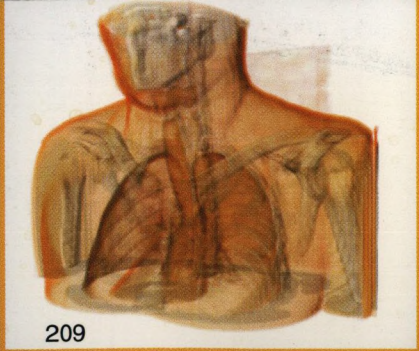


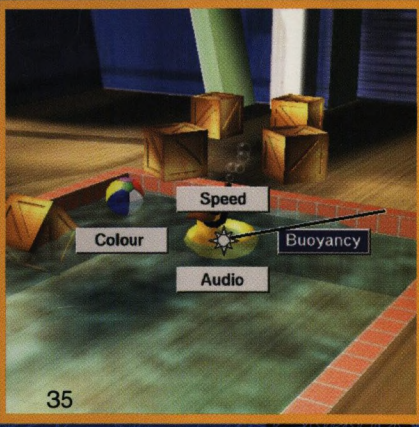
209



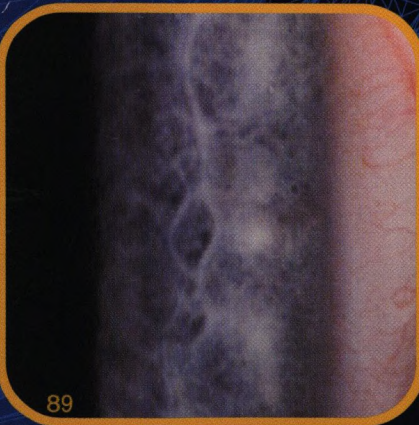
99



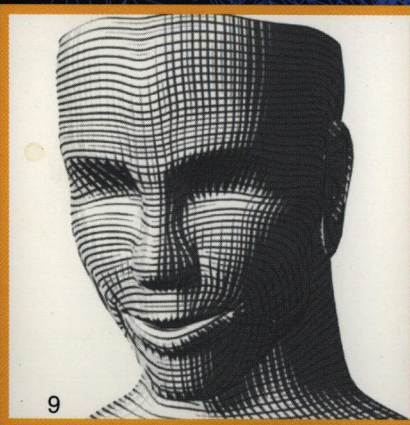
35



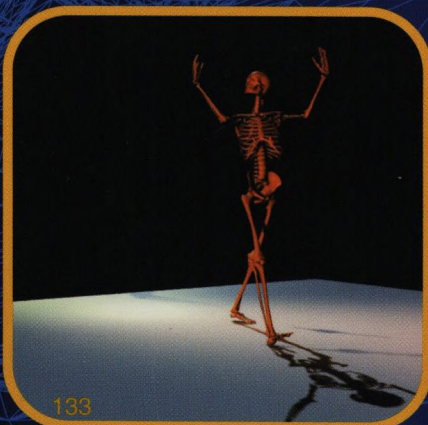
89



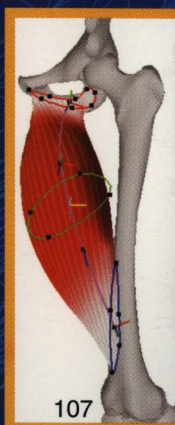
9



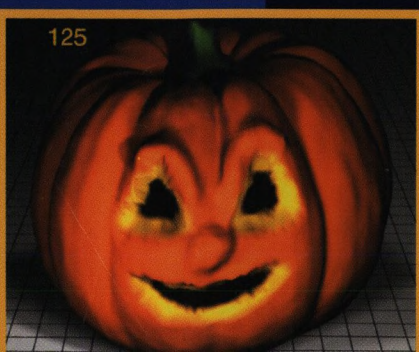
133



107



125



Proceedings Graphics Interface 2002

27-29 May 2002

Calgary, Alberta

Canadian Human-Computer
Communications Society

```

if defined(HAVE_GL_GLEXT_H) ** defined(USE_NV20) ** defined(USE_TEXTURE_SHADER)
glActiveTextureARB(GL_TEXTURE0_ARB);
glEnable(GL_TEXTURE_2D);
#ifdef USE_HILO
glBindTexture(GL_TEXTURE_2D, gid_ctx->dmap->normal_hilo_bind);
#else
glBindTexture(GL_TEXTURE_2D, gid_ctx->dmap->normal_bind);
#endif /* USE_HILO */
glActiveTextureARB(GL_TEXTURE2_ARB);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, gid_ctx->lightmap_bind);
glActiveTextureARB(GL_TEXTURE3_ARB);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, gid_ctx->omap_bind);
glClientActiveTextureARB(GL_TEXTURE0_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(3, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, u));
glClientActiveTextureARB(GL_TEXTURE1_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(3, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, v));
glClientActiveTextureARB(GL_TEXTURE2_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(3, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, b));
glClientActiveTextureARB(GL_TEXTURE3_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(3, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, r));
glCombiners_NV20_3 Texture();
glDrawArrays(GL_TRIANGLES, 0, 3*vb->count);
#endif /* HAVE_GL_GLEXT_H ** USE_NV20 ** USE_TEXTURE_SHADER */
}
static void __gldRender_NV20_3 Specular(gid_ctx_t *ctx, gid_vertex_t *vb)
{
if defined(HAVE_GL_GLEXT_H) ** defined(USE_NV20) ** defined(USE_TEXTURE_SHADER)
glActiveTextureARB(GL_TEXTURE0_ARB);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, gid_ctx->dmap->normal_hilo_bind);
glClientActiveTextureARB(GL_TEXTURE1_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(2, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, u));
glCombiners_NV20_3 Texture();
glDrawArrays(GL_TRIANGLES, 0, 3*vb->count);
#endif /* HAVE_GL_GLEXT_H ** USE_NV20 ** USE_TEXTURE_SHADER */
}
static void __gldRender_NV20_3 Diffuse(gid_ctx_t *ctx, gid_vertex_t *vb)
{
if defined(HAVE_GL_GLEXT_H) ** defined(USE_NV20) ** defined(USE_TEXTURE_SHADER)
glActiveTextureARB(GL_TEXTURE0_ARB);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, gid_ctx->dmap->normal_hilo_bind);
glClientActiveTextureARB(GL_TEXTURE1_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(2, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, u));
glCombiners_NV20_3 Texture();
glDrawArrays(GL_TRIANGLES, 0, 3*vb->count);
#endif /* HAVE_GL_GLEXT_H ** USE_NV20 ** USE_TEXTURE_SHADER */
}
static void __gldRender_NV20_3 Specular(gid_ctx_t *ctx, gid_vertex_t *vb)
{
if defined(HAVE_GL_GLEXT_H) ** defined(USE_NV20) ** defined(USE_TEXTURE_SHADER)
glActiveTextureARB(GL_TEXTURE0_ARB);
glEnable(GL_TEXTURE_2D);
glBindTexture(GL_TEXTURE_2D, gid_ctx->dmap->normal_hilo_bind);
glClientActiveTextureARB(GL_TEXTURE1_ARB);
glEnableClientState(GL_TEXTURE_COORD_ARRAY);
glTexCoordPointer(2, GL_FLOAT, GLD_VERTEX_STRIDE, vb->data+offsetof(gid_vertex_t, u));
glCombiners_NV20_3 Texture();
glDrawArrays(GL_TRIANGLES, 0, 3*vb->count);
#endif /* HAVE_GL_GLEXT_H ** USE_NV20 ** USE_TEXTURE_SHADER */
}
}

```



Proceedings

Graphics **Interface** 2002

Wolfgang Stürzlinger and Michael D. McCool
Program Co-Chairs

www.graphicsinterface.org

Calgary, Alberta
27-29 May 2002



Copyright © 2002 by the Canadian Information Processing Society

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

Papers are reproduced here from electronic files prepared by the authors.

ISSN 0713-5424
ISBN 1-56881-183-7

Proceedings Graphics Interface 2002, Wolfgang Stürzlinger and Michael D. McCool (Program Co-Chairs), Calgary, Alberta, 27–29 May 2002. Published by the Canadian Human-Computer Communications Society and A K Peters Ltd.

Graphics Interface is sponsored by:
The Canadian Human-Computer Communications Society (CHCCS)

Membership Information for CHCCS is available from:

Canadian Information Processing Society (CIPS)
2800 Skymark Avenue, Suite 402
Mississauga, Ontario L4W 5A6
Canada
Telephone: (905) 602-1370
Fax: (905) 602-7884
Web: <http://www.cips.ca/>

Additional copies of the proceedings are available from:

A K Peters Ltd.
63 South Avenue
Natick, MA 01760
Web: <http://www.akpeters.com/>

Published by the Canadian Human-Computer Communications Society and A K Peters Ltd.
Distributed by A K Peters Ltd.
Printed in Canada by Graphics Services at the University of Waterloo, Waterloo, Ontario.

Cover Credits

Wireframes: Front cover, Kevin Moule. Back cover: Marco Tarini, Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel (89); Hendrik Kück, Christian Vogelgsang, and Günther Greiner (81). Computer Program: Kevin Moule. Colour images: Left to right (starting on back left): Yefei He, James Cremer, and Yiannis Papelis (151); Hendrik Kück, Christian Vogelgsang, and Günther Greiner (81); Eric Paquette, Pierre Poulin, and George Drettakis (59); Brendan Lane and Przemyslaw Prusinkiewicz (69); Marco Tarini, Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel (89); Oleg Veryovka (9); Sageev Oore, Demetri Terzopoulos, and Geoffrey Hinton (133); Victor Ng-Thow-Hing and Eugene Fiume (107). Top to bottom (skipping joint): Michael Meißner, Stefan Guthe, and Wolfgang Straßer (209); John McDonald, Karen Alkoby, Roymieco Carter, Juliet Christopher, Mary Jo Davidson, Dan Ethridge, Jacob Furst, Damien Hinkle, Glenn Lancaster, Lori Smallwood, Nedjla Ougouag-Tiouririne, Jorge Toro, Shuang Xu, and Rosalee Wolfe (99); Michael McGuffin, Nicolas Burtnyk, and Gordon Kurtenbach (35); Robert Jagnow and Julie Dorsey (125). Design: Christine Goucher and Abby Van Dongen.

Message from the Program Co-Chairs

Wolfgang Stürzlinger
Department of Computer Science
York University

Michael D. McCool
School of Computer Science
University of Waterloo

Welcome to Graphics Interface (GI) 2002, a conference that combines coverage of original research results in both Human-Computer Interaction and Graphics. The conference took place in Calgary, Alberta, over 27–29 May 2002, and was held in conjunction with the Artificial Intelligence 2002 and Vision Interface 2002 conferences. GI 2002 is the 28th instance of the longest running conference series in human-computer interaction and computer graphics. This event has previously been held two times in Calgary: in 1977 and in 1991.

The program co-chairs received 96 submissions in all areas of human-computer interaction and computer graphics. The large number of submissions was a pleasant surprise; however, the number of submissions also necessitated an enlargement of the program committee on short notice. We are very grateful to all the additional program committee members who agreed to take time out of their busy schedules late in 2001. By agreeing to serve, these additional members reduced the workload on other committee members and permitted us to maintain our high reviewing standards.

The overall quality of the submissions was very high, which made the selection process difficult. After considerable deliberation, the program committee selected 25 papers for publication. The international program committee consisted of 20 people from around the world. Each paper received at least 4 reviews, two of which were from members of the program committee. The reviewing process was double-blind, and the identity of the authors was known only to program co-chairs and the program committee member responsible for choosing external reviewers for each submission. The program committee members were often able to solicit reviews from some of the topmost experts in a particular area of research. We greatly appreciate the effort of the members of the program committee. We would like to extend additional thanks to the 13 members of the program committee who attended the meeting at York University, Toronto, Canada on 16 February 2002 and funded their own travel.

Graphics Interface customarily has several invited speakers. This year three invited speakers were: Saul Greenberg, Professor at the University of Calgary; John Buchanan, Electronic Arts Canada and Adjunct Professor at the University of Alberta; and David Kirk, Chief Scientist at NVIDIA. We extend our gratitude to them for sharing their inspiration in their respective fields.

We would like to thank James Stewart for his work on the PCS electronic submission and reviewing system. His help behind the scenes made our job a lot easier! We would also like to thank Pierre Poulin and Kelly Booth for handling the liaison with AI and VI conference organizers, and Kelly Booth again for additional valuable advice. We thank further all referees for their voluntary work, Ravin Balakrishnan for handling the posters, Sara Diamond for organizing the video show, and Fred Peet, treasurer of the Canadian Human-Computer Communication Society, for keeping the finances straight, and Graphics Services at the University of Waterloo for doing such an excellent job on the proceedings. Last but not least, we send a very big thanks to Camille Sinanan, Sheelagh Carpendale, Mario Costa-Sousa, and Joerg Denzinger for the local organization of the joint conferences at the University of Calgary. Without their work, this conference would simply not have been possible.

For further information about the conference series we invite you to visit our web site:

<http://www.graphicsinterface.org/>

Organization

Conference and Program Chairs

Wolfgang Stürzlinger, York University
Michael McCool, University of Waterloo

Local Organizers

Sheelagh Carpendale, University of Calgary
Mario Costa-Sousa, University of Calgary
Joerg Denzinger, University of Calgary
Camille Sinanan, University of Calgary

Posters Chair

Ravin Balakrishnan, University of Toronto

Videos Chair

Sara Diamond, Banff Arts Centre

Student Volunteer Organizer

Oliver St. Cyr, University of Waterloo

Online Services

James Stewart, Queen's University

Proceedings Editor

Michael McCool, University of Waterloo

Program Committee

Ravin Balakrishnan, University of Toronto
Gladimir Baranoski, University of Waterloo
Thomas Ertl, University of Stuttgart
Michael Garland, University of Illinois at Urbana-Champaign
Craig Gotsman, Technion
Carl Gutwin, University of Saskatchewan
Christopher Healey, North Carolina State University
Wolfgang Heidrich, University of British Columbia
Jessica Hodgins, Carnegie Mellon University
Ted Kirkpatrick, Simon Fraser University
Yoshifumi Kitamura, Osaka University
Paul Lalonde, Electronic Arts Canada
Hanspeter Pfister, MERL Cambridge Research
Pierre Poulin, Université de Montréal
Holly Rushmeier, IBM T.J. Watson Research Center
Peter-Pike Sloan, Microsoft Research
Thomas Strothotte, University of Magdeburg
Michiel van de Panne, University of British Columbia

CHCCS Treasurer

Fred G. Peet, Canadian Forest Service

Referees

Johnny Accot	George Fitzmaurice	Paolo Magillo	Annette Scheel
John Amanatides	Eugene Fiume	Tom Malzbender	Eric Schenk
Nina Amenta	Michael Garland	Regan Mandryk	Stefan Schlechtweg
Alexis Angelidis	Michael Gleicher	Dinesh Manocha	Alla Sheffer
Ian Ashdown	Stephane Gobron	Steve Marschner	Mikio Shinya
Ravin Balakrishnan	Michael Goesele	Masood Masoodian	Peter Shirley
Gladimir Baranoski	Craig Gotsman	Toshiyuki Masui	Maryann Simmons
Gill Barequet	François Guimbretiere	Nelson Max	Peter-Pike Sloan
Lyn Bartram	Stefan Gumhold	Dave McAllister	Michiel Smid
Rui Bastos	Damrong Guoy	Michael McCool	A. Augusto Sousa
Thomas Baudel	Igor Guskov	Joanna McGrenere	Milos Sramek
Ben Bederson	Carl Gutwin	Ann McNamara	Robert St. Amant
Ian Bell	Jörg Haber	Michael Meißner	Marc Stamminger
Thomas Berlage	James Hahn	Stan Melax	James Stewart
Jules Bloomenthal	Eric Haines	Mark Meyer	Lisa Streit
Kellogg Booth	Jason Harrison	Tim Miller	Norbert Streitz
Kadi Bouatouch	John Hart	Jun-yong Noh	Thomas Strothotte
Stefan Brabec	Knut Hartmann	Haruo Noma	Wolfgang Stürzlinger
Chris Bregler	Helwig Hauser	Manuel M. Oliveira	Kelvin Sung
John Buchanan	Li-wei He	Dan Olsen	Haruo Takemura
Chris Buehler	Christopher Healey	Sageev Oore	Roger Tam
Tom Calvert	Wolfgang Heidrich	Victor Ostromoukhov	Hai Tao
Sheelagh Cappendale	Ken Hinckley	Eric Paquette	Russell Taylor
Anthony Chee-Hung	Koichi Hirota	Sergey Parilov	Greg Turk
Fang	Jessica Hodgins	Valerio Pascucci	Michiel van de Panne
Min Chen	Kenneth Hoff	Mark Pauly	Nicolas Vidot
Yiorogs Chrysanthou	Masahiko Inami	Hanspeter Pfister	Vadym Voznyuk
Mary Czerwinski	Poorang Irani	Jeff Pierce	Andrew Walenstein
Luiz Henrique de	Jan Kautz	Nancy Pollard	Jason Walter
Figueiredo	Ted Kirkpatrick	Jean Ponce	Marcelo Walter
Leila De Floriani	Yoshifumi Kitamura	Stuart Pook	Justin Wan
Ralph Deters	Reinhard Klein	Pierre Poulin	Greg Ward
Oliver Deussen	James Klosowski	Gilles Pouliquen	Benjamin Watson
Prasun Dewan	Leif Kobbelt	Simon Premoze	Ruediger Westermann
Jean-Michel Dischler	Kevin Kreeger	Werner Purgathofer	Stephen Westin
Michael Doggett	Tomohiro Kuroda	Andreas Raab	Jane Wilhelms
Julie Dorsey	Gordon Kurtenbach	Patrick Ratto	Craig Wittenbrink
Sarah Douglas	Hideaki Kuzuoka	Soraia Raupp Musse	David Wong
Steven Drucker	David Laidlaw	Lionel Reveret	Yin Wu
Mark Duchaineau	Paul Lalonde	Christof Rezk-Salama	Granier Xavier
Fredo Durand	Hugh Lauer	Penny Rheingans	Yasuyuki Yanagida
Phil Dutre	Jed Lengyel	Theresa-Marie Rhyne	Michal Young
Gershon Elber	Hendrik Lensch	Thomas Rist	Michael Young
Thomas Ertl	Peter Lindstrom	Stefan Roettger	Qinxin Yu
Irfan Essa	David Luebke	Jon Rokne	Yizhou Yu
Anath Fischer	I. Scott MacKenzie	Jarek Rossignac	Matthias Zwicker
		Holly Rushmeier	

Table of Contents

Non-Photorealistic and Image-Based Rendering

<i>Layered Environment-Map Impostors for Arbitrary Scenes</i>	1
Stefan Jeschke, Michael Wimmer, and Heidrunn Schumann	
<i>Animation with Threshold Textures</i>	9
Oleg Veryovka	
<i>A Fresh Perspective</i>	17
Karan Singh	

Interaction

<i>Constraint-Based Automatic Placement for Scene Composition</i>	25
Ken Xu, James Stewart, and Eugene Fiume	
<i>FaST Sliders: Integrating Marking Menus and the Adjustment of Continuous Values</i>	35
Michael McGuffin, Nicolas Burtnyk, and Gordon Kurtenbach	
<i>Traces: Visualizing the Immediate Past to Support Group Interaction</i>	43
Carl Gutwin	

Natural Phenomena

<i>Image-Based Hair Capture by Inverse Lighting</i>	51
Stéphane Grabli, François X. Sillion, Stephen R. Marschner, and Jerome E. Lengyel	
<i>The Simulation of Paint Cracking and Peeling</i>	59
Eric Paquette, Pierre Poulin, and George Drettakis	
<i>Generating Spatial Distributions for Multilevel Models of Plant Communities</i>	69
Brendan Lane and Przemyslaw Prusinkiewicz	
<i>Simulation and Rendering of Liquid Foams</i>	81
Hendrik Kück, Christian Vogelgsang, and Günther Greiner	

The Body

<i>Texturing Faces</i>	89
Marco Tarini, Hitoshi Yamauchi, Jörg Haber, and Hans-Peter Seidel	
<i>A Direct Method for Positioning the Arms of a Human Model</i>	99
John McDonald, Karen Alkoby, Roymieco Carter, Juliet Christopher, Mary Jo Davidson, Dan Ethridge, Jacob Furst, Damien Hinkle, Glenn Lancaster, Lori Smallwood, Nedjla Ougouag-Tiouririne, Jorge Toro, Shuang Xu, and Rosalee Wolfe	
<i>Application-Specific Muscle Representations</i>	107
Victor Ng-Thow-Hing and Eugene Fiume	

Input Devices

<i>A Model of Two-Thumb Text Entry</i>	117
I. Scott MacKenzie and R. William Soukoreff	
<i>Virtual Sculpting with Haptic Displacement Maps</i>	125
Robert Jagnow and Julie Dorsey	
<i>A Desktop Input Device and Interface for Interactive 3D Character Animation</i>	133
Sageev Oore, Demetri Terzopoulos, and Geoffrey Hinton	
<i>Laser Pointers as Collaborative Pointing Devices</i>	141
Ji-Young Oh and Wolfgang Stuerzlinger	

Surfaces and Meshes

<i>Real-Time Extendible-Resolution Display of On-Line Dynamic Terrain</i>	151
Yefei He, James Cremer, and Yiannis Papelis	
<i>Compressing Polygon Mesh Connectivity with Degree Duality Prediction</i>	161
Martin Isenburg	
<i>Efficient Bounded Adaptive Tessellation of Displacement Maps</i>	171
Kevin Moule and Michael D. McCool	
<i>Automatic Generation of Subdivision Surface Head Models from Point Cloud Data</i>	181
Won-Ki Jeong, Kolja Kähler, Jörg Haber, and Hans-Peter Seidel	

Reflectance and Lighting

<i>A BRDF Database Employing the Beard-Maxwell Reflection Model</i>	189
Harold B. Westlund and Gary W. Meyer	
<i>Coherent Bump Map Recovery from a Single Texture Image</i>	201
Jean-Michel Dischler, Karl Maritaud, and Djamchid Ghazanfarpour	
<i>Interactive Lighting Models and Pre-Integration for Volume Rendering on PC Graphics Accelerators</i>	209
Michael Meißner, Stefan Guthe, and Wolfgang Straßer	
<i>Single Sample Soft Shadows Using Depth Maps</i>	219
Stefan Bräbec and Hans-Peter Seidel	

Invited Speakers

<i>Video Games: What the heck?</i>	229
John W. Buchanan	
<i>Rapid Prototyping of Physical User Interfaces</i>	231
Saul Greenberg	
<i>How Long can Graphics Chips Exceed Moore's Law?</i>	235
David Kirk	

Video Games: What the heck?

John W. Buchanan
Electronic Arts Canada

The video game industry is growing up. Over the last 7 years the content delivery media for games has gone from 64 megabytes to 9 gigabytes. The processors and associated graphics cards have increased in power. It is now feasible to render fairly realistic characters. It is also realistic to render cartoon characters. In short we can now focus on the story and the experience rather than on the technology. We can start to answer the question, what is an intelligent avatar?

It used to be the case that the game engine was the crucial part of a game. This is no longer the case; the task of building an efficient game engine is well understood. The primary task before a game team is now content production. Thus the engineering task in game building is focused on tools for the artist and the art pipeline that bakes the data for the game. What are the tools that we need? Well, we basically need the same tools that the movie industry needs, and then some. We need to manage transition trees between animations that may have 10000 transitions in them. We need to blend between arbitrary frames on animations. We need our artist to be able to instantly see his/her work on the target platform.

In this talk I give a historical perspective of the games industry. I talk about the changes that have occurred over the last 7 years and I talk about our future directions and what our biggest technical challenges are.



How Long can Graphics Chips Exceed Moore's Law?

David Kirk
NVIDIA

A few short years ago, single-chip PC 3D graphics solutions arrived on the market at performance levels that rivaled professional workstations with multi-chip graphics pipelines. Since then, graphics performance has grown at a rate approaching doubling every 6 months, far exceeding Moore's Law. How is this possible? Will it be sustainable? There is evidence that this geometric performance growth is not only possible, but inevitable.



Author Index

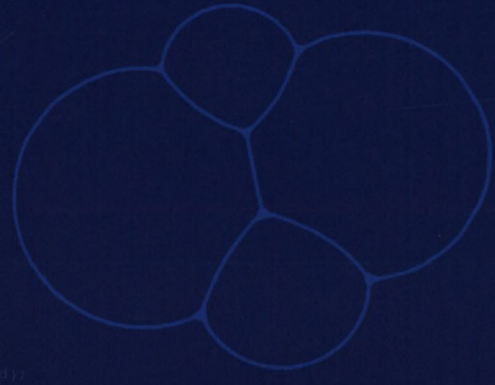
Alkoby, Karen	99	McCool, Michael D.	171
Brabec, Stefan	219	McDonald, John	99
Burtnyk, Nicolas	35	McGuffin, Michael	35
Carter, Roymieco	99	Meißner, Michael	209
Christopher, Juliet	99	Meyer, Gary W.	189
Cremer, James	151	Moule, Kevin	171
Davidson, Mary Jo	99	Ng-Thow-Hing, Victor	107
Dischler, Jean-Michel	201	Oh, Ji-Young	141
Dorsey, Julie	125	Oore, Sageev	133
Drettakis, George	59	Ougouag-Tiouririne, Nedjla	99
Ethridge, Dan	99	Papelis, Yiannis	151
Fiume, Eugene	25,107	Paquette, Eric	59
Furst, Jacob	99	Poulin, Pierre	59
Ghazanfarpour, Djamchid	201	Prusinkiewicz, Przemyslaw	69
Grabli, Stéphane	51	Schumann, Heidrun	1
Greiner, Günther	81	Seidel, Hans-Peter	89,181,219
Guthe, Stefan	209	Sillion, François X.	51
Gutwin, Carl	43	Singh, Karan	17
Haber, Jörg	89,181	Smallwood, Lori	99
He, Yefei	151	Soukoreff, R. William	117
Hinkle, Damien	99	Stewart, James	25
Hinton, Geoffrey	133	Straßer, Wolfgang	209
Isenburg, Martin	161	Stuerzlinger, Wolfgang	141
Jagnow, Robert	125	Tarini, Marco	89
Jeong, Won-Ki	181	Terzopoulos, Demetri	133
Jeschke, Stefan	1	Toro, Jorge	99
Kähler, Kolja	181	Veryovka, Oleg	9
Kück, Hendrik	81	Vogelsgang, Christian	81
Kurtenbach, Gordon	35	Westlund, Harold B.	189
Lancaster, Glenn	99	Wimmer, Michael	1
Lane, Brendan	69	Wolfe, Rosalee	99
Lengyel, Jerome E.	51	Xu, Ken	25
MacKenzie, I. Scott	117	Xu, Shuang	99
Maritaud, Karl	201	Yamauchi, Hitoshi	89
Marschner, Stephen R.	51		




```

static void __glRender_NV15_Diffuse(qid_t *vb) {
    #ifdef HAVE_GL_GLEXT_H
    glActiveTextureARB(GL_TEXTURE0_ARB);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, __gl_ctx->deap->normal_bind);
    glClientActiveTextureARB(GL_TEXTURE0_ARB);
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);
    glVertexAttribPointer(3, GL_FLOAT, __gl_ctx->vertex_stride,
        vb->data_offset+qid_t::vertex_t, 1);
    __gl_ctx->glDrawArrays(GL_TRIANGLES, 0, 1*vb->count);
    #endif // HAVE_GL_GLEXT_H
}

```



```

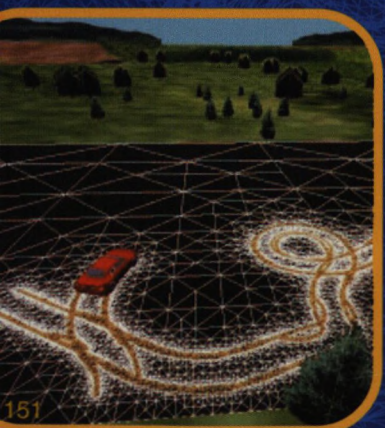
static void __glRender_NV15_Specular(qid_t *vb) {
    #ifdef HAVE_GL_GLEXT_H
    glActiveTextureARB(GL_TEXTURE0_ARB);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, __gl_ctx->deap->normal_bind);
    glClientActiveTextureARB(GL_TEXTURE0_ARB);
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);
    glVertexAttribPointer(3, GL_FLOAT, __gl_ctx->vertex_stride,
        vb->data_offset+qid_t::vertex_t, 1);
    __gl_ctx->glDrawArrays(GL_TRIANGLES, 0, 1*vb->count);
    #endif // HAVE_GL_GLEXT_H
}

```

```

static void __glRender_NV15_Specular(qid_t *vb) {
    #ifdef HAVE_GL_GLEXT_H
    glActiveTextureARB(GL_TEXTURE0_ARB);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, __gl_ctx->deap->normal_bind);
    glClientActiveTextureARB(GL_TEXTURE0_ARB);
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);
    glVertexAttribPointer(3, GL_FLOAT, __gl_ctx->vertex_stride,
        vb->data_offset+qid_t::vertex_t, 1);
    __gl_ctx->glDrawArrays(GL_TRIANGLES, 0, 1*vb->count);
    #endif // HAVE_GL_GLEXT_H
}

```



151



81



59



69

```

static void __glRender_NV20_2(s)R_zbuffer_t *vb) {
    #ifdef HAVE_GL_GLEXT_H
    glActiveTextureARB(GL_TEXTURE0_ARB);
    glEnable(GL_TEXTURE_2D);
    glBindTexture(GL_TEXTURE_2D, __gl_ctx->deap->normal_bind);
    glClientActiveTextureARB(GL_TEXTURE0_ARB);
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);
    glVertexAttribPointer(3, GL_FLOAT, __gl_ctx->vertex_stride,
        vb->data_offset+qid_t::vertex_t, 1);
    __gl_ctx->glDrawArrays(GL_TRIANGLES, 0, 1*vb->count);
    #endif // HAVE_GL_GLEXT_H && USE_NV20
}

```

www.graphicsinterface.org
www.akpeters.com



A K PETERS LTD.

ISSN 0713-5424
ISSN 1-56881-183-7

ISBN 156881183-7

9 781568 811833