering effects is typically limited to a subset of global parameters. Most consumer photo-editing products only rudimentary support to adjust local parameters via masking and blending. Because the proposed paint texture synthesis is local and fast, a system for per-pixel parameterization is provided that supports the modification and correction of (pre-)computed or intermediate filtering results. In contrast to placing dynamic or static rendering primitives (Schwarz *et al.* 2007), the concept of specialized local parameterizations (Anjyo *et al.* 2006; Todo *et al.* 2007) is extended to a generalized brush-based painting within effect-parameter spaces. At this, the proposed approach injects local parameters into image processing steps, a concept prominently used for WYSIWYG painting in the intensity (Hanrahan & Haeberli 1990) and gradient domain (McCann & Pollard 2008) to locally adjust stylized renderings (W. V. Baxter & Lin 2004)—which is often coupled with user-defined textures that can be interactively merged, intersected and overlapped (Ritter *et al.* 2006). The provided generalized approach enables to aggregate single filter operations and parameters, e.g., brush sizes, smoothing kernels, to high-level brush tools to explicitly control the LoA of the filtering effects, which is exemplified for local orientation and shading corrections of paint textures—for example to simulate the glossiness of water features. For a compendium on the synthesis of brush strokes, the reader is referred to the work by DiVerdi (2013).

2. Method

An overview of the stylization technique is shown in Figure VII.9. The processing starts with extracting a user-defined number of dominant colors from the input image. Next, a recolorization based on the optimization approach by Levin *et al.* (2004) is performed to quantize the chrominance and luminance channels using the derived color palette. In parallel, contour lines are extracted from the input image via DoG and LoG filtering, where the latter is used for parameterizing the computation of a flow field. The computation is based on the structure tensor (Brox *et al.* 2006b), which is adaptively smoothed according to the derived contour lines to avoid filtering across feature boundaries. An eigenanalysis is then used to provide gradient and tangent information for paint texture synthesis. The synthesis performs in real-time, is based on bump mapping and Phong shading (Phong 1975), and produces outputs that are blended with the quantized image to compose the final result. The filtering stages are presented in more detail in the following sections, each followed by a discussion as well as methods for dynamic parameterization to achieve content-based LoA effects.

Dominant Color Extraction

To synthesize the way artists paint with a limited number of base colors, dominant (and contemporary) colors need to be derived from the input image. Common approaches use global optimization schemes that cluster similar colors and determine the base colors from the largest clusters such as the median-cut algorithm (Heckbert 1982). These approaches, however, may produce false colors or absorb contemporary colors

The provided approach computes a color palette $\mathcal{P}$ with colors derived from local image regions of an image I. A scoring system is applied to classify image regions according to their normalized entropy by penalizing the extraction from features with similar colors. The color extraction is performed incrementally, starting with an empty palette. Extracted colors are inserted after each iteration.

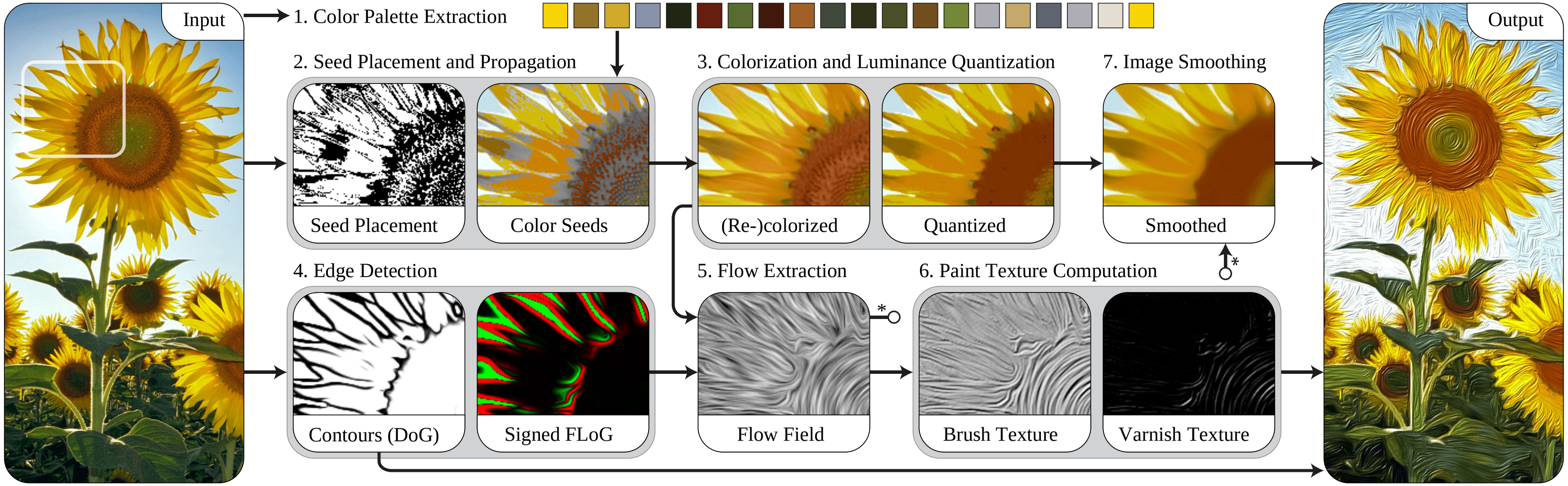


Figure VII.9.: Overview of the different stages performed for the proposed stylization technique that automatically transforms **(left)** color images to **(right)** filtered variants with an oil paint look. For a detailed description of the filtering stages the reader is referred to Section 2.

To find a new color for a given color palette $\mathcal{P}$, the approach seeks for an image region R of minimal region score

$$\mathcal{S}\left(R\right)=\frac{\frac{\eta(R)}{|R|}}{\frac{\bar{\omega}}{|R|}\cdot\frac{\bar{L}}{|R|}}=\frac{\eta\left(R\right)}{\bar{\omega}\cdot\bar{L}}\cdot|R|\,. \tag{VII.1}$$

$|R|$ denotes the area of the region and is used to normalize the weights: image entropy $\eta(R)$, lightness $\bar{L}$, and color distance $\bar{\omega}$. The entropy of the image is computed from the probabilities $p_R\left(c\right)$ that a binned color c appears in the image region R:

$$\eta\left(R\right)=-\sum_{c\in R}p_R\left(c\right)\log_2 p_R\left(c\right). \tag{VII.2}$$

The entropy weight is used to favor the color extraction from regions with constant color tones that preferably belong to a single image feature. At this, the probability functions are discretized using color bins of constant size. For all examples, the bin size is set to 256. To favor vivid colors, the region score is weighted according to the lightness $\bar{L}=\sum_{p\in R}L\left(p\right)$ — using sRGB gamma corrected intensities. Finally, to favor colors not yet present, the region score is weighted according to the minimum color distance to colors of the current palette $\mathcal{P}$:

$$\bar{\omega}=\sum_{p\in R}\omega\left(p\right)^2\quad\text{with}\quad\omega\left(p\right)=\min_{c\in\mathcal{P}}\Delta E\big(I\left(p\right),c\big)\,. \tag{VII.3}$$

This way, more priority is given to extracting palettes with diverging color tones. In Equation VII.1, the respective weights are divided by $|R|$ to yield the normalized entropy, average lightness, and average color distance.

To find a (rectangular) region with minimum score the approach proceeds heuristically, using the following algorithm. The steps are performed iteratively in CIE-Lab color space, n-times in total for a target palette size of n:

1. **Color Difference Mask**: To avoid recomputations in the subsequent steps, ω is precomputed via the minimal color distance (ΔE) for pixels in I to $\{c\in\mathcal{P}\}$, and the results are buffered in a color difference mask.

2. **Region Search**: Regions are computed for the horizontal and vertical scanlines of the input image. The color is extracted from the region with the better score (Figure VII.10):

   a) **First Pass**: $\mathcal{S}(R)$ is computed for all regions of width $\tau>0$ along the horizontal/vertical scanlines. The scanline $\mathcal{M}_x$ with minimal $\mathcal{S}(R)$ is determined. All examples set $\tau=3$.

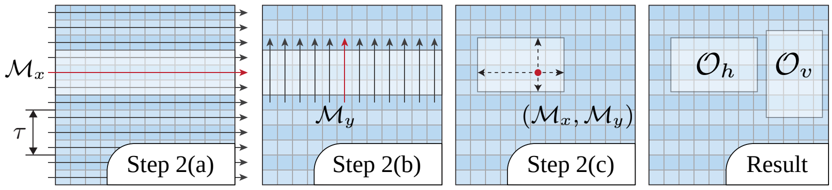


Figure VII.10: Schematic overview of the region selection (horizontal pass): (a) horizontal and (b) vertical scanlines with optimal score are determined, followed by (c) region growing to select $\mathcal{O}_h$—next to $\mathcal{O}_v$ for the vertical pass.

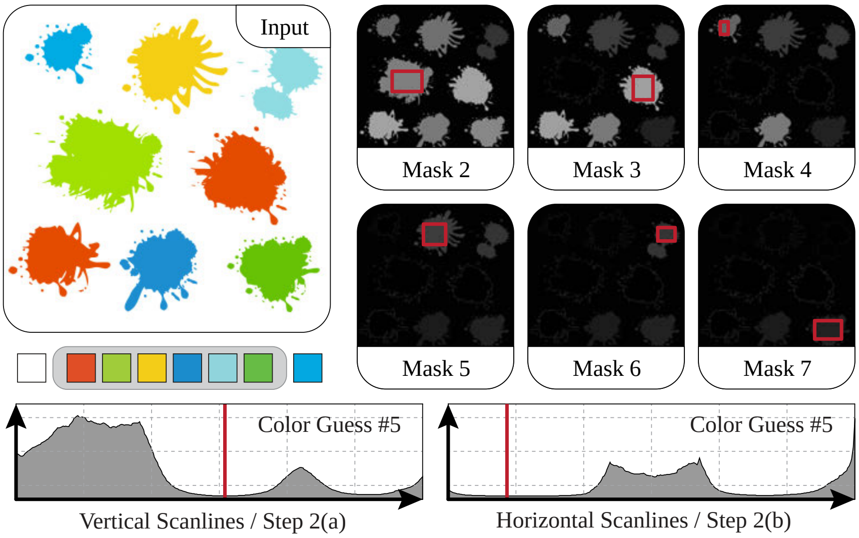


Figure VII.11: Color difference masks computed for an image with 8 colors (including white). The overlaid rectangles indicate the respective regions with minimum score used for the color extraction. The region scores for the fifth color guess are shown at the bottom.

b) **Second Pass**: The first pass is repeated for the orthogonal scanlines, bounded by the region that is defined by the scanline $\mathcal{M}_x$. Again, the scanline $\mathcal{M}_y$ with minimal $\mathcal{S}(R)$ is selected for further processing.

c) **Area Computation**: $\mathcal{S}(R)$ is determined iteratively for growing horizontal and vertical boundaries around the pixel position $(\mathcal{M}_x, \mathcal{M}_y)$ until a minimum value is reached.

3. **Color Extraction**: Once a region with minimum score has been identified, a representative color is extracted from the region and inserted into the palette $\mathcal{P}$. Thereby, the representative color is computed by finding a mode in the box-filtered histograms of the chrominance channels.

An example of the iterative computation of dominant colors, color difference masks, and optimal region selection is given in Figure VII.11. In a final step, the colors in $\mathcal{C}$ are sorted by their weighted count in the input image (i.e., thresholding ΔE) and color difference to previously sorted colors.

Discussion. The parameter $\xi = \Delta E^*_{ab} = 7$ was empirically determined as a good default value for thresholding, using the CIE76 formula

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2} \qquad \text{(VII.4)}$$

to compute the difference between two colors, where $\Delta E^*_{ab} \approx 2.3$ corresponds to a just noticeable color difference (Mahy *et al.* 1994). The proposed algorithm is compared to the median-cut algorithm in Figure VII.12. Here, colors of single features are accurately represented and not merged with colors of other features, e.g., the butterfly in Figure VII.12a, the eagle's beak in Figure VII.12c, the guitar in Figure VII.12d, and the red tones in Figure VII.12e. Further, it is observed that the provided approach is more resistant to noise as one can notice by the green background in Figure VII.12c, where a clustering approach may also derive false colors. Figure VII.13a demonstrates this stability by an image that has been artificially corrupted with Gaussian and impulse noise, where only small changes for the derived contemporary colors are noticeable when using the proposed approach. In addition, Figure VII.13b demonstrates

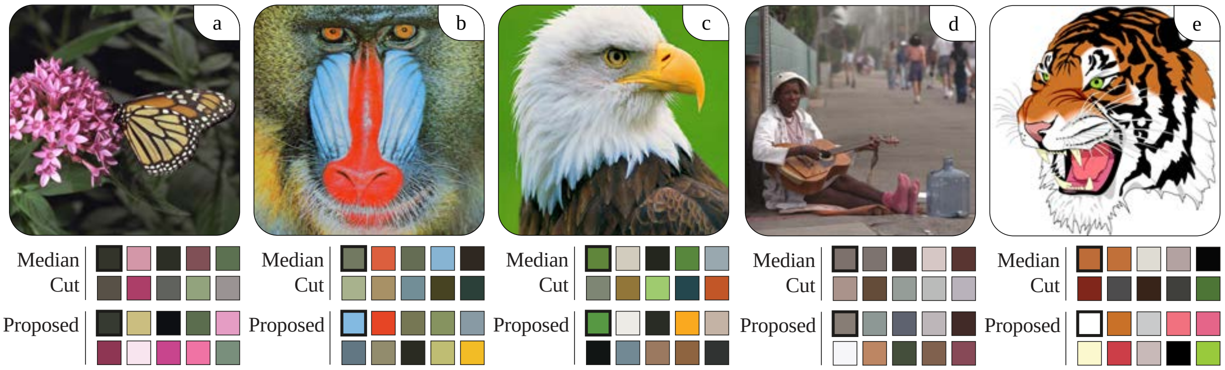


Figure VII.12.: Comparison of the median-cut algorithm to the proposed approach for dominant color extraction. The 10 dominant colors are sorted from left to right.

the stability for different image resolutions, where a down-sampled image still leads to plausible palettes with stable estimates for the pre-dominant colors. Finally, it is observed that the number of color extractions significantly affects if image features are accurately represented or not. To this end, one could use a metric to control the color coverage, e.g., to derive colors until the maximum and/or mean of the color difference mask (ω) falls below a given threshold. This way, more colors could be derived for colorful images without the need for content-dependent parameter adjustments such as the palette size. Here, the accuracy of the proposed algorithm can be further improved when using a generalized region growing approach that derives colors from feature-aligned regions that are not limited to be of rectangular shape.

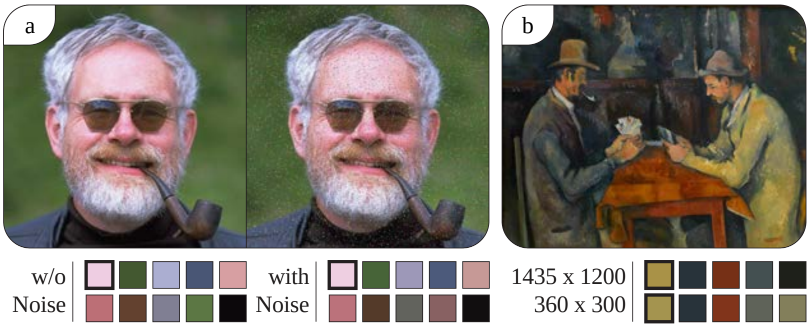
3. Image Quantization using Colorization

The goal is to quantize the input image I using the extracted dominant colors. This task is formulated as an optimization problem, performing a recolorization (Levin *et al.* 2004): Given the intensity of the input image and a number of colored seed pixels, the colors should be propagated to the remaining pixels such that pixels with similar intensities have similar colors.

1st Pass: Colorization. The optimization is performed with the constraint that *seed pixels* $c(\mathbf{r_i})$ are set to the respective color component of the dominant color c_p in $\mathcal{P}$ with minimal distance:

$$c(\mathbf{r_i}) = \operatorname*{arg\,min}_{c_p \in \mathcal{P}} \Delta E(c_p, I(\mathbf{r})) \tag{VII.5}$$

Figure VII.13: Stability tests of the dominant color extraction: (a) applied to an image corrupted with 2% Gaussian and 5% impulse noise, (b) applied to a down-sampled image of the painting *The Cardplayers* by P. Cézanne (1892). In all cases, plausible palettes with stable pre-dominant colors are derived.



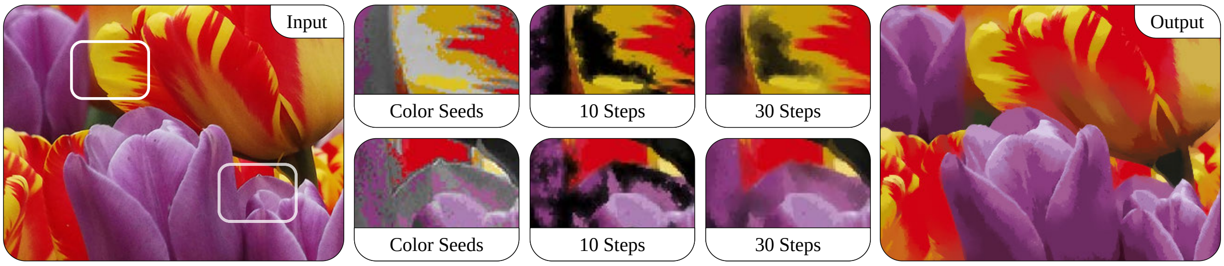


Figure VII.14.: Image quantization using the algorithm described in Section 3 with a palette of 26 colors and $\alpha = 5.5$ for automatic seed placement. The optimization problem was iteratively solved with the "generalized minimum residual" method (Saad & Schultz 1986).

if and only if the minimal color distance falls below a threshold $\alpha > 0$. This replaces the interactive placement of "scribbles" described by Levin *et al.* (2004). For a given color channel $\mathcal{C}$, the recolorized channel $\mathcal{C}'$ is then computed via the objective function

$$\begin{aligned} &\arg\min_{\mathcal{C}} \sum_{\mathbf{r}} \Big(\mathcal{C}(\mathbf{r}) - \sum_{\mathbf{s}\in\mathcal{N}(\mathbf{r})} w_{\mathbf{rs}}\, \mathcal{C}(\mathbf{s})\Big)^2, \\ &\text{subject to } \mathcal{C}(\mathbf{r}_i) = c(\mathbf{r}_i) \text{ for } i \in \text{Seeds} \end{aligned} \tag{VII.6}$$

where $w_{\mathbf{rs}}$ denotes the squared difference between the luminance values of the pixels $\mathbf{r}$ and $\mathbf{s}$, and $\mathcal{N}(\mathbf{r})$ being the 8-connected neighborhood of $\mathbf{r}$. The objective function yields a large sparse system of linear equations that is solved for both chrominance channels of the input image in CIE-Lab color space using the "generalized minimum residual" method (Saad & Schultz 1986) (Figure VII.14).

2nd Pass: Luminance Quantization. A second pass is introduced for luminance quantization, where the objective function is used to recompute the luminance channel. To this end, the weighting function $w_{\mathbf{rs}}$ is reformulated to compute the squared difference between the two *hue* values at pixels $\mathbf{r}$ and $\mathbf{s}$ of the recolorized image. The *hue* value is obtained by converting the recolorized image to CIE-LCh color space—as cylindrical version of the CIE-Lab color space. The recomputed luminance channel is then combined with the color channels of the first pass to yield a recolorized image with potential color transitions.

3rd Pass: Flow-based Gaussian Smoothing. Finally, the quantized image is post-processed by a Gaussian filter with standard deviation σ_q that is adapted to the local structure of image features, derived from the smoothed structure tensor (Kyprianidis & Döllner 2008). This creates smoothed outputs at curved boundaries and, in general, a more painterly look (Figure VII.15).

Discussion. Examples using the provided method are presented in Figure VII.14, Figure VII.15 and Figure VII.16. Contrary to schemes based on global optimization methods, it is observed that the provided method produces outputs with better feature contrasts but higher variances to the original images (e.g., Figure VII.15). This behavior can be explained by deriving colors from local image regions—e.g., instead of prioritizing a global variance minimization—and the usage of color differences as weighting factors in the scoring system introduced in subsection 2. The threshold for automatic seed placement should be set greater or equal to the threshold used for the color extraction to use all colors of the derived palette. Here, the parameter $\alpha = \xi = 7$ was empirically determined as

Figure VII.15: Comparison of the proposed quantization method with the median-cut algorithm. The provided approach preserves feature contrasts at a better scale. The smoothed quantization result is based on a flow-based Gaussian filter (Kyprianidis & Döllner 2008).

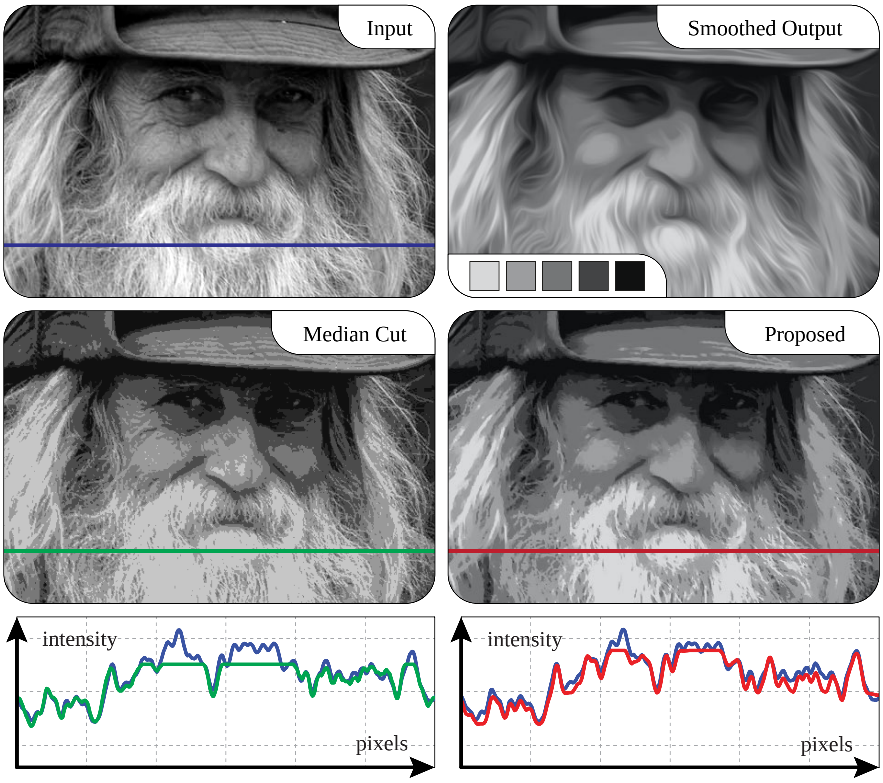
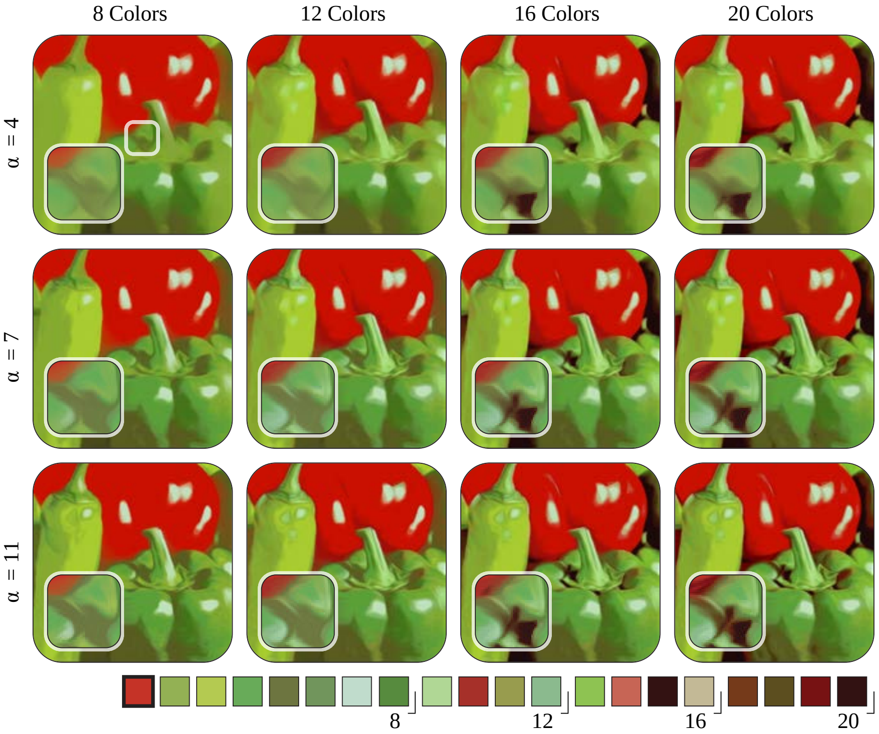


Figure VII.16: Example for adjusting the seed placement threshold α and the number of colors for the derived palette to adjust the LoA. The results include a post-processing step by flow-based Gaussian smoothing.



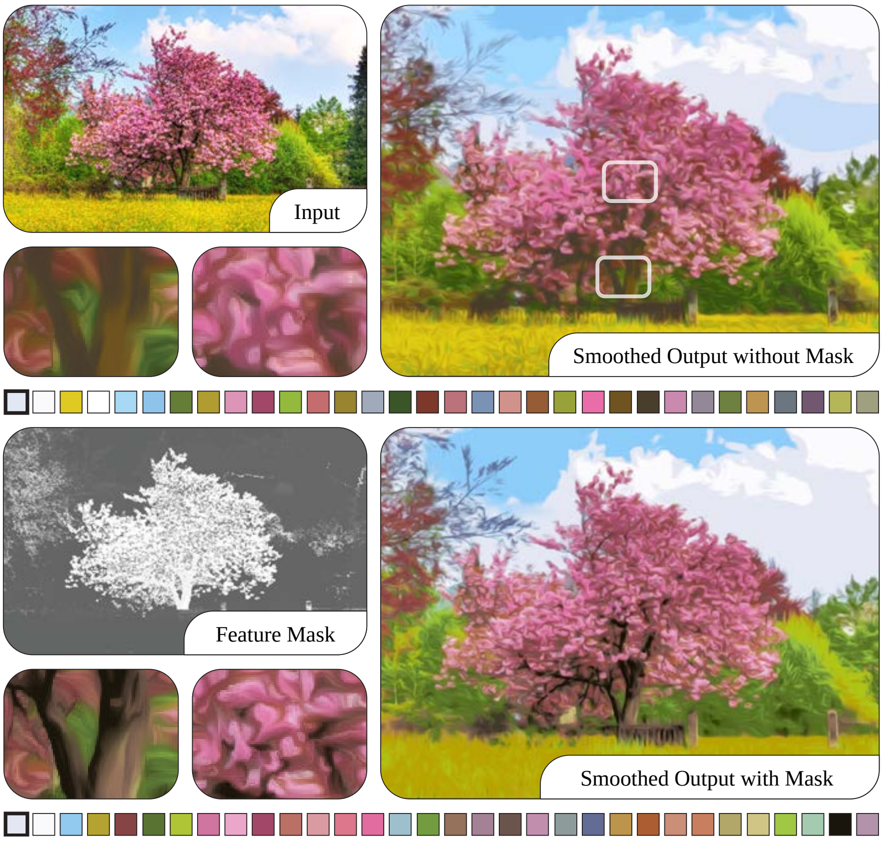


Figure VII.17: Using image masks to weight the color extraction according to important or salient image features. Here, the derived color palette for the modified weighting (bottom) represents the tree more accurately than the standard procedure (top), while the environment becomes more abstract. The salient region detection is based on the algorithm of Cheng *et al.* (2013).

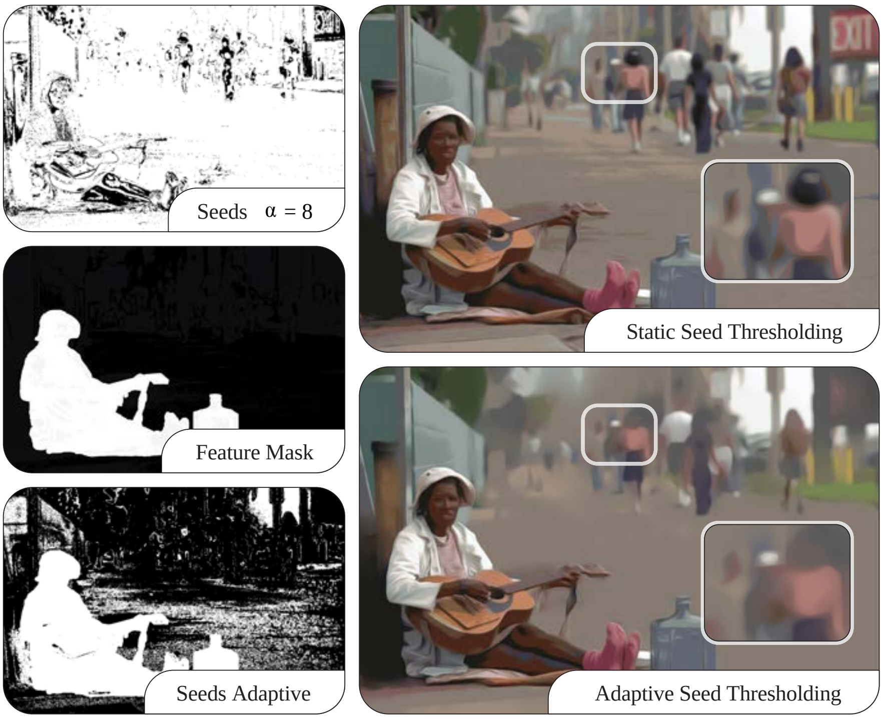


Figure VII.18: Using adaptive thresholds for the placement of color seeds to control the level of abstraction for image features. (top) Static thresholds $\alpha = 8$, (bottom) adaptive thresholds $\alpha \in [3, 11]$ selected according to an importance mask to increase the LoA for features in the background.

a good default setting. The threshold may also be set lower to initially place fewer color seeds and thus induce more color blendings, or may be set higher to result in more details and crisp boundaries between image features. In addition, the thresholding may be combined with reducing the number of colors in the derived palette to further adjust the LoA. Figure VII.16 illustrates the mutual impact of these two parameterization possibilities.

Adaptive Image Quantization. An advantage of the proposed quantization scheme becomes apparent when the LoA should be adjusted according to image contents, e.g., based on feature semantics, image saliency, or a foreground/background separation. According to Figure VII.16, two possibilities are considered for local-based adjustments: on the one hand, user-defined weights can be injected into Equation VII.1 to adapt the color extraction, and on the other hand the seed threshold α can be adaptively computed. Both adjustments were evaluated by using importance masks—explicitly defined prior to processing—to guide the color extraction to features of interest. Figure VII.17 shows a result where more contemporary colors are derived from a salient image feature to depict it with more details—using the algorithm of Cheng *et al.* (2013) for saliency estimation—without changing the number of color extractions. To this end, the score computation defined in Equation VII.1 is changed to:

$$\mathcal{S}(R) = \frac{\eta(R)}{\bar{\omega} \cdot \bar{L} \cdot \bar{\Omega}} \cdot |R| \,, \tag{VII.7}$$

where $\bar{\Omega}$ refers to the importance for pixels in region R as defined by a normalized input mask. In addition, Figure VII.18 shows a result where seed thresholds $\alpha \in [\alpha^{+}, \alpha^{-}]$ are linearly interpolated according to an importance mask to increase the LoA in the background regions of an image.

4. Paint Texture Synthesis

Oil painting is a time-consuming process that often comprises multiple layers of paint and drying phases. During finishing, thin protective layers (*varnish*) may be coated onto the paint for protection against dirt and dust, and to even out its final appearance. This yields two characteristics of oil paint textures: (1) reliefs of varying thickness according to the used brushes and applied number of layers with (2) a matte or glossy tint. The first effect may be perceived as subtle shading that is caused by external, off-image illumination, and the second effect as specular highlighting. To simulate both effects, first, a flow field is synthesized by using the local orientation information obtained from the smoothed structure tensor (Kyprianidis & Döllner 2008), afterwards, paint textures are synthesized by using the flow field for shading.

Flow Field Computation. Local orientation information is derived from an eigenanalysis of the smoothed structure tensor (Brox *et al.* 2006b), a method that provides stable estimates and can be computed in real-time (Kyprianidis & Döllner 2008). Line integral convolution (Cabral & Leedom 1993) is then performed along the stream lines defined by the minor eigenvector field of the smoothed structure tensor. The obtained flow field, however, may contain singularities and blurred feature boundaries leading to visual artifacts in the paint textures. To this end, the following enhancements are used (Figure VII.19):

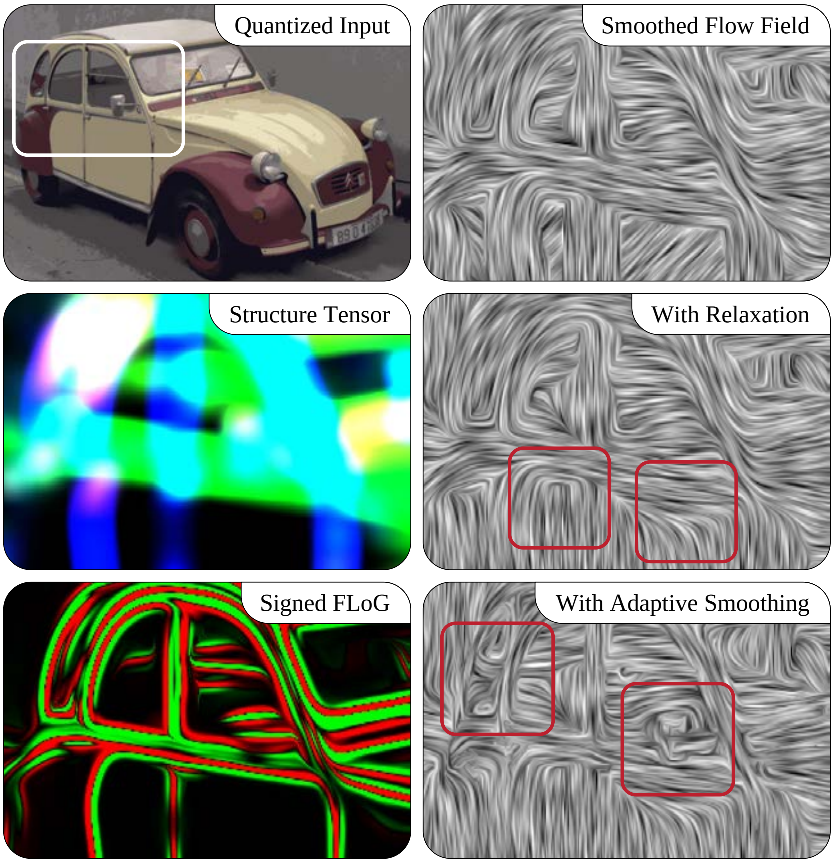


Figure VII.19: Enhancements for the smoothed structure tensor (here: $\sigma_s = 8$) to derive flow fields.

(top) input and original flow computation

(middle) visualized structure tensor (black regions refer to singularities) and relaxation to avoid singularities

(bottom) visualized sign of the FLoG which is thresholded for adaptive smoothing of the structure tensor

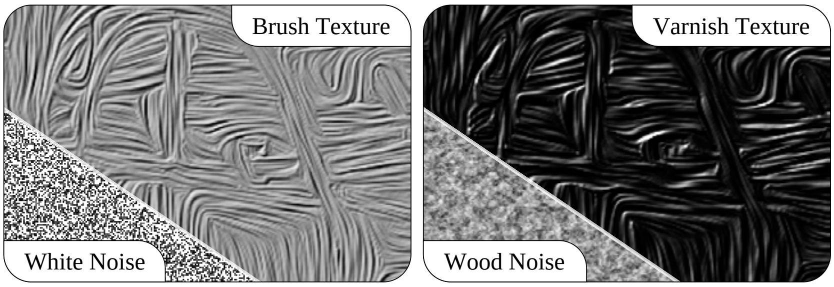


Figure VII.20: Paint textures computed for the flow field in Figure VII.19 using parameters: $\sigma_b = 8.0$, $k_{scale} = 10.0$, $k_{specular} = 3.0$, $k_{shininess} = 8.0$.

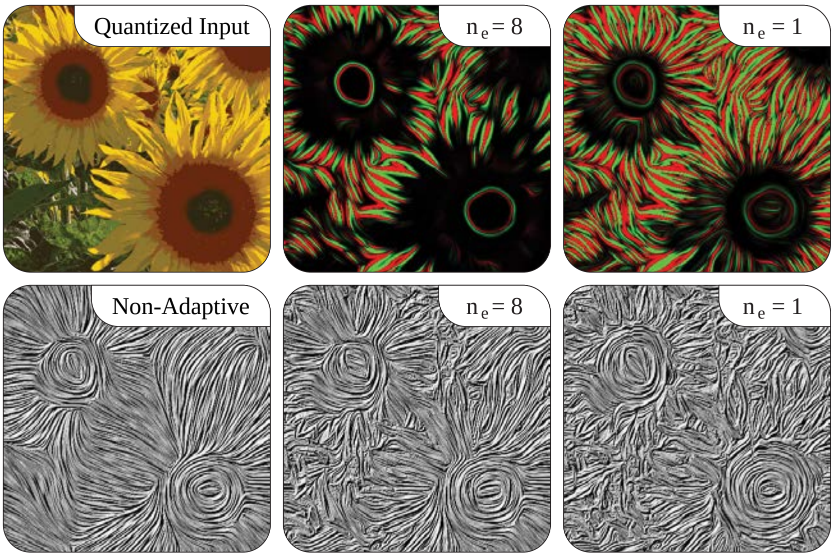


Figure VII.21: Varying the number of iterations for orientation-aligned bilateral filtering (n_e) of the quantized image, used for FLoG filtering.

(top) visualized sign of the FLoG filter

(bottom) brush texture with and without adaptive smoothing

1. *Relaxation*: The quantized image may provide large areas of solid color tones where gradient information are unreliable or undefined. To this end, structure tensors with low gradient magnitudes are replaced by inpainted information via relaxation (Figure VII.19 middle).
2. *Adaptive Smoothing*: The structure tensor is adaptively smoothed to avoid filtering over feature boundaries (Figure VII.19 bottom). Here, the sign of the flow-based FLoG—derived from the quantized color image—is used for thresholding: a Gaussian smoothing with standard deviation σ_s is adapted to exclude pixel values from weight averaging when the sign of the FLoG flips while crossing feature boundaries.

For a detailed description on the relaxation and FLoG computation, the interested reader is referred to the work by Kyprianidis (2011).

Paint Texture Synthesis. Procedural monochromatic noise is blurred in gradient flow direction defined by the minor eigenvector field of the adaptively smoothed structure tensor. This results in flow images similar to those produced by line integral convolution (Cabral & Leedom 1993), but using a flow-based Gaussian filter with standard deviation σ_b to elongate the brush reliefs. The blurred noise is interpreted as fine-grained height field with normals N and illuminated using a directional light source L. Principles of Phong shading (Phong 1975) are used to render a brush texture $\mathcal{T}_B$ and a varnish texture $\mathcal{T}_V$ with pixels p:

$$\begin{aligned} \mathcal{T}_B(p) &= 0.5 + N(p) \cdot L, \\ \mathcal{T}_V(p) &= k_{specular} \cdot (N(p) \cdot L)^{k_{shininess}}. \end{aligned} \tag{VII.8}$$

At this, $k_{specular}$ and $k_{shininess}$ are used to adjust the varnish texture. An additional factor k_{scale} scales the height field to control the relief strength (Figure VII.20). The standard deviation σ_b used for noise filtering is a key parameter to control the LoA of the paint textures. Here, it is observed that the provided enhancement for adaptive smoothing of the structure tensor is crucial to preserve salient feature curves. Figure VII.21 shows how to further adjust the LoA when pre-processing the quantized image—used as input for FLoG filtering—by an orientation-aligned bilateral filter (Kyprianidis & Döllner 2008) with a varying number of iterations n_e. This approach has also proven to be effective in cartoon-like filtering (Winnemöller *et al.* 2006) to adjust contour lines. In addition, different noise implementations were evaluated, and it is observed that high frequency noise simulates brush characteristics quite naturally (Figure VII.20), but may also be based on lower frequencies and amplitudes to adjust the brush size (Figure VII.22).

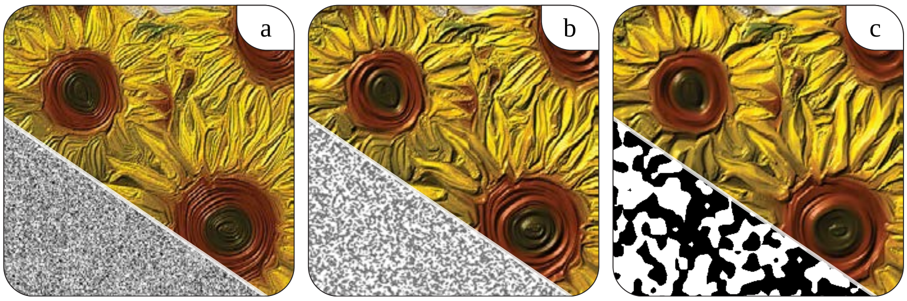


Figure VII.22: Noise configurations used to adjust the virtual brush. $\sigma_b = 20.0$, $k_{scale} = 20.0$, $k_{specular} = 0.5$, $k_{shininess} = 20.0$.

(a) black and white noise

(b) grayscale noise

(c) wood noise

Table VII.1.: Overview of parameters with value ranges used to adjust the color image quantization and paint texture synthesis. Left-end values typically refer to a low LoA, whereas right-end value refer to a high LoA.

	Parameter	Value Range	Effect
Quantization	n	36 – 8	Size of color palette
	α	11.0 – 4.0	Color seed threshold
	σ_s	2.0 – 20.0	Structure tensor smoothing (std. dev.)
	σ_q	0.0 – 20.0	Quantized image smoothing (std. dev.)
Paint Textures	n_e	0 – 10	Iterations of bilateral filtering for FLoG
	σ_b	2.0 – 20.0	Noise smoothing (std. dev.)
	k_{scale}	20.0 – 0.0	Relief strength
	$k_{specular}$	5.0 – 0.3	Specularity for varnish
	$k_{shininess}$	30.0 – 10.0	Shininess for varnish

Image Composition. Finally, the brush texture is multiplied with the smoothed color image, and the intermediate result is blended with the varnish texture using *linear dodge* as blend mode. Optionally, contour lines are enhanced by using a flow-based DoG filter (H. Kang *et al.* 2009) and a canvas texture is blended with the output to further enhance the sensation of depth. For the latter, the paint textures may also be filtered in image regions of low saturation to imitate layering at certain feature boundaries (e.g., Figure VII.7).

Adaptive Paint Texture Synthesis. The paint texture synthesis is local and fast (refer to the evaluation section), and thus may also be adaptively computed according to user interaction or image contents. Table VII.1 summarizes the parameters used to control the LoA on a per-pixel basis, where $\mathcal{K}_\mathcal{Q} = (n, \alpha, \sigma_s, \sigma_q)$ adjusts the image quantization and abstraction, and $\mathcal{K}_\mathcal{T} = (n_e, \sigma_b, k_{scale}, k_{specular}, k_{shininess})$ adjusts the brush and varnish textures. Figure VII.24 demonstrates a depth-dependent linear interpolation between two parameter sets to depict fewer or more details in the background regions of an image. In addition, Figure VII.23 shows an approach where image saliency is used to direct the granularity of the brush and varnish textures. Additional effects can be implemented, e.g., based on light field data to produce stylized depth of field effects (Bousseau 2009) or feature semantics with qualitative parameter sets for content-aware filtering.

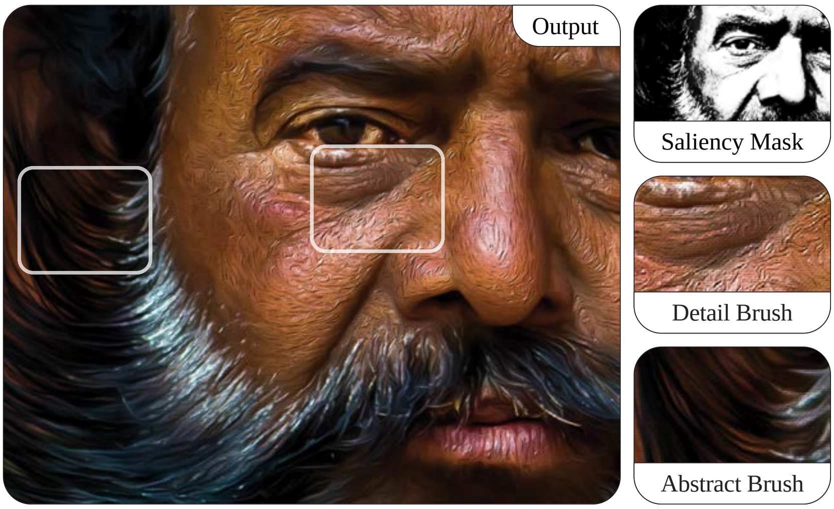


Figure VII.23: Saliency-based filtering output of a portrait with detailed paint textures in image regions with facial skin.

(High saliency)
$\mathcal{K}_\mathcal{Q} = (\text{NA}, \text{NA}, 4.0, 6.0)$ and $\mathcal{K}_\mathcal{T} = (0, 6.0, 7.0, 0.8, 20.0)$

(Low saliency)
$\mathcal{K}_\mathcal{Q} = (\text{NA}, \text{NA}, 4.0, 16.0)$ and $\mathcal{K}_\mathcal{T} = (8, 16.0, 2.0, 0.3, 20.0)$

For the computation of the saliency mask, the algorithm of Cheng *et al.* (2013) was used.

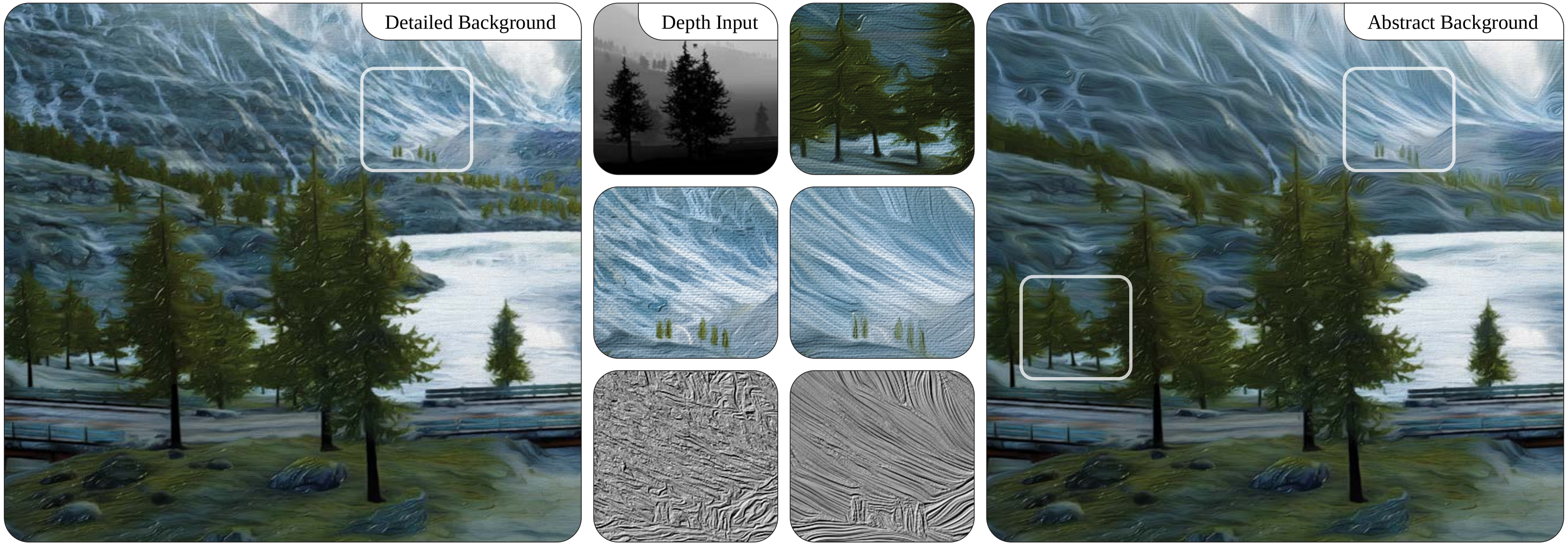


Figure VII.24.: Depth-dependent synthesis (color and depth input rendered with Unreal Engine 4) of paint textures to vary the LoA in the background region of a rendered image. Foreground: $\mathcal{K}_{\mathcal{Q}} = (\mathrm{NA}, \mathrm{NA}, 4.0, 20.0)$ and $\mathcal{K}_{\mathcal{T}} = (0, 20.0, 10.0, 0.8, 20.0)$, background (left): $\mathcal{K}_{\mathcal{Q}} = (\mathrm{NA}, \mathrm{NA}, 1.0, 8.0)$ and $\mathcal{K}_{\mathcal{T}} = (0, 8.0, 6.0, 0.8, 20.0)$, background (right): $\mathcal{K}_{\mathcal{Q}} = (\mathrm{NA}, \mathrm{NA}, 10.0, 20.0)$ and $\mathcal{K}_{\mathcal{T}} = (2, 20.0, 10.0, 0.8, 20.0)$. The visualized brush textures do not include additional filtering by saturation.

5. Interactive Painting

Users strive for a global parameterization-trade-off that corresponds to different visual requirements in local image regions. Recently, consumer photo-editing products started to extend the concept of non-destructive per-pixel parameterizations from alpha masking to a small set of image computations (adjustments) by means of parameter masks. This approach is extended by exposing filter parameters and intermediate filtering results as parameter maps that can be manipulated via painting metaphors. This enables (1) artistic control over the stylization process, and (2) the modification of intermediate filter outputs of inadequate local parameterizations (Figure VII.25).

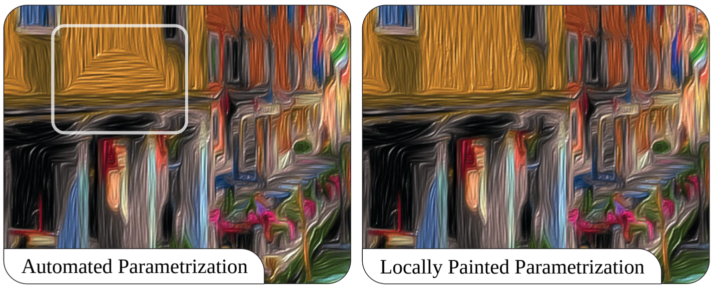


Figure VII.25.: "Failure case" of a filtering result and its correction using the painting interface for per-pixel parameterization.

Local Parameter Painting

The proposed method extends the concept of a specialized, locally computed parameterization (Todo *et al.* 2007) to a generalized configuration by brush-painting within parameter spaces of local image filters. At this, parameters are encoded as *parameter maps* and adjusted via *virtual brush models* according to well-defined action sequences. Thereby, the following system is denoted:

- A *parameter map* is a typed map that substitutes either a uniform filter parameter or an intermediate computational result. These maps are usually aligned with the input image, but might also cover sub-regions and have different image resolutions.

- An *action* defines locally typed computations on parameter maps, e.g., *replace*, *add*, *scale*, or *blur*. It is filter-independent and can be assigned to equally-typed parameter maps. The type usually depends on the number of a parameter's coefficients.

- A *brush shape* specifies rules and parameters for the dynamic creation of two-dimensional weight masks. Thereby, atomic shapes—either by functional definition or shapes from vector graphics—are (randomly) placed and encoded as distance maps while satisfying specific constraints, e.g., with respect to softness, jittering, or scattering.

- A *brush stroke* is derived from a set of sequential user inputs, e.g., attributed paths with interpolated position, orientation, and pressure values. Here, a temporal input mask is created by computing dynamic brush shapes along the path, enabling per-pixel weighting of actions.

- A *brush* maps a sequence of actions to parameter maps. While drawing, these actions are applied and weighted by the temporal brush stroke mask.

This system provides a generic interface and is used to locally adjust filter parameters and intermediate filter results through painting (schematized in Figure VII.26). The brush implementation is hardware-accelerated and is used, in particular, to locally adjust the parameters defined by $\mathcal{K}_\mathcal{Q}$ and $\mathcal{K}_\mathcal{T}$ (Figure VII.27).

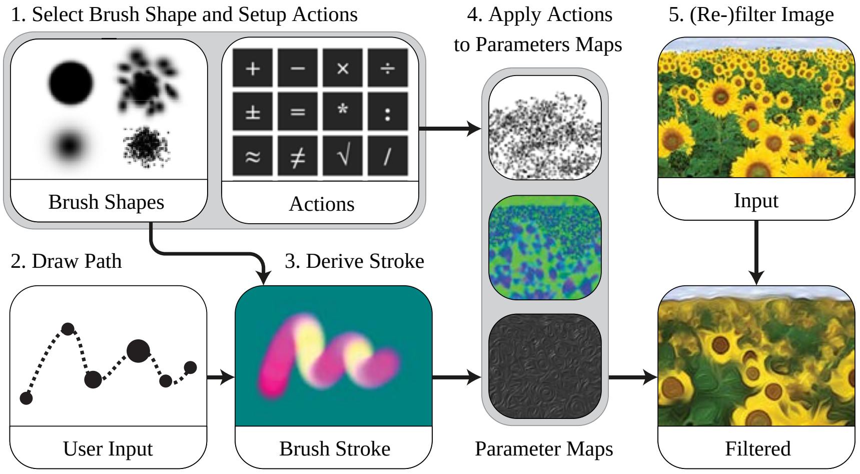


Figure VII.26: Schematic overview of the proposed interface for per-pixel parametrization: brush strokes are derived from drawn paths and applied to selected parameter maps. The maps' modified local parameters are then used to (re-)filter the image.

Brushes

The shape and actions of a brush can be parameterized by constant values or dynamically mapped to user inputs, e.g., the pressure and orientation of a digital pen, or gestures for touch-enabled devices. Technically, the implementation enables users to create and customize brushes at run-time, but which demands a detailed understanding of the underlying filtering stages. To this end, a number of predefined brushes is provided:

- A *relief brush* increases or decreases the relief strength by applying a multiply action to the height scale k_{scale}.

- A *varnish brush* allows to adjust the specularity and shininess of the varnish texture. It applies two multiply actions to the parameter maps of $k_{shininess}$ and $k_{specular}$.

- Two *LoA brushes*: one to adjust the structure tensor smoothing σ_s and another to perform bilateral filtering and apply unsharp masking effects.

- A *flow brush* to adjust the tangential information of the structure tensor, which is especially helpful to exaggerate or fix inadequate stroke directions.

- A *colorization brush* that allows to fade between the color output and a grayscale version.

- An *eraser brush* that reverts to initial painting states for all parameter maps or those of specific brushes.

Additional brushes, e.g., for painting bristle structures by scaling the noise frequency or adjusting additional filter kernels for smoothing (σ_q and σ_b) may also be implemented. To simplify the mapping of actions to parameter maps, a single texture for every parameter or intermediate result is used. For large images, the application of actions is restricted to sub-regions of the parameter maps—according to the bounds of a brush stroke—to maintain a responsive painting system.

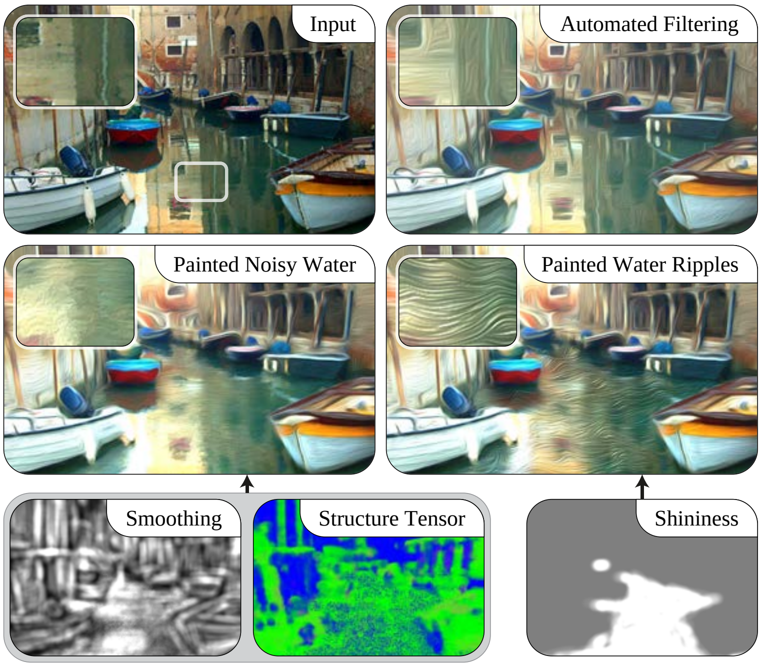


Figure VII.27: Example of manual corrections made for parameter layers to adjust the paint textures: smoothing of the flow field, flow direction, and shininess.

Image Warping

Artists often exaggerate figures and shapes to amplify the mood of their paintings. Prominent examples are art works from the Expressionism era such as Edvard Munch's *The Scream* (1893–1910). Image filtering, however, is generally less suited for intentional shape abstraction. Here, one approach is to use a complementary brush tool that performs local image warping. Starting with a virtual, regular grid, the user is able to shift local grid points by brush-based painting to create effects of local compression and distortion (Figure VII.28). A similar approach was used before with facial constraints to create caricatures from photographs (B. Gooch *et al.* 2004). The adapted grid is then used for texture parameterization and to resample the quantized image by bilinear interpolation. Finally, the warped image serves as input for the paint texture synthesis—which may also be performed during warping for immediate visual feedback. Alternatively, image segmentation and mass-spring systems may be used to create outputs with more deliberate shape abstractions as demonstrated by J. Li & Mould (2015).

6. Results

The dominant color extraction was implemented using C++, the colorization and filtering stages on the GPU with CUDA, and the painting interface with Qt. All images were processed on an Intel® Xeon™ 4× 3.06 GHz and NVidia® GTX 760 GPU with 4 GByte VRAM. The painting system was tested with a 85" multitouch monitor with Ultra-HD resolution (Figure VII.29). A 800 × 600 pixel image is processed in 50 seconds for a palette with 25 colors. Here, the color extraction is currently the limiting stage, followed by the colorization, and the paint texture synthesis that performs in real-time for images with HD resolution. As demonstrated in Figure VII.13, the color extraction also provides stable estimates when processing downscaled images—up to the second pyramid level to speed up the processing by a factor of two. To enable interactive performance during painting,

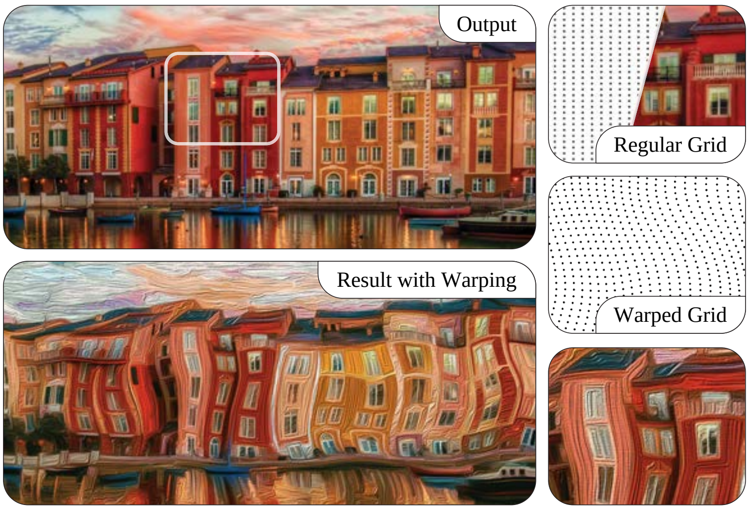


Figure VII.28: Image warping using grid-based resampling, parameterized by a virtual brush tool. The warping is performed prior to paint texture synthesis.

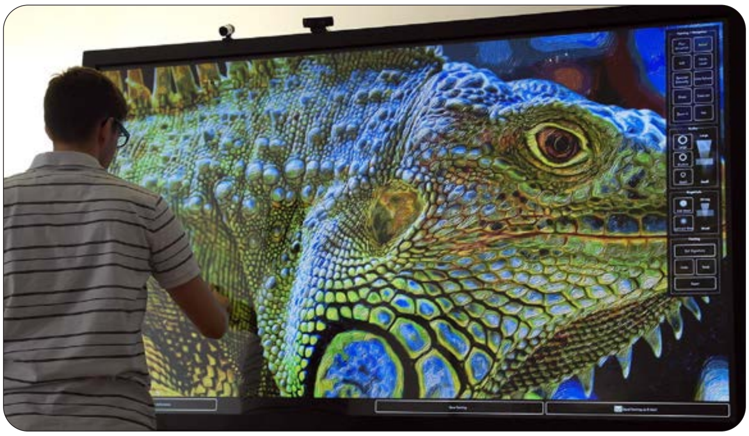


Figure VII.29: Touchscreen and user interface for the proposed painting system.

the stages after the image quantization are optimized to process only those image regions that require a recomputation. At this, the processing performs three steps per render cycle: 1) local regions defined by the virtual brush are buffered and extracted from the quantized image and parameter layers (stored in main memory) and are transferred to GPU memory, 2) the sub-images are processed according to the brush mode and the results are *blit* to the framebuffer for immediate visual feedback, 3) the filtered sub-images are transferred back to main memory. Using pitch linear memory on the described test system, this procedure enables to interactively adjust images up to 100 MP.

The proposed color quantization gives control to adjust the LoA. A comparison to a non-quantized version of an image is shown in Figure VII.30, and demonstrates how the quantization is able to filter detail information and produce large areas of solid color tones. In particular, the latter effect yields varying scales of the paint texture, i.e., to simulate thick brushes. Figure VII.31 and Figure VII.32 show comparisons of the proposed technique to previous works. In contrast to stroke-based rendering, the proposed method produces outputs with more soft color blendings (Figure VII.31) but is also able to simulate reliefs of varying thickness and strong abstraction (e.g., the background in Figure VII.32). In addition, the local parameterization capabilities are mandatory to provide artistic control over the depiction of single image features (e.g., the face in Figure VII.32), i.e., to provide an adaptive LoA that is similar to hybrid stylization techniques,

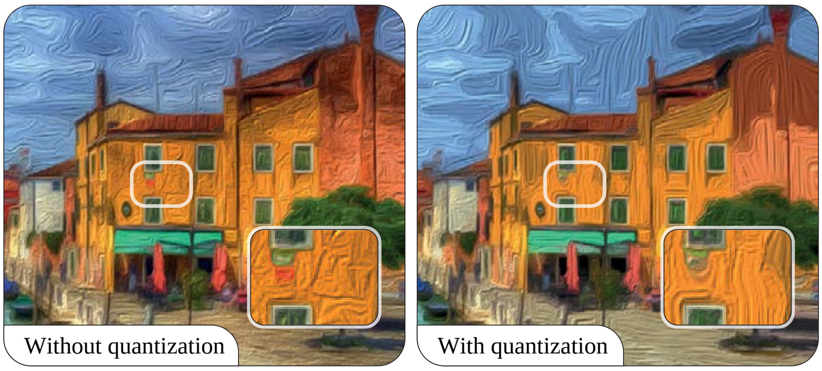


Figure VII.30: Two stylized versions: (left) without prior quantization, (right) with prior quantization. The quantized version produces a more abstract look with respect to the color range and scale of the paint texture.

Parameters used:
$\mathcal{K}_{\mathcal{Q}} = (40, 8.0, 3.0, 14.0)$,
$\mathcal{K}_{\mathcal{T}} = (0, 14.0, 5.0, 1.0, 16.0)$.

e.g., as is used in the work of Zeng *et al.* (2009). More images processed by the proposed method are shown in Figure VII.33, where the LoA is adapted to the image contents to have more distinct colors in colorful images (e.g., *Venice*) and wide filter kernels for Gaussian smoothing to obtain soft color blendings (e.g., the *landscape*).

Limitations

An inherent limitation of the proposed method is that it does not reach the qualities of shape abstraction as provided by top-down stroke-based rendering techniques such as the method of Zhao & Zhu (2010). The proposed painting system with complementary warping gives users some local control to adjust the LoA. Further, image features that should be represented with soft color blendings may be filtered with hard transitions, depending on the seed color placement. Eventually, this requires manual effort to locally adjust the thresholds for the color quantization. Finally, the performance of the colorization currently does not enable interactive color refinements. One approach to alleviate this issue is to visualize intermediate results of the iterative solver, or to accelerate the colorization using fast intrinsic distance computations (Yatziv & Sapiro 2006). Alternatively, a color grading may use lookup tables (LUTs) to enable real-time performance (Selan 2004), but which does not involve color quantization—see Figure VII.34 for results made with a mobile implementation.

3. Summary

This chapter provides an approach for automated, image-based abstraction that handles massive, tiled aerial photography, and an approach for transforming images into filtered variants with an oil paint look. For oil paint filtering, the proposed color extraction and colorization methods enable to quantize color images according to their dominant color palette. Results show that the proposed quantization scheme is able to represent selected image features accurately and provide homogeneous outputs in the color domain. The flow-based image abstraction and proposed paint texture synthesis simulate paint with soft to moderate color blendings, perform in real-time to enable interactive refinements, and facilitate per-pixel parameterizations to direct the LoA to user-defined or salient image regions. Several results demonstrate the manifold application of the provided techniques to different types of aerial and terrestrial photography.

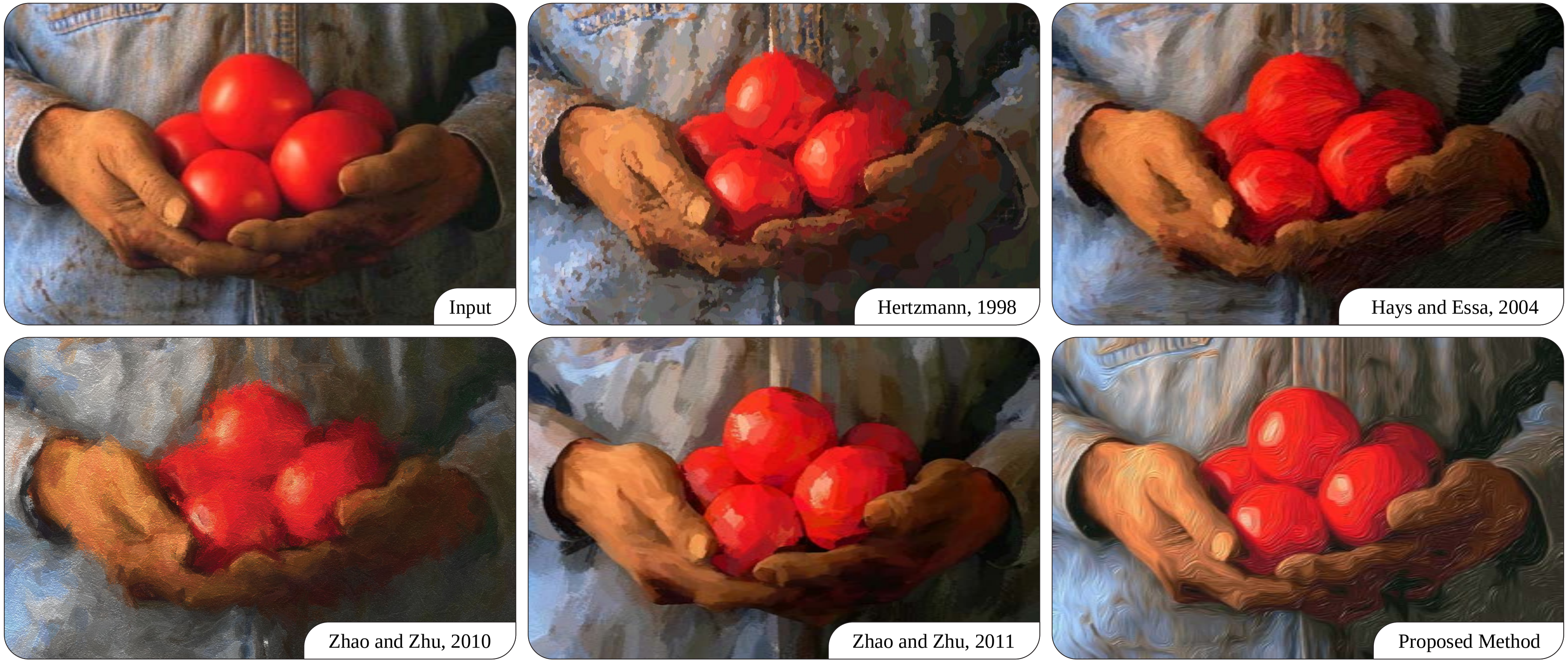


Figure VII.31.: Comparison of the proposed method with stroke-based rendering techniques. LRTB: input, from Hertzmann (1998), from Hays & Essa (2004), from Zhao & Zhu (2010), from Zhao & Zhu (2011a), and proposed method with $\mathcal{K}_{\mathcal{Q}} = (34, 7.0, 6.0, 10.0)$ and $\mathcal{K}_{\mathcal{T}} = (4, 10.0, 3.0, 1.0, 10.0)$. Each method produces visually distinct outputs of varying expressiveness, texture, and feature alignment. For instance, the method of Hertzmann (1998) aligns brush strokes in a color-matching scheme, but tends to overdraw features in regions of low contrast. Hays & Essa (2004) use real brush stroke textures that are feature-aligned with globally interpolated orientations, yet their approach lacks complementary colors in neighboring strokes. This effect is simulated in the method of Zhao & Zhu (2011a) to explicitly emphasize feature contrasts. Zhao & Zhu (2010) simulate the perceptual ambiguities known from abstract art, where shape-simplifying abstraction plays a major role. Finally, the proposed method produces soft color blendings with no visible borders between brush strokes, yet without the capability for explicit shape abstraction.

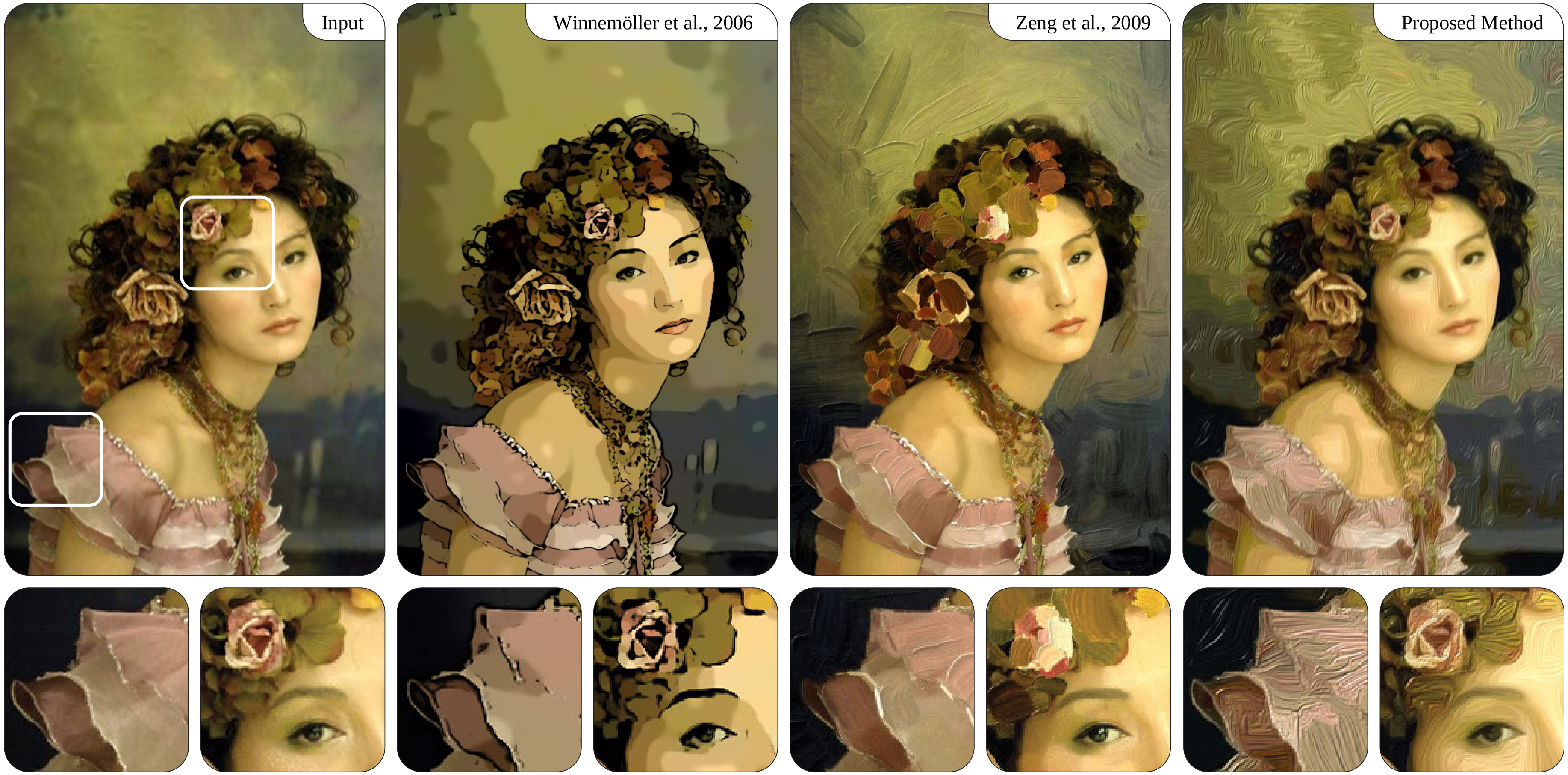


Figure VII.32.: Comparison of the proposed method with the image filtering technique of Winnemöller *et al.* (2006) and the stroke-based rendering technique of Zeng *et al.* (2009). Parameters: $\mathcal{K}_{\mathcal{Q}} = (38, 7.0, 16.0, 16.0)$, base: $\mathcal{K}_{\mathcal{T}} = (0, 16.0, 5.0, 1.0, 16.0)$ and skin: $\mathcal{K}_{\mathcal{T}} = (0, 16.0, 1.0, 1.0, 10.0)$ with no color quantization in facial regions..

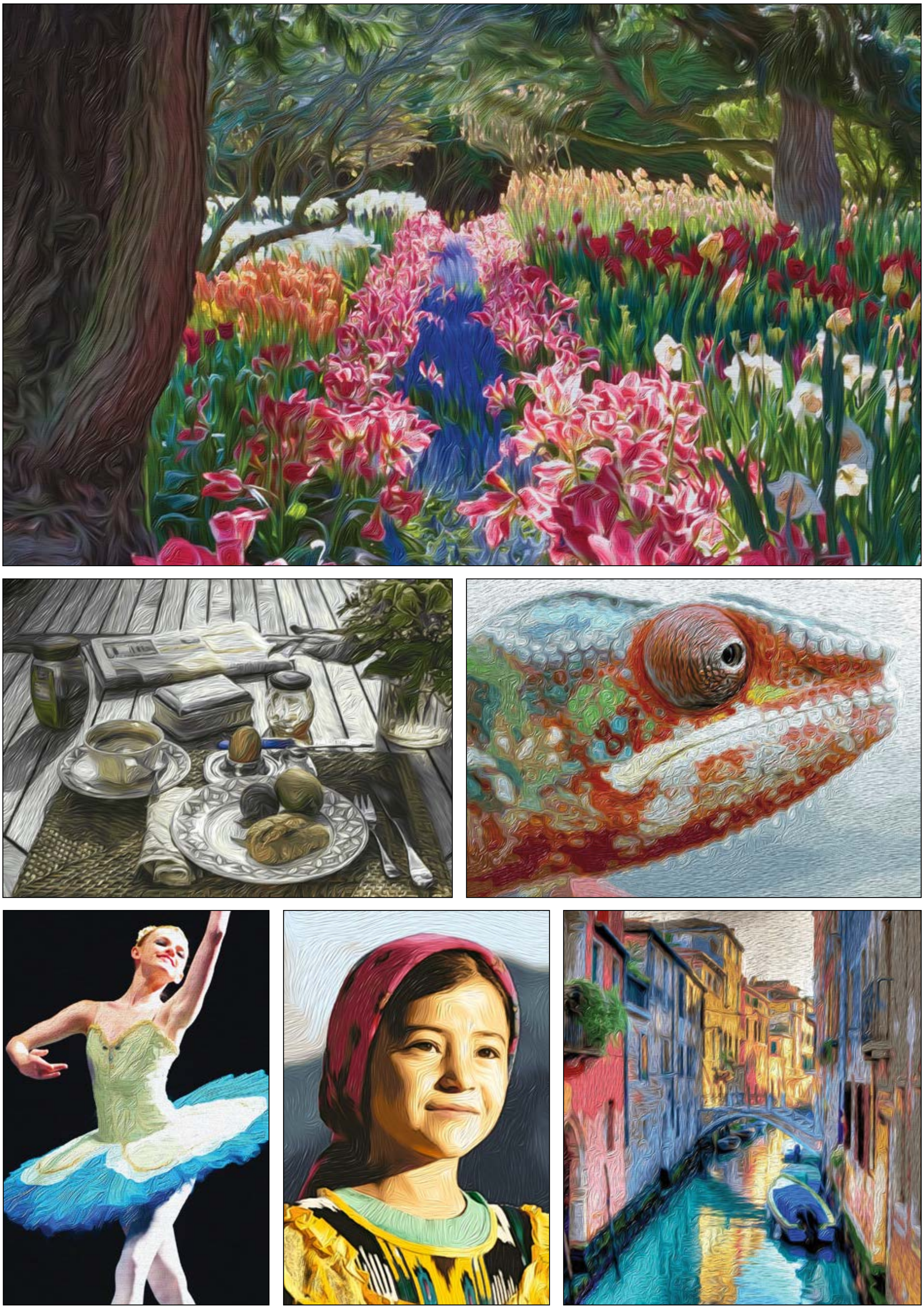


Figure VII.33.: Image stylization results produced with the proposed methods for oil paint filtering.

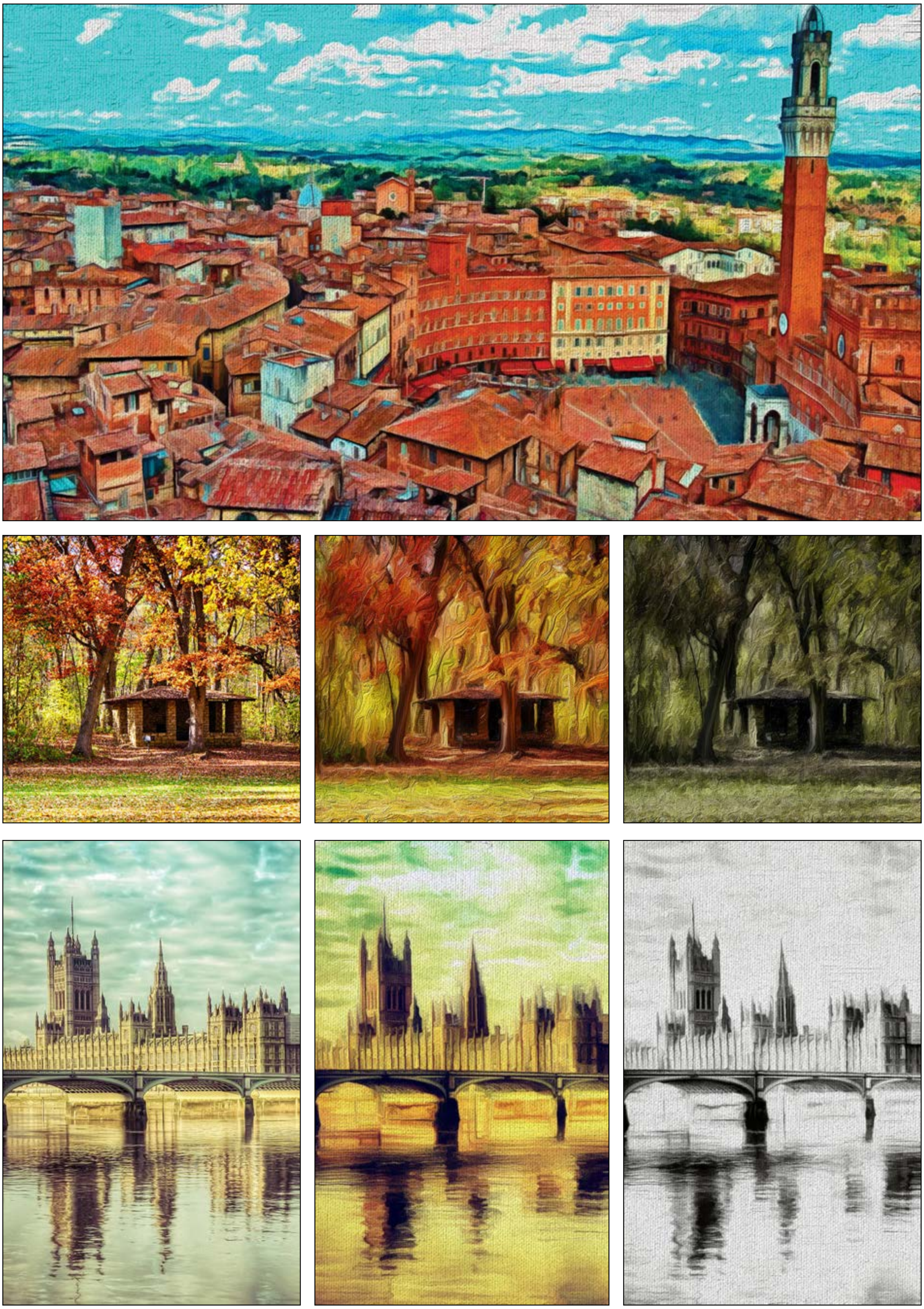


Figure VII.34.: Additional image stylization results produced with a mobile implementation of the proposed methods for iOS devices (i.e., iPhone and iPad). Here, color lookup tables were used for color grading (Selan 2004). A color quantization was not performed to maintain interactive frame rates. The mobile implementation is based on OpenGL ES 2.0 and Objective-C, and processes images up to 8 megapixel in size.

VIII

Conclusions and Future Research

How should visualization cope with today's increasingly detailed and complex 3D geospatial data to maintain effective information transfers to users? This thesis seeks answers to challenges in the interactive, context-dependent visual design of 3D geospatial information, and states that non-photorealistic rendering (NPR) provides an effective visualization category—situated between computer graphics and cartography—which serves these demands. The validity of this statement is founded in abstraction as a key concept of NPR and cartography to support visual communication. In this respect, the proposed techniques use principles of abstraction and 3D semiotics to address the drawbacks of photorealistic rendering, which were identified in Chapter I. Hence, an extended 3D semiotic model was developed that integrates cartographic design aspects into the general, interactive visualization pipeline (Chapter II) and provides a taxonomy for 3D geospatial information visualization (Chapter III). It can be used in conjunction with cartographic principles to design and implement specialized visualization techniques for geospatial feature type entities, such as water surfaces, buildings, and infrastructure networks (Chapter V). Furthermore, generalized methods were proposed for level-of-abstraction (LoA) texturing by means of image filtering and deferred shading (Chapter IV) and LoA transitions by means of image blending and geometric transformations (Chapter VI), with each using context-based and view-dependent information to selectively highlight features or regions of interest. Finally, methods for image-based abstraction and artistic rendering by means of oil paint filtering were presented for aerial and terrestrial images (Chapter VII), which constitute an integral part of 3D geospatial information visualization. From these concepts and techniques, the following conclusions are drawn to affirm the thesis statement.

Conclusions

1. **Cognitive load:** Cluttered information displays were encountered by the abstraction techniques when attempting to highlight important or prioritized information encoded in 3D scene textures. Here, image filtering and core mechanisms of the real-time rendering pipeline, such as mipmapping, provided effective means for interactive LoA texturing. In particular, evaluations indicated that the proposed methods, combined with a cartography-oriented design, could reduce visual clutter and improve figure-ground perception. Applications using these techniques effectively directed the viewer's gaze by salient stimuli attraction and eased the cognitive burden in the fields of illustrative visualization and focus+context visualization.

2. **Integration of non-realistic information:** Configurations of the proposed methods for LoA texturing, based on image-based edge enhancements, are particularly suited for extracting feature silhouettes and contour lines.

Evaluations concluded that the obtained results can be augmented with color-encoded thematic information to maintain structural information and depth cues, which are required for landmark identification and mental modeling. To this end, the approach provides essential means for combining non-realistic and appearance information modeled in texture space.

3. **Visualization of uncertainty:** Several applications of the proposed NPR techniques mimicked the appeal of artistic media to express uncertainty, e.g., using digitized hand-drawn strokes or flow-based image filtering. Similar directions were followed by Wood *et al.* (2012) in the domain of information visualization, where evaluations suggested that sketchy styles encourage user engagement with data visualization. It is supposed that the provided applications for image filtering and shading will show similar effects, while the general imprecision of 3D geospatial data can be obscured, which requires further validation, e.g., when using the proposed method for blueprint rendering in urban planning.

4. **Visualization on small displays:** The abstraction of textures on the mipmap level, coupled with view-dependent shading, opens up possibilities for reducing *visual noise* induced by 3D perspective views. Here, the proposed system for LoA transitions provides generalized approaches for context-aware information visualization. Combined with non-linear projections, such as bended maps, it is particularly suited to reducing information compression, increasing screen-space utilization, and reducing occlusion.

5. **Memory resources:** In addition to optimizing the information transfer to a user, the proposed NPR techniques may also optimize the memory transfer in service-oriented environments. For instance, the cartographic 3D rendering technique for building models is able to reduce memory consumption up to a factor of 10. Similar potentials may be expected for image compression, e.g., when processing images with the noise-reducing bilateral filter.

Another focus of this thesis was designing and implementing NPR techniques that adapt to a user's context while maintaining real-time image synthesis required for interactive visualization—characteristics that many previous techniques for 3D geospatial data have lacked. To achieve this goal, the proposed techniques integrate into hardware-accelerated rendering pipelines, utilizing texturing and shading technology of GPUs. Applications and performance evaluations have shown that this technical approach fulfills the requirements for interactivity. Moreover, the techniques and systems are designed for generic application; thus, they can be integrated into contemporary visualization systems or services and applied to other visualization domains such as medical visualization.

Future Research

There are multiple technical and conceptual aspects to consider for future research:

- **Effect parameterization:** The configuration of NPR techniques in visualization systems and services raises questions regarding how the level of abstraction can be parameterized with minimum effort to ensure optimal results for different viewing situations. The proposed focus+context and zooming interfaces are promising concepts for providing these constraints for 3D geospatial information visualization; however, they demand a generalized interaction and rendering interface that is sensitive to the environment, e.g., to ease a context-aware abstraction for in-car navigation and mobile mapping.

- **Indoor visualization:** Applications of the techniques are primarily motivated by outdoor visualization scenarios; notably, they may be extended to cover indoor mapping scenarios as well. Based on CityGML LoD-4 models or building information model standards, this may include cutaway and explosion viewing techniques. For instance, the proposed system for LoA transitions operates in a 3D space; thus, it can be used to enhance the x-ray volumetric lens effect for indoor visualization.

- **Service-oriented architecture:** The integration of NPR techniques into service-oriented architecture is of particular interest for cloud computing and the support of devices with low computing power such as mobile devices. One approach is to perform complex rendering passes that include geometry processing—in particular the synthesis of a *G-buffer*—on a server and image-based operations such as post-processing effects and image compositing on a client, e.g., based on a *G-buffer* cube map approach that decouples "3D scene complexity from data transmission complexity" (Döllner *et al.* 2012).

- **Evaluation:** Qualitative and quantitative user studies remain an important topic to assess the impact of the proposed techniques for geospatial applications. For instance, saliency maps indicate that the system for LoA texturing and transitions are feasible for drawing attention to important or prioritized information, but they require further validation to confirm significant effects in orientation, navigation, and analysis tasks performed within 3D geovirtual environments. Here, an evaluation for limited screen sizes, such as on mobile devices, is of particular interest, e.g., to measure the impact of abstraction techniques on a user's task performance when configured in a focus+context setup.

Finally, it is hoped that the presented systematization of rendering and visualization techniques via the extended 3D semiotic model may form a basis for the development of future concepts, tools, and systems that integrate aspects of cartographic design into 3D geospatial visualization.

List of Publications

The work in this manuscript appeared previously in the following publications:

Journal Articles

1. Pasewaldt, S., Semmo, A., Trapp, M. & Döllner, J. "Multi-Perspective 3D Panoramas". *International Journal of Geographical Information Science* **28,** 2030–2051 (2014).
2. Semmo, A. & Döllner, J. "Interactive Image Filtering for Level-of-Abstraction Texturing of Virtual 3D Scenes". *Computers & Graphics* **52,** 181–198 (2015).
3. Semmo, A., Hildebrandt, D., Trapp, M. & Döllner, J. "Concepts for Cartography-Oriented Visualization of Virtual 3D City Models". *Photogrammetrie - Fernerkundung - Geoinformation,* 455–465 (2012a).
4. Semmo, A., Limberger, D., Kyprianidis, J. E. & Döllner, J. "Image Stylization by Interactive Oil Paint Filtering". *Computers & Graphics* **55,** 157–171 (2016).
5. Semmo, A., Trapp, M., Jobst, M. & Döllner, J. "Cartography-Oriented Design of 3D Geospatial Information Visualization - Overview and Techniques". *The Cartographic Journal* **52,** 95–106 (2015b).
6. Semmo, A., Trapp, M., Kyprianidis, J. E. & Döllner, J. "Interactive Visualization of Generalized Virtual 3D City Models using Level-of-Abstraction Transitions". *Computer Graphics Forum* **31,** 885–894 (2012b).

Conference Papers

1. Engel, J., Semmo, A., Trapp, M. & Döllner, J. *Evaluating the Perceptual Impact of Rendering Techniques on Thematic Color Mappings in 3D Virtual Environments.* in *Proc. Vision, Modeling & Visualization* (2013), 25–32.
2. Kyprianidis, J. E., Semmo, A., Kang, H. & Döllner, J. *Anisotropic Kuwahara Filtering with Polynomial Weighting Functions.* in *Proc. EG UK Theory and Practice of Computer Graphics* (2010a), 25–30.
3. Pasewaldt, S., Semmo, A., Trapp, M. & Döllner, J. *Towards Comprehensible Digital 3D Maps.* in *Proc. Service-Oriented Mapping* (2012), 261–.276.
4. Semmo, A. & Döllner, J. *An Interaction Framework for Level-of-Abstraction Visualization of 3D Geovirtual Environments.* in *Proc. ACM SIGSPATIAL Workshop on MapInteraction* (2014a), 43–49.
5. Semmo, A. & Döllner, J. *Image Filtering for Interactive Level-of-Abstraction Visualization of 3D Scenes.* in *Proc. Computational Aesthetics* (2014b), 5–14.
6. Semmo, A., Kyprianidis, J. E. & Döllner, J. *Automated Image-Based Abstraction of Aerial Images.* in *Proc. AGILE* (2010), 359–378.

7. Semmo, A., Kyprianidis, J. E., Trapp, M. & Döllner, J. *Real-Time Rendering of Water Surfaces with Cartography-Oriented Design.* in *Proc. Computational Aesthetics* (2013), 5–14.

8. Semmo, A., Limberger, D., Kyprianidis, J. E. & Döllner, J. *Image Stylization by Oil Paint Filtering using Color Palettes.* in *Proc. Computational Aesthetics* (2015a), 149–158.

9. Semmo, A., Trapp, M. & Döllner, J. *Ansätze zur kartographischen Gestaltung von 3D-Stadtmodellen.* in *31. Wissenschaftlich-Technische Jahrestagung der DGPF* (2011), 473–482.

10. Trapp, M., Semmo, A. & Döllner, J. *Interactive Rendering and Stylization of Transportation Networks Using Distance Fields.* in *Proc. International Conference on Computer Graphics Theory and Applications* (2015), 207–219.

Posters

1. Kyprianidis, J. E., Semmo, A., Kang, H. & Döllner, J. *Anisotropic Kuwahara Filtering with Polynomial Weighting Functions.* in *NPAR Poster Session* (2010b).

2. Semmo, A. & Döllner, J. *Oil Paint Filtering Using Color Palettes For Colorization.* in *Expressive Poster Session* (2014c).

Bibliography

1. Akenine-Möller, T. & Haines, E. *Real-Time Rendering* (Taylor & Francis Ltd., 2008).
2. Albertz, J., Zimmer, A. C., Seitelberger, F., Freitag, U., Mattenklott, G., Förstner, W., Neumann, H. & Stiehl, H. S. "Wahrnehmung und Wirklichkeit - Wie wir unsere Umwelt sehen, erkennen und gestalten". *Schriftenreihe der Freien Akademie* **17** (1997).
3. Anjyo, K.-i., Wemler, S. & Baxter, W. *Tweakable Light and Shade for Cartoon Animation.* in *Proc. NPAR* (2006), 133–139.
4. Bailey, R., McNamara, A., Sudarsanam, N. & Grimm, C. "Subtle Gaze Direction". *ACM Trans. Graph.* **28,** 100:1–100:14 (2009).
5. Barla, P., Thollot, J. & Markosian, L. *X-toon: An Extended Toon Shader.* in *Proc. NPAR* (2006), 127–132.
6. Barla, P., Thollot, J. & Sillion, F. X. *Geometric Clustering for Line Drawing Simplification.* in *Proc. ACM SIGGRAPH Sketches* (2005).
7. Baxter, W. V. & Lin, M. C. *A Versatile Interactive 3D Brush Model.* in *Proc. Pacific Graphics* (2004), 319–328.
8. Baxter, W., Wendt, J. & Lin, M. C. *IMPaSTo: A Realistic, Interactive Model for Paint.* in *Proc. NPAR* (2004), 45–148.
9. Bederson, B. & Boltman, A. *Does Animation Help Users Build Mental Maps of Spatial Information?* in *Proc. IEEE InfoVis* (1999), 28–35.
10. Bell, B., Feiner, S. & Höllerer, T. *View Management for Virtual and Augmented Reality.* in *Proc. ACM UIST* (2001), 101–110.
11. Bénard, P., Bousseau, A. & Thollot, J. *Dynamic Solid Textures for Real-time Coherent Stylization.* in *Proc. ACM I3D* (2009), 121–127.
12. Bénard, P., Bousseau, A. & Thollot, J. "State-of-the-Art Report on Temporal Coherence for Stylized Animations". *Comput. Graph. Forum* **30,** 2367–2386 (2011).
13. Bénard, P., Lu, J., Cole, F., Finkelstein, A. & Thollot, J. *Active Strokes: Coherent Line Stylization for Animated 3D Models.* in *Proc. NPAR* (2012), 37–46.
14. Bénard, P., Cole, F., Kass, M., Mordatch, I., Hegarty, J., Senn, M. S., Fleischer, K., Pesare, D. & Breeden, K. "Stylizing Animation by Example". *ACM Trans. Graph.* **32,** 119:1–119:12 (2013).
15. Bertin, J. *Graphics and graphic information processing* (Walter de Gruyter, 1981).
16. Bier, E. A., Stone, M. C., Pier, K., Buxton, W. & DeRose, T. D. *Toolglass and Magic Lenses: The See-through Interface.* in *Proc. ACM SIGGRAPH* (1993), 73–80.
17. Bier, E., Stone, M. & Pier, K. "Enhanced Illustration Using Magic Lens Filters". *IEEE Computer Graphics and Applications* **17,** 62–70 (1997).

18. Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S. & Çöltekin, A. "Applications of 3D City Models: State of the Art Review". *ISPRS International Journal of Geo-Information* **4,** 2842–2889 (2015).

19. Bishop, I. D. *The Role of Visual Realism in Communicating and Understanding Spatial Change and Process.* in *Visualization in Geographical Information Systems* (Wiley, 1994), 60–64.

20. Boreskov, A. & Shikin, E. *Computer Graphics: From Pixels to Programmable Graphics Hardware* (Chapman and Hall, 2013).

21. Borji, A. & Itti, L. "State-of-the-Art in Visual Attention Modeling". *IEEE Trans Pattern Anal Mach Intell.* **35,** 185–207 (2013).

22. Bousseau, A. *Non-Linear Aperture for Stylized Depth of Field.* in *Proc. ACM SIGGRAPH Talks* (2009), 57:1–57:1.

23. Bousseau, A., Kaplan, M., Thollot, J. & Sillion, F. X. *Interactive Watercolor Rendering with Temporal Coherence and Abstraction.* in *Proc. NPAR* (2006), 141–149.

24. Brambilla, A., Carnecky, R., Peikert, R., Viola, I. & Hauser, H. *Illustrative flow visualization: State of the art, trends and challenges.* in *Proc. Eurographics State of the Art Reports* (2012).

25. Bratkova, M., Shirley, P. & Thompson, W. B. "Artistic Rendering of Mountainous Terrain". *ACM Trans. Graph.* **28,** 102:1–102:17 (2009).

26. Brewer, C. A. "Color Use Guidelines for Mapping and Visualization". *Visualization in Modern Cartography* **2,** 123–148 (1994).

27. Brodersen, L. *Paradigm Shift from Cartography to Geo-communication.* in *Proc. International Cartographic Conference* (ICA, 2007).

28. Brooks, S. "Mixed Media Painting and Portraiture". *IEEE Trans Vis Comput Graphics* **13,** 1041–1054 (2007).

29. Brox, T., Weickert, J., Burgeth, B. & Mrázek, P. "Nonlinear structure tensors". *Image and Vision Computing* **24,** 41–55 (2006a).

30. Brox, T., Boomgaard, R., Lauze, F., Weijer, J., Weickert, J., Mrázek, P. & Kornprobst, P. "Adaptive Structure Tensors and their Applications". *Visualization and Processing of Tensor Fields,* 17–47 (2006b).

31. Bruckner, S., Gröller, M. E., Mueller, K., Preim, B. & Silver, D. *Illustrative Focus+Context Approaches in Interactive Volume Visualization.* in *Scientific Visualization: Advanced Concepts* (2010). Chap. 10.

32. Bruneton, E. & Neyret, F. "Real-Time Rendering and Editing of Vector-based Terrains". *Comput. Graph. Forum* **27,** 311–320 (2008).

33. Buchin, K., Sousa, M. C., Döllner, J., Samavati, F. & Walther, M. *Illustrating Terrains using Direction of Slope and Lighting.* in *Proc. ICA Mountain Carthography Workshop* (2004), 259–269.

34. Buchroithner, M., Schenkel, R. & Kirschenbauer, S. "3D Display Techniques for Cartographic Purposes: Semiotic Aspects". *International Archives of Photogrammetry and Remote Sensing* **33,** 99–106 (2000).

35. Cabral, B. & Leedom, L. C. *Imaging Vector Fields Using Line Integral Convolution.* in *Proc. ACM SIGGRAPH* (1993), 263–270.

36. Canny, J. "A computational approach to edge detection". *IEEE Trans. Pattern Anal. Mach. Intell.* **8,** 679–698 (6 1986).

37. Cao, T.-T., Tang, K., Mohamed, A. & Tan, T.-S. *Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU.* in *Proc. ACM I3D* (2010), 83–90.

38. Cartwright, W. & Peterson, M. P. *Multimedia cartography* (Springer, 2007).

39. Chang, M.-W. & Collins, C. M. *Exploring entities in text with descriptive non-photorealistic rendering.* in *Proc. IEEE PacificVis* (2013), 9–16.

40. Chen, J., Paris, S. & Durand, F. "Real-time Edge-aware Image Processing with the Bilateral Grid". *ACM Trans. Graph.* **26,** 103:1–103:10 (2007).

41. Chen, J., Chen, Y., Granier, X., Wang, J. & Peng, Q. *Importance-Driven Composition of Multiple Rendering Styles.* in *Proc. CAD/Graphics* (2011), 79–86.

42. Chen, J., Pappas, T. N., Mojsilovic, A. & Rogowitz, B. "Adaptive Perceptual Color-Texture Image Segmentation". *IEEE Trans Image Processing* **14,** 1524–1536 (2005).

43. Cheng, M.-M., Warrell, J., Lin, W.-Y., Zheng, S., Vineet, V. & Crook, N. *Efficient Salient Region Detection with Soft Image Abstraction.* in *Proc. ICCV* (2013), 1529–1536.

44. Cho, H., Lee, H., Kang, H. & Lee, S. "Bilateral Texture Filtering". *ACM Trans. Graph.* **33,** 128:1–128:8 (2014).

45. Christensen, A. H. "A Reflection on the Waterlining Technique in Relation to the History of Map Ornamentation". *The Cartographic Journal* **45,** 68–78 (2008).

46. Christensen, A. H. "The Revival of a Victorian Art: Waterlining with a Computer". *The Cartographic Journal* **36,** 31–41 (1999).

47. Chu, N., Baxter, W., Wei, L.-Y. & Govindaraju, N. *Detail-preserving Paint Modeling for 3D Brushes.* in *Proc. NPAR* (2010), 27–34.

48. Cignoni, P., Montani, C. & Scopigno, R. *MagicSphere: an insight tool for 3D data visualization.* in *Comput. Graph. Forum* **13** (1994), 317–328.

49. Cipriano, G. & Gleicher, M. "Text Scaffolds for Effective Surface Labeling". *IEEE Trans. Vis. Comput. Graphics* **14,** 1675–1682 (2008).

50. Cockburn, A., Karlson, A. & Bederson, B. B. "A Review of Overview+Detail, Zooming, and Focus+Context Interfaces". *ACM Comput. Surv.* **41,** 2:1–2:31 (2009).

51. Coconu, L., Deussen, O. & Hege, H. *Real-Time Pen-and-Ink Illustration of Landscapes.* in *Proc. NPAR* (2006), 27–35.

52. Cole, F. & Finkelstein, A. "Two Fast Methods for High-Quality Line Visibility". *IEEE Trans Vis Comput Graphics* **16,** 707–717 (2010).

53. Cole, F., DeCarlo, D., Finkelstein, A., Kin, K., Morley, K. & Santella, A. *Directing Gaze in 3D Models with Stylized Focus.* in *Proc. EGSR* (2006), 377–387.

54. Collomosse, J. & Hall, P. *Genetic Paint: A Search for Salient Paintings.* in *Applications of Evolutionary Computing* **3449** (Springer Berlin Heidelberg, 2005), 437–447.

55. Comaniciu, D., Meer, P. & Member, S. "Mean Shift: A Robust Approach Toward Feature Space Analysis". *IEEE Trans. Pattern Anal. Mach. Intell.* **24,** 603–619 (2002).

56. Cong, L., Tong, R. & Dong, J. "Selective Image Abstraction". *Vis. Comput.* **27,** 187–198 (2011).

57. Criminisi, A., Sharp, T., Rother, C. & Pérez, P. "Geodesic Image and Video Editing". *ACM Trans. Graph.* **29,** 134:1–134:15 (2010).

58. Curtis, C. J., Anderson, S. E., Seims, J. E., Fleischer, K. W. & Salesin, D. H. *Computer-Generated Watercolor.* in *Proc. ACM SIGGRAPH* (1997), 421–430.

59. Cutting, J. E. & Vishton, P. M. *Perceiving layout and knowing distances: The interaction, potency, and contextual use of different information about depth.* in *Perception of Space and Motion* (San Diego: Academic Press, 1995), 69–117.

60. Danielsson, P.-E. "Euclidean Distance Mapping". *Computer Graphics and Image Processing* **14,** 227–248 (1980).

61. Darles, E., Crespin, B., Ghazanfarpour, D. & Gonzato, J. "A Survey of Ocean Simulation and Rendering Techniques in Computer Graphics". *Comput. Graph. Forum* **30,** 43–60 (2011).

62. DeCarlo, D., Finkelstein, A., Rusinkiewicz, S. & Santella, A. "Suggestive Contours for Conveying Shape". *ACM Trans. Graph.* **22,** 848–855 (2003).

63. DeCarlo, D. & Santella, A. "Stylization and Abstraction of Photographs". *ACM Trans. Graph.* **21,** 769–776 (2002).

64. Decaudin, P. *Cartoon Looking Rendering of 3D Scenes.* Research Report 2919 (INRIA, 1996).

65. DeCoro, C., Cole, F., Finkelstein, A. & Rusinkiewicz, S. *Stylized Shadows.* in *Proc. NPAR* (2007), 77–83.

66. Degener, P. & Klein, R. "A Variational Approach for Automatic Generation of Panoramic Maps". *ACM Trans. Graph.* **28,** 2:1–2:14 (2009).

67. Deng, H., Zhang, L., Mao, X. & Qu, H. "Interactive Urban Context-Aware Visualization via Multiple Disocclusion Operators". *IEEE Trans Vis Comput Graphics* **PP** (2015).

68. Deng, H., Zhang, L., Han, C., Ren, Y., Zhang, L. & Li, J. "Efficient occlusion-free visualization for navigation in mountainous areas". *Computers & Geosciences* **52,** 389–397 (2013).

69. Deriche, R. *Recursively implementating the Gaussian and its derivatives.* in *Proc. Int. Conf. Image Processing* (1993), 263–267.

70. Deussen, O. & Isenberg, T. *Halftoning and Stippling.* in *Image and Video-Based Artistic Stylisation* (Springer London, 2013), 45–61.

71. Deussen, O. & Strothotte, T. *Computer-Generated Pen-and-Ink Illustration of Trees.* in *Proc. ACM SIGGRAPH* (2000), 13–18.

72. DiVerdi, S. *A Brush Stroke Synthesis Toolbox.* in *Image and Video-Based Artistic Stylisation* (Springer, 2013), 23–44.

73. Döllner, J., Baumann, K. & Buchholz, H. *Virtual 3D City Models as Foundation of Complex Urban Information Spaces.* in *Proc. International Conference on Urban Planning and Spatial Development in the Information Society* (2006), 107–112.

74. Döllner, J. & Buchholz, H. *Expressive Virtual 3D City Models.* in *Proc. International Cartographic Conference* (2005a).

75. Döllner, J. & Buchholz, H. *Non-Photorealism in 3D Geovirtual Environments.* in *Proc. AutoCarto* (2005b), 1–14.

76. Döllner, J., Hagedorn, B. & Klimke, J. *Server-based Rendering of Large 3D Scenes for Mobile Devices Using G-buffer Cube Maps.* in *Proc. Web3D* (2012), 97–100.

77. Döllner, J. & Kyprianidis, J. E. *Approaches to Image Abstraction for Photorealistic Depictions of Virtual 3D Models.* in *Cartography in Central and Eastern Europe* (Springer, 2010), 263–277.

78. Döllner, J. & Walther, M. *Real-Time Expressive Rendering of City Models.* in *Proc. IEEE IV* (2003), 245–250.

79. Donnelly, W. & Lauritzen, A. *Variance Shadow Maps.* in *Proc. ACM I3D* (2006), 161–165.

80. Dowson, K. *Towards Extracting Artistic Sketches and Maps from Digital Elevation Models.* PhD thesis (University of Hull, 1994).

81. Durand, F. *An Invitation to Discuss Computer Depiction.* in *Proc. NPAR* (2002), 111–124.

82. Dykes, J., MacEachren, A. & Kraak, M. *Exploring Geovisualization* (Pergamon, 2005).

83. Dykes, J. A., Moore, K. E. & Fairbairn, D. *From Chernoff to Imhof and beyond: VRML and cartography.* in *Proc. VRML* (1999), 99–104.

84. Earls, I. *Renaissance Art: A Topical Dictionary* (ABC-CLIO, 1987).

85. Eden, A. M., Bargteil, A. W., Goktekin, T. G., Eisinger, S. B. & O'Brien, J. F. *A Method for Cartoon-Style Rendering of Liquid Animations.* in *Proc. ACM Graphics Interface* (2007), 51–55.

86. Eisemann, E., Schwarz, M., Assarsson, U. & Wimmer, M. *Real-Time Shadows* (CRC Press, 2011).

87. Elias, B., Paelke, V. & Kuhnt, S. *Concepts for the Cartographic Visualization of Landmarks.* in *Proc. LBS and Telecartography* (2005), 149–155.

88. Ellis, G., Bertini, E. & Dix, A. *The Sampling Lens: Making Sense of Saturated Visualisations.* in *Proc. ACM CHI* (2005), 1351–1354.

89. Ellis, G. & Dix, A. "A Taxonomy of Clutter Reduction for Information Visualisation". *IEEE Trans. Vis. Comput. Graphics* **13,** 1216–1223 (2007).

90. Ellis, S. R. "What are virtual environments?" *IEEE Computer Graphics and Applications* **14,** 17–22 (1994).

91. Elmqvist, N., Dragicevic, P. & Fekete, J.-D. "Color Lens: Adaptive Color Scale Optimization for Visual Exploration". *IEEE Trans Vis Comput Graphics* **17,** 795–807 (2011).

92. Engel, J. & Döllner, J. "Effiziente Verschattungsberechnung für die Solarpotenzialanalyse durch bildbasierte 3D-Analyse". *Proceedings of the GeoInformatik* (2010).

93. Ewins, J., Waller, M., White, M. & Lister, P. "MIP-Map Level Selection for Texture Mapping". *IEEE Trans Vis Comput Graphics* **4,** 317–329 (1998).

94. Fabrikant, S. I., Hespanha, S. R. & Hegarty, M. "Cognitively Inspired and Perceptually Salient Graphic Displays for Efficient Spatial Inference Making". *Annals of the Association of American Geographers* **100,** 13–29 (2010).

95. Farbman, Z., Fattal, R., Lischinski, D. & Szeliski, R. "Edge-Preserving Decompositions for Multi-Scale Tone and Detail Manipulation". *ACM Trans. Graph.* **27,** 67:1–67:10 (2008).

96. Fattal, R., Agrawala, M. & Rusinkiewicz, S. "Multiscale Shape and Detail Enhancement from Multi-light Image Collections". *ACM Trans. Graph.* **26,** 51:1–51:9 (2007).

97. Foerster, T., Stoter, J. E. & Kobben, B. *Towards a formal classification of Generalization operators.* in *Proc. International Cartographic Conference* (2007).

98. French, T. *A manual of engineering drawing for students and draftsmen* (McGraw-Hill book company, 1918).

99. Freudenberg, B. *Real-time stroke textures.* in *Proc. ACM SIGGRAPH Technical Sketches* (2001).

100. Frisken, S. F., Perry, R. N., Rockwood, A. P. & Jones, T. R. *Adaptively Sampled Distance Fields: A General Representation of Shape for Computer Graphics.* in *Proc. ACM SIGGRAPH* (2000), 249–254.

101. Fuhrmann, A. & Gröller, E. *Real-Time Techniques for 3D Flow Visualization.* in *Proc. IEEE Visualization* (1998), 305–312.

102. Furnas, G. W. *Generalized Fisheye Views.* in *Proc. ACM CHI* (1986), 16–23.

103. Garlandini, S. & Fabrikant, S. I. *Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization.* in *Spatial Information Theory* (Springer, 2009), 195–211.

104. Gastal, E. S. L. & Oliveira, M. M. "Domain Transform for Edge-Aware Image and Video Processing". *ACM Trans. Graph.* **30,** 69:1–69:12 (2011).

105. Gasteiger, R., Neugebauer, M., Beuing, O. & Preim, B. "The FLOWLENS: A Focus-and-Context Visualization Approach for Exploration of Blood Flow in Cerebral Aneurysms". *IEEE Trans Vis Comput Graphics* **17,** 2183–2192 (2011).

106. Gatys, L. A., Ecker, A. S. & Bethge, M. "A Neural Algorithm of Artistic Style". *arXiv:1508.06576* (2015).

107. Geisthövel, R. & Hurni, L. *Automatic Rock Depiction via Relief Shading.* in *Proc. International Cartographic Conference* (2015).

108. Gerl, M. & Isenberg, T. "Interactive Example-based Hatching". *Computers & Graphics* **37,** 65–80 (2013).

109. Gershon, N. *From Perception to Visualization,* 129–139 (Academic Press, New York, 1994).

110. Gervautz, M. & Purgathofer, W. *A Simple Method for Color Quantization: Octree Quantization.* in *New Trends in Computer Graphics* (Springer Berlin Heidelberg, 1988), 219–231.

111. Gibson, J. J. *The Ecological Approach to Visual Perception* (Routledge, 1986).

112. Giegl, M. & Wimmer, M. "Unpopping: Solving the Image-Space Blend Problem for Smooth Discrete LOD Transitions". *Comput. Graph. Forum* **26,** 46–49 (2007).

113. Giller, V., Tscheligi, M., Schrammel, J., Fröhlich, P., Rabl, B., Kosara, R., Miksch, S. & Hauser, H. *Experimental Evaluation of Semantic Depth of Field, a Preattentive Method for Focus+Context Visualization.* in *Proc. INTERACT* **3** (2001), 888–891.

114. Girshick, A., Interrante, V., Haker, S. & Lemoine, T. *Line Direction Matters: An Argument for the Use of Principal Directions in 3D Line Drawings.* in *Proc. NPAR* (2000), 43–52.

115. Glander, T. & Döllner, J. "Abstract Representations for Interactive Visualization of Virtual 3D City Models". *Computers, Environment and Urban Systems* **33,** 375–387 (2009).

116. Glander, T., Trapp, M. & Döllner, J. *3D Isocontours – Real-time Generation and Visualization of 3D Stepped Terrain Models.* in *Proc. Eurographics Shortpaper* (2010), 17–20.

117. Glander, T., Trapp, M. & Döllner, J. *A Concept of Effective Landmark Depiction in Geovirtual 3D Environments by View-Dependent Deformation.* in *Proc. International Symposium on LBS and Telecartography* (2007).

118. Glanville, R. S. *Texture Bombing.* in *GPU Gems* (Addison-Wesley, 2004), 323–338.

119. Goldstein, E. B. *Sensation and Perception* (Wadsworth Publishing Company, 2010).

120. Gooch, A. A., Long, J., Ji, L., Estey, A. & Gooch, B. S. *Viewing Progress in Non-photorealistic Rendering through Heinlein's Lens.* in *Proc. NPAR* (2010), 165–171.

121. Gooch, A., Gooch, B., Shirley, P. & Cohen, E. *A Non-photorealistic Lighting Model for Automatic Technical Illustration.* in *Proc. ACM SIGGRAPH* (1998), 447–452.

122. Gooch, B., Coombe, G. & Shirley, P. *Artistic Vision: Painterly Rendering Using Computer Vision Techniques.* in *Proc. NPAR* (2002), 83–ff.

123. Gooch, B. & Gooch, A. *Non-Photorealistic Rendering* (AK Peters Ltd, 2001).

124. Gooch, B., Reinhard, E. & Gooch, A. "Human Facial Illustrations: Creation and Psychophysical Evaluation". *ACM Trans. Graph.* **23,** 27–44 (2004).

125. Goodwin, T., Vollick, I. & Hertzmann, A. *Isophote Distance: A Shading Approach to Artistic Stroke Thickness.* in *Proc. NPAR* (2007), 53–62.

126. Götzelmann, T., Ali, K., Hartmann, K. & Strothotte, T. *Form Follows Function: Aesthetic Interactive Labels.* in *Proc. CAe* (2005), 193–200.

127. Grabler, F., Agrawala, M., Sumner, R. W. & Pauly, M. *Automatic Generation of Tourist Maps.* in *Proc. ACM SIGGRAPH* (2008), 100:1–100:11.

128. Grabli, S., Turquin, E., Durand, F. & Sillion, F. X. "Programmable Rendering of Line Drawing from 3D Scenes". *ACM Trans. Graph.* **29,** 18:1–18:20 (2010).

129. Grabli, S., Turquin, E., Durand, F. & Sillion, F. X. *Programmable Style for NPR Line Drawing.* in *Proc. EGSR* (2004), 33–44.

130. Green, C. *Improved Alpha-tested Magnification for Vector Textures and Special Effects.* in *Proc. ACM SIGGRAPH Courses* (2007), 9–18.

131. Häberling, C. *Symbolization in topographic 3D maps: Conceptual aspects for user-oriented design.* in *Proc. International Cartographic Conference* (1999), 1037–1044.

132. Häberling, C., Bär, H. & Hurni, L. "Proposed Cartographic Design Principles for 3D Maps: A Contribution to an Extended Cartographic Theory". *Cartographica* **43,** 175–188 (2008).

133. Häberling, C. & Hurni, L. "Mountain cartography: revival of a classic domain". *ISPRS Journal of Photogrammetry and Remote Sensing* **57,** 134–158 (2002).

134. Haeberli, P. "Paint by Numbers: Abstract Image Representations". *Computer Graphics* **24,** 207–214 (1990).

135. Haeberli, P. & Segal, M. *Texture Mapping as a Fundamental Drawing Primitive.* in *Proc. Eurographics Workshop on Rendering* (1993), 259–266.

136. Hake, G., Grünreich, D. & Meng, L. *Kartographie: Visualisierung raumzeitlicher Informationen* (DeGruyter, 2002).

137. Haklay, M. & Weber, P. "OpenStreetMap: User-Generated Street Maps". *IEEE Pervasive Computing* **7,** 12–18 (2008).

138. Halper, N., Mellin, M., Herrmann, C., Linneweber, V. & Strothotte, T. *Psychology and Non-Photorealistic Rendering: The Beginning of a Beautiful Relationship.* in *Mensch & Computer 2003* **57** (Vieweg+Teubner Verlag, 2003), 277–286.

139. Hanrahan, P. & Haeberli, P. "Direct WYSIWYG Painting and Texturing on 3D Shapes". *Computer Graphics* **24,** 215–223 (1990).

140. Harel, J., Koch, C. & Perona, P. "Graph-Based Visual Saliency". *Advances in Neural Information Processing Systems* **19,** 545–552 (2007).

141. Hays, J. & Essa, I. *Image and Video Based Painterly Animation.* in *Proc. NPAR* (2004), 113–120.

142. He, K., Sun, J. & Tang, X. "Guided Image Filtering". *IEEE Trans Pattern Anal Mach Intell.* **35,** 1397–1409 (2013).

143. Heckbert, P. "Color Image Quantization for Frame Buffer Display". *SIGGRAPH Comput. Graph.* **16,** 297–307 (1982).

144. Heer, J. & Robertson, G. "Animated Transitions in Statistical Data Graphics". *Proc. IEEE InfoVis* **13,** 1240–1247 (6 2007).

145. Hegde, S., Gatzidis, C. & Tian, F. "Painterly rendering techniques: a state-of-the-art review of current approaches". *Comp. Anim. Virtual Worlds* **24,** 43–64 (2013).

146. Hermosilla, P. & Vázquez, P. *Single Pass GPU Stylized Edges.* in *Proc. Iberoamerican Symposium in Computer Graphics* (2009).

147. Hertzmann, A. "A Survey of Stroke-Based Rendering." *IEEE Computer Graphics and Applications,* 70–81 (2003).

148. Hertzmann, A. *Fast Paint Texture.* in *Proc. NPAR* (2002), 91–96.

149. Hertzmann, A. *Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines.* in *Proc. ACM SIGGRAPH Courses* (1999), 21–26.

150. Hertzmann, A. *Non-Photorealistic Rendering and the Science of Art.* in *Proc. NPAR* (2010), 147–157.

151. Hertzmann, A. *Painterly Rendering with Curved Brush Strokes of Multiple Sizes.* in *Proc. ACM SIGGRAPH* (1998), 453–460.

152. Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B. & Salesin, D. H. *Image Analogies.* in *Proc. ACM SIGGRAPH* (2001), 327–340.

153. Hertzmann, A. & Zorin, D. *Illustrating Smooth Surfaces.* in *Proc. ACM SIGGRAPH* (2000), 517–526.

154. Hirono, D., Wu, H.-Y., Arikawa, M. & Takahashi, S. *Constrained Optimization for Disoccluding Geographic Landmarks in 3D Urban Maps.* in *Proc. IEEE PacificVis* (2013), 17–24.

155. Hoppe, H. "Smooth View-Dependent Level-of-Detail Control and Its Application to Terrain Rendering". *Proc. IEEE Vis,* 35 (1998).

156. Howard, I. P. & Rogers, B. J. *Perceiving in Depth, Volume 3: Other Mechanisms of Depth Perception.* **29** (Oxford University Press, 2012).

157. Hu, J., You, S. & Neumann, U. "Approaches to Large-Scale Urban Modeling". *IEEE Computer Graphics and Applications* **23,** 62–69 (2003).

158. Huffman, D. P. "On Waterlines: Arguments for their Employment, Advice on their Generation". *Cartographic Perspectives* **66,** 23–30 (2010).

159. Hughes, J. F., Van Dam, A., Foley, J. D. & Feiner, S. K. *Computer Graphics: Principles and Practice* (Pearson Education, 2013).

160. Hurni, L., Dahinden, T. & Hutzler, E. "Digital Cliff Drawing for Topographic Maps: Traditional Representations by Means of New Technologies". *Cartographica* **38,** 55–65 (2001).

161. Hurtut, T., Landes, P.-E., Thollot, J., Gousseau, Y., Drouillhet, R. & Coeurjolly, J.-F. *Appearance-guided Synthesis of Element Arrangements by Example.* in *Proc. NPAR* (2009), 51–60.

162. Imhof, E. *Cartographic relief presentation* (Walter de Gruyter, Berlin, 1982).

163. Imhof, E. *Thematische Kartographie* (Walter de Gruyter, 1972).

164. Imhof, E. "Positioning Names on Maps". *The American Cartographer* **2,** 128–144 (1975).

165. Interrante, V., Fuchs, H. & Pizer, S. *Enhancing Transparent Skin Surfaces with Ridge and Valley Lines.* in *Proc. IEEE Visualization* (1995), 52–59.

166. Isenberg, T., Freudenberg, B., Halper, N., Schlechtweg, S. & Strothotte, T. "A Developer's Guide to Silhouette Algorithms for Polygonal Models". *IEEE Computer Graphics and Applications* **23,** 28–37 (2003).

167. Isenberg, T., Masuch, M. & Strothotte, T. *3D Illustrative Effects for Animating Line Drawings.* in *Proc. IEEE IV* (2000), 413–418.

168. Isenberg, T. "Visual Abstraction and Stylisation of Maps". *The Cartographic Journal* **50,** 8–18 (2013).

169. Isenberg, T., Halper, N. & Strothotte, T. *Stylizing Silhouettes at Interactive Rates: From Silhouette Edges to Silhouette Strokes.* in *Comput. Graph. Forum* **21** (2002), 249–258.

170. Itti, L., Koch, C. & Niebur, E. "A Model of Saliency-based Visual Attention for Rapid Scene Analysis". *IEEE Trans. Pattern Anal. Mach. Intell.* 1254–1259 (1998).

171. Jahnke, M., Meng, L., Kyprianidis, J. & Döllner, J. "Nonphotorealistic Rendering on Mobile Devices and its Usability Concerns". *Virtual Geographic Environment* (2008).

172. Jahnke, M., Berger, T., Donaubauer, A. & Krisp, J. "Nicht fotorealistische Darstellung von 3D-Stadtmodellen". *HMD Praxis der Wirtschaftsinformatik* **48,** 101–112 (2011).

173. Jahnke, M., Krisp, J. M. & Kumke, H. *Typification of Facade Elements for Virtual Three-Dimensional City Models.* in *Proc. International Cartographic Conference* (2013).

174. Jankowski, J. & Hachet, M. "Advances in Interaction with 3D Environments". *Comput. Graph. Forum* **34,** 152–190 (2014).

175. Jenny, B. "An Interactive Approach to Analytical Relief Shading". *Cartographica* **38,** 67–75 (2001).

176. Jenny, B., Hutzler, E. & Hurni, L. "Point Pattern Synthesis". *The Cartographic Journal* **47,** 257–261 (2010a).

177. Jenny, B., Hutzler, E. & Hurni, L. "Scree Representation on Topographic Maps". *The Cartographic Journal* **47,** 141–149 (2010b).

178. Jenny, H. & Jenny, B. "Challenges in Adapting Example-Based Texture Synthesis for Panoramic Map Creation: A Case Study". *CaGIS* **40,** 297–304 (2013).

179. Jenny, H., Jenny, B., Cartwright, W. E. & Hurni, L. "Interactive Local Terrain Deformation Inspired by Hand-painted Panoramas". *The Cartographic Journal* **48,** 11–20 (2011).

180. Jenny, H., Jenny, B. & Hurni, L. "Interactive Design of 3D Maps with Progressive Projection". *The Cartographic Journal* **47,** 211–221 (2010).

181. Jobst, M. *Ein semiotisches Modell für die kartographische Kommunikation mit 3D.* PhD thesis (Vienna University of Technology, Insitute for Geoninformation and Cartography 127/2, 2008).

182. Jobst, M. & Döllner, J. *3D City Model Visualization with Cartography-Oriented Design.* in *Proc. REAL CORP* (2008a), 507–516.

183. Jobst, M. & Döllner, J. *Better Perception of 3D-Spatial Relations by Viewport Variations.* in *Proc. International Conference on Visual Information Systems* (2008b), 7–18.

184. Jobst, M., Kyprianidis, J. E. & Döllner, J. *Mechanisms on Graphical Core Variables in the Design of Cartographic 3D City Presentations.* in *Geospatial Vision* (Springer, 2008), 45–59.

185. Judd, T., Durand, F. & Adelson, E. "Apparent Ridges for Line Drawing". *ACM Trans. Graph.* **26** (2007).

186. Kada, M., Wichmann, A. & Hermes, T. "Smooth transformations between generalized 3D building models for visualization purposes". *CaGIS* **42,** 306–314 (2015).

187. Kagaya, M., Brendel, W., Deng, Q., Kesterson, T., Todorovic, S., Neill, P. & Zhang, E. "Video Painting with Space-Time-Varying Style Parameters". *IEEE Trans. Vis. Comput. Graphics* **17,** 74–87 (2011).

188. Kalogerakis, E., Nowrouzezahrai, D., Breslav, S. & Hertzmann, A. "Learning hatching for pen-and-ink illustration of surfaces". *ACM Trans. Graph.* **31,** 1:1–1:17 (2012).

189. Kang, D., Chung, J.-M., Seo, S.-H., Choi, J.-S. & Yoon, K.-H. *Detail-Adaptive Toon Shading Using Saliency.* in *Proc. IEEE VIZ* (2009), 16–20.

190. Kang, H. & Lee, S. "Shape-simplifying Image Abstraction". *Comput. Graph. Forum* **27,** 1773–1780 (2008).

191. Kang, H., Lee, S. & Chui, C. K. *Coherent Line Drawing.* in *Proc. NPAR* (2007), 43–50.

192. Kang, H., Lee, S. & Chui, C. K. "Flow-Based Image Abstraction". *IEEE Trans. Vis. Comput. Graphics* **15,** 62–76 (2009).

193. Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R. & Wu, A. Y. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation". *IEEE Trans. Pattern. Anal. Mach. Intell.* **24,** 881–892 (2002).

194. Karacan, L., Erdem, E. & Erdem, A. "Structure-preserving Image Smoothing via Region Covariances". *ACM Trans. Graph.* **32,** 176:1–176:11 (2013).

195. Karnick, P., Cline, D., Jeschke, S., Razdan, A. & Wonka, P. "Route Visualization Using Detail Lenses". *IEEE Trans. Vis. Comput. Graphics* **16,** 235–247 (2010).

196. Kass, M. & Solomon, J. "Smoothed Local Histogram Filters". *ACM Trans. Graph.* **29,** 100:1–100:10 (2010).

197. Kennelly, P. "Cross-Hatched Shadow Line Maps". *The Cartographic Journal* **49,** 135–142 (2012).

198. Kennelly, P. J. *Hill-shading Techniques to Enhance Terrain Maps.* in *Proc. International Cartographic Conference* (2009), 15–21.

199. Kennelly, P. J. & Kimerling, A. J. "Desktop Hachure Maps from Digital Elevation Models". *Cartographic Perspectives,* 78–81 (2000).

200. Kennelly, P. J. & Kimerling, A. J. "Non-Photorealistic Rendering and Terrain Representation". *Cartographic Perspectives,* 35–54 (2006).

201. Kennelly, P. & Kimerling, A. J. "Modifications of Tanaka's Illuminated Contour Method". *CaGIS* **28,** 111–123 (2001).

202. Kersting, O. & Döllner, J. *Interactive 3D Visualization of Vector Data in GIS.* in *Proc. ACM GIS* (2002), 107–112.

203. Kim, D., Son, M., Lee, Y., Kang, H. & Lee, S. "Feature-guided Image Stippling". *Comput. Graph. Forum* **27,** 1209–1216 (2008).

204. Kim, S., Woo, I., Maciejewski, R. & Ebert, D. *Automated Hedcut Illustration Using Isophotes.* in *Proc. Smart Graphics* (2010), 172–183.

205. Kim, T.-h., Ahn, J. & Choi, M. G. "Image Dequantization: Restoration of Quantized Colors". *Comput. Graph. Forum* **26,** 619–626 (2007).

206. Kim, Y., Yu, J., Yu, X. & Lee, S. "Line-art illustration of dynamic and specular surfaces". *ACM Trans. Graph.* **27,** 156:1–156:10 (2008).

207. Klein, A. W., Li, W., Kazhdan, M. M., Corrêa, W. T., Finkelstein, A. & Funkhouser, T. A. *Non-photorealistic Virtual Environments.* in *Proc. ACM SIGGRAPH* (2000), 527–534.

208. Knödel, S., Hachet, M. & Guitton, P. *Interactive Generation and Modification of Cutaway Illustrations for Polygonal Models.* in *Proc. Smart Graphics* (2009), 140–151.

209. Kol, T. R., Liao, J. & Eisemann, E. *Real-time Canonical-angle Views in 3D Virtual Cities.* in *Vision, Modeling & Visualization* (2014), 55–62.

210. Kolacny, A. "Cartographic Information – a Fundamental Concept and Term in Modern Cartography". *The Cartographic Journal* **6,** 47–49 (1969).

211. Kolbe, T. H. *Representing and Exchanging 3D City Models with CityGML.* in *Proc. Int. Workshop on 3D Geo-Information* (2009), 20.

212. Kolomenkin, M., Shimshoni, I. & Tal, A. "Demarcating Curves for Shape Illustration". *ACM Trans. Graph.* **27,** 157:1–157:9 (2008).

213. Kopf, J., Agrawala, M., Bargeron, D., Salesin, D. & Cohen, M. "Automatic Generation of Gestination Maps". *ACM Trans. Graph.* **29,** 158:1–158:12 (2010).

214. Kopf, J., Cohen-Or, D., Deussen, O. & Lischinski, D. "Recursive Wang Tiles for Real-time Blue Noise". *ACM Trans. Graph.* **25,** 509–518 (2006).

215. Kosara, R., Miksch, S. & Hauser, H. "Focus+Context Taken Literally". *IEEE Computer Graphics and Applications* **22,** 22–29 (2002).

216. Kosara, R., Miksch, S. & Hauser, H. *Semantic Depth of Field.* in *Proc. IEEE InfoVis* (2001), 97–104.

217. Kraak, M. *Computer-assisted cartographical 3D imaging techniques* (Taylor & Francis, London, 1989).

218. Kraak, M. & Ormeling, F. *Cartography: Visualization of Geospatial Data* (Pearson Education, 2003).

219. Kröger, J., Schiewe, J. & Weninger, B. *Analysis and Improvement of the OpenStreetMap Street Color Scheme for Users with Color Vision Deficiencies.* in *Proc. International Cartographic Conference* (2013).

220. Krüger, R., Thom, D., Wörner, M., Bosch, H. & Ertl, T. *TrajectoryLenses – A Set-based Filtering and Exploration Technique for Long-term Trajectory Data.* in *Comput. Graph. Forum* **32** (2013), 451–460.

221. Kuwahara, M., Hachimura, K., Eiho, S. & Kinoshita, M. *Processing of RI-angiocardiographic images.* in *Digital processing of biomedical images* (Plenum Press, 1976).

222. Kyprianidis, J. E. *Image and Video Abstraction by Multi-scale Anisotropic Kuwahara Filtering.* in *Proc. NPAR* (2011), 55–64.

223. Kyprianidis, J. E., Collomosse, J., Wang, T. & Isenberg, T. "State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video". *IEEE Trans. Vis. Comput. Graphics* **19,** 866–885 (2013).

224. Kyprianidis, J. E. & Döllner, J. *Image Abstraction by Structure Adaptive Filtering.* in *Proc. EG UK TPCG* (2008), 51–58.

225. Kyprianidis, J. E. & Kang, H. "Image and Video Abstraction by Coherence-Enhancing Filtering". *Comput. Graph. Forum* **30,** 593–602 (2011).

226. Lake, A., Marshall, C., Harris, M. & Blackstein, M. *Stylized Rendering Techniques for Scalable Real-time 3D Animation.* in *Proc. NPAR* (2000), 13–20.

227. Lawonn, K., Glaber, S., Vilanova, A., Preim, B. & Isenberg, T. "Occlusion-free Blood Flow Animation with Wall Thickness Visualization". *IEEE Trans Vis Comput Graphics* **22,** 728–737 (2015).

228. Lawonn, K., Mönch, T. & Preim, B. *Streamlines for Illustrative Real-Time Rendering.* in *Comput. Graph. Forum* **32** (2013), 321–330.

229. Lee, A. W. F., Dobkin, D., Sweldens, W. & Schröder, P. *Multiresolution Mesh Morphing.* in *Proc. ACM SIGGRAPH* (1999), 343–350.

230. Lee, H.-C. "Review of image-blur models in a photographic system using the principles of optics". *Optical Engineering* **29,** 405–421 (1990).

231. Lee, Y., Markosian, L., Lee, S. & Hughes, J. F. "Line Drawings via Abstracted Shading". *ACM Trans. Graph.* **26** (2007).

232. Leonowicz, A. M., Jenny, B. & Hurni, L. "Terrain Sculptor: Generalizing Terrain Models for Relief Shading". *Cartographic Perspectives,* 51–60 (2010).

233. Lesage, P.-L. & Visvalingam, M. "Towards Sketch-Based Exploration of Terrain". *Computers & Graphics* **26,** 309–328 (2002).

234. Leung, Y. K. & Apperley, M. D. "A Review and Taxonomy of Distortion-Oriented Presentation Techniques". *ACM Transactions on Computer-Human Interaction* **1,** 126–160 (1994).

235. Levin, A., Lischinski, D. & Weiss, Y. "Colorization Using Optimization". *ACM Trans. Graph.* **23,** 689–694 (2004).

236. Li, J. & Mould, D. *Image Warping for a Painterly Effect.* in *Proc. CAe* (2015), 131–140.

237. Li, W., Ritter, L., Agrawala, M., Curless, B. & Salesin, D. *Interactive Cutaway Illustrations of Complex 3D Models.* in *ACM Trans. Graph.* **26** (2007).

238. Lillesand, T., Kiefer, R. W. & Chipman, J. *Remote sensing and image interpretation* (John Wiley & Sons, 2014).

239. Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L. F., Faust, N. & Turner, G. A. *Real-time, Continuous Level of Detail Rendering of Height Fields.* in *Proc. ACM SIGGRAPH* (1996), 109–118.

240. Lorenz, H., Trapp, M., Jobst, M. & Döllner, J. *Interactive Multi-Perspective Views of Virtual 3D Landscape and City Models.* in *Proc. AGILE* (2008), 301–321.

241. Loya, A., Adabala, N., Das, A. & Mishra, P. *A Practical Approach to Image-Guided Building Facade Abstraction.* in *Proc. Computer Graphics International* (2008).

242. Lu, J., Barnes, C., DiVerdi, S. & Finkelstein, A. "RealBrush: Painting with Examples of Physical Media". *ACM Trans. Graph.* **32,** 117:1–117:12 (2013).

243. Lu, J., Sander, P. V. & Finkelstein, A. *Interactive Painterly Stylization of Images, Videos and 3D Animations.* in *Proc. ACM I3D* (2010), 127–134.

244. Luft, T., Colditz, C. & Deussen, O. "Image Enhancement by Unsharp Masking the Depth Buffer". *ACM Trans. Graph.* **25,** 1206–1213 (2006).

245. Luft, T. & Deussen, O. *Real-time Watercolor Illustrations of Plants Using a Blurred Depth Test.* in *Proc. NPAR* (2006), 11–20.

246. Lynch, K. *The Image of the City* (MIT Press, 1960).

247. Ma, K.-L., Hertzmann, A., Interrante, V. & Lum, E. B. *Recent Advances in Non-Photorealistic Rendering for Art and Visualization.* in *Proc. ACM SIGGRAPH Course Notes* (2002).

248. Maass, S. & Döllner, J. *Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes.* in *Proc. Smart Graphics* (2006), 1–12.

249. Maass, S. & Döllner, J. *Seamless Integration of Labels into Interactive Virtual 3D Environments Using Parameterized Hulls.* in *Proc. CAe* (2008), 33–40.

250. Maass, S., Jobst, M. & Döllner, J. *Depth Cue of Occlusion Information as Criterion for the Quality of Annotation Placement in Perspective Views.* in *Leading the Way with Geo-information* (Springer Berlin Heidelberg, 2007), 473–486.

251. MacEachren, A. M. & Kraak, M.-J. "Research Challenges in Geovisualization". *CaGIS* **28,** 3–12 (2001).

252. MacEachren, A. M., Gahegan, M., Pike, W., Brewer, I., Cai, G., Lengerich, E. & Hardistry, F. "Geovisualization for Knowledge Construction and Decision Support". *IEEE Computer Graphics and Applications* **24,** 13–17 (2004).

253. MacEachren, A. M., Edsall, R., Haug, D., Baxter, R., Otto, G., Masters, R., Fuhrmann, S. & Qian, L. *Virtual environments for geographic visualization: Potential and challenges.* in *Proc. ACM NPIVM* (1999), 35–40.

254. MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M. & Hetzler, E. "Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know". *Cartography and Geographic Information Science* **32,** 139–160 (2005).

255. MacEachren, A. *How Maps Work* (Guilford Press, 1995).

256. MacEachren, A., Roth, R., O'Brien, J., Li, B., Swingley, D. & Gahegan, M. "Visual Semiotics & Uncertainty Visualization: An Empirical Study". *IEEE Trans Vis Comput Graphics* **18,** 2496–2505 (2012).

257. Magdics, M., Sauvaget, C., García, R. J. & Sbert, M. *Post-processing NPR Effects for Video Games.* in *Proc. ACM VRCAI* (2013), 147–156.

258. Mahy, M., Eycken, L. & Oosterlinck, A. "Evaluation of Uniform Color Spaces Developed after the Adoption of CIELAB and CIELUV". *Color Research & Application* **19,** 105–121 (1994).

259. Mannan, S. K., Ruddock, K. H. & Wooding, D. S. "The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images". *Spatial Vision* **10,** 165–188 (1996).

260. Marr, D. & Hildreth, E. "Theory of Edge Detection". *RoyalP* **B-207,** 187–217 (1980).

261. McCabe, D. & Brothers, J. "DirectX 6 Texture Map Compression". *Game Developer Magazine* **5,** 42–46 (1998).

262. McCann, J. & Pollard, N. S. "Real-time Gradient-domain Painting". *ACM Trans. Graph.* **27,** 93:1–93:7 (2008).

263. McGuire, M., Halén, H., St-Amour, J.-F., Thibault, A. & Martel, B. *Stylized Rendering in Games.* in *Proc. ACM SIGGRAPH Courses* (2010).

264. McMaster, R. B. & Shea, S. K. *Generalization in digital cartography.* Tech. rep. (Washington, D.C., USA, 1992).

265. McNamara, A., Bailey, R. & Grimm, C. *Improving Search Task Performance Using Subtle Gaze Direction.* in *Proc. ACM APGV* (2008), 51–56.

266. Meng, L. "How can 3D Geovisualization Please Users Eyes Better?" *Geoinformatics Magazine for Geo-IT Professionals* **5,** 34–35 (2002).

267. Merian, M. *Topographia Germaniae* (2005).

268. Montello, D. R. "Cognitive Map-Design Research in the Twentieth Century: Theoretical and Empirical Approaches". *CaGIS* **29,** 283–304 (2002).

269. Möser, S., Degener, P., Wahl, R. & Klein, R. "Context Aware Terrain Visualization for Wayfinding and Navigation". *Comput. Graph. Forum* **27,** 1853–1860 (2008).

270. Mould, D. *A Stained Glass Image Filter.* in *Proc. EGRW* (2003), 20–25.

271. Mould, D. *Texture-preserving Abstraction.* in *Proc. NPAR* (2012), 75–82.

272. Mower, J. E. "Supporting Automated Pen and Ink Style Surface Illustration with B-Spline Models". *CaGIS* **38,** 174–183 (2011).

273. Myers, K. & Bavoil, L. *Stencil Routed A-Buffer.* in *Proc. ACM SIGGRAPH Sketches* (2007).

274. Nehab, D., Maximo, A., Lima, R. S. & Hoppe, H. "GPU-efficient Recursive Filtering and Summed-area Tables". *ACM Trans. Graph.* **30,** 176:1–176:12 (2011).

275. Neumann, P., Isenberg, T. & Carpendale, S. *NPR Lenses: Interactive Tools for Non-photorealistic Line Drawings.* in *Proc. Smart Graphics* (Springer, 2007), 10–22.

276. Nienhaus, M. & Döllner, J. *Blueprints - Illustrating Architecture and Technical Parts using Hardware-Accelerated Non-Photorealistic Rendering.* in *Proc. Graphics Interface* (2004a), 49–56.

277. Nienhaus, M. & Döllner, J. "Edge-Enhancement - An Algorithm for Real-Time Non-Photorealistic Rendering". *Journal of WSCG* **11,** 346–353 (2003).

278. Nienhaus, M. & Döllner, J. *Sketchy Drawings.* in *Proc. ACM AFRIGRAPH* (2004b), 73–81.

279. Nivala, A.-M. & Sarjakoski, L. T. *Need for Context-Aware Topographic Maps in Mobile Devices.* in *Proc. ScanGIS* (2003), 15–29.

280. Norman, D. A. *The design of everyday things: Revised and expanded edition* (Basic books, 2013).

281. O'Donovan, P. & Mould, D. *Felt-based Rendering.* in *Proc. NPAR* (2006), 55–62.

282. Ohlarik, D. & Cozzi, P. *A Screen-Space Approach to Rendering Polylines on Terrain.* in *Proc. ACM SIGGRAPH Posters* (2011), 68.

283. Olsen, S. C., Maxwell, B. A. & Gooch, B. *Interactive Vector Fields for Painterly Rendering.* in *Proc. Graphics Interface* (2005), 241–247.

284. Orzan, A., Bousseau, A., Barla, P. & Thollot, J. *Structure-preserving Manipulation of Photographs.* in *Proc. NPAR* (2007), 103–110.

285. Pan, B., Zhao, Y., Guo, X., Chen, X., Chen, W. & Peng, Q. "Perception-motivated visualization for 3D city scenes". *The Visual Computer* **29,** 277–286 (2013).

286. Papari, G., Petkov, N. & Campisi, P. "Artistic Edge and Corner Enhancing Smoothing". *IEEE Transactions on Image Processing* **16,** 2449–2662 (2007).

287. Parkhurst, D. J. & Niebur, E. "Scene content selected by active vision". *Spatial vision* **16,** 125–154 (2003).

288. Pasewaldt, S., Trapp, M. & Döllner, J. "Multiscale Visualization of 3D Geovirtual Environments Using View-Dependent Multi-Perspective Views". *Journal of WSCG* **19,** 111–118 (2011).

289. Patterson, T. *A View From on High: Heinrich Berann's Panoramas and Landscape Visualization Techniques For the US National Park Service.* in *Cartographic Perspectives* (2000), 38–65.

290. Patterson, T. & Kelso, N. V. "Hal Shelton Revisted: Designing and Producing Natural-Color Maps with Satellite Land Cover Data". *Cartographic Perspectives,* 28–55 (2004).

291. Pegg, D. *Design Issues with 3D Maps and the Need for 3D Cartographic Design Principles.* Tech. rep. (Online, 2012).

292. Perona, P. & Malik, J. "Scale-space and edge detection using anisotropic diffusion". *IEEE Trans Pattern Anal Mach Intell.* **12,** 629–639 (1990).

293. Peterson, M. P. *Interactive and Animated Cartography* (Prentice Hall, 2005).

294. Petrovic, D. "Cartographic Design in 3D Maps". *Civil Engineering,* 10–16 (2003).

295. Pfautz, J. D. *Depth Perception in Computer Graphics.* PhD thesis (University of Cambridge, 2000).

296. Pham, T. & van Vliet, L. *Separable bilateral filtering for fast video preprocessing.* in *Proc. IEEE ICME* (2005), 1–4.

297. Phong, B. T. "Illumination for Computer Generated Pictures". *Commun. ACM* **18,** 311–317 (1975).

298. Pietriga, E. & Appert, C. *Sigma Lenses: Focus-context Transitions Combining Space, Time and Translucence.* in *Proc. ACM CHI* (2008), 1343–1352.

299. Polis, M. F., Gifford, S. J. & McKeown, D. M. "Automating the Construction of Large-Scale Virtual Worlds". *IEEE Computer* **28,** 57–65 (1995).

300. Porter, T. & Duff, T. "Compositing Digital Images". *SIGGRAPH Comput. Graph.* **18,** 253–259 (1984).

301. Pratt, W. K. *Digital Image Processing* (John Wiley & Sons, Inc., 2001).

302. Praun, E., Hoppe, H., Webb, M. & Finkelstein, A. *Real-Time Hatching.* in *Proc. ACM SIGGRAPH* (2001), 581–586.

303. Praun, E., Finkelstein, A. & Hoppe, H. *Lapped textures.* in *Proc. ACM SIGGRAPH* (2000), 465–470.

304. Qu, H., Wang, H., Cui, W., Wu, Y. & Chan, M.-Y. "Focus+Context Route Zooming and Information Overlay in 3D Urban Environments". *IEEE Trans. Vis. Comput. Graphics* **15,** 1547–1554 (2009).

305. Quillet, J.-C., Thomas, G., Granier, X., Guitton, P. & Marvie, J.-E. *Using Expressive Rendering for Remote Visualization of Large City Models.* in *Proc. ACM Web3D* (2006), 27–35.

306. Raskar, R. & Cohen, M. *Image Precision Silhouette Edges.* in *Proc. ACM I3D* (1999), 135–140.

307. Raskin, J. *The Humane Interface: New Directions for Designing Interactive Systems* (Addison-Wesley Professional, 2000).

308. Rautek, P., Bruckner, S., Gröller, E. & Viola, I. "Illustrative Visualization – New Technology or Useless Tautology?" *ACM SIGGRAPH Computer Graphics* **42,** 4 (2008).

309. Redmond, N. & Dingliana, J. *A Hybrid Approach to Real-Time Abstraction.* in *Proc. Eurographics Ireland* (2009a).

310. Redmond, N. & Dingliana, J. *Adaptive Abstraction of 3D Scenes in Real-Time.* in *Proc. Eurographics Short Papers* (2007), 77–80.

311. Redmond, N. & Dingliana, J. *Investigating the Effect of Real-time Stylisation Techniques on User Task Performance.* in *Proc. APGV* (2009b), 121–124.

312. Reichenbacher, T. *The concept of relevance in mobile maps.* in *Location Based Services and TeleCartography* (Springer, 2007), 231–246.

313. Reichenbacher, T. & Swienty, O. *Attention-Guiding Geovisualisation.* in *Proc. AGILE* (2007), 8–11.

314. Ritter, L., Li, W., Curless, B., Agrawala, M. & Salesin, D. *Painting With Texture.* in *Proc. EGSR* (2006), 371–376.

315. Robertson, G. G., Card, S. K. & Mackinlay, J. D. "Information Visualization Using 3D Interactive Animation". *Commun. ACM* **36,** 57–71 (4 1993).

316. Robinson, A. H., Morrison, J. L., Muehrcke, P. C., Kimerling, A. J. & Guptill, S. C. *Elements of Cartography* (New York: John Wiley & Sons, 1995).

317. Rong, G. & Tan, T.-S. *Jump Flooding in GPU with Applications to Voronoi Diagram and Distance Transform.* in *Proc. ACM I3D* (2006), 109–116.

318. Ropinski, T., Hinrichs, K. H. & Steinicke, F. *A Solution for the Focus and Context Problem in Geo-Virtual Environments.* in *Proc. ISPRS DMGIS* (2005), 144–149.

319. Ropinski, T., Praßni, J.-S., Roters, J. & Hinrichs, K. *Internal Labels as Shape Cues for Medical Illustration.* in *Proc. VMV* (2007), 203–212.

320. Rosenholtz, R., Li, Y. & Nakano, L. "Measuring Visual Clutter". *Journal of Vision* **7,** 1–22 (2007).

321. Rosin, P. L. & Lai, Y.-K. *Non-photorealistic Rendering with Spot Colour.* in *Proc. CAe* (2013), 67–75.

322. Rusinkiewicz, S., Burns, M. & DeCarlo, D. "Exaggerated Shading for Depicting Shape and Detail". *ACM Trans. Graph.* **25,** 1199–1205 (2006).

323. Rusinkiewicz, S., Cole, F., DeCarlo, D. & Finkelstein, A. *Line Drawings from 3D Models*. in *Proc. ACM SIGGRAPH Classes* (2008), 39:1–39:356.

324. Saad, Y. & Schultz, M. H. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM J. Sci. and Stat. Comput.* **7,** 856–869 (1986).

325. Saito, T. & Takahashi, T. *Comprehensible Rendering of 3-D Shapes*. in *Proc. ACM SIGGRAPH* (1990), 197–206.

326. Salisbury, M. P., Wong, M. T., Hughes, J. F. & Salesin, D. H. *Orientable Textures for Image-based Pen-and-ink Illustration*. in *Proc. ACM SIGGRAPH* (1997), 401–406.

327. Samsonov, T. "Morphometric Mapping of Topography by Flowline Hachures". *The Cartographic Journal* **51,** 63–74 (2014).

328. Santella, A. *The Art of Seeing: Visual Perception in Design and Evaluation of Non-Photorealistic Rendering*. PhD thesis (Rutgers University, New Brunswick, New Jersey, USA, 2005).

329. Santella, A. & DeCarlo, D. *Visual Interest and NPR: an Evaluation and Manifesto*. in *Proc. NPAR* (2004), 71–78.

330. Schneider, M., Guthe, M. & Klein, R. *Real-time Rendering of Complex Vector Data on 3D Terrain Models*. in *Proc. VSMM* (2005), 573–582.

331. Schneider, M. & Klein, R. "Efficient and Accurate Rendering of Vector Data on Virtual Landscapes". *Journal of WSCG* **15,** 59–66 (2007).

332. Schwarz, M., Isenberg, T., Mason, K. & Carpendale, S. *Modeling with Rendering Primitives: An Interactive Non-photorealistic Canvas*. in *Proc. NPAR* (2007), 15–22.

333. Scott, M. *Oil Painter's Bible* (Chartwell Books, 2005).

334. Seitz, S. M. & Dyer, C. R. *View morphing*. in *Proc. ACM SIGGRAPH* (1996), 21–30.

335. Selan, J. *Using Lookup Tables to Accelerate Color Transformations*. in *GPU Gems* (Addison-Wesley, 2004), 381–392.

336. Sester, M. *3D Visualization and Generalization*. in *Photogrammetric Week* (2007), 285–295.

337. Shanmugam, P. & Arikan, O. *Hardware Accelerated Ambient Occlusion Techniques on GPUs*. in *Proc. ACM I3D* (2007), 73–80.

338. Shesh, A. & Chen, B. *Efficient and Dynamic Simplification of Line Drawings*. in *Comput. Graph. Forum* **27** (2008), 537–545.

339. Shiraishi, M. & Yamaguchi, Y. *An Algorithm for Automatic Painterly Rendering Based on Local Source Image Approximation*. in *Proc. NPAR* (2000), 53–58.

340. Shishkovtsov, O. "Deferred Shading in S.T.A.L.K.E.R." *GPU Gems* **2,** 143–166 (2005).

341. Shneiderman, B., Plaisant, C., Cohen, M. & Jacobs, S. *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (Pearson, 2009).

342. Shneiderman, B. *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.* in *Proc. IEEE Symposium on Visual Languages* (1996), 336–343.

343. Shreiner, D., Sellers, G., Kessenich, J. M. & Licea-Kane, B. M. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3* (Addison-Wesley, 2013).

344. Smallman, H. S. & John, M. S. "Naive Realism: Misplaced Faith in Realistic Displays". *Ergonomics in Design: The Quarterly of Human Factors Applications* **13,** 6–13 (2005).

345. Smith, K., Landes, P.-E., Thollot, J. & Myszkowski, K. "Apparent Greyscale: A Simple and Fast Conversion to Perceptually Accurate Images and Video". *Comput. Graph. Forum* **27,** 193–200 (2008).

346. Strothotte, T. & Schlechtweg, S. *Non-Photorealistic Computer Graphics. Modeling, Rendering, and Animation* (Morgan Kaufmann Publishers, 2002).

347. Subr, K., Soler, C. & Durand, F. "Edge-preserving Multiscale Image Decomposition based on Local Extrema". *ACM Trans. Graph.* **28,** 147:1–147:9 (2009).

348. Surdick, R. T., Davis, E. T., King, R. A., Corso, G. M., Shapiro, A., Hodges, L. & Elliot, K. *Relevant Cues for the Visual Perception of Depth: Is Where You See it Where it is?* in *Proc. Hum. Fact. Ergon. Soc. Annu. Meet.* **38** (1994), 1305–1309.

349. Takahashi, S., Yoshida, K., Shimada, K. & Nishita, T. "Occlusion-Free Animation of Driving Routes for Car Navigation Systems". *IEEE Trans. Vis. Comput. Graphics* **12,** 1141–1148 (2006).

350. Tanaka, K. "The Relief Contour Method of Representing Topography on Maps". *Geographical Review,* 444–456 (1950).

351. Todd, J. T. "The visual perception of 3D shape". *Trends in Cognitive Sciences* **8,** 115–121 (2004).

352. Todo, H., Anjyo, K.-i., Baxter, W. & Igarashi, T. "Locally Controllable Stylized Shading". *ACM Trans. Graph.* **26,** 17:1–17:7 (2007).

353. Tomasi, C. & Manduchi, R. *Bilateral Filtering for Gray and Color Images.* in *Proc. ICCV* (1998), 839–846.

354. Tominski, C., Gladisch, S., Kister, U., Dachselt, R. & Schumann, H. *A Survey on Interactive Lenses in Visualization.* in *Proc. EuroVis - STARs* (2014), 43–62.

355. Trapp, M., Beesk, C., Pasewaldt, S. & Döllner, J. *Interactive Rendering Techniques for Highlighting in 3D Geovirtual Environments.* in *Proc. 3D GeoInfo Conference* (2010), 197–210.

356. Trapp, M., Glander, T., Buchholz, H. & Döllner, J. *3D Generalization Lenses for Interactive Focus+Context Visualization of Virtual City Models.* in *Proc. IEEE IV* (2008), 356–361.

357. Tufte, E. R. *The Visual Display of Quantitative Information* (Graphics press Cheshire, CT, 1983).

358. Tufte, E. R. *Visual Explanations: Images and Quantities, Evidence and Narrative* (Graphics Press Cheshire, CT, 1997).

359. Tversky, B., Morrison, J. B. & Betrancourt, M. "Animation: can it facilitate?" *Int. Journal of Human-Computer Studies* **57,** 247–262 (4 2002).

360. Tyner, J. *Principles of Map Design* (Guilford Press, 2010).

361. Vaaraniemi, M., Treib, M. & Westermann, R. "High-Quality Cartographic Roads on High-Resolution DEMs". *Journal of WSCG* **12,** 41–48 (2011).

362. Vaaraniemi, M., Freidank, M. & Westermann, R. *Enhancing the Visibility of Labels in 3D Navigation Maps.* in *Progress and New Trends in 3D Geoinformation Sciences* (Springer, 2013), 23–40.

363. Vaaraniemi, M., Görlich, M. & in der Au, A. "Intelligent Prioritization and Filtering of Labels in Navigation Maps". *Journal of WSCG* **22,** 11–20 (2014).

364. Van Der Zwan, M., Telea, A. & Isenberg, T. *Continuous Navigation of Nested Abstraction Levels.* in *Proc. EuroVis Shortpapers* (2012), 13–17.

365. Vázquez, P.-P., Feixas, M., Sbert, M. & Heidrich, W. "Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modeling". *Comput. Graph. Forum* **22,** 689–700 (2004).

366. Vergne, R., Vanderhaeghe, D., Chen, J., Barla, P., Granier, X. & Schlick, C. "Implicit Brushes for Stylized Line-based Rendering". *Comput. Graph. Forum* **30,** 513–522 (2011).

367. Viega, J., Conway, M. J., Williams, G. & Pausch, R. *3D Magic Lenses.* in *ACM UIST* (1996), 51–58.

368. Viola, I., Kanitsar, A. & Groller, M. E. *Importance-Driven Volume Rendering.* in *Proc. IEEE Visualization* (2004), 139–146.

369. Viola, I., Gröller, M. E., Hadwiger, M., Bühler, K., Preim, B., Sousa, M. C., Ebert, D. & Stredney, D. *Illustrative Visualization.* in *Proc. IEEE Visualization Tutorials* (2005).

370. Wagner, D. *Terrain Geomorphing in the Vertex Shader.* in *ShaderX2* (Wordware Publishing, 2003).

371. Wang, F., Li, Y., Sakamoto, D. & Igarashi, T. *Hierarchical Route Maps for Efficient Navigation.* in *Proc. ACM UIST* (2014), 169–178.

372. Wang, L., Zhao, Y., Mueller, K. & Kaufman, A. *The Magic Volume Lens: An Interactive Focus+Context Technique for Volume Rendering.* in *Proc. IEEE Visualization* (2005), 367–374.

373. Wang, S., Pan, B., Guo, X., Wang, Z. & Peng, Q. *2.5D Focus+Context Map Visualization.* in *Proc. CAD/Graphics* (2011), 389–396.

374. Wang, T., Collomosse, J., Hunter, A. & Greig, D. *Learnable Stroke Models for Example-based Portrait Painting.* in *Proc. British Machine Vision Conference* (2013), 36.1–36.11.

375. Wanger, L., Ferwerda, J. & Greenberg, D. "Perceiving Spatial Relationships in Computer Generated Images". *IEEE Computer Graphics and Applications* **12,** 44–58 (1992).

376. Ware, C. *Information Visualization: Perception for Design* (Morgan Kaufmann Publishers Inc., San Francisco, 2004).

377. Wartell, Z., Kang, E., Wasilewski, T., Ribarsky, W. & Faust, N. *Rendering Vector Data over Global, Multi-resolution 3D Terrain.* in *Proc. Data Visualisation* (2003), 213–222.

378. Webb, M., Praun, E., Finkelstein, A. & Hoppe, H. *Fine Tone Control in Hardware Hatching.* in *Proc. NPAR* (2002), 53–58.

379. Weber, A. & Benner, J. *Interactive Generation of Digital Terrain Models Using Multiple Data Sources.* in *Digital Earth Moving* (Springer, 2001), 60–64.

380. Wei, L., Lefebvre, S., Kwatra, V. & Turk, G. *State of the Art in Example-based Texture Synthesis.* in *Proc. Eurographics State-of-the-Art Reports* (2009), 93–117.

381. Weickert, J. *Anisotropic Diffusion in Image Processing* (Teubner Stuttgart, 1998).

382. Wen, F., Luan, Q., Liang, L., Xu, Y.-Q. & Shum, H.-Y. *Color Sketch Generation.* in *Proc. NPAR* (2006), 47–54.

383. Whelan, J. C. & Visvalingam, M. *Formulated Silhouettes for Sketching Terrain.* in *Proc. TPCG* (2003), 90–96.

384. Willett, W., Jenny, B., Isenberg, T. & Dragicevic, P. *Lightweight Relief Shearing for Enhanced Terrain Perception on Interactive Maps.* in *Proc. ACM CHI* (2015), 3563–3572.

385. Williams, L. *Pyramidal Parametrics.* in *Proc. ACM SIGGRAPH* **17** (1983), 1–11.

386. Winkenbach, G. & Salesin, D. H. *Computer-Generated Pen-and-Ink Illustration.* in *Proc. ACM SIGGRAPH* (1994), 91–100.

387. Winnemöller, H., Kyprianidis, J. E. & Olsen, S. C. "XDoG: an extended difference-of-Gaussians compendium including advanced image stylization". *Computers & Graphics* **36,** 740–753 (2012).

388. Winnemöller, H., Olsen, S. C. & Gooch, B. "Real-Time Video Abstraction". *ACM Trans. Graph.* **25,** 1221–1226 (2006).

389. Wolff, M. & Asche, H. *Geospatial Crime Scene Investigation - From Hotspot Analysis to Interactive 3D Visualization.* in *Computational Science and Its Applications - ICCSA 2009* **5592** (Springer Berlin Heidelberg, 2009), 285–299.

390. Wood, J., Kirschenbauer, S., Döllner, J., Lopes, A. & Bodum, L. *Using 3D in Visualization.* in *Exploring Geovisualization* (Pergamon Press, 2005), 295–312.

391. Wood, J., Isenberg, P., Isenberg, T., Dykes, J., Boukhelifa, N. & Slingsby, A. "Sketchy Rendering for Information Visualization". *IEEE Trans Vis Comput Graphics* **18,** 2749–2758 (2012).

392. Xie, X., He, Y., Tian, F., Seah, H.-S., Gu, X. & Qin, H. "An Effective Illustrative Visualization Framework Based on Photic Extremum Lines (PELs)". *IEEE Trans Vis Comput Graphics* **13,** 1328–1335 (2007).

393. Xu, K., Cohen-Or, D., Ju, T., Liu, L., Zhang, H., Zhou, S. & Xiong, Y. "Feature-aligned shape texturing". *ACM Trans. Graph.* **28,** 108:1–108:7 (2009).

394. Xu, L., Lu, C., Xu, Y. & Jia, J. "Image Smoothing via L_0 Gradient Minimization". *ACM Trans. Graph.* **30,** 174:1–174:12 (2011).

395. Yatziv, L. & Sapiro, G. "Fast image and video colorization using chrominance blending". *IEEE Trans Image Processing* **15,** 1120–1129 (2006).

396. Yoeli, P. "Topographical Relief Depiction by Hachures with Computer and Plotter". *The Cartographic Journal* **22,** 111–124 (1985).

397. Yoon, J.-C., Lee, I.-K. & Kang, H. "Video Painting Based on a Stabilized Time-Varying Flow Field". *IEEE Trans. Vis. Comput. Graphics* **18,** 58–67 (2012).

398. Yu, J., Jiang, X., Chen, H. & Yao, C. "Real-time Cartoon Water Animation". *Computer Animation and Virtual Worlds* **18,** 405–414 (2007).

399. Yu, J., Lu, C. & Sato, Y. *Sparsity-based Color Quantization with Preserved Image Details.* in *SIGGRAPH Asia 2014 Posters* (2014), 32:1–32:1.

400. Yue, X., Miao, D., Cao, L., Wu, Q. & Chen, Y. "An efficient color quantization based on generic roughness measure". *Pattern Recognition* **47,** 1777–1789 (2014).

401. Zander, J., Isenberg, T., Schlechtweg, S. & Strothotte, T. *High Quality Hatching.* in *Comput. Graph. Forum* **23** (2004), 421–430.

402. Zanola, S., Fabrikant, S. I. & Çöltekin, A. *The Effect of Realism on the Confidence in Spatial Data Quality in Stereoscopic 3D Displays.* in *Proc. International Cartographic Conference* (2009).

403. Zeng, K., Zhao, M., Xiong, C. & Zhu, S.-C. "From Image Parsing to Painterly Rendering". *ACM Trans. Graph.* **29,** 2:1–2:11 (2009).

404. Zhang, E., Hays, J. & Turk, G. "Interactive Tensor Field Design and Visualization on Surfaces". *IEEE Trans. Vis. Comput. Graphics* **13,** 94–107 (2007).

405. Zhang, L., He, Y., Xia, J., Xie, X. & Chen, W. "Real-Time Shape Illustration Using Laplacian Lines". *IEEE Trans. Vis. Comput. Graphics* **17,** 993–1006 (2011).

406. Zhao, M. & Zhu, S.-C. *Customizing Painterly Rendering Styles Using Stroke Processes.* in *Proc. NPAR* (2011a), 137–146.

407. Zhao, M. & Zhu, S.-C. *Portrait Painting Using Active Templates.* in *Proc. NPAR* (2011b), 117–124.

408. Zhao, M. & Zhu, S.-C. *Sisley the Abstract Painter.* in *Proc. NPAR* (2010), 99–107.

Eidesstattliche Erklaerung
Declaration of Academic Honesty

Hiermit versichere ich, dass ich die vorliegende Dissertation ohne Hilfe Dritter und ohne Zuhilfenahme anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I hereby declare in lieu of an oath that this thesis has been written by myself without any external unauthorized help, that it has been neither presented to any institution for evaluation nor previously published in its entirety or in parts.

Potsdam, Germany

March 23, 2016

Amir Semmo