Image-based Synthesis and Re-Synthesis of Viewpoints Guided by 3D Models

Konstantinos Rematas[1], Tobias Ritschel[2,3], Mario Fritz[2], and Tinne Tuytelaars[1]

[1]KU Leuven, iMinds
[2]Max Planck Institute for Informatics
[2]Saarland University

Abstract

We propose a technique to use the structural information extracted from a set of 3D models of an object class to improve novel-view synthesis for images showing unknown instances of this class. These novel views can be used to "amplify" training image collections that typically contain only a low number of views or lack certain classes of views entirely (e. g. top views).

We extract the correlation of position, normal, reflectance and appearance from computer-generated images of a few exemplars and use this information to infer new appearance for new instances. We show that our approach can improve performance of state-of-the-art detectors using real-world training data. Additional applications include guided versions of inpainting, 2D-to-3D conversion, super-resolution and non-local smoothing.

1. Introduction

Given a single view of a car, humans get a pretty good idea how it will look from other angles. How is it possible for us to hallucinate the rest of the car without ever having seen it? The answer lies in the knowledge of structural information about regularities, shapes, materials and symmetries of objects that we use to build an informed hypothesis of novel viewpoints in our mind.

Predicting how an object would appear from one view when given a 2D image taken from another view (novel view synthesis) has both technical and esthetical applications. Technically, the new views can be used to simulate binocular stereo, depth of field, motion blur [4], or to improve the performance of detectors, as we propose in this work. Esthetically, a change of view has its own value – "seeing something from a different view" – and is used routinely by artists to convey presence, most well-known as the Ken Burn's effect [15].

In contrast, the predominant paradigm in computer vision is to present all possible viewpoints in order to arrive at a model that is robust to out-of-plane rotations. The most prominent detection models lack the domain knowledge that would give them an understanding of how generalization across viewpoints can be achieved from a single-view example. A dense sampling across viewpoints and intra-category variation is tedious to achieve. Recent analysis of such detectors has pointed out that rare cases in viewpoint is indeed one of the frontiers on which there is still significant room to push the state-of-the-art in object detection [14].

In this work, we show how to improve novel view synthesis by making use of correlations observed in 3D models and applying them to new image instances. Intuitively, if we observe a certain appearance at one position, orientation and material, a similar position, orientation and material will have a similar appearance, even if it was not visible in the original image and irrespectively of the appearance itself. A simple example are the colors of the windows of a car: they may differ between exemplars, but for a particular car they are likely identical. A more advanced example is shading: if all surfaces oriented in a certain direction are dark in the original view (*i.e.* directions facing away from the sun), they will also be dark in the novel view for similar orientations. This reasoning is only possible if an approximate 3D model is roughly aligned with the image, which is achieved manually in our experiments, but could be automated using [30, 12].

We show the application of our method to synthesis and re-synthesis of training data for object detectors. In synthesis, we are able to generate new viewpoints and thereby "amplify" a training set. We show improved detection rates – with strong improvements on rare viewpoints. In re-synthesis, we show how to exploit the structural knowledge in order to denoise, inpaint, and upsample images as well as generate stereo-pairs from single images. We exemplify

Project page:
http://homes.esat.kuleuven.be/~krematas/imgSynth/

the use of resynthesis for learning object detectors from corrupted data (e.g. noise, low-resolution).

2. Previous work

View Synthesis Generating novel views from 2D images is an image-based rendering problem, with several applications in computer vision [2] and computer graphics [4, 15]. View interpolation including complex shading is possible by using surface light fields [37] or lumigraphs [11]. When deforming and blending images, detecting holes and filling them is the biggest challenge [6], usually solved either by stretching foreground over background or by inpainting [5, 1], but no approach we are aware of uses the structure of a 3D template to guide this inpainting. Appearance that depends on a shape template was proposed for the diffuse case and manual intervention to capture the appearance of art in the "Lit Sphere" metaphor [32]. Our approach can be seen as the automatic and continuous generalization from one to many, context-dependent Lit Spheres. Image Analogies [13] learns the transformation from a pair of images, to apply to a novel image. Similarly, we learn the relation between two rendered 3D views of an image to generate appearance in new views from a single image for general appearance and account for the relation of 3D geometry and materials [17]. Simpler, joint bilateral upsampling [19] has been used to reconstruct a high-resolution image from a low-resolution image using a high-resolution guidance signal. Our approach uses multiple synthesized high-resolution images to guide view synthesis, including upsampling. For 3D reconstruction of occluded areas in scanned objects, approaches based on templates are routinely used [25] to fill holes in surfaces, but these operate on 3D surfaces and do not account for appearance such as we do. Another approach to reconstruct occluded regions is [35], but the method requires visibility of the region or its symmetric in the video sequence.

Training from synthetic data There is an increasing interest to train computer vision models from synthesized data in order to achieve more scalable learning schemes. Work on fully synthetic data has shown mostly successful in depth data [31, 20, 36], shape-based representations [33, 22], textures [34, 21] and scenes [18]. In contrast, we take an image-based approach in order to leverage real-world image statistics as well as the intra-class variation available to us in image data. Previous image-based work synthesizes training images by recombining poses and appearance of objects in order to create new instances [8, 28, 29, 27, 38]. In contrast, our work focuses on synthesis across viewpoints and deals with disocclusions that are not addressed in previous work.

3. Guiding novel-view synthesis by a 3D model

We pose novel view synthesis as reconstructing of appearance as a function of scene attributes from observed samples. Input to our algorithm is an image and an aligned 3D model and output is an image from a novel view. First, we notice, that both the original and the novel view of the 3D model can be rendered easily. This allows to put pixel locations from the original to the novel view into correspondence (flow) and to detect disocclusions, *i.e.* pixels in the novel view that were not present in the original view. Simply copying pixels along their flow will result in the classic projective texturing used in image-based rendering [6]. The challenge is to consistently fill the disocclusions by inpainting. The most simple solution would be to replace disoccluded pixels with pixels from the rendered view. This however, cannot produce a consistent appearance as precise materials, lighting and geometry are unknown. The key observation is, that an image rendered from the novel view never has disocclusions. Therefore, we also know all 3D properties, such as position, normal, reflectance, for all disoccluded pixels. We can use this information to guide the inpainting of appearance for such pixels. To this end, we copy appearance from the original view, that is similar in terms of position, normal and reflectance.

3.1. Sampling appearance

Input to our system is appearance in the form of a 2D RGB image $f_1 : \mathbb{R}^2 \to \mathbb{R}^3$ and a 3D model $\mathcal{M} \subseteq \mathbb{R}^3$ with Phong reflectance $\mathcal{F} : \mathcal{M} \to \mathbb{R}^9$ defined on it (diffuse color, specular color and glossiness). We want to compute the image f_2 that shows the scene from a view different by the matrix $\mathsf{T} \in \mathbb{R}^{4\times4}$.

The 3D model typically contains the object in question, as well as its context, e. g. a car standing on a planar ground, which is very common for cars. The particular type (triangular, (NURBS) patches, procedural) and technical properties of the 3D model (face count, orientability, UV coordinates) are not relevant for our approach, we only require that the model can be rendered into an image from a view.

First, let $p_1 : \mathbb{R}^2 \to \mathbb{R}^3$ be the *position image*, which is the projection of $\mathcal{M}$ in the original view after hidden-surface removal. We produce such images using deferred shading [7] with z-buffering, i. e. generating positions instead of shaded colors in rendering. Using deferred shading, we also compute $n_1 : \mathbb{R}^2 \to \mathbb{R}^3$, the *normal image* computed from the derivative of $\mathcal{M}$: $r_1 : \mathbb{R}^2 \to \mathbb{R}^9$, the *reflectance image*, and $L_1 : \mathbb{R}^2 \to \mathbb{R}^3$, the radiance (Fig. 1). Position, normal and reflectance images p_2, n_2, r_2 and L_2 from the novel view T are produced in the same way.

Instead of aligning the 3D model $\mathcal{M}$ to the original image f in 3D, we simply deform the original 2D image [2] to align it to the images p_1, n_1 and r_1 of the 3D model $\mathcal{M}$.

Next, we compute the flow $w : \mathbb{R}^2 \to \mathbb{R}^2$ between the

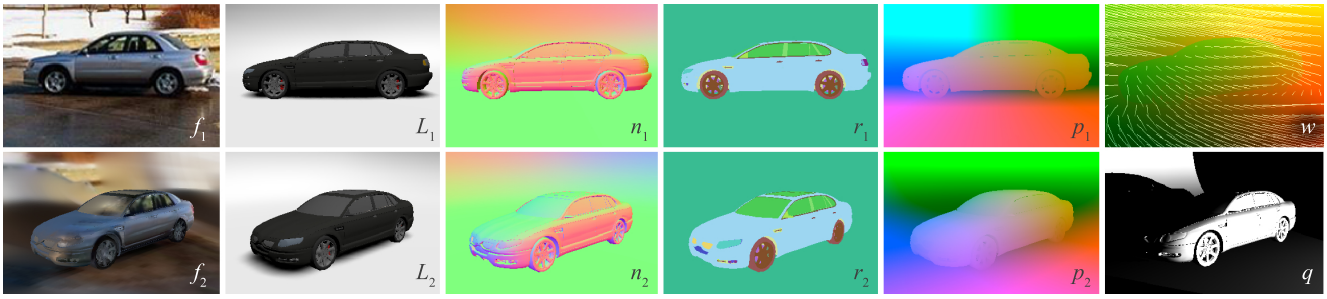

Figure 1. We compute a novel view image f_2 from the input image f_1 using 3D information of an aligned 3D template (Left to right: radiance, normals, reflectance, positions) as guidance, even if its renderings L_1 and L_2 have appearance largely different from f_1.

novel view and the original view, i. e. where every pixel in the novel view is coming from in the original view. The flow is undefined for positions that were occluded in the original view. We again rasterize $\mathcal{M}$ from the view T but store $w(\mathbf{x}) = \rho(p(\mathbf{x})) - \rho(p(\mathsf{T}p(\mathbf{x})))$ at the pixel with location $\mathbf{x}$, where $\rho : \mathbb{R}^3 \to \mathbb{R}^2$ is the (perspective) projection from world to image space. Additionally we compute sampling quality and occlusion in a z-buffer-like test, formalized by a function

$$q(\mathbf{x}) = \begin{cases} \max(0, n(\mathbf{x}) \cdot v(\mathbf{x})) & \text{if } z(p(\mathbf{x}) - \mathsf{T}p(\mathbf{x})) > \epsilon \\ 0 & \text{else,} \end{cases}$$

where $v(\mathbf{x}) = \mathsf{T}^{-1}\mathbf{x}/||\mathsf{T}^{-1}\mathbf{x}||_2^2$ is the normalized viewing direction and z returns the depth component of a vector.

Further we define a metric on all attributes as follows (Fig. 2): As a distance on positions Δ_p we use Euclidean distance; for normals, we use the dot product as the distance Δ_p; for reflectance, we apply a perceptually linear Phong BRDF distance Δ_r similar to [26]; radiance and locality of two pixels Δ_L and Δ_l is again measured using Euclidean distance.

Using all the above, we can now write the probability of assigning the appearance from pixel position $\mathbf{x}_1$ to the novel appearance at location $\mathbf{x}_2$ as

$$\begin{aligned} c(\mathbf{x}_1, \mathbf{x}_2) = q(\mathbf{x}_1)/(&w_\mathrm{p}\Delta_\mathrm{p}(p_1(\mathbf{x}_1), p_1(\mathbf{x}_2))+ \\ &w_\mathrm{n}\Delta_\mathrm{n}(n_1(\mathbf{x}_1), n_1(\mathbf{x}_2))+ \\ &w_\mathrm{r}\Delta_\mathrm{r}(r_1(\mathbf{x}_1), r_1(\mathbf{x}_2)))+ \\ &w_\mathrm{L}\Delta_\mathrm{L}(L_1(\mathbf{x}_1), L_1(\mathbf{x}_2)))+ \\ &w_\mathrm{l}\Delta_\mathrm{l}(\mathbf{x}_1, \mathbf{x}_2)). \end{aligned}$$

3.2. Reconstructing appearance

All pixels in the result image f_2 are reconstructed independently for every location $\mathbf{x}_2$, as

$$f_2(\mathbf{x}_2) = \int c(\mathbf{x}_1, \mathbf{x}_2)^s f_1(\mathbf{x}_1) \mathrm{d}\mathbf{x}_1 \Big/ \int c(\mathbf{x}_1, \mathbf{x}_2)^s \mathrm{d}\mathbf{x}_1 \ , \tag{1}$$

where s is a sharpness parameter. If s is low, the reconstructed appearance is combined from many sources. It is more reliable, but also more smooth. If s gets larger, fewer, but higher-quality observations dominate the solution.

For discrete images the integral above turns into a sum in practice. We do not need to iterate over the entire image but only over a local neighborhood. If $\mathbf{x}_2$ is the novel image pixel position, we run over a fixed-size neighborhood around location $\mathbf{x}_2 + w(\mathbf{x}_2)$. This is because more correlated pixels are found in the neighborhood of the pixel position, that a world space position had in the original view. We use a GPU to produce all guide images using deferred shading and to evaluate Eq. 1 on a 512×512 image using a 128×128 neighborhood in less than a second, allowing to interactively explore novel views by moving a virtual camera.

Figure 3. Contribution of distances to the reconstruction *(See text)*.

For reconstruction, the weights w are tuned by visual inspection of the result, which is easy thanks to the interactive feedback (Fig. 3). We now discuss the individual terms: First (Fig. 3,+n), if all weights except normal are zero, our approach is equivalent to the Lit Sphere approach of Sloan *et al.* [32]. All details are missing here, except a global dependency on surface orientation. Second (Fig. 3,+r), adding a reflectance term is equivalent to a Lit Sphere for each material. This is equivalent to a "continuous" Lit Sphere depending on material parameters. Third (Fig. 3,+L), the radiance term prefers appearance that is similar if multiple appearances are equivalent. This is the case for the ground plane, that is either shadows or unshadowed in L_1 and L_2, which is transferred from f_1. Finally (Fig. 3,+p), adding a dependency on position prefers local appearance if everything else is similar. In our example, the cobblestone pattern that is not visible in any feature, but correlates with position is reproduced. Note, that when adding one component, no other component is degraded.

The 3D model is fit into the unit cube to normalize

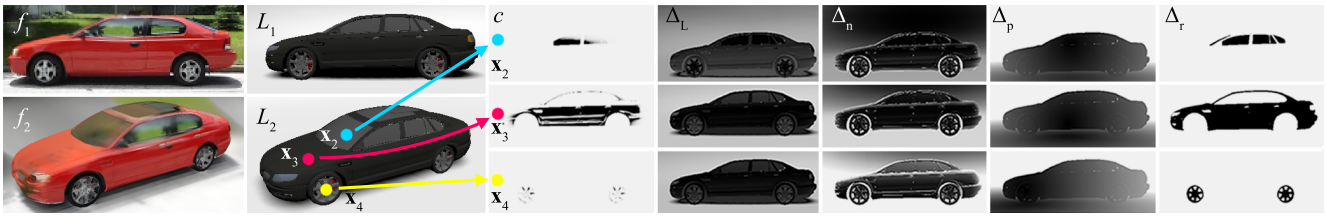

Figure 2. Weights c and distances $\Delta_{\{\mathrm{L,n,p,r}\}}$ of three output positions $\mathbf{x}_{\{2,3,4\}}$ in respect to all positions from the input image.

positions and make them comparable for different scenes. The settings in Fig. 3 are $w_\mathrm{n} = 1$, $w_\mathrm{r} = 2$, $w_\mathrm{L} = 0.1$, $w_\mathrm{p} = 0.01$, $w_\mathrm{l} = 1$. The sharpness is set to $s = 3$. We keep those constant for all results reported throughout the paper.

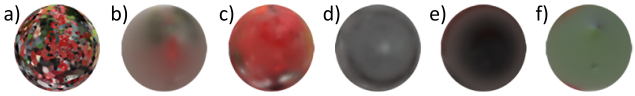

Figure 4. Gauss sphere visualizations (Lit Spheres) of the input image f_1 seen in Fig. 3: *a)* Reconstructed using nearest-neighbor. Note the increasing density towards frontal views. *b)* Our reconstruction without locality ($w_\mathrm{n} > 0, w_{\{\mathrm{r,L,p}\}} = 0$). *c–f)* Our reconstruction with locality ($w_{\{\mathrm{n,r,L,p}\}} > 0$) for a position on the body, on the rim, on the tire and on the wind shield.

A visualization of the sampling in the directional domain is seen in Fig. 4. First, it can be noticed, that several normals are only encountered rarely, in particular no normals facing away from the viewer are observed (Fig. 4, a). We extrapolate information for such areas using our smooth reconstruction. Next, we see how appearance does depend on orientation (Fig. 4, b): front-facing directions tend to be red in this example. Finally, using different spheres for different parts of the image (Fig. 4, c–f), results in different orientation-dependent appearances.

Regarding illumination, the reference images are illuminated using one "representative" lighting (skylight with ambient occlusion), that can be assumed to also be present in the 2D image for cars. For background, we use a large white background sphere for all 3D models such that the function p, n, r, L, *etc*. are always defined. When evaluating Eq1 on background, distance to foreground is so large that it does not contribute, so similarity is based on 3D position and normal alone. Effectively the background is projected onto a sphere, including proper occlusion.

4. Experiments

Our approach has applications ranging from amplification of training data over 2D-to-3D conversion, inpainting to super-resolution and feature-aware smoothing. In all results reported, we assume a 3D template[1] was selected and aligned with the view in the results to follow. We perform this step – which can be automated [30, 12] – manually here.

4.1. Training data amplification

Object detection and classification approaches have seen substantial improvements over the last decade. One driving factor is the availability of training data that is representative for the test scenarios of interest. However, the construction of such data sets is tedious and yet does not capture all aspects of variability in the classes that it contains. In particular the sampling of untypical examples or viewpoints is often lacking [14]. More specifically, the popular PASCAL VOC benchmark [9] provides a good sampling of intra-class variations - yet there is no exhaustive view-point sampling. Other data sets like EPFL Cars [24] provide dense view sampling (without azimuth) but do not capture intra-class variation well.

Next we will show how to "amplify" the PASCAL VOC training data in order to represent the intra-class variation together with a better viewpoint sampling that even includes atypical views. Our study focuses on the "car" class.

Synthetic Viewpoint Dataset We base our dataset amplification on 26 sideview images which we manually align with the 3D models (about 2 h effort). Then we apply our approach for novel view synthesis to generate for each image 9 synthetic views by sampling the viewing sphere. Examples of the synthesized data are shown in Fig. 5. Note how effects of global illumination on the vehicle as well as shadow below the car are preserved in many images. The disocclusion areas are filled in in a plausible and natural way. The transition between the visible and "hallucinated" part is seamless. Given this augmented dataset we run a series of experiment to underline the validity of our approach. In all experiments, we use the state-of-the-art Deformable Part Model (DPM version 5) [10].

Pilot study: Resynthesis As an initial test, we investigate how much the synthesized views affect the performance, compared with the real images and direct renderings of the 3D models. We perform this study on the 26 sideviews which are resynthesized by treating the visible part of the car as a disocclusion. Using this data we train a DPM detector and we test on the whole VOC test set. Ta-

[1] We used 3 CAD models in total (SUV, Sedan, Compact).

Figure 5. Novel views used to amplify the PASCAL VOC input training image data set *(1st and 5th column)*.

Table 1. Performance of the DPM detector when trained in different data and with different number of components (N).

Data	Avg. precision (%)		#Views	
	$N = 1$	$N = 3$	Real	Synth.
Side	15.4	16.2	26	-
Side+rend.	11.5	12.7	-	26
Side+synth	15.0	14.5	-	26

ble 1 shows the performance numbers in average precision (PASCAL VOC standard criterion) for different number of components (columns) in the DPM. The first line represents training on the real 26 sideviews. Then we repeat the training but in this case we have removed the car using the "Context Aware"-tool from Adobe Photoshop and we replaced it with a rendering of the 3D model that corresponds to that type of car (Sedan, SUV or Compact). We observe that the performance is much lower, due to the lack of variation in the car appearance. The last row corresponds to training on the 26 resynthesized cars using our approach. The performance is very close to the training on the real cars which provides first evidence that our method is indeed able to generate the kind of realism that is needed to successfully train an object detection algorithm without a strong loss in performance that is often observed in such settings.

PASCAL VOC data set We continue by using our whole synthetic viewpoint dataset and mixing it with different portions of real data. Quantitative results varying the number of components in the DPM are shown in Tbl. 2 and performance plots in Fig. 7. We first focus on the upper half of Table 2 where we test on the standard PASCAL VOC test set. We start by training only on the 26 sideviews (Side) that we consider for amplification and compare the performance to a model on the same 26 views amplified by 728 views synthesized by our approach (Side+synth). We observe improvements between 14 to 16.5% by adding our synthesized views. We now compare the model trained on the full VOC training set (Full) to a version to which we add our synthetic views. In this setting we do not observe an improvement in average precision, but rather comparable performance. However, if we inspect the associated precision recall curve in Fig. 7 (left) more closely, we observe that our model improves in the high precision regime. It produces no single false positive until 15% of the positives are detected (recall).

In order to generate further insights into our model, we perform a study similar to the one proposed in [14]. Here we focus on rare viewpoints of the cars by selecting a subset of the PASCAL VOC testset where the top of the car becomes visible. Our reasoning is that those cases are difficult for the standard model as this part of the viewpoint distribution is poorly sampled. Fig. 6 confirms out intuition about the view

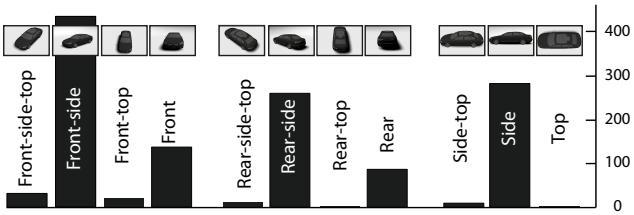

Figure 6. Object side visibility for car VOC2007 test data set.

Table 2. Performance of the DPM detector on PASCAL VOC dataset with varying training set and different number of components (N).Evaluation is performed on the full test set as well as a subset of rare viewpoints.

Test	Train	Avg. precision (%)			#Views	
		$N = 3$	$N = 4$	$N = 5$	Real	Synth.
VOC 2007	Side	16.2	18.4	16.7	26	-
	Side+synth	30.2	31.4	33.2	26	728
	Full	51.7	53.4	50.7	1250	-
	Full+synth	50.2	53.1	50.9	1250	728
VOC rare	Side	11.9	11.6	10.3	26	-
	Side+synth	23.2	30.2	32.9	26	728
	Full	51.9	52.5	51.8	1250	-
	Full+synth	55.0	57.3	53.1	1250	728

distribution in the VOC dataset. The statistic shows that all views involving the visibility of the top are underrepresented. The lower part of Tbl. 2 performs an analysis for detection of such rare viewpoints. We see strong improvements on those rare viewpoints of up to 22.6% for training from sideviews and up to 4.8% for the full training set.

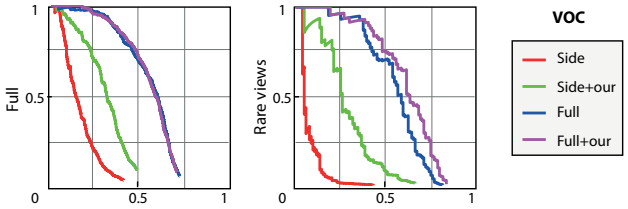

Figure 7. Performance of the "side" and "full" versions of PASCAL VOC data set with and without novel views produced using our approach. Evaluation is performed on the full PASCAL VOC test set (left) as well as a subset of rare viewpoints (right).

UCLA data set In order to get a more realistic estimate of the performance across viewpoints we turn to the UCLA cars data set [16] which has been designed with a more uniform viewpoint sampling in mind. The test set consists of 200 images that cover better the viewing sphere.

Table 3. Performance of the DPM detector on UCLA dataset with varying training set and different number of components (N).Evaluation is performed on the full test set as well as a subset of rare viewpoints.

Test	Train	Avg. precision (%)			#Views	
		$N=3$	$N=4$	$N=5$	Real	Nov.
UCLA	Side	41.5	43.5	41.1	26	-
	Side+synth	83.0	82.4	85.4	26	728
	Full	75.4	78.7	78.3	1250	-
	Full+synth	84.0	86.0	85.2	1250	728
UCLA rare	Side	40.5	44.2	39.7	26	-
	Side+synth	82.2	81.9	85.1	26	728
	Full	69.8	73.7	72.5	1250	-
	Full+synth	81.4	83.6	83.3	1250	728

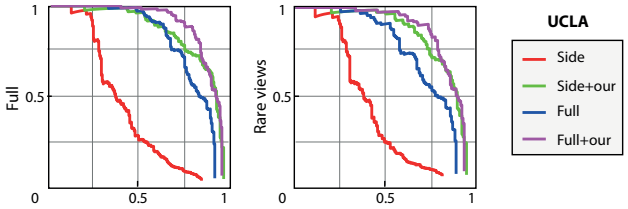

Figure 8. Performance of the "side" and "full" versions of training set with and without novel views synthesized by approach. Evaluation is performed on the full UCLA test set (left) as well as a subset of rare viewpoints (right).

In Tab. 3 and Fig. 8 we show the performance plots of the DPM detectors trained on the same dataset as in the previous section. For the full UCLA test set (upper half of Table 3) we observe drastic improvements by adding our synthesized data. For the sideviews the improvement is between 38.9 and 44.3% and for the full set between 6.9 and 9.6%. Remarkably, our detector trained on the amplified 26 sidesviews (Side+synth) outperforms the model trained on the full PASCAL dataset (Full) which had access to 48 times more real training examples (26 sideviews vs. 1250 real examples). The improvements can be seen more clearly in Fig. 8. We also provide the rare viewpoint analysis for this dataset. Here the improvements are even more pronounced. From the precision recall curve we see that the model trained on the amplified sideviews (Side+synth) is already approaching the model trained on the full set with our sideviews. In this case 26 real training examples in combination with our amplification method is enough to get close to the best performance on this data (Full+ours).

4.2. 2D-to-3D conversion

2D-to-3D conversion is a special case of novel-view synthesis, where the two novel views are the cameras of a stereo rig and the original image is the cyclopean view [23]. We demonstrate binocular stereo images created from 2D images of cars from an Internet site selling used cars in Fig. 9.

Figure 9. Conversion of a low-resolution 2D image *(Left)* into a high-quality anaglyph stereo image *(Right)*.

4.3. Inpainting

As we use our inpainting to fill holes due to disocclusions, we can achieve general inpainting, just by marking regions as disocclusions and performing the inpainting there. Fig. 10 shows examples of removing unwanted parts of an image in front of a car compared with another inpainting approach.

4.4. Super-resolution

Our formulation easily allows to reconstruct f_2 at a higher resolution than f_1 just by rendering p_2, n_2 and r_2 in an arbitrary resolution. This allows for super-resolution from a very coarse image of a car, as shown in Fig. 11.

4.5. Feature-aware smoothing

In a similar spirit, we can construct a special non-local smoothing filter [3], which uses our guide distance for

Figure 10. Inpainting of red scribbles that remove entire parts of the car such as the wheel *(Left)*. Using OpenCV inpainting does not preserve structures *(Middle)*. Our reconstruction *(Right)* repairs entire structures, such as the wheel while preserving consistent appearance.

Figure 11. Super-resolution from an input image *(Top)* using our approach *(Bottom)* for three different resolutions 20×8 *(Left)*, 40×16 *(Middle)* and the original size 160×128 *(Right)*. Note how even small features such as the the break light upsample to plausible structures.

computing the weighting of samples. While the original non-local means uses the image itself to infer locally similar patches, we can use the guide signals to do so (Figure Fig. 10).

Figure 12. Smoothing of an image that contains noise and distortions using *(Left)* using Adobe Photoshop *(Middle)* and our approach *(Right)*. While subtle echoes of the distortions are visible, the overall appearance is much more plausible. Note, that distortions are unknown while they are known for inpainting Fig. 10.

5. Discussion

While our results focus on well-aligned cars, our approach is also applicable to other classes such as airplanes or ships (Fig. 13). The challenge remains alignment, where errors lead to false matches between virtual and real information, *e.g.* at the airplane wings that never fit the template perfectly. Consequently, parts of the sky appear on the airplane and vice versa. Replacing the weighting in Eq. 1 with robust statistics might overcome such difficulties. Handling of specular and transparent surface appearance could be naturally incorporated in Eq. 1. Our results are perceptually plausible as they combine *consistency* with *details* while absence of either appears artificial. We avoid re-synthesizing appearance and reproduce also complex effects like global illumination to remain consistent with the context, even when hallucinating details in disocclusions.

6. Conclusion

We have presented a novel method for viewpoint synthesis and resynthesis. In particular, we address the challenging problem of filling in disocclusion areas. The results are visually pleasing and have shown useful in a series of applications ranging from dataset amplification for improved recognition across viewpoints, denoising, inpainting, 2D-3D-conversion and super-resolution.

Acknowledgement: This work was funded by the ERC grand Cognimund.

References

[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *SIGGRAPH*, 2009.

[2] T. Beier and S. Neely. Feature-based image metamorphosis. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 35–42, 1992.

[3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65. IEEE, 2005.

[4] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Proc. SIGRAPH*, pages 279–288. ACM, 1993.

[5] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *CVPR*, volume 2, pages II–721. IEEE, 2003.

[6] P. Debevec, Y. Yu, and G. Borshukov. *Efficient view-dependent image-based rendering with projective texture-mapping*. Springer, 1998.

Figure 13. View synthesis of planes and ships *(originals in 1st row)*.

[7] M. Deering, S. Winner, B. Schediwy, C. Duffy, and N. Hunt. The triangle processor and normal vector shader: a VLSI system for high performance graphics. *SIGGRAPH Comput. Graph.*, 22(4):21–30, June 1988.

[8] M. Enzweiler and D. M. Gavrila. A mixed generative-discriminative framework for pedestrian classification. In *CVPR*, 2008.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2010.

[11] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. SIGGRAPH*, pages 43–54. ACM, 1996.

[12] T. Hassner. Viewing real-world faces in 3D. In *ICCV*, 2013.

[13] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. SIGGRAPH*, Proc. SIGGRAPH, pages 327–340. ACM, 2001.

[14] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[15] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proc. SIGGRAPH*, pages 225–232. ACM, 1997.

[16] W. Hu. Learning 3D object templates by hierarchical quantization of geometry and appearance spaces. In *CVPR*, pages 2336–43, 2012.

[17] A. Jain, T. Thormählen, T. Ritschel, and H.-P. Seidel. Material memex: automatic material suggestions for 3d objects. *ACM Trans. Graph.*, 31(6):143:1–143:8, Nov. 2012.

[18] B. Kaneva, A. Torralba, and W. Freeman. Evaluation of image features using a photorealistic virtual world. In *ICCV*, 2011.

[19] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26(3):96, 2007.

[20] K. Lai and D. Fox. 3D laser scan classification using web data and domain adaptation. In *Proceedings of Robotics: Science and Systems*, 2009.

[21] W. Li and M. Fritz. Recognizing materials from virtual examples. In *ECCV*, 2012.

[22] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010.

[23] B. Mendiburu. *3D movie making: stereoscopic digital cinema from script to screen*. Focal Press, 2009.

[24] M. Ozuysal, V. Lepetit, and P.Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[25] M. Pauly, N. J. Mitra, J. Giesen, M. H. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symp. on Geometry Processing*, pages 23–32, 2005.

[26] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. In *Proc. SIGGRAPH*, pages 55–64, 2000.

[27] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D Geometry to Deformable Part Models. In *CVPR*, 2012.

[28] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormaehlen, and B. Schiele. Learning people detection models from few training samples. In *CVPR*, 2011.

[29] L. Pishchulin, A. Jain, C. Wojek, T. Thormaehlen, and B. Schiele. In good shape: Robust people detection based on appearance and shape. In *British Machine Vision Conference (BMVC)*, September 2011.

[30] V. A. Prisacariu, A. V. Segal, and I. Reid. Simultaneous monocular 2d segmentation, 3d pose recovery and 3d reconstruction. In *ACCV*, 2012.

[31] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipmanand, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011.

[32] P.-P. J. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: A model for capturing npr shading from art. In *Graphics interface*, pages 143–150, 2001.

[33] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVC*, 2010.

[34] A. T. Targhi, J. M. Geusebroek, and A. Zisserman. Texture classification with minimal training images. In *ICPR*, 2008.

[35] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr. Videotrace: Rapid interactive scene modelling from video. In *Proc. SIGGRAPH*, 2007.

[36] W. Wohlkinger and M. Vincze. 3D object classification for mobile robots in home-environments using web-data. In *Int. Conf. on Cognitive Systems*, 2010.

[37] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3D photography. In *Proc. SIGRAPH*, pages 287–296. ACM, 2000.

[38] M. Zobel, M. Fritz, and I. Scholz. Object tracking and pose estimation using light-field object models. In *Vision, Modeling, and Visualization Conference (VMV)*, 2002.