TESIS DE LA UNIVERSIDAD
DE ZARAGOZA

2012

27

Fernando Navarro Gil

# Optimization techniques for computationally expensive rendering algorithms

Departamento

Informática e Ingeniería de Sistemas

Director/es

Gutiérrez Pérez, Diego
Serón Arbeloa, Francisco

ISSN 2254-7606

Prensas de la Universidad
Universidad Zaragoza

Tesis Doctoral

# OPTIMIZATION TECHNIQUES FOR COMPUTATIONALLY EXPENSIVE RENDERING ALGORITHMS

Autor

Fernando Navarro Gil

Director/es

Gutiérrez Pérez, Diego
Serón Arbeloa, Francisco

**UNIVERSIDAD DE ZARAGOZA**
Informática e Ingeniería de Sistemas

2012

Universidad Zaragoza

1542

Escuela de Ingeniería y Arquitectura
Universidad Zaragoza

1542

# OPTIMIZATION TECHNIQUES FOR COMPUTATIONALLY EXPENSIVE RENDERING ALGORITHMS

LUIS FERNANDO NAVARRO GIL

Supervisors:
DR. DIEGO GUTIÉRREZ
DR. FRANCISCO JOSÉ SERÓN

Grupo de Informática Gráfica Avanzada
Dpto. de Ingeniería de Sistemas e Informática
Universidad de Zaragoza

April 2012

# ACKNOWLEDGMENTS

I have been lucky enough to meet exceptional people, whose knowledge and personal values are first-class. That is why there are many individuals that have been instrumental in this thesis and should share part of the merit.

The people from Lionhead Studios, Microsoft Games Studios and Telefonica I+D have been extremely welcoming, encouraging and have offered invaluable technical advises. A list of names will surely be incomplete, that is why I would rather thank all of them, instead of leaving a single one out.

The members of the Graphics and Imaging Lab of the University of Zaragoza, for their support, advises and insightful comments. Among them, I want to thank the directors of this thesis, Diego Gutiérrez and Francisco Serón, for accepting my proposal, steering my efforts and adapting their schedules to the pace imposed by my other compromises.

To my friends for their supportive attitude. Even after my frequently unspecific explanations, they managed to show interest and make fun of what I decided it was going to be a serious hobby for the next few years.

My parents, for patiently listening the details of what was occupying my spare time. They helped me see the bigger picture when my reports were limited to news of a somehow unclear advance. And to my brother, Joaquin and his fiancée Xenia, for making up amusing conversations that gave me the opportunity to think about so many other different topics.

Finally, to Alicia, who has provided limitless support during these years of work. She has being considerate and patient even when my time and attention were diverted. You understood my passion and figure out were the value of beautiful images reside.

# ABSTRACT

Advances in rendering technology allow handling complex materials, novel lighting methods, physics simulation and vast amounts of geometry. However, they frequently come at the cost of significant increments in the computational resources required to generate images. It is not infrequent to listen that, in modern productions, a single frame can take many hours to render. As glamorous as it may sound, this has a direct impact on production budgets and schedules.

This thesis focuses on a group of rendering methods known by their high computational requirements. We analyse them in detail and reduce their cost using a set of conceptually different approaches.

We first focus on rendering time-varying participating media. We propose a modified formulation of the rendering equation and implement several optimizations to the ray marching algorithm. Our GPU based framework can generate photo-realistic images using high dynamic range lighting at interactive rates.

We also analyse two different aspects of the generation of anti-aliased images.

The first one is targeted to rendering screen-space anti-aliasing and reducing image artifacts. We propose a real time implementation of the morphological antialiasing algorithm that is efficient to evaluate, has a moderate impact and can be easily integrated into existing pipelines.

The final part of the thesis takes a radically different approach and studies the responses of the Human Visual System to motion blurred stimuli. Using psychophysical experiments, we analyse the limits with respect to the perception of temporally antialiased images.

Results, both for standard sequences and stereoscopic footage, suggest that human observers have notable tolerance to image artifacts like strobbing, excessive blur and noise. In some cases, images rendered with low quality settings may be indistinguishable from a gold standard. Based on these insights, we provide examples of how render settings can be used to reduce computation times without degradation of visual quality.

In summary, this thesis describes novel algorithmic optimizations as well as introduces aspects related to human perception that can be leveraged to design more efficient rendering methods.

# RESUMEN

Avances en tecnología de render de imágenes permiten la manipulación de materiales complejos, nuevos métodos de iluminación, simulaciones basadas en física y cantidades ingentes de geometría. Sin embargo, estas mejoras vienen asociadas con incrementos significativos en los recursos computacionales necesarios para cada imagen. No es infrecuente escuchar que en producciones actuales, un solo frame puede requerir varias horas de calculo. Independientemente del interés que este tipo de informaciones puedan suscitar entre los aficionados, el impacto en los presupuestos y las planificaciones de producción no es desdeñable.

Esta tesis se centra en un grupo de métodos de render conocidos por sus altos requerimientos computacionales. A lo largo de este trabajo, los analizaremos en detalle con el objeto de reducir su coste mediante un conjunto de enfoques conceptualmente diferentes.

En primer lugar, nos centramos en la generación de imágenes a partir de medios participativos que varían en el tiempo. Proponemos una formulación alternativa de la ecuación de render e implementamos varias optimizaciones al algoritmo de ray-marching. Utilizando un framework basado en GPU, nuestro algoritmo es capaz de generar imágenes foto-realistas con iluminación de alto rango dinámico en tiempos interactivos.

También analizamos dos aspectos diferentes relacionados con la generación de imágenes con anti-aliasing.

El primero esta orientado al render de anti-aliasing en espacio de pantalla y la reducción de los errores asociados. Proponemos una implementación en tiempo real del algoritmo de antialiasing morfológico que es eficiente, tiene un impacto moderado y puede ser integrado fácilmente en pipelines existentes.

La parte final de esta tesis, toma una aproximación completamente diferente y estudia la respuesta del sistema visual humano. Utilizando experimentos basados en psicofísica, analizamos los limites perceptuales con respecto a un conjunto de imágenes que han sido generadas utilizando algoritmos de antialiasing temporal y que como consecuencia contienen motion blur.

Los resultados, tanto para secuencias estándar como para imágenes estereoscópicas, sugieren que los observadores humanos tienen una tolerancia notable con respecto a deficiencias como excesiva suavidad de la imagen, ruido o problemas estroboscópicos. En algunos casos, imágenes de baja calidad pueden ser indistinguibles de referencias de alta calidad. Basándonos en esto, detallamos ejemplos de como

ciertos parámetros de render pueden ser utilizados para reducir los tiempos de cálculo sin degradar la calidad percibida de las imágenes resultantes.

En resumen, esta tesis describe nuevas optimizaciones a algoritmos existentes además de introducir aspectos relacionados con la percepción humana. Los resultados derivados de ambos metodos pueden ser utilizados para diseñar métodos de render más eficientes.

# PUBLICATIONS AND MERITS

This thesis describes techniques and results previously exposed in several printed works. These ideas have generated three different articles in publications included in the Journal Citation Report (JCR) and one chapter in a peer-reviewed book:

Part i, Accelerating Volume Rendering, contains research published as:

> Fernando Navarro, Diego Gutiérrez, and Francisco J. Serón. Interactive HDR lighting of dynamic participating media. *Vis. Comput.*, 25(4):339–347, February 2009. Visual Computer is a JCR indexed publication with an impact factor of 0.919.

Part ii, Accelerating Antialiasing Rendering, is based on the results previously published on GPU Pro 2, a peer-reviewed book that is one of the main references in the state-of-the-art of real time graphics.

> Jorge Jiménez, Belen Masía, Jose I. Echevarría, Fernando Navarro, and Diego Gutiérrez. *GPU Pro 2*, chapter Practical Morphological Anti-Aliasing. AK Peters Ltd., 2011

My contribution was focused on providing guidance to the main authors during the design and implementation of the technique, as well as supporting the test phase under the constraints defined in a game company.

The MLAA technique has been reviewed in a SIGGRAPH 2011 course [106] and several international industry publications: Digital Foundry, Game Developers Magazine and GamesIndustry.biz. The method has been integrated in engines such as Mesa, Torque 3D and several commercial games. The front and back cover of the book was composed with images computed with the technique.

Part iii, Accelerating Motion Blur Rendering, includes a compendium of the results previously contained in two publications and a third one that is expected to be submitted in the near future.

> Fernando Navarro, Francisco J. Serón, and Diego Gutiérrez. Motion blur rendering: State of the art. *Computer Graphics Forum*, 30 (1):3–26, 2011. Computer Graphics Forum is part of the Journal Citation Report with an impact factor of 1.455.

> Fernando Navarro, Susana Castillo, Francisco J. Serón, and Diego Gutiérrez. Perceptual considerations for motion blur rendering.

*ACM Trans. Appl. Percept.*, 8(3):20:1–20:15, August 2011. Transactions on Applied Perception is also part of the Journal Citation Report with an impact factor of 1.447. Images of the publication were selected as the front cover of the journal.

Fernando Navarro, Susana Castillo, Francisco J. Serón, and Diego Gutiérrez. Motion blur perception and stereo rendering. To be submitted to ACM Transactions on Applied Perception.

There are some other relevant contributions that are indirectly related to the main topic of this thesis. They are listed below.

A second chapter included in a peer reviewed book on game pipelines:

Fernando Navarro. *Game Development Tools*, chapter Taming the Beast: Managing Complexity in Game Build Pipelines. AK Peters Ltd./CRC Press, 2011.

Simultaneously to the completion of this thesis, I have lead different R&D groups performing research on computer graphics and pipelines on different studios. The productions where they took part were awarded with the following distinctions:

Fable II, Bafta Awards 2009, Best Game. Senior Programmer.

Fable II, Bafta Awards 2009, Best Action and Adventure Game. Senior Programmer.

Donkey Xote, Goya Awards 2009, Nomination to the Best Animation Film. Credited as R&D Supervisor.

Nocturna, Goya Award 2008, Best Animated Film. Credited as R&D Supervisor.

Perez (The hairy tooth fairy), Goya Award, Best Animated Film, 2007. Credited as R&D Supervisor.

Gisaku, Goya Awards 2006, Nomination to the Best Animation Film. Credited as R&D Supervisor.

P3K: Pinocchio 3000, Goya Award 2005, Best Animated Film. Credited as R&D Supervisor.

El Cid: The Leyend. Goya Award 2004, Best Animated Film. Credited as R&D Supervisor.

La gran aventura de Mortadelo y Filemon, Goya Awards 2003, Best special effects. R&D Supervisor.

800 balas, Goya Awards 2002, Best special effects. R&D Supervisor.

Los otros, Goya Awards 2001, Nomination to the Best special effects. R&D Supervisor.

In association with Microsoft Games Studios, I have generated intellectual property that is the process of being patented:

IPCOM000195648D, Rig proportion tool: Tool to reproportion rigs and reuse animations. May 2010.

I have been lecturer at several conferences on game and visual effects development:

What they did not tell you about game development: Lessons learnt with Fable. VII MillaGame, Milla Digital, Ayuntamiento de Zaragoza. December 2011.

Current trends in game development, *and* How to get a job in a game company. MAaster in Video game production, Universidad de Zaragoza. December 2011.

Animation methods. Autodesk 3December launch party, London. December 2008.

Animation and rendering. Universidad de La Coruña. September 2005.

VFX in The Incredibles, Star Wars and The Lord of the rings: Why and how. Universidad de verano de Jaca. July 2005.

Management and Optimization in 3D production with proprietary tools: SoftBren. Mundos digitales, La Coruña. July 2005.

Development of a 3D production. Universidad de verano de Teruel. June 2005

3D Animation. Xuventude Galicia Net, Santiago de Compostela. April 2004.

Technical aspects in visual effects and animation. Universidad de Santiago de Compostela. October 2005

VFX creation and new technologies. Centro Cultural Ibercaja, Zaragoza. April 2005.

Engineers in animation and visual effects. JETAI 2003. Centro Politécnico Superior, Universidad de Zaragoza. March 2003.

Lecturer at several editions of the courses: Computer Graphics, User-machine interaction and Visual modeling and animation. Universidad de Zaragoza, between 2001 and 2006.

Two open source developments have attracted the attention of the professional visual effects community:

fnPeltMapper, first UV automatic unwrapping tool for Autodesk Softimage XSI. Based on [137]. 15000 downloads from highend3d.com.

fnDualQuaternion, first dual quaternion deformation plugin for Autodesk Softimage XSI. Based on [113]. 5000 downloads from highend3d.com

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# LISTINGS

# 1 INTRODUCTION

Computer graphics is a not an unexplored field anymore. Over the last decades, thousands of researchers have proposed new algorithms to extend the types of phenomena that can be accurately simulated. As a result, there are countless methods to handle complex materials, global illumination, novel lighting methods, physics based animation, complex motion and massive amounts of geometry.

Those advances frequently come at the cost of significant increments in the computational resources required to complete a render. It is not infrequent to listen that, in modern productions, a single frame can take many hours to render. Those times do not even include the time needed to build and configure the scenes themselves. The number of installed CPUs, the efficiency of the networks and the size of the data sets are flagships that set the level of technical achievement of the production. However, this unavoidable number crunching has a direct impact on the cost of the productions.

## 1.1 THE PROPOSALS

Concerned with this uncontrolled race, this thesis selects a set of rendering algorithms with obvious relevance to the industry and study them in detail.

We first target a set of real time methods. Our interest is to determine their respective strengths, limitations and, based on them, made proposals to improve their efficiency. The study results in a new method to render time varying participating media, like animated smoke and clouds, in real time. The algorithm performs high dynamic range illumination in a few miliseconds, something that was not described before.

In a second example, the author has teamed with a group of researchers to implement a real time post-process that computes antialiasing on a prerendered image. In this case, the contribution has been focused on providing guidance to the main authors during the design and implementation of the technique, as well as supporting the test phase under the constraints defined in a game company. The quality of the result and the efficiency of the implementation allows to include the method in an existing graphic pipeline. In a field in continuous evolution, our proposal competes with state-of-the-art methods accepted and implemented by the graphics community.

On a complementary study, we analyse another challenging problem: how to render motion blurred scenes within reasonable time ranges. Simulating motion blur rendering is one of the most expensive rendering processes and in many situations, it can only be performed offline. The cost of producing a single image is important, but it may be prohibitive when stereo footage is generated. The mathematical framework that is solved poses a series of difficulties that makes the proposal of efficient methods a very challenging task.

Our approach does not try to tackle these issues, but takes an alternative approach by leveraging the limitations of the Human Visual System. The question we tried to answer is: What are the limitations of the observers' visual system with respect to motion blur perception, and how can be uses to reduce the time to render temporally antialiased images? In simpler terms, can we simplify a render and still produce appealing images? The answer is yes, and in some cases, there is plenty of space to fine tune renders without affecting their visual quality.

Our research has been oriented to provide direct practical applications. The proposals include improvements that come hand in hand with better algorithms, but also, our suggestions try to make better allocation of the resources already available. In both cases, this thesis answers questions that may open the door to new methods to efficiently compute beautiful images.

## 1.2 STRUCTURE

This document has the following organization:

In Part i we present a real time framework capable of rendering volumetric data of participating media. We discuss existing offline and real time methods, suggest an alternative formulation of the rendering equation and describe a set of optimizations. As a result, high quality images can be efficiently rendered without requiring any pre-computations.

Part ii introduces a new algorithm that implements screen space antialiasing based on recovering and rasterizing the edges found on a prerendered image. We describe an implementation of morphological antialiasing that can be evaluated in real time, with very moderate memory and time consumption.

Part iii, includes an in-depth research focused on motion blur rendering methods. Chapter 4 contains a complete review of the existing algorithms to render motion blurred images. Chapter 5 studies the perceptual implications of using temporally antialiased footage and proposes several approaches to reduce computation times. In

Chapter 6 we extend this knowledge to stereo rendering and stereoscopic perception. The discussion contains insights that allow reducing the resources required to render temporally antialiased images, both monoscopic and stereo pairs.

The last section of the thesis, Part iv, provides an overview of our findings, summarizes the results and describes the main contributions. It finalises with a set of possible directions for further research.

Part I

# ACCELERATING VOLUME RENDERING

In this part of the thesis, we present a real time framework capable of rendering volumetric data of inhomogeneous participating media.

We extend a GPU implementation of the ray maything algorithm with two optimization techniques. The first one depends on the media while the second takes advantage of the position of the observer.

The method computes single scattering in time-varying isotropic participating media with the incident light field being modeled as a high dynamic range (HDR) environment map. Images can be generated at interactive rates, using dynamic lighting and free camera movement without relying on any pre-computations.

# 2 GPU ACCELERATED RENDERING OF PARTICIPATING MEDIA

Rendering participating media is one of the most computationally intensive processes in computer graphics. Light transport relies on the simulation of scattering, absorption and emission and is commonly based on the Radiative Transfer Equation (RTE) [38] [102]. However, this model is expensive to evaluate and in order to make it tractable, it needs to be approximated using simplified versions of the original phenomena.

Just a few approaches can efficiently solve the mathematical expressions associated with the RTE. Among them, the ray marching algorithm is probably the most popular. This method solves the multidimensional integral equations of the RTE incrementally. However, even in cases where it is accelerated using techniques such as volume photon mapping, it cannot be evaluated in real time. Alternatively, interactive rendering has relied on ad-hoc methods that rely on precomputation or impose severe restrictions to the underlying model. Except for a few exceptions, high quality results are reserved to offline frameworks.

In this chapter, we describe an interactive system to light and render participating media by solving an approximation of the Radiance Transfer Equation. We use two novel optimizations that allow evaluating the ray marching algorithm as GPU shaders executed as a series of passes. Interactive frame rates are achieved using a view dependent technique that reduces the number of pixels processed, and a view independent empty space skipping technique that allows speeding up volume traversals. The method is original as it uses HDR maps to define environment lighting. These are efficiently sampled by transforming them into a light constellation.

In our method, participating media is characterized as volumetric data obtained by sweeping a laser sheet through a volume of real smoke using the method of Hawkins, Einarsson and Debevec [93]. However, since it is general enough, it has also been applied to data generated using numerical simulation.

None of these steps are required to be precomputed nor are based on predetermined illumination models. As a consequence, the system can handle time-varying participating media, single scattering including self shadowing, free camera movement and dynamic lighting where the incident light field can be arbitrarily rotated.

The following sections are organized as follows. In section 2.1, we briefly present an overview of traditional volume rendering techniques and recent GPU based implementations. Section 2.2 describes the lighting model based on the Radiative Transfer Equation. The implementation details are covered in section2.3. Finally, section 2.4 and 2.5 present the results, a brief discussion on the limitations and future directions.

## 2.1 PREVIOUS WORK

Light transport in participating media has been a topic of intense research. An important part of the effort has been focused on different methods for solving the radiative transfer equation (RTE) [38] [102].

### 2.1.1 *Traditional models*

A number of analytic solutions have been described, but they are frequently based on strong simplifications such as assuming homogeneous [14] and optically thick media [165] or infinitelly distant light sources [151].

Numerical simulation approaches are based on evaluating the radiance transport integral [111] [212] [232]. By sampling points inside the media, stochastic methods like Monte Carlo path tracing [231], bidirectional path tracing [131] or photon mapping [104] can generate high quality results. They are capable of modelling complex phenomena including multiple scattering in inhomogeneous media but their cost is usually high: worst cases may involve computations of several hours per frame.

Performance can be increased using precomputation combined with efficient storage. With these methods, results of certain calculations that depend on the light, the media or their interactions are baked so they can be reused at render time. By using pre-computed radiance transfer, Sloan et al. [228] handle complex lighting phenomena like soft shadows and scattering. Other methods use pre-calculated forward multiple scattering [91], pre-integrated lighting [257], spherical harmonics for fourier volume rendering [67], precalculated transmission integrals [96] or reuse light scattering paths in a particle system [239]. Recently, Zhou uses a set of radial basis functions to convert the media into an equivalent sparse representation [287]. With all the benefits of these methods, interactivity is limited to static camera positions, specific transfer functions or prefixed light types. Furthermore, rendering data sets containing a sequence of frames requires precomputation on each of them.

An overview of the existing techniques and the mathematical background of the scattering phenomena can be found elsewhere [88].

### 2.1.2 *Real time rendering*

Implementations of interactive and real time volume rendering methods have become increasingly more popular after recent advances in graphics hardware.

Since analytic solutions evaluate a closed set of mathematical formulas, they can be efficiently implemented using GPU shaders [236] [286]. These methods rely on precomputation, but other solutions are based on 2D texturing [66] or are not physically based [126].

There are a number of acceleration techniques designed to improve the efficiency of media sampling. Kruger and Westermann [126] describe an efficient streaming model over GPUs and implemented different acceleration techniques for volume ray-casting. Early ray termination is based on space classification structures and compression techniques: Guthe et al. use hierarchical wavelets [86], LaMar, Hamman and Joy [132] rely in an octree; Westermann and Senenich [271] use CPU precalculated 2D textures. Li, Mueller and Kaufman's [138] empty space skipping technique avoids sampling areas that contain no data. Spatial and temporal data coherence has also been exploited [245] [118]. All these techniques set up acceleration structures in a precomputation step.

Our intention is efficiently rendering time dependent participating media without significant pre-computation. This precludes the application of the previous methods. The solution we describe in the following sections relies on a physically based light transfer model capable of calculating single scattering with self shadowing. Two optimizations, based on the media rendered and the camera position, provide the extra acceleration required. Moreover, they do not use any pre-computed lighting nor depends on the illumination model or the environment. Acceleration structures and lighting can be calculated on the fly, therefore position and orientation of the environment and camera can be interactively changed.

## 2.2 LIGHTING MODEL

### 2.2.1 *Lighting environment*

High dynamic range maps are one of the most popular methods to represent lighting environments [203]. Several image-based algorithms have been developed based on well known Monte Carlo methods [112]. However, since they rely on intensive sampling of the irradiance map, they cannot be efficiently implemented on a GPU.

Different techniques can convert these maps into spherical harmonics [196], light constellations [42] [148] or wavelets [172]. An environment represented with these alternatives can be efficiently sampled at the cost of using similar but not identical lighting conditions. Except for wavelets, these representations can only include the low frequency components of the environment. For our purposes, this is a mild limitation as the phenomena involved in participating media tends to blur the effects of lighting [172].

The methods that produce light constellations replace areas of the irradiance image with individual light emitters, usually point or infinite directional lights. Among others, Cohen and Debevec's [42] mediancut technique extracts a set of equal energy point lights, each of them representing areas with different sizes. Alternatively, K-means clustering [148] generates lights with different energy but with more regular spatial distribution. Similar algorithms have also been implemented in real time [120] [180].

Conversion to a finite number of sources benefits from deterministic sampling without the storage and complex representations of traditional techniques. At the same time, noise, the principal problem associated with stochastic methods, is eliminated and a desirable coherence over time is added. With such a representation, computing the light received from the environment is converted to repeating a finite number of times the same sampling that is required for a single emitter . All these characteristics make them specially suitable for a GPU based implementation.

### 2.2.2 *Light transport*

Once the light field has been computed, a solution to the light transport needs to be provided. Using the integral form of the RTE [38] [102], the radiance L at a point $x$ in direction $\overline{\omega}$ can be written as:

$$\begin{aligned} L(x,\overline{\omega}) &= \tau(x_0,x)L(x_0,\overline{\omega}) \\ &+ \int_{x_0}^{x} \tau(u,x)\kappa_t(u)(1-\Omega(u))L_e(u,\overline{\omega})du \\ &+ \int_{x_0}^{x} \tau(u,x)\kappa_t(u)\Omega(u)\int_S L(u,\overline{\omega}_i)\rho(u,\overline{\omega},\overline{\omega}_i)d\sigma_{\omega i}du \end{aligned} \tag{2.1}$$

where $x$ and $x_0$ are points in the $\Re^3$ space, $\kappa_t$ denotes the extinction coefficient and is the sum of the scattering $\kappa_s$ and absorption $\kappa_a$ coefficients, $\Omega(x) = \kappa_s(x)/\kappa_t(x)$ is the scattering albedo and $L_e(x,\overline{\omega})$ is the radiance emitted by the media at point $x$ in direction $\overline{\omega}$. S represents the set of directions on the sphere around point $x$, and the normalized phase function $\rho(x,\overline{\omega},\overline{\omega}_i)$ determines the amount of the incident light arriving at $x$ in direction $\overline{\omega}$ that is scattered in direction $\overline{\omega}_i$. $\tau(x_0,x)$, the transmittance factor along the segment from $x_0$ to $x$, is expressed as

$$\tau(x_0,x) = e^{-\int_{x_0}^{x} \kappa_t(\xi)d\xi} \tag{2.2}$$

The cost of computing Equation 2.1 can be reduced by: first, discarding multiple scattering and considering a single scattering event at every point of the volume; second, assuming the media is isotropic; and third, using non emissive media. Single scattering makes the model accurate for low albedo or optically thin media only. An isotropic phase function $\rho(x, \overline{\omega}, \overline{\omega}_i) = \frac{1}{4\pi}$ excludes participating media such as fog, smoke or clouds that show a strong forward scattering behaviour. However, the model could be easily extended to support anisotropic phase functions with reduced run-time cost [13]. Finally, common participating media does not emit light so the emission term $L_e(x, \overline{\omega})$ can be safely discarded.

With the previous assumptions and given that the lighting environment has been replaced by a finite set of N light emitters, Equation 2.1 can be transformed to:

$$\begin{aligned} L(x, \overline{\omega}) &= \tau(x_0, x) L(x_0, \overline{\omega}) \\ &+ \int_{x_0}^{x} \tau(u, x) \kappa_t(u) \frac{\Omega(u)}{4\pi} \sum_{n=1}^{N} L_{ri}(u, \overline{\omega}_n) du \end{aligned} \tag{2.3}$$

Where $L_{ri}(u, \overline{\omega}_n)$ is the reduced incidence radiance [36] arriving at $u$ in the direction of the $n$th light emitter .

Finally, using small integration steps $\Delta x = x - x_0$, the integral Equations 2.3 and 2.2 can be discretized as

$$\begin{aligned} L(x, \overline{\omega}) &\approx \tau(x_0, x) L(x_0, \overline{\omega}) \\ &+ \tau(x_0, x) \kappa_t(x_0) \frac{\Omega(x_0)}{4\pi} \sum_{n=1}^{N} L_{ri}(x_0, \overline{\omega}_n) \Delta x \end{aligned} \tag{2.4}$$

$$\tau(x_0, x) = \tau(x_0, x_0 + \Delta x) \approx e^{-\kappa_t(x_0) \cdot \Delta x} \tag{2.5}$$

This Equation calculates the radiance arriving at a point $x$ as a function of the radiance and media properties at a near point $x_0$. If $eye$ is the viewer position and $\overline{\omega}_{ij}$ is the direction from a pixel with coordinates $(i, j)$ to the observer, then $L(eye, \overline{\omega}_{ij})$ can determine the radiance arriving at the observer from each of the image pixels. Given that the initial conditions at the furthest position in the media can be established, this formulation can be mapped to traditional back to front ray marching using $\Delta x$ increments. In the following sections we will explain the details on how Equation 2.4 can be solved following this schema.

## 2.3 IMPLEMENTATION DETAILS

CPU based ray marching algorithms have been commonly used to display volumetric data. However GPU based implementations are

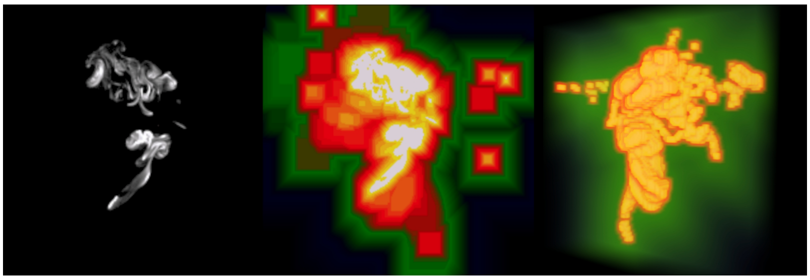

Figure 2.1: Distance map optimization: Left: One of the 256 slices of smoke from a volume data set. Middle: Distance map corresponding to this slice. Brighter regions corresponds to lower distances. Right: Volumetric representation of the whole distance map.

not so frequent due to restrictions in the logic complexity and number of instructions in fragment programs. To overcome these limitations we use an OpenGL and Cg [176] multi-pass approach. The following section describes its implementation details.

As an overview, our system performs the following steps: Initially, volumetric data is loaded from CPU to GPU memory. This data is represented as a 3D array of floating point voxels containing the density of the media. Its structure is analysed and a view independent distance map is computed. This map is used to accelerate media traversals. Taking into account camera position, a different pass builds a validity mask that helps eliminating computations on areas that will be empty in the final image. Once these two acceleration methods are set-up, a succession of passes solve Equation 2.4 using back to front ray marching. Independently, every time the lighting is changed the existing light constellation is replaced by a new one that approximates the illumination of the new environment. In the absence of changes in the environment, the corresponding set of lights will be reused for different frames. The previous steps are performed using floating point arithmetic so a final tone mapping pass is required to display the final image.

### 2.3.1 *Distance based optimization*

The speed of memory accesses is one of the principal bottlenecks in existing GPU architectures . Shader performance is drastically reduced when more than just a few texture fetches are executed. Since the ray marching algorithm is based on intensively sampling the media and accesses lack spatial coherence, brute force implementations fail to run at interactive frame rates. In this section we describe a novel use of an optimization technique that, based on a distance map, drastically reduces the bandwidth requirements by skipping volume regions containing no data.

A distance map determines, for each position $x$ inside the volume, the distance $\delta(x)$ to the nearest non empty voxel. Other authors have used related approaches to calculate surface displacement [61] or compute geometry in shell space [207]. The novelty in our case is we apply this technique to optimize lighting calculations by accelerating primary and shadow ray traversals. Without loss of generality our implementation calculates the distance map as a preprocess but existing algorithms can calculate it in real time [49].

At runtime, for every sampling point, the next visited position is determined jumping a distance given by $\delta(x)$ in the sampling direction. The map stores scalar values without any directional information, so no cues on the exact location of the nearest non-empty voxel can be inferred. As such, each $\delta(x)$ identifies the radius of a sphere centered at $x$ which contains empty data only. This is a conservative optimization, but given the memory constraints of the hardware framework and since we have observed important speed ups (see section 2.4), this seems to be a good tradeoff.

Talking about data layout, the original density data and the distance map are stored in the red and green channels of the same 3D texture. This option doubles the memory footprint, but allows retrieving both values in the same instruction at no extra cost. The outcome between use floating-point or integer values is also clear. The images computed using low or high precision representations show neglectable differences but the performance of the rendering algorithm drops an order of magnitude when the most accurate representation is selected.

### 2.3.2 *Validity mask*

In this section we explain an optimization, implemented as a new pass, that uses the volume of data and the camera settings to determine which areas of the final image will be empty. The method follows each of the primary rays cast from the eye through the media, flagging the traversals that find non-transparent data. Once computed, a full rendering loop is executed for this subset of image pixels. This approach does not require performing any lighting calculations and uses the optimization described in the previous section. As a consequence it is lightweight to evaluate while the speed-ups associated are significant.

Initially, using the method described by Kruger and Westermann [126], each pixel with coordinates $(i, j)$ is assigned a pixel-to-eye direction $\overline{\omega}_{ij}$. A traditional slabs test [114] is used to determine $P_{near}$ and $P_{far}$, the nearest and furthest intersections with the bounding box of the volume. Any pixels which do not define intersections are discarded for further computations.

The sampling strategy is based on using shells or spherical slices centered at the camera position [84]. Although this technique eliminates the need for perspective correction, it requires building proxy

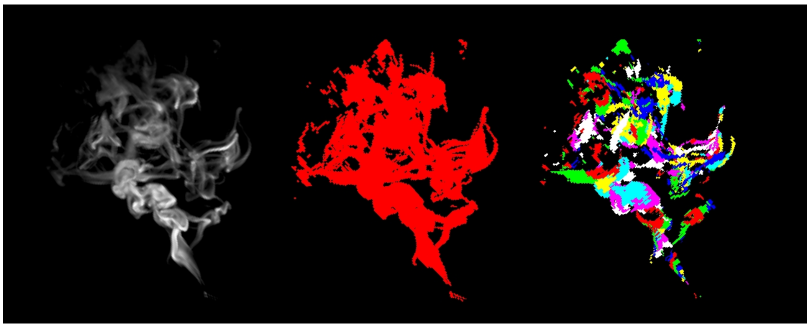

Figure 2.2: Values generated by pixel validity mask pass: From left to right: Smoke volume being rendered, pixel validity mask showing red pixels for those fragments containing valid information and color representation of index of first slice containing non transparent data ($\mathtt{iFirst}$).

geometry containing segmented data. That is one of the reasons why it has been traditionally reserved to cases where extreme viewports were used [66]. Our method does not create any geometry at all as shells are implicitly represented in the mathematical formulae that calculates the position of each sampling point. Each of these positions are found after intersecting a ray with an sphere using their corresponding analytic representations.

Slices are $i\Delta slice$ units away from the camera position, with $i$ being the index of the slice and $\Delta slice$ the distance between them. To prevent aliasing artifacts $\Delta slice$ is chosen to be smaller than half the side of a voxel. However, given the data varies smoothly, is trilinearly filtered by the hardware and the mask is not directly visible, this value can be relaxed and more sparse sampling can be used. We have found that, in some cases, up to one slice every five voxels can yield good results.

$$iNear = ceil(|P_{near} - eye|/\Delta slice) \tag{2.6}$$

$$iFar = floor(|P_{far} - eye|/\Delta slice) \tag{2.7}$$

In order to determine which ray directions will generate empty image pixels, ray marching is performed in back to front order. The indexes of the nearest and furthest slices inside the bounding box are calculated using Equations 2.6 and 2.7. During the traversal from $iFar$ to $iNear$, a direction is tagged as containing data as soon as a density value above a given threshold are found. The index of the corresponding slice is noted as $iFirst$. To improve performance the process is accelerated using the technique explained in section 2.3.1.

This pass outputs a half floating point RGBA image composed of each pixel's normalized ray direction $\overline{\omega}_{ij}$ (RGB components) and $iFirst$ index (alpha component). The later is set to -1 for pixels with no data nor bounding box intersection associated. In Figure 2.2 we can see a false color representation of the validity mask and the first slice index.

It is worth noting that both the validity mask and the distance map make no assumptions on the underlying physical model and are calculated before any light computations are started. This is an important improvement respect to other acceleration methods that only generate time reductions at the cost of making them dependent on specific types of physical phenomena. The single scattering model explained in previous sections it is just a convenient example and a proof of concept. Since our optimizations focus on speeding up traversals and reducing the number of fragments to be calculated, the heavier the computation imposed by the lighting model, the bigger will be the speed-ups associated with our technique.

### 2.3.3 *Ray marching passes*

The radiance arriving to the observer from each of the pixels in the image plane $L(eye, \overline{\omega}_{ij})$ is the main magnitude that needs to be determined to solve the lighting of the participating media. This section explains the details of the Cg fragment program that computes it.

The initial steps of the lighting pass perform a quick test on the validity mask to avoid undesired computations. As soon the pixel is known not to contain data, evaluation of the shader is stopped and a transparent pixel is returned. For the remaining pixels, a primary ray is marched from the position corresponding to $iFirst$ slice in the $\overline{\omega}_{ij}$ direction in $\Delta slice$ increments. This traversal is continued until the ray exits the bounding box. At each point $x$, a shadow ray is followed from each of the $N$ light emitters to the sampling point itself. Since this ray accounts for absorption and scattering, self shadowing effects are included in the final image. We assume the space surrounding the bounding box is empty so the target radiance $L(eye, \overline{\omega}_{ij})$ is $L(P_{near}, \overline{\omega}_{ij})$, the radiance measured where the ray exits the bounding box.

Program 2.1 shows pseudocode that performs lighting calculations on the slice with index $iSlice$. The parameters `vol3D`, `rayDir` and `pathRad` refer respectively to the 3D texture containing the density data and distance maps, ray direction $\overline{\omega}_{ij}$ and path radiance accumulated in previous slices. `x` and `x_uvw` are the sampling point global and texture space coordinates. The function `GetIntersection` returns the intersection of the slice `iSlice` and the ray starting at `eye` with direction `rayDir`. $\tau(x, x + \Delta x)$ is returned by `Transmittance`, whilst $\kappa_t(x)$ and $\Omega(x)$ are calculated by `Extinction` and `ScattAlbedo`. All three functions are internally implemented as lookup tables, calculated once $\Delta x$ is known. Function `OneLightRad` calculates the radiance arriving from the `n`th emitter and is based on a shadow ray traversal. `SkipSlices` returns the number of slices that are skipped according to the value stored in the distance map. Both `iSlice` and `pathRad` are output in a render target.

Listing 2.1: Slice sampling pseudocode.

```
function SampleSlice( vol3D,rayDir,iSlice,pathRad )
{
   x = GetIntersection(iSlice, eye, rayDir);
   x_uvw = ToTxtCoords(x);
   volValue,distValue = tex3D(vol3D,x_uvw);

   // accumulate each light contribution
   lightsRad = (0,0,0);
   for (n=0; n<N; n++)
   {
      lightsRad += OneLightRad(x,n);
   }
   lightsRad *= ScattAlbedo(volValue)/(16*pi*pi);
   lightsRad *= Extinction(volValue);

   // add current sample to path radiance
   pathRad += lightsRad;
   pathRad *= Transmittance(volValue,deltaX);

   // determine next sampled slice
   iSlice += SkipSlices(distValue);

   return (iSlice,pathRad);
}
```

The OpenGL/Cg FP40 fragment profile, the most flexible available at the date of the implementation, imposes a limit of 65.535 instructions per fragment program. Evaluation of shaders containing more instructions are truncated and generate visible artifacts. The nested loops that are used to compute primary and shadow ray traversals easily reach this limit. To workaround this issue, we use an approach that is similar to manual loop unrolling. In this case, the main loop is replaced by a series of passes each of them evaluating $m$ slices. The renderer performs as many of them as are required until the volume is fully covered. $m$ is assigned an empirical value selected on a per-scene basis, depending on the number of light sources and the distance between slices. To reduce the overhead, $m$ is chosen to be the highest possible value without exceeding the limit (in our tests, values between 3 and 15). We use the OpenGL framebuffer object extension [177] in conjunction with MRT (Multiple Render Targets) to pipe the results from each pass as the initial status for the next pass.

## 2.4 RESULTS

The figures in this chapter have been rendered using two different volume data sets. Images 2.3, 2.4 and 2.5a to 2.5h have been rendered with data scanned from real smoke using the optical system of Hawkins et al. [93]. This data set, referred to as $(i)$, is composed of 120 full frame volume scans where each frame has been resized to a resolution of 256x256x64 voxels. Images 2.5i to 2.5p have been calculated from voxel arrays simulated using the algorithm of Shi and Yu [223]. In this data set, named $(ii)$, each of the 450 frames have been resized to a resolution of 128x128x128 voxels. For display purposes we have implemented the interactive tone mapper by Goodnight et al. [78] based on Reinhard's photographic tone mapper [204]. This pass provides the conversion from the high dynamic range images that result from the ray marching process to an 8 bit/channel low dynamic range that can be displayed in a standard monitor.

Each environment map has been processed using the algorithm of Cohen and Debevec [42] and light constellations of up to 20 lights have been generated. Other alternatives exist, including several real time conversion techniques [120] [180]. Each light constellation is preloaded before the render of each frame begins and apart from being a set of point lights, there are no other assumptions associated with this representation. As such, our system allows free rotation of the HDR environment by rigidly reorienting the constellation. Even more, our system can handle dynamic light environments based on sequences of HDR images given each frame generates its own constellation.

In terms of storage, apart from the volumetric data being rendered and its corresponding distance map, there are no significant memory requirements. One RGBA framebuffer is required to store the valid-

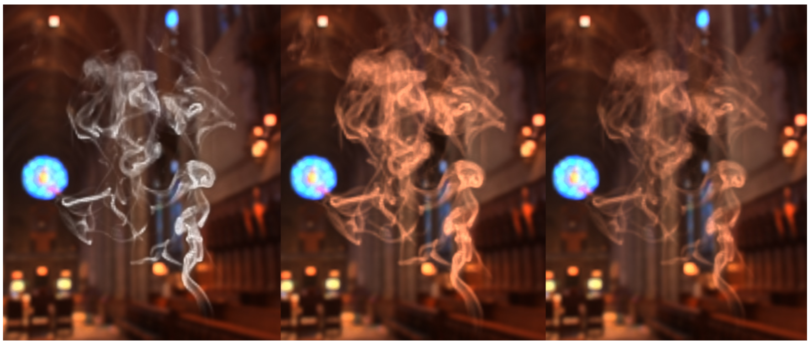

Figure 2.3: Lighting models, left to right: Simple back to front alpha compositing, primary rays, primary rays and shadow rays.

ity mask as well as two extra ones to store the rendered image and perform ping-pong buffering . Light constellations use only 6 floating point values (color and position) per emitter. For low frequency lighting, these requirements are comparable to the equivalent of spherical harmonics and wavelets. More significant is the absence of heavy pre-computation and storage for lighting and transfer functions. If we consider this in the context of displaying sequences of volumetric data sets, these advantages become important improvements respect to other techniques.

Figure 2.3 shows one frame rendered using different lighting models: back to front alpha compositing, single scattering limited to primary rays and single scattering using primary and shadow rays. Note how in the second and third images the light constellation successfully captures the overall lighting conditions even with a low number of light emitters. However, in the middle image, as the emitters illuminate every point with the same strength, the media looks excessively bright. The most realistic result in the rightmost image can be attributed to the attenuation and self shadowing effects performed during the light-to-sampling-point traversals. This example shows that for optically thin media, assuming a single scattering model does not prevent getting images that look physically correct.

Figure 2.4 shows the exact same volumes rendered using different extinction coefficients. In this case, the first image, optically thin media with low extinction coefficient; and the second one, optically thick media with high extinction coefficient are correctly depicted.

Finally, Figure 2.5 shows a range of lighting environments representing diverse conditions including interior and exteriors as well as high contrast to smoothly changing conditions. In all the cases, volumetric data is lit without visible artifacts and is well integrated in the overall scene.

Algorithm performance is represented in Table 2.1. For each data set (i and ii) and lighting method (alpha, primary and self shadowing), values represent the average number of frames rendered per second. Each combination is evaluated using three different running modes:

| | | ALPHA | PRIMARY | SELF SHADOWS |
|---|---|---|---|---|
| No optimiz. | i | 11 | 4.1 | 0.5 |
| | ii | 11.5 | 4.7 | 0.6 |
| Mask | i | 10 (0.9) | 4.7 (1.1) | 1.2 (2.4) |
| | ii | 9.3 (0.8) | 4.5 (0.9) | 1.1 (1.8) |
| Mask + dist. | i | 36.5 (3.3) | 16.3 (3.9) | 4.1 (8.2) |
| | ii | 35 (3.0) | 14.8 (3.1) | 4.0 (6.6) |

Table 2.1: Frames per second (speed ups)

brute force ray marching without optimizations, optimized using validity mask and optimized using validity mask and distance maps. Speed-ups respect to the brute force method are shown in brackets. All the images have been rendered with a NVidia Quadro FX3500 graphic card, at a resolution of 512x512 pixels. The projection of the bounding box for data sets (i) and (ii) approximately covers 50% of the screen where 20% and 40% of the pixels correspond to non transparent smoke respectively. In the absence of any optimizations, the algorithm is capable of generating interactive frame rates, ranging from 11.5 to 0.5 fps, which is a poor performance compared to texture based methods. The second data set performs slightly better than the first, probably due to its smaller size.

Performance gains up to 2.4 are obtained when the validity mask is enabled. Higher improvements are obtained with data set (i) whose mask contains less pixels to be processed. Note, the cost of this pass makes the algorithm run slower with the simplest lighting model (0.8 speedup). However, when the complexity of the model increases (primary and self shadowing), this overhead is surpassed by the improvements associated with the absence of processing in empty image areas.

Finally, activating the distance map optimization results in the best performance. As both the validity mask and render passes apply this method, the resulting speed-up raises to 8.2. Up to 4 frames per second can be rendered using the full lighting model. Given this optimization takes advantage of volume sparsity, the algorithm performs slightly better with data set (i). These results show that our implementation is more efficient than other algorithms that include self shadowing without the need of including heavy pre-computation.

## 2.5 CONCLUSIONS

In this section, we have presented a framework capable of rendering inhomogeneous participating media at interactive frame rates. Adapting the ray marching technique to the GPU, our method models the

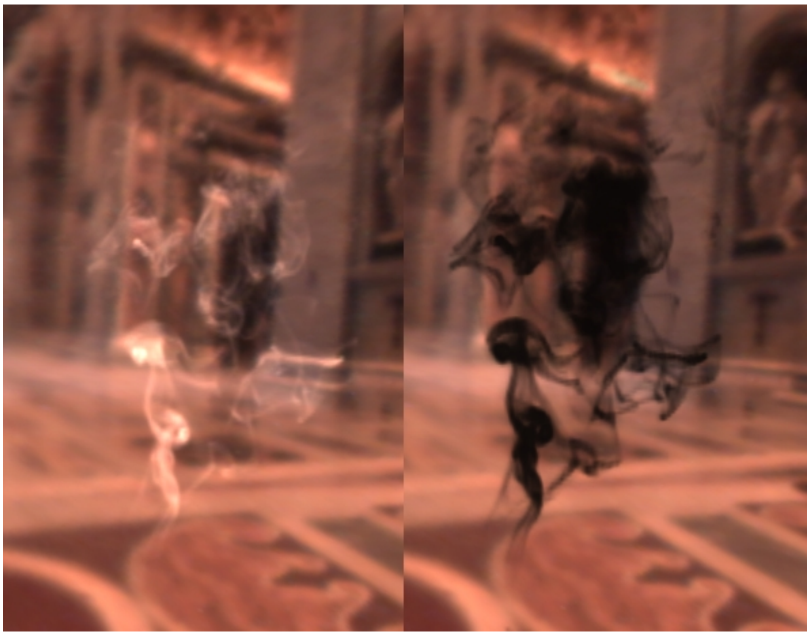

Figure 2.4: Image shows same dataset rendered with high (left) and low (right) extinction coefficients.

incoming light field as a light constellation obtained from a HDR environment map. It allows for dynamic media, lighting and camera motion and any changes in these elements are efficiently considered using real time recalculation.

By combining two novel optimization techniques, a view-dependent validity pass and a view-independent distance map, we achieve speedups up to 8 times respect to a brute force implementation. We have used a model based on single scattering with self shadowing. However, since the optimizations do not make any assumptions about the nature of the light interactions, they can be applied to other lighting models. A GPU implementation of those algorithms is expected to be bounded by the complexity of per-fragment calculations and the memory intensive media traversals. Both elements are addressed by these techniques.

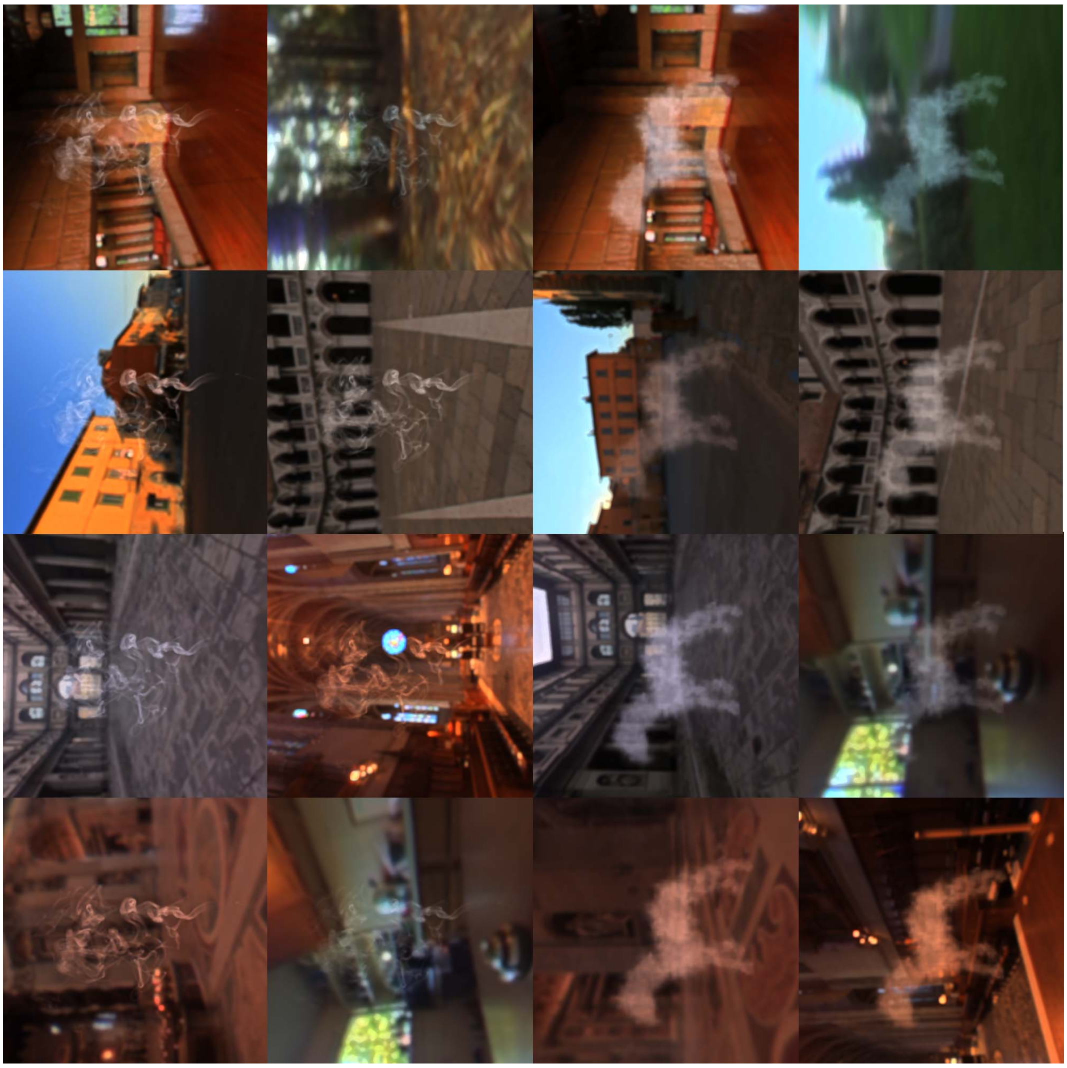

Figure 2.5: Smoke rendered using eight different light probes. Images a to p from left-right and top-bottom order

# Part II

# ACCELERATING ANTIALIASING RENDERING

Aliasing is an intrinsic limitation of sampling methods. In the case of computer rendered images, processes such as rasterization tend to generate artifacts in the form of staircase lines and jaggies.

While hardware vendors have provided their own proprietary solutions, they frequently rely on expensive techniques such as multisampling.

In order to overcome these costs, a number of postproduction methods have been described. One of the most recent is morphological anti-aliasing, MLAA. This algorithm is simple to implement, and it is capable of producing images of reasonable quality. However, the original implementation is not GPU amenable.

In this section we describe a novel method inspired on Reshetov's original algorithm that is specially designed to leverage the parallelism available on the GPU. Our method can be easily integrated in an existing rendering pipeline, it is temporally stable and produces high quality anti-aliasing at a very modest cost.

# 3 MORPHOLOGICAL ANTIALIASING

In general terms, aliasing is a known issue of signal processing. It is produced when two different signals are handled by a process that makes them indistinguisable. Aliasing may also be the result of a reconstruction process that fail to produce the original signal from a set of samples. Aliasing and the issues associated have been studied in depth and are known for a variety of scientific fields [179].

A computer generated image is the result of sampling and reconstruction of the continuous optical signals that represent the natural interactions between light, geometry and materials. Virtual cameras deliver a pixel-based discrete representation of the original signals. Aliasing results from insufficient sampling levels respect to their bandwidth. In this case, the reconstructed image differs from the original representation of the scene.

While this description of the phenomena is generic, it is frequently found in two specific ways:

- Edges and textures with sharp transitions are defined by signals with a infinitely wide frequency spectra. In these cases, no sampling level provides enough data to reconstruct the originals. Edges generate staircase and jagged lines, and textures lose part of the original detail which in some cases will add up to moire patterns. These issues are commonly described as screen-space or spatial aliasing.

- In the case of functions that evolve along time, insufficient sampling will not only produce the issues described before, but will also generate flickering and strobbing artifacts. The problem of temporal aliasing and its perceptual implications will be studied in chapters 4, 5 and 6

In this section we study the implications of spatial aliasing and suggest a solution to improve the visible quality of any rendered images. Our method is based on morphological anti-aliasing [206], which relies on detecting certain image patterns. The shape of the original line before rasterization is reconstructed using a non-linear filter. The original implementation is designed run in a CPU and relies on memory structures that can hardly be represented in a real time environment.

Our method leverage the parallelism available in a GPU. The quality of our results lies between 4x and 8x MSAA, considered the current gold standard for real time antialiasing rendering, at a fraction of the time and memory consumption.

## 3.1 PREVIOUS WORK

For many years, the aliasing problem has been solved using different approaches and as soon as in 1977 it was covered in the context of synthetic imagery [48]. The most salient methods are covered in the following section.

### 3.1.1 *Edge antialiasing*

In some cases, the solution comes from an improved method to convert the geometrical description of the objects in the scene to image pixels. This process is called generically as rasterization. Line segments can be antialiased using improved versions of Bresenham's algorithm such as Wu's line algorithm [277].

The method of Cook et al. performs antialiasing by converting the geometry of the scene into a set of tiny quadrilaterals [46]. Their shading is stochastically sampled and filtered down to produce an image pixel. This has certain advantages such as allowing lower shading sampling levels respect to the pixel sampling levels.

### 3.1.2 *Aliasing in textures*

In the same way that geometry edges produce high frequency signals, textures can also contain sharp transitions and fine detail. This produces similar challenges to any sampling method, but at at difference from the previous methods, it is not supported by a geometrical description.

One of the options consists of pre-filtering the textures so high frequencies from the original signal are removed before it is sampled. Low pass filtering removes any undesired signal contents while leaving any frequencies under the Nyquist level intact [175]. In this case, the difficulty is how to reduce image blur and keep enough detail while eliminating the highest range of frequencies.

Clamping is a special filtering method that is frequently applied to textures that are generated by spectral synthesis [64]. Since texture generation explicitly adds each frequency component to the texture profile, the procedure ignores any undesired frequency components.

A final alternative relies on using enhanced filter functions. The simplest box filter averages every sample with equal weight, which in most cases produces excessive blur. Second-order linear, third-order quadratic and four-order cubic filters provide better approximations to the optimal sync-filter. The latter as well as the Catmull-Rom and Mitchell filters contain negative lobes, which tend to produce sharper texture patterns although in some cases can make ringing artifacts visible [33, 161].

In most cases, the design of the previous techniques make them suitable for environments where there is no limitations in the latency where each image is produced. In the following sections we will describe approaches that are applicable in all cases, but will focus on the alternatives that can be implemented as part of a real-time framework.

### 3.1.3 *Real time multisampling*

Multisampling methods perform several computations per visible pixel and the final color is determined by a filtered down version of these samples. While this method is popular in the context of offline rendering [170], the difficulties associated with the use of limited hardware have reduced the introduction into real-time frameworks.

The earliest instance of multisample rendering taking advantage of hardware acceleration is the accumulation buffer algorithm [90]. This method, performs several full renders with slightly different camera parameters. The final image comes from the addition of those images into the accumulation buffer.

In the current generation of hardware graphics units, multisampling anti-aliasing (MSAA) remains the most extended solution and the one that offers the best results. The precursor of this technique, FSAA (Full Scene Antialiasing), performs a full render of the scene using an oversized resolution framebuffer that is downsampled to generate the final image. MSAA uses different optimizations whereas certain components of the image such as stencil and depth buffers are supersampled while others, ie. lighting and shading, are not.

Different flavors of this technique have been introduced by each hardware vendor. However, techniques such as CSAA, Qintix, ... are based on similar concepts. CSAA, short for coverage antialiasing [281], reduces shading complexity after decoupling color and coverage samples. In this case, the image results from the combination of higher number of coverage samples than shading/z/stencil samples.

These methods allow better image quality where imaging was traditionally based on non-antialiased methods. However, there are important drawbacks to the use of multisampling, specifically MSAA, in certain scenarios:

- It is roughly equivalent to computing a higher resolution image that is sampled down. The increase in processing time and storage is not negligible [242].

- In some cases activating multisampling prevents the use of multiple render targets (MRT). This is specially relevant as MRTs are widely used and are the basis of fundamental techniques such as deferred shading [226, 122].

- Implementation of MSAA is neither trivial nor cost-free [242]. In fact, the current generation of consoles does not offer a simple and complete solution. In the case of the Xbox 360, memory constraints force the use of CPU-based tiling techniques to handle high resolution frame buffers in conjunction with MSAA; whereas on the PS3 multisampling has not been widely applied.
- MSAA is uncapable of smoothing non-geometric edges. This has an impact on vegetation rendering for example, where fine details are generated using rough geometry and alpha channels. In these cases, antialiasing can only computed if an adhoc technique such as alpha to coverage is added [174].

### 3.1.4 *Postproduction*

Postproduction methods are an interesting alternative to multisampling. This family of methods uses the information contained in a rendered image and applies different enhancement filters to reduce the visibility of the aliasing artifacts.

The most popular approaches rely on a feature detection stage that analyses the information present at the neighbourhood of each pixel and detects those that can potentially be the result of a rasterized edge. A blurring stage uses the pixels in the image to generate a new antialiased value. This imposes certain quality compromises but it is efficient to evaluate and only requires information that is produced at render time.

In its simplest form, edge detection can be based on the color of each pixel but some methods also take advantage of normal and depth information. In S.T.A.L.K.E.R [226], edge detection is performed using differences in the 8-neighbourhood depth values and the 4-neighbourhood normal angles; then, edges are blurred using a cross-shaped sampling pattern. A similar, improved scheme is used in Tabula Rasa [122], where edge detection uses threshold values that are resolution independent, and the full 8-neighbourhood of the pixel is considered for differences in the normal angles. In Crysis [229], edges are detected using depth values and texture lookups using bilinear filtering. Finally, in Killzone 2 samples are rendered into a double horizontal resolution G-buffer. The lighting pass averages two samples from the G-buffer to produce a pixel in the final image buffer. However, this implies executing the lighting shader several times for each final pixel.

Morphological antialiasing [206] is a recent improvement in this area. Using the neighbourhood of each pixel in the rendered image, certain patterns are detected and based on them, the original topology of the edges is reconstructed. The image is blurred using an adaptive filter that smooths the image based on this reconstructed geometry. The original CPU based implementation has been successfully ported to the cell-processor architecture of the PS3 [99]. In the following sections we will describe the first GPU based implementation available to the date of publication.

## 3.2 OVERVIEW

Reshetov searches for specific patterns (U-shaped, Z-shaped and L-shaped patterns) which are then decomposed into simpler ones. This approach, while feasible in architectures where random memory accesses have low impact on performance, it is impractical for GPU based stream frameworks.

We have found that the pattern type and the consequent anti-aliasing parameterization, only depends on a small set of values that can be obtained for each edge pixel. For brevity, in the following, we will call *edgel* to any pixel belonging to a detected edge. Furthermore, this approach allows to handle all pattern types in a symmetric way so the need for decomposing them to simpler ones is avoided. The original algorithm is transformed so that it uses texture framebuffers instead of lists (see Figure 3.1).

In addition, pre-computing certain values and storing the results as textures allows for an even faster implementation. Section 3.5 discusses this optimization.

Finally, in order to accelerate calculations, we make extensive use of hardware bilinear interpolation. This allows fetching several blended values without the need of explicit computation in a single operation. Moreover, we store certain magnitudes in a way that the neighbourhood that correspond to certain pixel values can be inferred without explicitly retrieving them. Section 3.6 gives details of this short-cut.

The combined speed-ups associated with this algorithm allows it to be efficiently executed by a GPU, has a moderate memory footprint and can be integrated in the standard rendering pipeline of any game architecture.

## 3.3 ALGORITHM STRUCTURE

The algorithm searches for patterns in edges. Based on this information, the original lines are constructed to be later rasterized using an improved antialiased profile. This can, in general terms, be seen as a re-vectorization of edges.

The method is implemented as a succession of three render passes:

1. The initial pass performs edge detection on the image that needs to be antialiased and yields a texture containing edgels (see Figure 3.1, *center left*).

2. In the second pass the blending weight $a$ is computed for each pixel adjacent to the edge being smoothed (see Figure 3.1, *center*

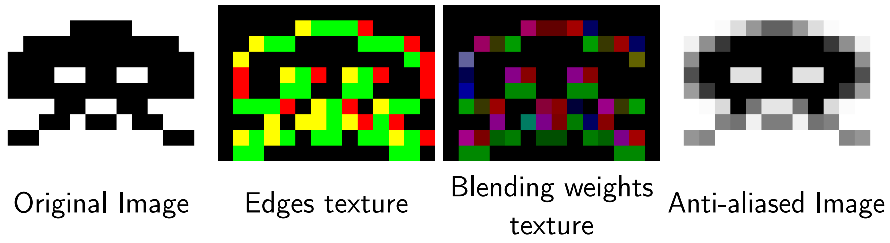

Figure 3.1: Starting from an aliased image (*left*), edgels can be detected and stored in a edges texture (*center left*). The color of each pixel represents the relative position of each pixel relative to the edge: green pixels have an edge at their top boundary, red pixels at their left boundary, and yellow pixels have edges at both boundaries. The edges texture is used in conjunction with the precomputed area texture to produce the blending weights texture (*center right*). This texture stores the weights for the pixels at each side of an edgel in the RGBA channels. In the third pass, blending is performed to obtain the final anti-aliased image (*right*).

*right*). The value is based on the pattern type for each line passing through the north and west boundaries or each pixel and is followed by a calculation of the distances to the *crossing edges*. Both values are used to query the precomputed area texture.

3. The third and final pass involves blending each pixel with its 4-neighborhood according to the blending weights from the previous pass. See Figure Figure 3.1, *right*.

## 3.4 PASS #1: DETECTING EDGES

Edge detection is performed using the data contained in the depth buffer of the original rendered image. For each pixel, the difference with respect to the pixel to the top and to the left left is computed. This difference is thresholded to obtain a binary value indicating whether an edge exists in a pixel boundary. The value stating the existence of left and top edges is stored in the red and green channels of the edges texture, respectively.

In most cases, depth information is enough to detect edges. However, in those locations where no depth discotinuity exists, for example where two different polygons meet at an angle, an additional condition based on the normals to the polygons.

In case no geometric information is available, thresholding needs to be performed using magnitudes other than depth or normals. In these cases, the final rendered image can be used to extract a luminance value per pixel using the CIE XYZ standard formulation:

$$L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \tag{3.1}$$

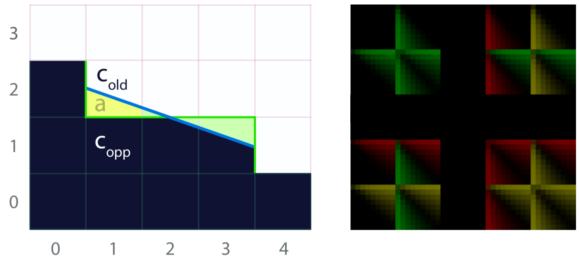

Figure 3.2: *Left:* Anti-aliasing process. Color $c_{opp}$ bleeds into $c_{old}$ according to the area $a$ below the reconstructed blue line. *Right:* Texture containing the precomputed areas. The texture uses two channels to store areas at each side of the edge, i.e. for a pixel and its opposite (pixels $(1,1)$ and $(1,2)$ on the left). Each $9 \times 9$ subtexture corresponds to a pattern type. Inside each of these subtextures $(u,v)$ coordinates encode distances to end of the left and to the right end of the line, respectively.

As an example, when dense vegetation is present, using luminances is faster than using depth (around 12% faster for the particular case shown in Figure 3.5, *bottom row*), due to the amount of edges detected using depth. Figure 3.1, *center left* shows the resulting image of the edge detection pass, in this particular case using luminance-based detection.

## 3.5 pass #2: blending weights

Those edgels that need to be modified are blended according to the following equation:

$$c_{new} = (1 - a) \cdot c_{old} + a \cdot c_{opp}, \tag{3.2}$$

where $c_{old}$ is the original color of the pixel, $c_{opp}$ is the color of the pixel on the other side of the line and $c_{new}$ is the new color of the pixel. The blending factor $a$ is a function of both the pattern type and the distances to both ends of the line.

In Figure 3.2, *left* represents the rasterized line and its respective crossing edges as blue and green segments respectively. The blending factor is displayed as a yellow area. In order to save processing time we precompute this area and store it as a two-channel texture that can be seen in Figure 3.2, *right*.

In order to calculate the blending weights, the second pass will compute the distances to both ends of the line which the edgel belongs to. Taking the edges texture as input, a linear search in the horizontal direction for the ends of the edge will provide this information (see Section 3.5.1). The distances are used to fetch the *crossing edges* at both ends of the line (see Section 3.5.2). These *crossing edges* indicate the type of pattern.

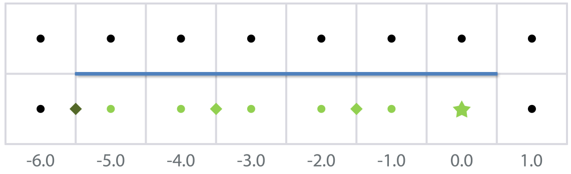

Figure 3.3: Distance computation relies on hardware bilinear filtering. The color of the dot at the center of each pixel represents the value of that pixel in the edges texture. In the example, distance search progresses from the the pixel marked with a star to the one to the leftmost end of blue line. Each access to the edges texture is marked with a rhombus. This allows us to travel double the distance with the same number of accesses as a standard pixel by pixel traversal.

The pass outputs a texture that contains, for each pixel, the areas at each side of its corresponding edgels. This produces two values for north edgels and two values for west edgels in the final blending weights texture. These values are directly used by the final pass to control pixel blending and produce the antialiased texture.

### 3.5.1 *Searching for Distances*

Distances are determined using an interative search algorithm that steps over the pixels of an horizontal edge. Each iteration simply checks whether the end of the line has been reached or not. Once the end of the line has been reached, the number of hops is returned as a distance map.

The process is accelerated by leveraging the fact that the information stored in the edges texture is binary –it simply encodes whether an edgel exists–. Since the texture is queried using hardware bilinear filtering, a query at a position placed between two pixels retrieves a value that represents the linear combination of the two pixels at their sides. (see Figure 3.3). Based on this, a fething the value in the mid point of the center of two pixels can only produce the following results:

(a) 0.0, which means that none of the pixels contain an edgel,

(b) 1.0, which implies an edgel exists in both pixels,

(c) 0.5, which is returned when just one of the two pixels contains an edgel

These values are used as stop condition for the search, that is, the end of the line has been found if the returned value is lower than one.

Searches are limited to a maximum distance. This makes execution times bounded and more predictable, independently of the complexity and the visual characteristics of the image. As expected, the highest the

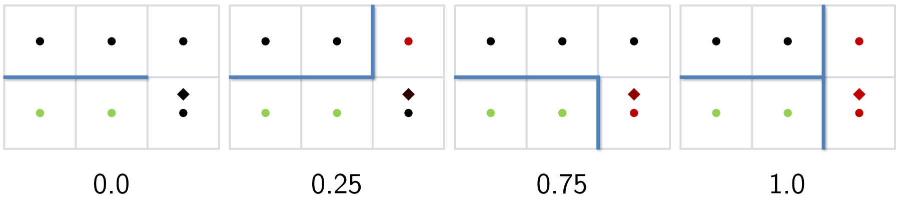

Figure 3.4: Examples of the four possible types of *crossing edge* and corresponding value returned by the bilinear query of the edges texture. The color of the dot at the center of each pixel represents the value of that pixel in the edges texture. The rhombuses, at a distance of 0.25 from the center of the pixel, indicate the sampling position, while their color represents the value returned by the bilinear access.

maximum length, the better the quality of the anti-aliasing. However, we have found that, for the majority of cases, distance values between 8 and 12 pixels are good trade-offs between quality and performance.

### 3.5.2 *Fetching Crossing Edges*

Once the distances to the ends of the line are determined, they are used to obtain the *crossing edges*.

A naive approach to fetch the *crossing edge* of an end-of-line would imply querying two edgels. Instead, using bilinear filtering in a similar way as with distance computation may provide significant perfomance gains. This allows fetching two edgels at a time. However, in this case the actual value of each edgel must be distinguished, so the offset is set to 0.25. The set of possible values is then 0.0, 0.25, 0.75 and 1.0 for samples finding none, one to the left, one to the right and two edgels, respectively. Figure 3.4 shows the *crossing edge* corresponding to each of the different values returned by the bilinear query.

The resulting blending weights texture stores area values for two adjacent pixels in a similar way as with the edges texture. As a consequence the precomputed area texture needs to be built on a per-edgel basis. Thus, each pixel stores two values, one for a pixel and another one for its opposite pixel. Again, $a$ is zero for one of them in all cases except for those pixels that lie at the center of a line with odd length.

## 3.6 pass #3: blending with the 4-neighborhood

The blending pass computes the color of each pixel in the final image by combining the colors from its four neighbours. The blending factor is determined according to the areas stored in the weights texture, shown in Figure 3.2.

The computation is based on three memory accesses to the weight texture: the current pixel, which gives us the north and west blending weights; as well as the pixels to the south and east.

Once more, four bilinear filtered accesses allow blending the current pixel with each of its four neighbours. One pixel can belong to four different lines, so the average between all contributing lines determines the final color. Figure 3.1, *right*, shows the resulting image.

## 3.7 RESULTS

Figure 3.5 shows a comparison of three images calculated using MLAA, 8x MSAA and no anti-aliasing. These scenes belong to the Unigine Heaven Benchmark. Visual inspection show MLAA algorithm lies between 4x and 8x MSAA, while only requiring a memory consumption of 1.5x the size of the backbuffer on PC and of 2x on Xbox 360.

Working as a postprocess, the MLAA algorithm cannot recover subpixel features. This is a drawback respect to other algorithms such as MSAA that work at the sample/subpixel level. In this case, the algorithm cannot recreate the detail has has been lost in the original render, as can be seen in the images in the bottom row of figure 3.5.

Our implementation can work as a command line tool on individual images. Using this interface, we have run it on a batch of screenshots from several commercial games. Figure 3.6 and 3.7 shows example images from Fable® III.

Screenshots were taken at $1280 \times 720$, which can be assumed to be the most frequent resolution in the current generation of games. Depth information was not available so the algorithm applied luminance-based edge detection. Table 3.1 shows the average time and standard deviation of our algorithm on different games and platforms (Xbox 360/DirectX 9 and PC/DirectX 10), as well as the speed-up factor with respect to MSAA. On average, our method provides a speed-up of 11.80x with respect to 8x MSAA.

## 3.8 CONCLUSION

In this chapter, we have presented an algorithm crafted for the computation of anti-aliased images. The method is inspired on the original Morphological Antialiasing by Reshetov and leverages the parallelism available in GPU based architectures.

Our technique can be efficiently executed and is usable in actual game environments. The times measured and the memory consumption improve those required with state-of-the-art techniques such as MSAA multisampling. Working as a postprocess, it has minimal impact on existing rendering pipelines. The quality of the images is equivalent to 4x and 8x MSAA.

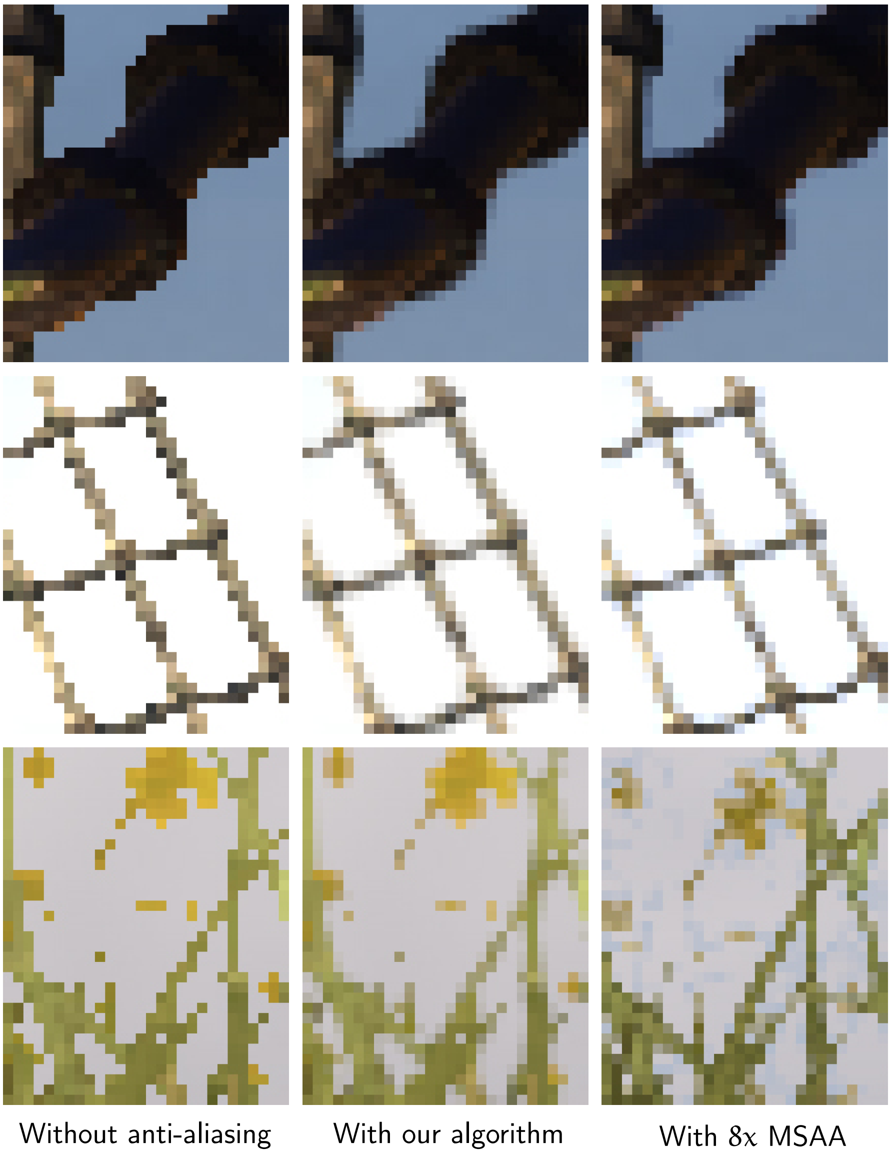

Figure 3.5: Examples of images without anti-aliasing, processed with our algorithm and with 8x MSAA. Our algorithm offers similar results to 8x MSAA. A special case is the handling of alpha textures (*bottom row*). Note that in the grass shown here, alpha to coverage is used when MSAA is activated, which provides additional detail, hence the different look. As the scene is animated, there might be slight changes in appearance from one image to another. Images from Unigine Heaven Benchmark courtesy of Unigine Corporation.

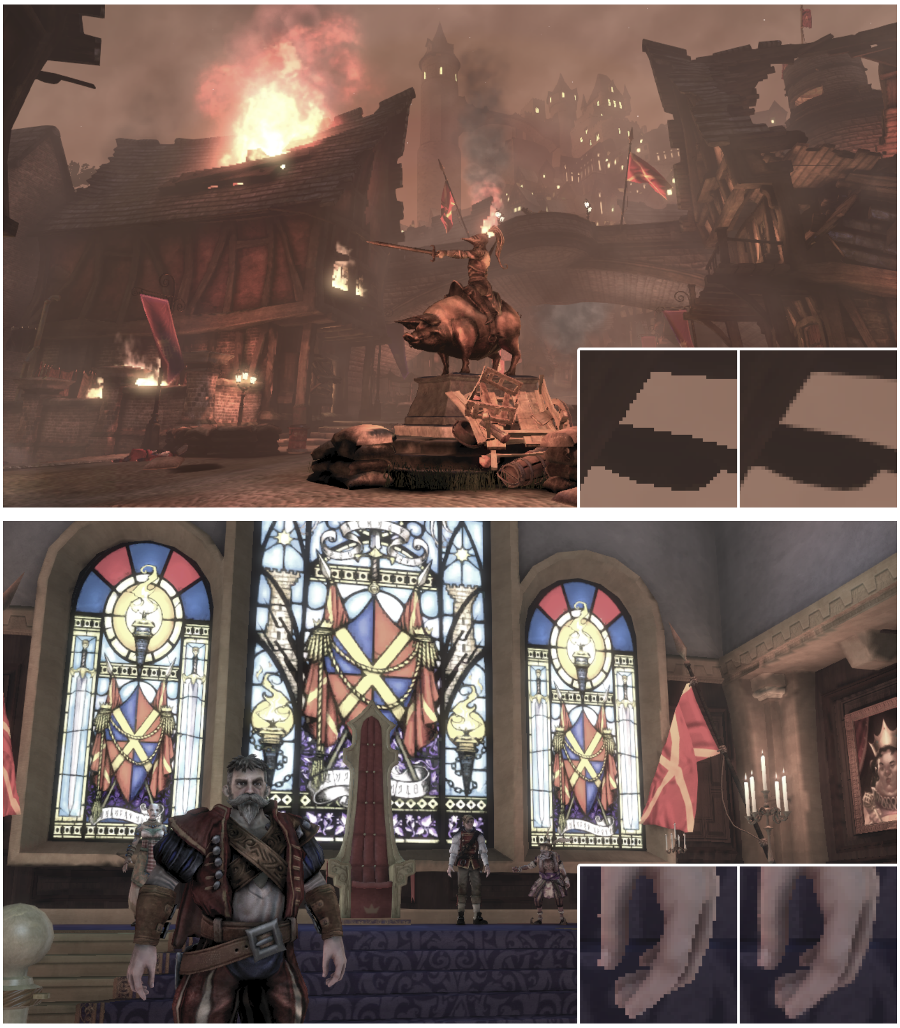

Figure 3.6: Images obtained with our algorithm. Insets show close-ups with no anti-aliasing at all (*left*) and processed with our technique (*right*). Images from Fable® III courtesy of Lionhead Studios.

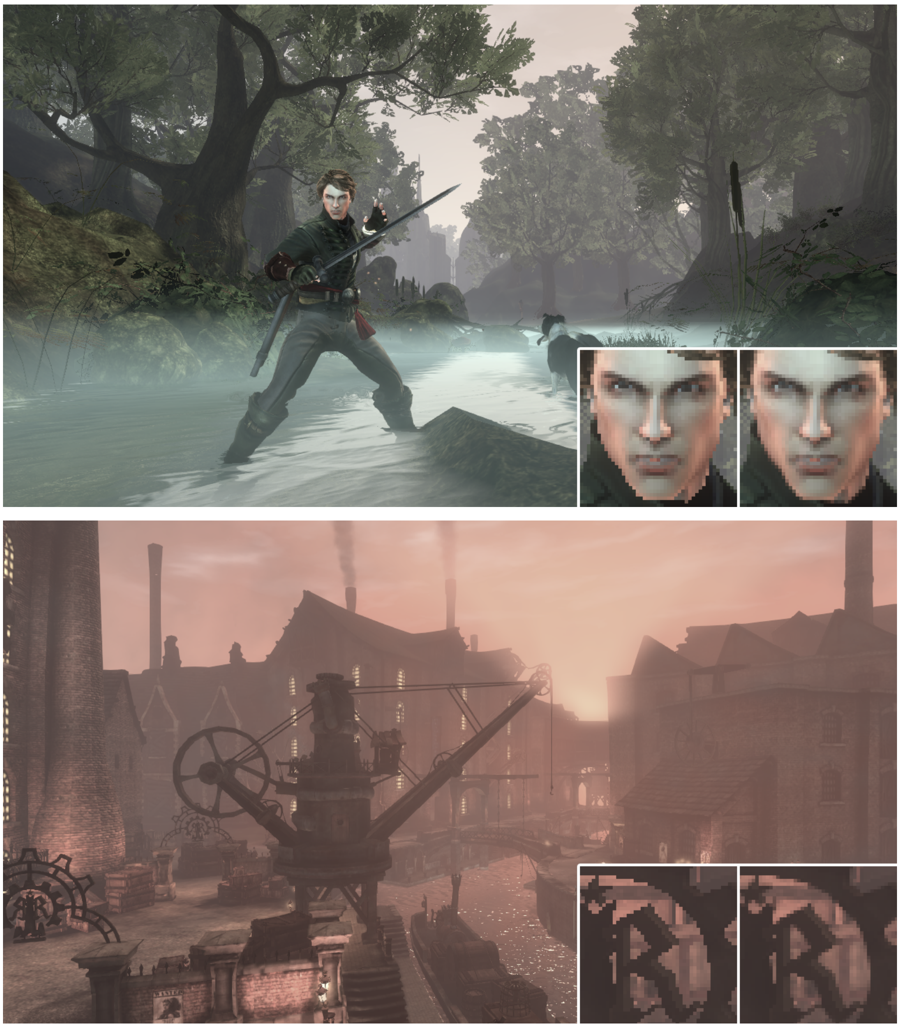

Figure 3.7: More images showing our technique in action. Insets show close-ups with no anti-aliasing at all (*left*) and processed with our technique (*right*). Images from Fable® III courtesy of Lionhead Studios.

| | Xbox 360 | | GeForce 9800 GTX+ | | |
|---|---|---|---|---|---|
| | **Avg.** | **Std. Dev.** | **Avg.** | **Std. Dev.** | **Speed-up** |
| Assasin's Creed | 4.37 ms | 0.61 ms | 0.55 ms | 0.13 ms | 6.31x* |
| Bioshock | 3.44 ms | 0.09 ms | 0.37 ms | 0.00 ms | n/a |
| Crysis | 3.92 ms | 0.10 ms | 0.44 ms | 0.02 ms | 14.80x |
| Dead Space | 3.65 ms | 0.45 ms | 0.39 ms | 0.03 ms | n/a |
| Devil May Cry 4 | 3.46 ms | 0.34 ms | 0.39 ms | 0.04 ms | 5.75x |
| GTA IV | 4.11 ms | 0.23 ms | 0.47 ms | 0.04 ms | n/a |
| Modern Warfare 2 | 4.38 ms | 0.80 ms | 0.57 ms | 0.17 ms | 2.48x* |
| NFS Shift | 3.54 ms | 0.35 ms | 0.42 ms | 0.04 ms | 14.84x |
| Split/Second | 3.85 ms | 0.27 ms | 0.46 ms | 0.05 ms | n/a |
| S.T.A.L.K.E.R. | 3.18 ms | 0.05 ms | 0.36 ms | 0.01 ms | n/a |
| **Grand Average** | 3.79 ms | 0.33 ms | 0.44 ms | 0.05 ms | 11.80x |

Table 3.1: Average times and standard deviations for a set of well-known commercial games. A column showing the speed-up factor of our algorithm with respect to 8x MSAA is also included for the PC/DirectX 10 implementation. Values marked with * indicate 4x MSAA, since 8x was not available, and the grand average of these includes only values for 8x MSAA.

The method improves many aspects of MSAA at a moderate cost. For example, it can antialias transparent textures such as the ones used in alpha testing for rendering vegetation, whereas MSAA can only smooth vegetation when using alpha to coverage.

Since it can produce image qualities that are comparable to 8x MSAA with times an order of magnitude smaller, it seems like a good choice for rendering high quality images in today game architectures.

# Part III

# ACCELERATING MOTION BLUR RENDERING

Motion blur is a fundamental cue in the perception of objects in motion. This phenomenon manifests as a visible trail along the trajectory of the object and is the result of the combination of relative motion and light integration taking place in film and electronic cameras.

Motion blur is also a frequent requirement for the generation of high quality animated images. Due to the nature of computer rendering methods, it needs to be explicitly simulated and this process is frequently resource hungy. Section 4 makes an in-depth review of the existing algorithms and categorizes them in the context of a formal model that highlights their differences, strengths and limitations.

Research has traditionally been focused on finding new methods to reduce the cost and alleviate sampling artifacts. They usually try to steer computation to the most important dimensions of the rendering equation.

In sections 5 and 6, we take a different approach and study monoscopic and stereoscopic images rendered with motion blur from the perspective of human perception. We evaluate the influence of high-level properties such as object material and speed, shutter time and sampling level. Based on several psychophysical experiments, we determine which ones are more influential in the perception of image quality. Interestingly, we have found that radically different motion blurred stimuli may be indistinguishable and even equivalent to a gold standard. This has direct implications on how scenes can be rendered and opens the door to time optimizations.

To our knowledge, this is the first time that the perceptual limits of the Human Visual System have been considered to accelerate motion blur rendering. What follows can be considered an important step in that direction.

# 4

# MOTION BLUR RENDERING: STATE OF THE ART

Motion blur is an effect that manifests as a visible streak generated by the movement of an object in front of a recording device. It is the result of combining apparent motion in the scene and an imaging media that integrates light during a finite exposure time. This relative motion can be produced from object movement and camera movement and is observed both in still pictures and image sequences. In general, sequences containing a moderate amount of motion blur are perceived as natural, while its total absence produces jerky and strobing movement.

This phenomenon is an integral effect of photography and film recording. It needs to be accounted for and in some cases compensated with the use of specific techniques [2]. Moreover, it has become part of the toolkit used by filmmakers and photographers [187] and is commonly used to increase dynamism in still images (see Figure 4.1).

Motion blur for synthetic images has been an active area of research from the early '80s. Unlike recorded footage that automatically integrates motion blur, rendering algorithms need to explicitly simulate it. These cases require accurate temporal integration to avoid aliasing artifacts that can otherwise be easily noticed by an untrained eye. This simulation is one of the most expensive processes in the production of high quality renders. The cost becomes more relevant knowing that the result is a blurred image whose high spatial frequencies have been removed.

In this work we describe the origin of the phenomenon and make a detailed discussion of the existing algorithmic solutions. In section 4.1 we start with a description of the physical phenomena that generate motion blur in a recording device. In section 4.2, it is mathematically formalized based on the models by Meredith [157] and Sung et al. [237]. In section 4.3 we review the techniques that can produce synthetic motion blur and classify them according to this formalization. Their similarities and differences are analyzed in section 4.4. Section 4.5 briefly discusses current research trends and future directions.

## 4.1 ORIGIN OF THE PHENOMENON

The amount of light entering a camera, independently of whether it uses photosensitive film or an electronic sensor, is limited by the diaphragm and shutter [2].

The diaphragm defines the size and shape of the aperture where the light enters the body of the camera, and it statically influences

Figure 4.1: Motion blur is frequently used to increase the perceived motion in photographic snapshots. As seen in the images, carefully controlling shutter timing, camera motion, illumination, lens and filter configurations can produce dramatic effects. Images reproduced with permission of their respective authors: ©Tony Margiocchi, flickr users Noodle Snacks and E01, Carl Rosendahl, Peter Heacox and Enrique Mandujano.

the amount of exposure of the film, depth of field, optical aberrations, vignetting and field of view.

Shutters are mechanical or electronic devices designed to limit the exposition to a finite range of time. Once opened, and since the media integrates all light hitting its surface, a continuous view of a moving object is projected to different areas of the image plane. Additionally, since transitioning from the closed to the open state and back takes a finite amount of time, its effective aperture size and shape evolve over time. The resulting interactions between light, diaphragm, shutter, exposed media and object motion produce motion blur.

The exposure process can be formalized using Equation 4.1. During scene capture, $I(\omega)$ represents the contents of the image plane when the scene is seen in the direction $\omega$. Captured light is the result of the integration of the incoming radiance $L$ during the exposition time when the shutter is open $\Delta T$. $f(\omega, t)$ models the influence of optics, shutter, aperture and film.

$$I(\omega) = \int_{\Delta T} f(\omega, t) \, L(\omega, t) \, dt \tag{4.1}$$

The previous equation already gives clues about the characteristics of the resulting images. If a rendering process generates each image in a sequence as an instantaneous snapshot ($\Delta T$ is zero), quickly moving objects will miss parts of their trajectories and the continuity of the sequence will be lost. Even if the shutter range is long enough, different exposures may be separated by gaps. This may result in double images and ghosting. In the previous equation, this situation is represented by $\Delta T$ bigger than zero but secuential $I(\omega)$ separated with unsampled time ranges.

| CATEGORY | | METHODS |
|---|---|---|
| Analytic | | [123] [81] |
| Geometric substitution | Generic | [200] [35] [76] |
| | Motion Hints | [275] [163] [240] [83] [85] [109] [216] |
| Texture clamping | | [175] [145] |
| Monte Carlo | Distributed Ray tracing | [45] [89] [65] [182] |
| | Evolutions | [251] [29] [92] [261] [171] [194] |
| | Accumulation buffer | [123] [90] |
| | Frameless rendering | [11] [215] [54] [55] |
| | Specific to a primitive type | [192] [117] [4] |
| Postprocessing | Generic | [220] [191] [152] [40] |
| | Motion fields | [225] [19] [224] [285] [209] [230] [256] |
| Hybrid | | [46] [237] |
| Mechanical and optical | | [220] [77] [139] [241] [183] |

Table 4.1: Motion blur rendering methods classified according to the categories explained in section 4.3

Rendered image sequences usually look more natural when they integrate motion blur. Three basic reasons are commonly accepted for this [77]:

- Firstly, images recorded using real cameras automatically incorporate motion blur. Even natural images as seen by our eyes contain a similar effect, motion smear [22]. As such, motion blur is part of what we expect to find in any moving image.

- Secondly, sequences of images lacking motion blur can contain strobing artifacts. This makes the predictions of the human visual system (HVS) difficult and object trajectories may not be understood as continuous paths.

- Finally, discretization of continuous signals is a by-product of many imaging algorithms. This results in different forms of aliasing artifacts that are usually easily noticed by the HVS. Reducing temporal aliasing by avoiding time undersampling is the main target of motion blur rendering techniques. An overview of these will be given in section 4.3.

Although this work focuses on the problems associated with the generation of motion blur, there are many complementary fields. Interested readers can find useful references for the algorithms used to simulate realistic camera lenses [10], depth of field [9], film emulsion grain and exposition [75] or the dynamic range of recording devices [203]. The inverse problems, such as image deblurring and restoration [127] have also been covered elsewhere.

## 4.2 FORMALIZATION OF THE MOTION BLUR PROBLEM

Different motion blur rendering algorithms have been proposed based on several formal models [220, 191, 152, 159, 225]. Probably, the most widely accepted describes a motion blurred image as a spatio-temporal integral that includes independent terms for the geometric and shading functions. Equation 4.2 mathematically states this idea. The expression follows the formulation of Sung et al. [237], but similar descriptions have been used by Meredith [157].

$$I_{xy} = \sum_{l} \int_{\Omega} \int_{\Delta T} r(\omega, t) g_l(\omega, t) L_l(\omega, t) dt d\omega \tag{4.2}$$

In this equation, $I_{xy}$ represents the contents of the image pixel with coordinates $(x, y)$ and $\Omega$ is its corresponding subtended solid angle. Independently of their geometrical representation, the overall contribution of all primitives in the scene is considered by iterating over each individual object $l$. $g_l(\omega, t)$ is a geometrical term that accounts for occlusions between objects. Its value is 1 if object $l$ is directly visible in the direction $\omega$, 0 otherwise. As we have seen, shutter shape and efficiency, lens aberrations and film influence the final image. The reconstruction filter $r(\omega, t)$ accounts for their overall effect. In general, this term cannot be split into pure spatial and temporal components. However, an approximation that is widely accepted is to replace it with the product of two filters $r(\omega, t) = r_s(\omega) r_t(t)$, where each term exclusively depends on one of the dimensions. $L_l(\omega, t)$ represents the radiance of object $l$ without explicitly establishing the method by which is calculated.

In order to account for the complex spatio-temporal relationships taking place in an animated scene, all terms are evaluated at an instantaneous time $t$ over the aperture time $\Delta T$, and over the solid angle $\Omega$. In some cases and depending on the desired filter footprint, $\Omega$ can represent narrower or wider solid angles than the one defined by the pixel.

Although $g_l(\omega, t)$ is a binary function that assumes that a single object is visible at $\omega$ at a specific instant, this should not be considered a limitation of the model since the radiance term $L_l(\omega, t)$ already accounts for cases where images of several objects can be seen simultaneously. Kajiya's rendering equation [110] is an example of formulation for light interaction that can be used as a method to resolve complex transparent, translucent, reflective, refractive and shadowed phenomena.

Different geometrical descriptions can be included in Equation 4.2, even when those are based on non traditional representations. The term $L_l$ simply accounts for the final value of the radiance of object $l$ independently of the nature and complexity of the phenomena. Something similar occurs with the geometrical term $g_l$. With this in mind, the original formulation can be used to render scenes where, for example, surfaces based on polygonal, analytical or implicit descriptions are mixed with objects built from liquids and gases.

Due to reasons we will discuss, the previous formulation is not always practical to calculate. In the following sections we will see how it can be adapted so it can be implemented as an algorithm suitable to be run in a computer.

## 4.3 motion blur rendering algorithms

Even if many methods can gather subpixel information to produce spatial antialiasing, they are not designed to find the subframe information on which temporal antialiasing relies. Any methods that are not aware of the time dimension, when applied to each frame of a sequence, may produce images that lack any temporal coherence, and display a myriad of different artifacts.

Motion blur rendering algorithms are designed to handle the degree of complexity associated with the addition of the time dimension. However, even if they can produce temporally correct images, their computational complexity may be unacceptable. As observed in the photographic snapshots of Figure 4.1, the amount of fine detail is drastically reduced when the images integrate motion blur. When considered from the perspective of the Fourier theory [65], common visual phenomena suffer similar transformations under the presence of motion blur and their spectra is confined to a specific region of the domain. This supports the initial intuition that motion blur produces a reduction of the spatial complexity and frequency contents with respect to an equivalent instantaneous unblurred image.

While the mathematical framework has been already described, different algorithms use alternative methods to evaluate the expressions associated. Knowledge of the physical phenomena may also be exploited. In all cases, it is desirable to reduce the impact of any simplifications, so it can be applied in a broader set of situations and still be accurate. In the next sections we will describe and group the existing methods based on an analysis of their assumptions, limitations and associated visual artifacts.

### 4.3.1 *Overview*

From the description in Equation 4.2, two different problems must be addressed. Firstly, a good approximation for the evolution of the geometry needs to be found; and secondly, the shading of the objects needs to be accurately simulated. By considering the approaches and assumptions used to solve each of these problems, the following categories can be established. They are also summarized on table 4.1.

- **Analytic methods**, while frequently relying on heavy assumptions, use a closed form solution that can be exactly evaluated. Section 4.3.2 covers them in detail.

- In section 4.3.3, **geometric substitution methods** use alternative primitives that represent the original geometry and its evolution along time. These methods are ideal to be implemented in real time frameworks or when approximate motion hints suffice.

- **Monte Carlo methods** have received a great deal of attention due to the fact that a wide range of phenomena can be modeled inside a flexible and technically simple framework. However, stochastic point sampling methods are not deterministic and tend to produce artifacts due to insufficient sampling. These techniques and its associated methods are described in section 4.3.5.

- **Postprocessing methods** extend the information of an image snapshots to the whole exposure range. Temporal information is extracted directly from the render engine or extrapolated from different frames of the sequence. A detailed description can be found in section 4.3.6

- A number of models have been combined together to solve specific aspects of the general problem. Section 4.3.7 contains a description of **hybrid algorithms** which have proved to be the best options when an open and unconstrained problem needs to be solved. This is probably the main reason for their wide acceptation in the visual effects and film industry.

- Finally, section 4.3.8 contains recent techniques that include **physically accurate models of the imaging device**. The availability of CMOS/CCD chips in digital video cameras and the need for better composition of rendered images into real footage are giving these solutions an increasing relevancy.

### 4.3.2 *Analytic methods*

Analytic methods are among the pioneering solutions to render motion blur. They replace Equation 4.2 with closed form expressions that efficiently provide an exact value of the pixel's radiance. In general, lighting equations are highly non-linear functions that may not have analytical equivalents so this family of methods are only valid under strong compromises.

The first example of this approach is the work of Korein and Badler [123]. Their algorithm finds a continuous function that represents the time intervals when the projection of an object covers a given image plane pixel. Once those visibility ranges $\tau_l$ are found, the problem is

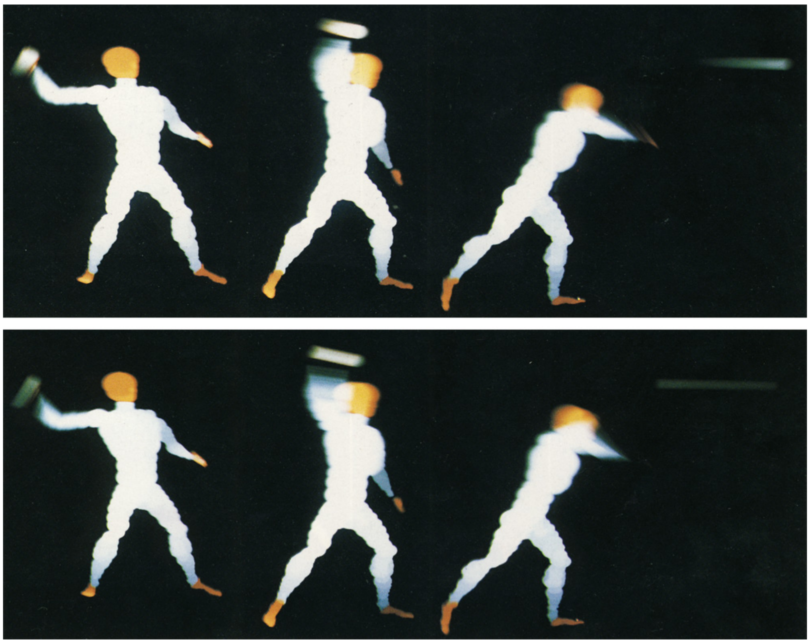

Figure 4.2: Method of Korein and Badler [123]. Top row images use a filter that emphasizes later movement, while the bottom row shows the same motion using a a temporal box filter.

reduced to determining a single shading sample for the pixel in $\omega$ at a time $t$. Equation 4.3 models this approach:

$$\begin{aligned} I_{xy} &= \sum_{l} \int_{\Delta T} r(\omega,t) g_l(\omega,t) L_l(\omega,t) dt \\ &= \sum_{l} \int_{\tau_l} r(\omega,t) L_l(\omega,t) dt = \sum_{l} \tau_l L_l(\omega,t) \end{aligned} \tag{4.3}$$

Since their implementation is limited to objects represented as spheres, polygons and polyhedra, and their trajectories are approximated using piecewise linear interpolation, the corresponding projections in motion are determined by simple primitives, see Figure 4.2. Occluded objects are stripped out from the calculations using depth sorting and a single object is considered for each time range and pixel. This method focuses on solving geometric temporal aliasing and, since it uses a single sample $L_l$, changes in shading are beyond the capabilities of the algorithm.

An alternative method, adapted for polygonal primitives, is described by Grant [81]. Dynamic 3D polygons are converted to static 4D polyhedra that describe the volumes swept by the geometry. Visibility is determined by scan-converting those continuous 4D (x,y,z,t) primitives into continuous visible 3D (x,y,t) geometry using a temporal extension of Catmull's visible surface algorithm [35]. Non occluded geometry is rendered using an extension of Feibush et al. analytical filtering algorithm [70]. Intersecting and non intersecting objects under image space translational and scaling motion can be represented with this approach.

Grant's polyhedral algorithm can also be included in the category of methods that use geometry substitution. Those methods are explained in the following section.

### 4.3.3 *Geometric substitution*

Methods that use geometric substitution are fundamented on the idea that, in order to render motion blur, each moving object can be replaced by static geometry built from the original primitives and their motion. Time-sampling is no longer needed, as the new geometry contains an implicit description of the temporal evolution of the shape, shading and trajectory. In general these methods can be described by adapting the formulation from Equation 4.2 to:

$$\begin{aligned} I_{xy} &= \sum_{l} \int_{\Omega} \int_{\Delta T} r(\omega, t) g_l(\omega, t) L_l(\omega, t) dt d\omega \\ &\approx \sum_{l'} \int_{\Omega} r_s(\omega) g'_{l'}(\omega) L'_{l'}(\omega) d\omega \end{aligned} \tag{4.4}$$

In Equation 4.4, each geometric object $l$ is replaced by an alternative primitive $l'$. Functions $r$, $g_l$ and $L_l$ are also swapped with $r_s$, $g'_{l'}$ and $L'_{l'}$, their time independent counterparts. This formulation is just an approximation that drops the integration over time while keeping the corresponding image space evaluation over the domain $\Omega$. Different algorithms make different interpretations of this expression and, as we will see, each approach is associated with a specific set of implications and varying flexibility.

One of the earliest proposals is made by Reeves. He establishes the fundamentals of particle systems [200]. Motion blur is achieved by replacing each moving particle by an antialiased line segment describing its trajectory during the exposure time. Their color is based on a single shading sample. Since they are rendered as point light sources, all particles projected to a given pixel can be simply added together without accounting for visibility, that is, $g'_{l'}$ is always one. The original algorithm is not applicable to particles that use different shapes or textured billboards and layer composition is needed to solve occlusion with other types of primitives. An image rendered with the original technique can be seen in Figure 4.3.

Catmull's visibility algorithm [35] focuses on finding which polygons are visible from a given pixel. Each primitive is decomposed into a set of micro-polygons whose shading is weighted with a circularly symmetric filter [70] and composited using depth sorting. Spatio-temporal filtering is achieved by using the original filter on geometry that has been deformed according to the speed of each of

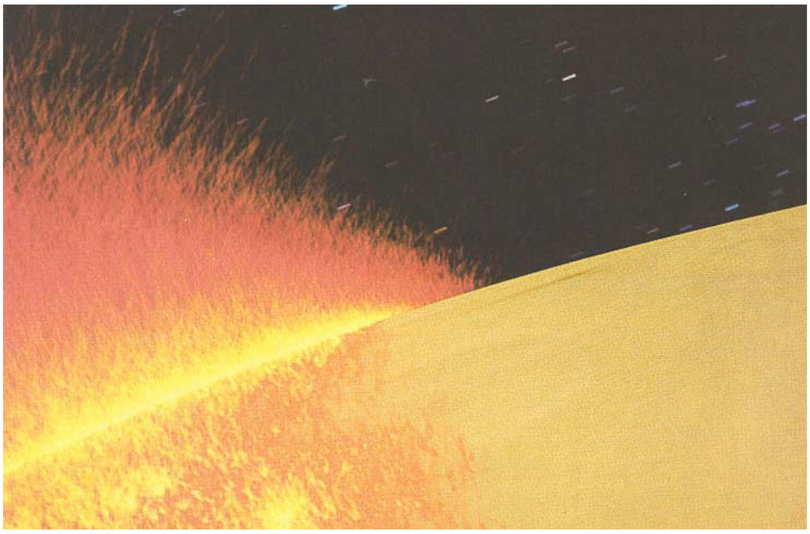

Figure 4.3: Particle systems using the motion blur technique described in [200]

its vertices. Primitives whose vertices are moving at different rates, following different paths or non-linear screen space speeds can be accurately handled.

In the space-time raytracer [76], scene primitives are represented as static entities in a four-dimensional space. Since the 4D representations implicitly contain the time dimension, the scene only needs to be evaluated during the algorithm's initialization where the evolution of each object can be fully determined. Intersection tests are based on tracing rays in the sampled spacetime direction. Since the method focuses on determining the $g_l'$, term, the evolution of the shading function needs to be independently handled. As in the original ray-tracing algorithm, motion blur is calculated by shooting rays at different times.

#### 4.3.3.1 *Motion hints*

A subset of the substitution methods, while not inspired by physically accurate models, are designed to produce visually appealing motion blur. They assume that, at high frame rates, observers cannot distinguish between correct blur and rough approximations.

These algorithms have certain advantages that make them interesting. Since they use simplified models for geometry, shading and visibility, they can be efficiently executed. The integration of these methods into existing pipelines is also simplified by the fact that original and replacement geometries use the same basic representations. These are the main reasons why these methods have been widely popular inside real time rendering frameworks.

Wloka and Zeleznik [275] use a representation of the volume described by the original moving geometry. Each object is divided into a leading and trailing polygons and a contour is built with the edges

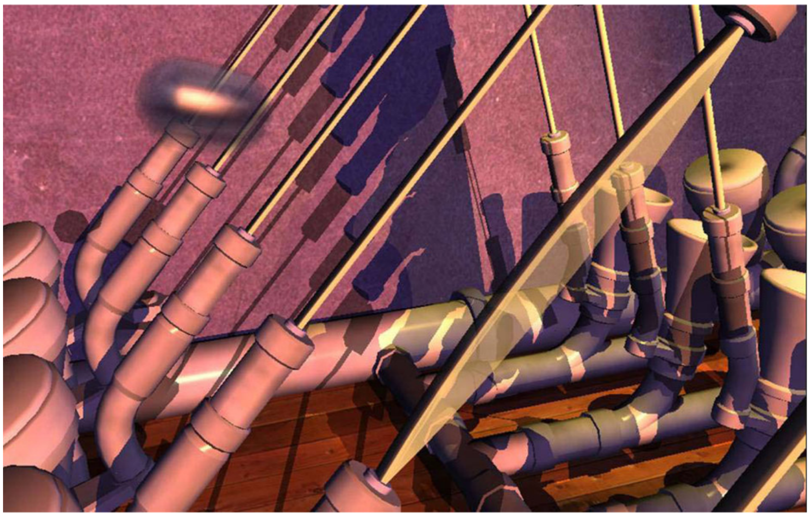

Figure 4.4: Real time motion blur using the technique of Tatarchuk et al. [240]. Specular highlights and environment reflections are spread according to the speed of ball. The original spherical shape is also elongated along the direction of motion.

connecting them. An in-between surface is generated by sweeping this contour along a piecewise linear approximation of the trajectory of the object. Leading polygons are drawn fully opaque, while trailing and connecting parts use transparency determined by their motion vectors. This method cannot properly handle general rotation, object scaling or independent vertex deformation.

An extension to rendering deforming polygonal meshes and handling arbitrary rotation is presented by Jones and Keyser [109]. A shell is built using the vertices of the motion silhouette at different time snapshots. Age decaying opacity is used to render it. Since this method is based on extruding the silhouette, it cannot handle textured geometry or complex shading. Visibility computation is replaced by depth tests in the raster engine which may result in backward facing polygons being rendered and self-intersections between object and trail.

Programmable motion effects [216] build bilinear patches defined by the trajectory of a set of seed points placed on the surface of the object. Based on this 4D representation, the concept of surface shader is extended to a flexible postprocess that can produce speed lines, stroboscopic effects, temporal offsetting as well as stylized and realistic motion blurring. Figure 4.5 shows a range of effects achieved with this technique.

The method of Green [83] uses multiple OpenGL passes. An initial pass renders a sharp image of the scene and a per pixel velocity field. These velocities are used to expand and render the original objects. The resulting image is blurred using multiple texture samples that are later applied to the replacement geometry.

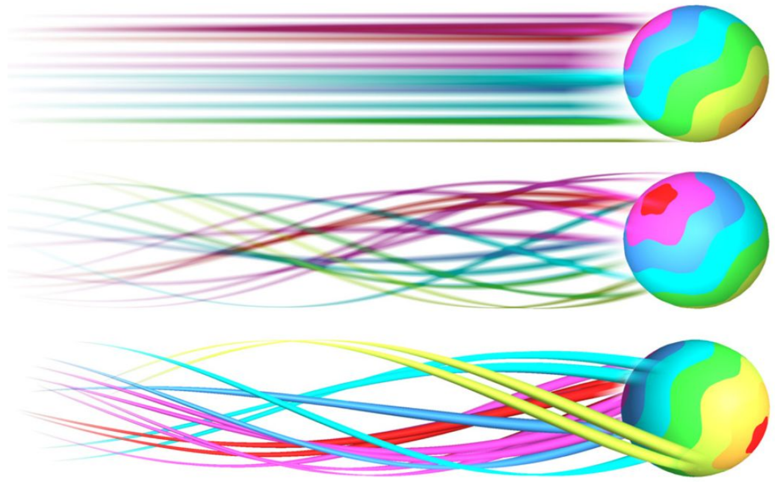

Figure 4.5: Different motion trail styles achieved with the technique of Schmid et al. [216]

A related approach is followed by Tatarchuk [240]. Moving geometry, in this case polygonal spheres, are replaced by capsules whose lengths are dependent on its image-space speed, with their orientations matching the direction of the motion. As seen in Figure 4.4, the original shading is modified to spread the energy of specular highlights among the pixels they would move across during the exposure time. Object transparency and environment mipmaps are also modulated based on the speed of the object. Each pixel is independently shaded with a single sample that accounts for the whole shutter range, with occlusion being handled by the hardware rasterizer.

Point based rendering can also integrate motion blur by using the method of Guan and Mueller [85]. Instead of motion blurring all the voxels of the volume [163], temporal antialiasing is performed on an isosurface extracted using a simplified version of the EWA algorithm [291]. During render, a Gaussian temporal filter is assumed and an ellipsoid represents the projection of the original 2D round splat and its linear motion trail. Figure 4.6 shows different effects that can be achieved by modifying the sizes, colors and number of ellipsoids generated per point. The original approach cannot resolve varying shading and situations where geometry is partially hidden, intersect or change its relative positions. Recently, the solution for time-varying point samples has been described [97]. Heinzle's et al. algorithm also assumes piecewise linear trajectories, but a continuous reconstruction of the spatio temporal functions is handled using 3D Gaussian kernels. While Guan's method renders a static image that is composited with the corresponding motion trails, the new method incorporates both the renders of the points and its trails into the sampling of the kernels.

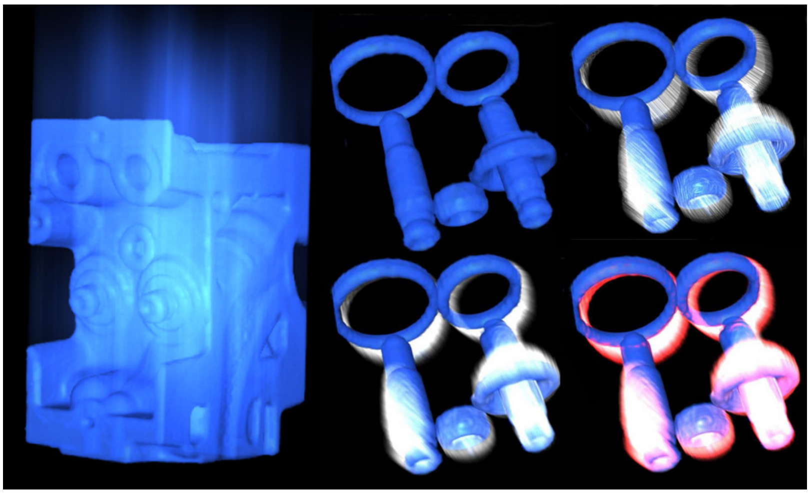

Figure 4.6: Motion blur applied to point based rendering [85]. Left image shows a volumetric model of an engine moving vertically with its corresponding motion blur. Images to the right show a different model rendered without motion blur (left top), motion as streak lines (right top), and two different artistic effects applied to the motion trails (bottom)

### 4.3.4 *Texture clamping*

Aliasing artifacts can be thought of as a mismatch between the frequency contents of the signal and the rate selected to sample it. Lighting functions, occlussions and their evolution in space and time produce sharp changes and discontinuities, which boost the frequency contents of the original shading functions. Techniques like supersampling handle these high frequencies by using extra samples (see section 4.3.5). A more efficient alternative consists in reducing the bandwidth of the original signal so a reconstruction that is free of aliasing can be achieved with less samples.

Textured objects can be anti-aliased with the method of Norton et al. [175]. Assuming perspective projections are linear transforms, the shape of an image-space box filter can be approximated by an object-space parallelogram. With this in mind, a texture can be convolved in the Fourier and spatial domains with a low-pass filter whose support is defined by the texels covered by the projection of each image pixel. This formulation extends to the temporal domain by taking into account the evolution of this projection. More recently, Loviscach uses anisotropic texture filtering to perform a similar procedure [145]. When a polygon in motion is rasterized, each of the resulting pixels define a parallelogram in texture space. By carefully selecting the size and orientation of the filter, hardware texture fetches can return values that incorporate motion blur.

It should be noted that none of these methods generate motion blur per se, but focus on alleviating the problems associated with temporal and spatial aliasing. Even if they can efficiently solve problems for textured geometry, shading or lighting functions can themselves produce aliasing. Filtering after shading may reduce these problems, however directionally dependent effects may need to be frequently re-evaluated to accommodate for scene updates. Also, occlusions are not considered, so situations where moving polygons overlap need to be solved with a complementary algorithm.

### 4.3.5 *Monte Carlo methods*

In those cases where no analytical expression or an alternative geometrical description can be used, Equation 4.2 can be approximated using numerical methods. Given the multidimensional nature and unpredictability of the integrands, standard methods relying on the smoothness of the functions cannot always be used. However, Monte Carlo methods are designed to handle high rates of change and discontinuities. Instead of regular artifacts that are usually easily noticeable, errors due to low levels of sampling are shown as random patterns and noise.

In the following section we will describe different stochastic algorithms grouped under the generic name of Monte Carlo methods. Since our focus is the production of motion blurred images, we assume a basic knowledge of the fundamentals of numerical methods in computer graphics. Interested readers are referred to the excellent descriptions that can be found, among others, in [129], [253] or [265].

#### 4.3.5.1 *Distributed ray tracing*

As seen in section 4.2, a visual phenomenon is parametrized in a highly multidimensional space. Traditional ray tracers [272] focus on producing image space antialiased pictures. For phenomena such as soft shadows, translucency, glossy reflections and so on; any additional dimensions need to be explicitly and independently sampled [43, 272, 273]. This is cumbersome and frequently results in an excessive number of samples being calculated. With Cook et al. distribution ray tracing [45] each ray is stochastically allocated so all dimensions are simultaneously sampled. Motion blur is solved by sampling the time domain and as seen in Figure 4.7, different effects can be modeled: glossy reflections, translucency, penumbras and depth of field are rendered by evaluating the specular distribution function, the directions of the transmitted rays, the solid angle of light sources or the camera lens area. Due to the elegance and simplicity of the method, distributed ray tracing has received extensive attention and it has become one of the most popular approaches in industry and academia.

In general, distribution ray tracing approximates Equation 4.2 using the following expression:

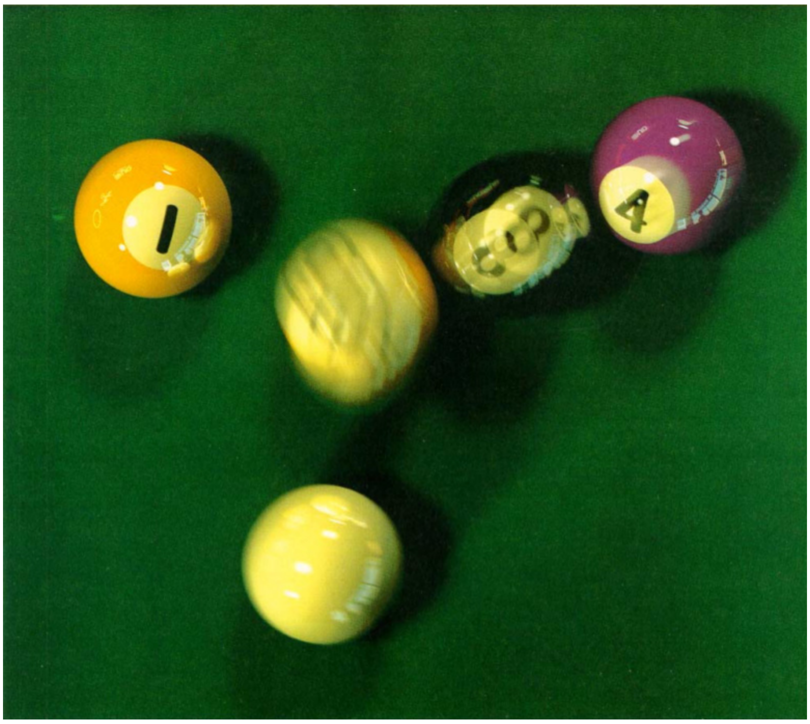

Figure 4.7: Image rendered using distribution ray tracing [45]. This technique can simulate high quality motion blur including shadows, penumbras and specular reflections where the movement is non linear during the exposure time. ©1984 Thomas Porter, Pixar

$$I_{xy} \approx \frac{1}{N_j} \sum_{j}^{N_j} i_d(\omega_j, t_j) \tag{4.5}$$

$$i_d(\omega_j, t_j) = \sum_{l} r(\omega_j, t_j) g_l(\omega_j, t_j) L_l(\omega_j, t_j) \tag{4.6}$$

In Equation 4.5, each pixel is calculated as a sum of $N_j$ discrete point samples. For simplicity, we will consider that motion blur is based on sampling locations in the spatio-temporal domain only. A sample $i_d(\omega_j, t_j)$ accounts for the contribution of each object $l$ as seen in the direction $\omega_j$ at an instantaneous time $t_j$. This value can be calculated by different means, but in general it is the result of evaluating the visibility $g_l$ and radiance $L_l$ functions for each of them. These contributions are weighted with the value determined by the filter $r(\omega_j, t_j)$. The resulting samples can be simply added together and averaged, or as we will see, more sophisticated methods can be applied.

Increased dimensionality accentuates the problems of point sampling. Since distribution ray tracing is not tied to a specific sampling technique, the problem of how and where samples are generated has received great deal of attention. Simple methods like uniform and regular adaptative sampling [47, 272] while successfully used

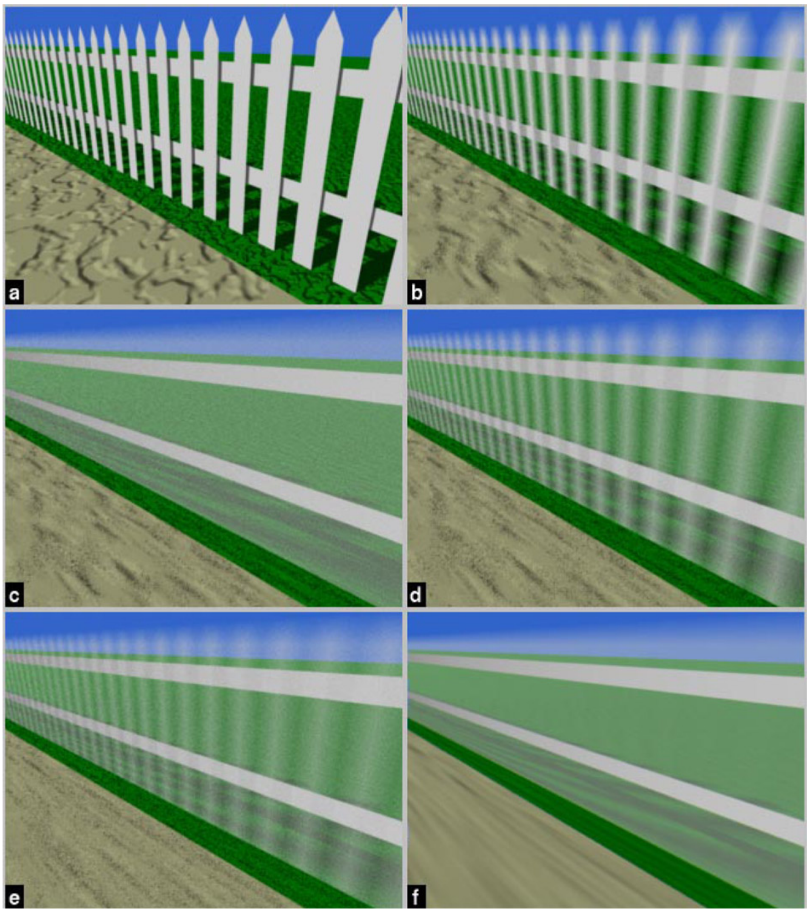

Figure 4.8: Scene rendered with a static camera (a). Images (b) to (e) use a camera that moves progressively faster. A box reconstruction filter creates aliasing patterns in images (d) and (e). The patterns are corrected in image (f) by using a gaussian filter. Reproduced from [50].

by standard ray tracing produce poor results with distributed ray tracing. Stochastic methods like minimum distance Poisson or jittered sampling [44, 60], precomputed sampling patterns [44], adaptative sampling [110, 135, 158], importance sampling [63] or stratified sampling [160] are designed to reduce those problems. As seen in Figure 4.8, alternative reconstruction filters can combine a given sample set resulting in varying qualities [161, 50]. The topic of sampling and reconstruction has been discussed in detail [189, 63].

Those methods which produce estimates as seen in image space may produce noisy reconstructions and miss image features. Alternative methods, like Metropolis light transport [251] or the recent Hachisuka's et al. [89] multidimensional adaptive sampling, place samples based on the changes of the functions in the original multidimensional space. The latter evaluates the contrast of the rendering equations to determine the sampling level and analytically reconstructs the original function in all but the image dimensions. A Riemann sum is used to integrate the illumination of each pixel.

Several studies have considered the rendering problem from the perspective of the frequency domain [60, 37, 62]. Recent results [65] have shown new methods to design sampling and reconstruction filters. In essence, for common effects such as object and texture motion, rotations of both the BRDF and lighting and non static shadows, the influence of motion blur can be approximated by a shear filter. The frequency contents and ideal sampling levels can be estimated for each image pixel based on their respective image space velocities. Based on the same data, an improved reconstruction can also be obtained by tuning the orientation and extent of the filter.

In contrast to previous methods that exhaustively sample the integral to reconstruct a smooth image, adaptive wavelet rendering [182] directly estimates a smooth function in the wavelet domain. With this multi-scale representation, coarse-scale wavelets represent large smooth regions while finer-scale ones represent edges and small details. Adaptive sampling can efficiently detect sharp changes in image-space as well as smooth image regions with high variance in the non-image-space dimensions. The algorithm reconstructs smooth images even in regions of high variance, scales well for high dimensional problems but it may show ringing and over smoothing artifacts.

#### 4.3.5.2 *Evolutions of naive distribution ray tracing*

The original distribution ray tracing method has been improved to produce results more efficiently with increased accuracy. In this section we will focus on exploring those techniques.

Metropolis light transport [251] is an evolution of the original bidirectional path tracing [130, 252]. Paths are built from the light sources to the eye, and different mutation strategies are used to generate variations of these. This method naturally extends to calculate motion blur when the mutations include the temporal dimension.

Caching techniques, like Jensen's photon mapping increase performance by storing and reusing lighting results [103]. However, since they produce estimates that are descriptive of a given value in the time domain, they cannot represent objects with changing shading, motion or shape. An extension of this technique [29], handles motion blur by assigning a random time to each photon and its descendants in the light path. The final reconstruction pass relies on a filter that averages the photons of the spatio-temporal neighbourhood of the pixel being calculated. In Figure 4.9, motion blur is incorporated to a set of shadowing, reflective, refractive and caustic effects.

The original Lightcuts algorithm [260] has been expanded to support temporal sampling [261]. The new method builds on the original discretization of the lighting integrals and graph partitioning. However, the temporal domain is represented with a fixed set of time instants

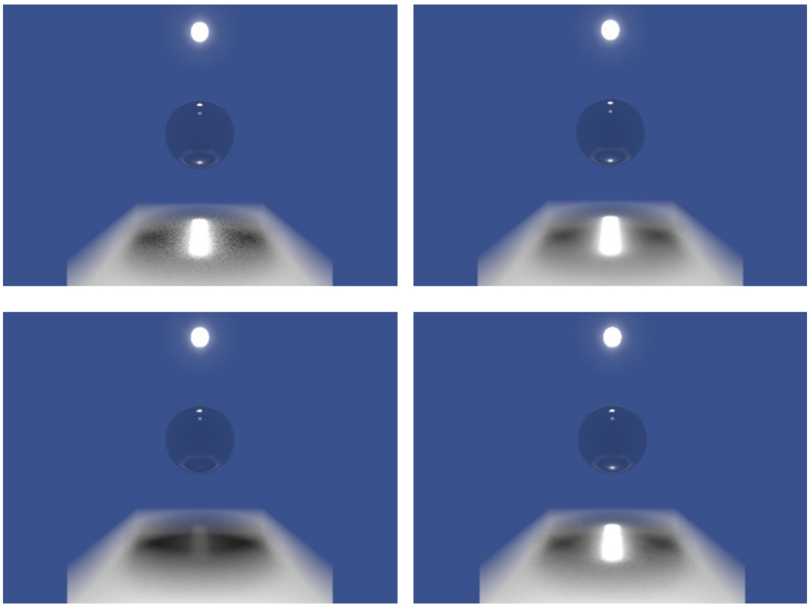

Figure 4.9: A transparent sphere projects a caustic on a diffuse cube that moves downwards. Left-top: Noisy image rendered using path tracing with 10000 random paths per pixel. Right-top: Accumulation buffer averaging 20 images placed at equal intervals. Bottom-left: Photon mapping using standard estimation that underestimates the final value of the radiance. Bottom-right: Photon mapping using the time dependent radiance estimation from [29]

and any light interactions are limited to those subintervals. Even with these simplifications, the method can provide high quality results while drastically reducing the cost and the noise produced. Figure 4.10 shows examples of the results.

Other methods focus on exploiting spatio-temporal coherence [92]. Since a given area of an object can usually be seen within several consecutive frames, novel views can be built by reprojecting existing samples. The algorithm of Cabral and Leedom can be used to account for the image space projection [26]. Motion blur is computed by determining the trajectory of each sample and accumulating their energy contribution over the motion path. Like similar methods [19], it uses linear constant speed motions, but in contrast, precise trajectories are computed from object space and not image space data. While reprojection clamps the shading functions, the technique is only available for non deformable objects, cannot directly model view dependent effects and due to the simplified nature of the paths, may produce occlusion artifacts.

Using a similar approach, real-time reprojection caching [171] allows supersampling at high frame rates. In this case, cached values are reused using hardware texture filtering. Noise is reduced while keeping storage requirements low by using a recursive low-pass filter that averages new samples with the existing ones.

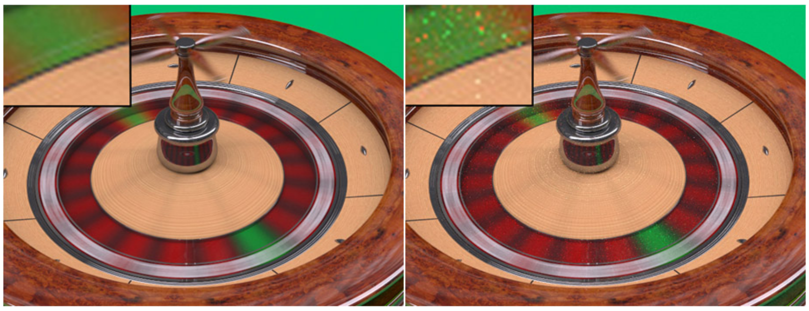

Figure 4.10: Comparison of two renders performed using multidimensional lightcuts (left) and traditional Metropolis light transport (right). Lightcuts' image is less noisy and can be rendered faster (15x). Reproduced from [261]

Decoupled shading [194] leverages the intrinsic coherence of depth and motion blurred scenes by using a shading cache. The technique densely evaluates the visibility function and reuses any previously calculated shading samples. The defining parameters of each visibility location are hashed so they deterministically map to a shading value. On a cache miss, the value is calculated on demand and stored so they can be later retrieved. This method is built on the assumption that shading does not change rapidly and it is view independent, so a single value is representative of the whole shutter range. Shading rates are reduced respect to other methods that shade before visibility or those in which visibility and shading samples have a one-to-one relation.

It should be noted that exploiting spatio-temporal coherence is not only applied to calculating motion blur. Other methods have relied on similar approaches for the rendering of sequences of images where the interest is obtaining high frame rates and not temporal supersampling [259, 244, 238, 227].

#### 4.3.5.3 *The accumulation buffer*

A number of methods have been designed to leverage the use of accumulation buffers to super-sample multidimensional spaces. These hardware framebuffers are specially suited to efficiently weight and combine samples in image space using high precision computations.

Korein and Badler [123] include the description of an algorithm that can be considered a precursor to the accumulation buffer method. It calculates temporal anti-aliasing by supersampling the temporal domain using several independent renders performed at different times. Full images are stored externally and filtered down to produce the resulting image.

Haeberli and Akeley [90] weight and add several passes computed using a single sample per pixel. By updating different parameters between passes, the resulting image is spatially antialiased, integrates

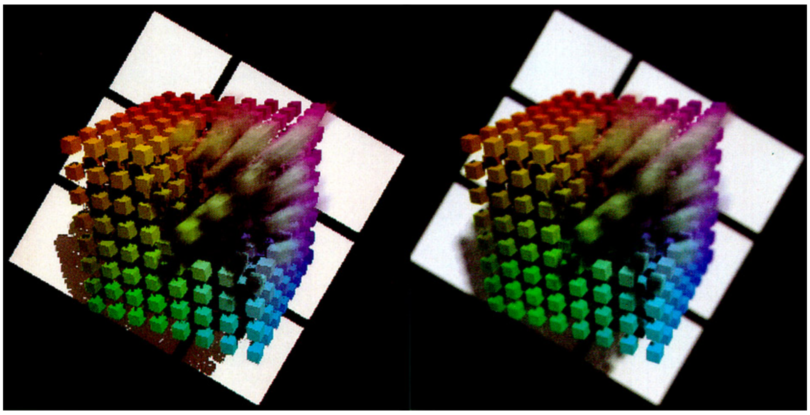

Figure 4.11: Images rendered with the original accumulation buffer method [90]. Left image contains a motion blurred image composed from 23 different passes. Right image uses 66 individual images to simultaneously render motion blur, depth of field and soft shadows.

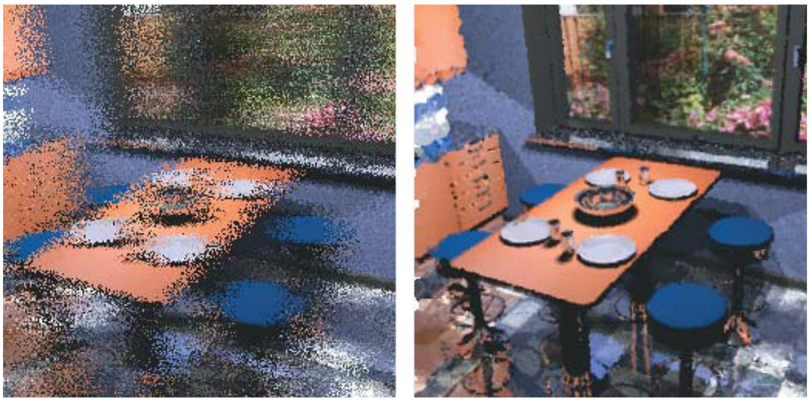

Figure 4.12: Images generated using frameless rendering. Left, naive frameless rendering where each sample is displayed as soon as is calculated. Right, the same image using adaptive frameless rendering [55]

depth of field, soft shadows, anisotropic reflection or filtered texture maps. Motion blur, as can be seen in Figure 4.11, is the result of combining scene renders at time instants are that equally placed in the shutter range. A final image is the result of N subframe passes or, in the case of repeated integration [94], every two passes.

Before the appearance of flexible programable GPUs, this was the preferred method to leverage the parallelism available in hardware frameworks. Since the solution comes by iteratively applying the original algorithm, the implementation is simple and avoids the overhead of elaborated control logics [213]. However this method shows poor results with scenes with high temporal correlation and does not scale well with scene complexity.

#### 4.3.5.4 *Frameless rendering*

Frameless rendering [11] is a paradigm designed to reduce the latency of interactive display systems. Unlike traditional approaches that expose frames after they have been fully computed, frameless rendering derives each pixel from the most recent instant in time and immediately shows it. As a consequence, high motion scenes show an effect that resembles motion blur, while static ones converge to the same solution as a traditional renderer does.

The original technique randomly selects the samples to be rendered. Further evolutions use a probability distribution function based on the age of the pixels [54] or even closed loop adaptative sampling to steer computation to regions undergoing significant changes [55]. Related techniques use one temporal sample per scanline [178], dynamically modify frame rate [276] or adapt frame rates for each individual object [202]. Perceptually based heuristics have also been envisaged [283, 215]. Image plane pixels can be reconstructed from a single sample [11], a temporally weighted set of samples corresponding to a slice of time or even from a 3D volume in which the filter is adapted to local sampling density and color gradients [55]. Examples of the resulting images can be seen in Figure 4.12.

Since computation speed is bounded, these methods trade temporal supersampling with spatial undersampling. Scenes with high spatio-temporal coherence are well suited for this schema. Otherwise quick changes are shown as a dissolve filter between scene snapshots. As a side effect, traditional strobing effects associated with temporal low sampling, are replaced by less perceptually disturbing noise.

#### 4.3.5.5 *Methods designed for specific types of primitives*

Distribution ray tracing has been traditionally oriented to handle polygonal meshes. With frequency, other types of primitives are rendered after a conversion to these primitives. This section covers the methods that have focused on using different geometric representations without requiring any previous transformations. In some cases, using those alternative models open the door to speed and quality improvements.

Preston implements motion blurred distribution tracing for NURBS surfaces [192]. His algorithm expands the original parametric representation so it represents the surface and its temporal evolution. The surface's control points are stored as slices representing different moments in time and any surface configuration is constructed as an in-between of two of them. Motion blur is achieved by sampling the resulting surface at different time lapses.

Fluids, based on non polygonal surfaces, can be rendered with Eulerian motion blur [117]. Since newtonian inertia is not a valid assumption, the status of the system cannot be inferred using inter-

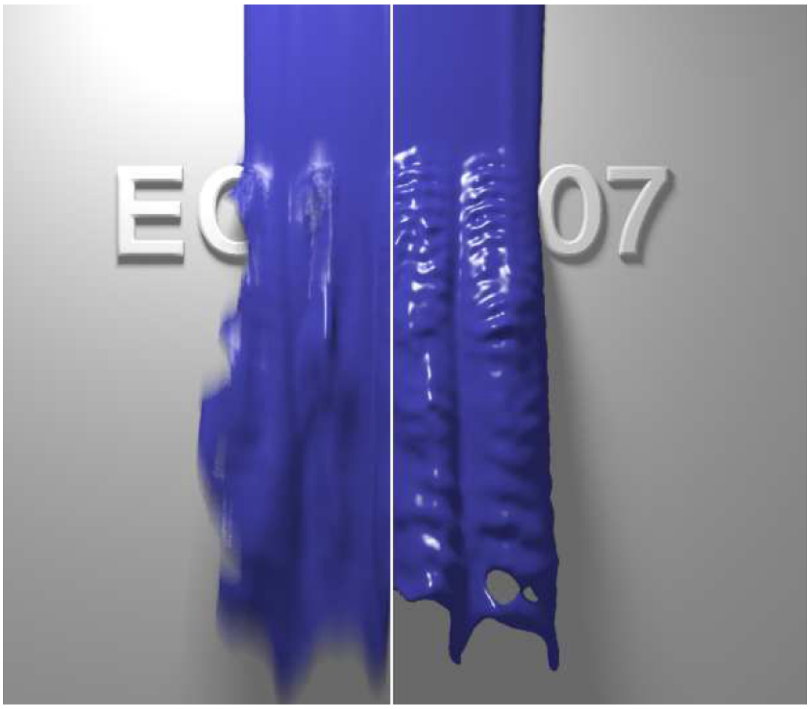

Figure 4.13: Images comparing an Eulerian system rendered with and without motion blur. Left image has been rendered using the method of Kim and Ko [117]

polation between known states. Kim's approach is based on defining intermediate states based on sparse level-sets, density data and the fluid's semi-lagrangian advection. From that point, existing Monte Carlo methods can be used to render the fluid [69, 173]. Figure 4.13 shows the result of the algorithm.

Rendering motion blur for voxel sets, rigid and time-varying point clouds has already been described in section 4.3.3.1. Guan and Mueller briefly introduce an alternative approach [85] using optical flow or mpeg-like motion estimation to determine motion flows inside the evolving voxel grids. However this method has not yet been implemented neither described in detail.

Akenine-Möller et al. implement real time rasterization of triangles [4] whose motion is described as a continuous function of their vertices at two moments in time. Using this representation, any surface attribute can be interpolated using GPU tiled rasterization, accumulation buffers and piecewise linear approximation of the trajectories. Motion blur, soft shadows and depth of field can be rendered using this framework. Inspired by the shadow mapping algorithm [273, 201], this method can render motion blurred shadows without relying on ray tracing. Unlike the original algorithms or deep shadow maps [141], see image 4.14, time dependent textures are not limited to static scenes or static shadow receivers.

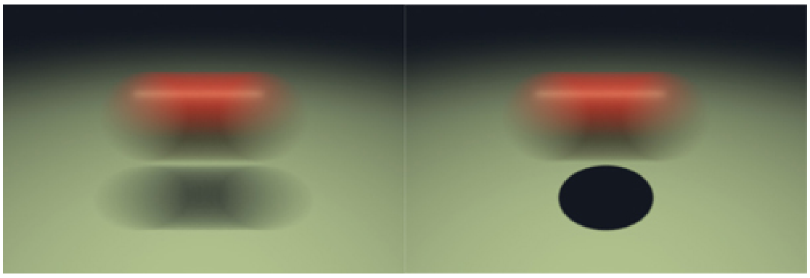

Figure 4.14: Left image shows a motion blurred shadow rendered using deep shadow maps [141]. This technique can only be used when the object receiving the shadow is stationary respect to the light emitter. Right image shows the same scene with static shadows. ©Pixar

### 4.3.6 *Postprocessing*

Postprocessing methods use one or several prerendered snapshots of the scene and blur them using motion information built from the images themselves or the objects' animation data. With this approach, motion blurring and rendering methods are fully decoupled. As they operate in image space, they are also independent of scene complexity. They tend to be quite efficient, but not everything are advantages: as we will see, they are also associated with certain quality compromises.

In general these methods can be modeled as in Equation 4.7. A motion blurred image is calculated by convolving each pixel $I(x, y, t_i)$ of an instantaneous image at time $t_i$, with a function $\Phi(t_0, t_\infty, t_i)$. This function is built based on the information available in the scene or its rendered images. Even if it can potentially use all image data available between $t_0$ and $t_\infty$, most methods will reduce the temporal scope of the analysis. In general, the details of the algorithm that computed the unblurred images are not needed, but in some cases it may provide guidance to setting up the convolution kernel.

$$I_{xy} \approx I(x, y, t_i) \otimes \Phi(t_0, t_\infty, t_i) \tag{4.7}$$

The first of the methods is described by Potmesil and Chakravarty [191] and extends their previous formulation of an aperture camera model [190]. In their work, Whitted's pinhole raytracer [272] is used to render static images of the world. The image formation process is modeled as a series of image degradation transformations represented by point spread functions (PSF). PSFs can be analytically calculated for shutters of different types [220] and are specific of each object's relative motion. In general, postprocessing do not properly handle effects that are baked into the image, but this method can blur reflections and refractions by explicitly convolving each object's rendering samples with the PSF. Figure 4.15 shows an image rendered with this technique.

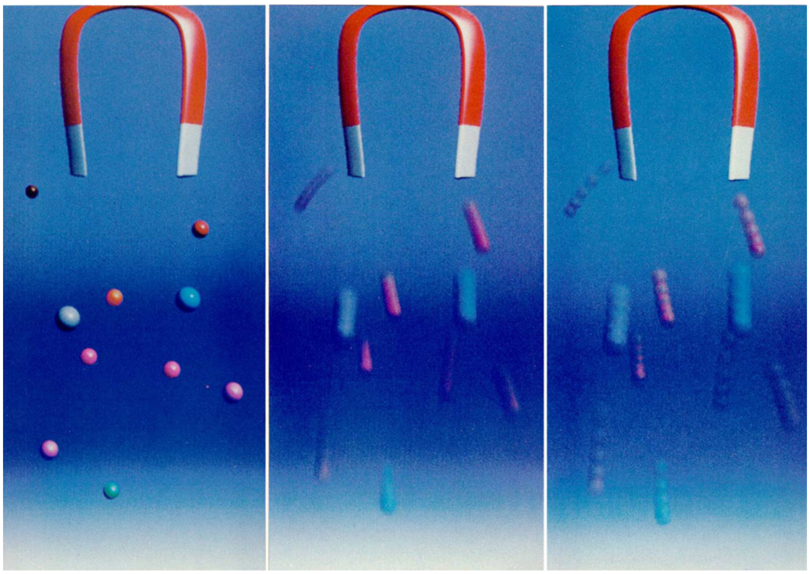

Figure 4.15: Method of Potmesil [191]. Left to right: Instantaneous snapshot, using extended exposure and simulating several exposures of finite length.

A similar model is proposed by Max and Lerner [152]. Objects having similar motion are skewed in the direction of maximum movement according to the intensity of the motion. Incremental summation is used to add blur. Blurred images are unskewed using the inverse of the original transformation before they are alpha composited. In this method, Potmesil's PSF is replaced by an equivalent blurring process, with the rest of the algorithm remaining similar. As a result, both methods have the same limitations.

Sharing common points with the reprojection methods explained in section 4.3.5.2, morphing methods exploit frame to frame coherence [40]. Motion estimation for each sample is described using precomputed morph maps that are constructed from the image pixels, their depth information and the camera transform. While the main focus of the technique is efficiently generating scene walkthroughs, images can incorporate motion blur by generating several temporal samples per rendered pixel. This method is not specific to CG imagery and can also be applied to any images given range data is available [39].

#### 4.3.6.1 *Methods based on motion fields*

An alternative approach is based on synthesizing vectorial fields containing the direction and speed of each image pixel. A motion field represents a snapshot of the dynamic status of the system and it is used as a replacement for the real movement during exposure time. Object motion can be intricate, so these methods are a compromise between simplicity and accuracy. However, some of them are capable of efficiently rendering production quality motion blur.

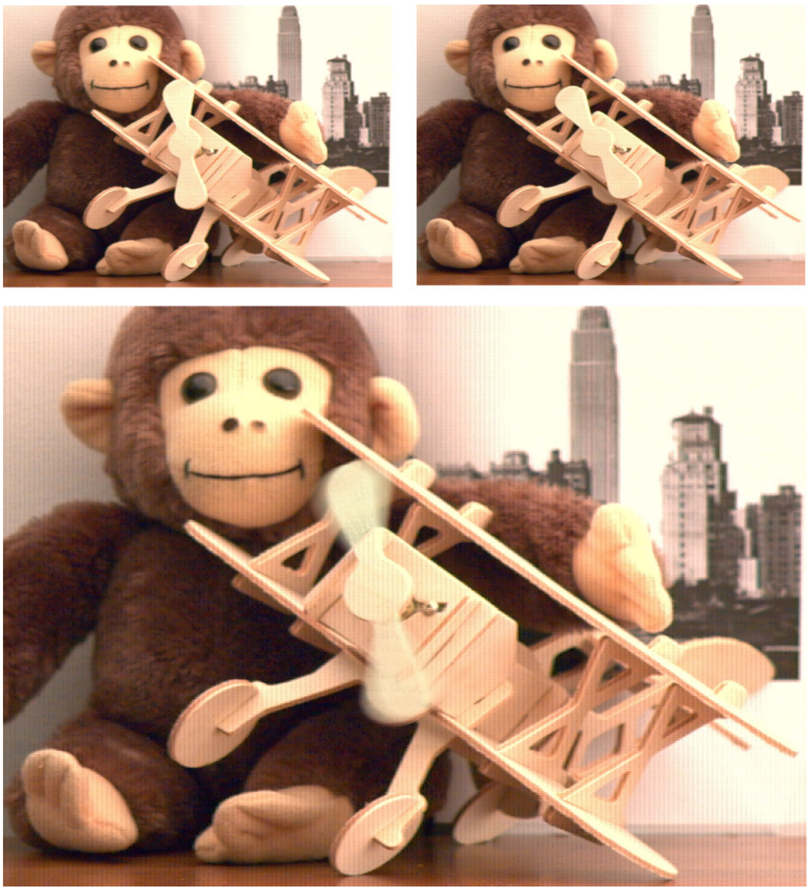

Figure 4.16: Motion blur using the technique of Brostow and Essa [19]. Images at the top are two instantaneous pictures taken from the original scene. Bottom image shows the resulting motion blur applied to the propeller.

Shinya [225] calculates spatial and temporal antialiasing by applying an image transformation that is built on the motion flow of a sequence of images. The pixel-tracing filter is adapted to gather information from sequences of different lengths, with linear computation times. Samples from each image are weighted and combined together so a trail of the object is produced.

The method proposed by Brostow and Essa [19] can be used when no information from the original scene is available. This is the case for stop motion animated sequences where each image is a photograph. Initially, foreground and background pixels are separated using a motion detection procedure. The moving foreground is then segmented into contiguous pixel blobs that are matched between consecutive frames using an exhaustive search. An initial estimate of the velocity field is locally refined using hierarchical optical flow [12] and the resulting velocity is used to smear each pixel assuming constant speed and linear trajectories during the simulated exposure. The results can be observed in Figure 4.16.

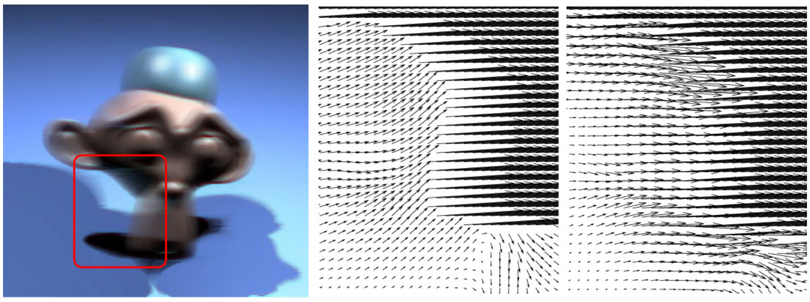

Figure 4.17: Motion blur calculated with the method of Zheng et al. [285]. Center image shows a motion field calculated using optical flow computation and, where available, information from the renderer has replaced the original field. Right image shows a better field that results from using renderer data as landmarks for the optical flow computation.

Previous methods provide smooth velocity fields that may generate inaccurate results when, for example, overlapping objects with different speeds are projected to the same screen area. The results are also greatly determined by the quality of the input. Bad lighting, camera shaking, the existence of shadows or the capture process itself can compromise the results.

Zheng et al. propose a hybrid approach that combines motion from the scene with pure image based optical flow [285]. An incomplete motion map can be easily determined at render time from the scene database. These motion vectors are independent of the illumination and can be incorporated as constraints or landmarks for the minimization step of the optical flow computation [100]. In Figure 4.17 we can see the resulting field, that smoothly blends exact values and image-based estimations where needed. Due to the fundamental nature of the algorithm, it does not completely solve the problems associated with occlusions and pattern blurring in areas containing shadows and reflections.

Shimizu, Shesh and Chen present a method that produces motion blurred polygonal geometry in real time [224]. It efficiently computes an approximation to the optic flow of the scene by using the motion vectors of the model's vertices. As seen in Figure 4.18, they are used as displacement offsets to produce lagging and leading trails of the original geometry that are rendered using different degrees of transparency. This method implicitly implements a line integral convolution [26].

Game engines have also benefited from postprocessing algorithms. A recent example is described by Sousa [230]. His method uses a velocity field that is calculated in an independent pass using the current and previous camera transforms. The original images are progressively blurred, while keeping near objects sharp, using two alternating ping-pong buffers. Deformable geometry can be handled by taking into account the motion of the skeleton bones. While it

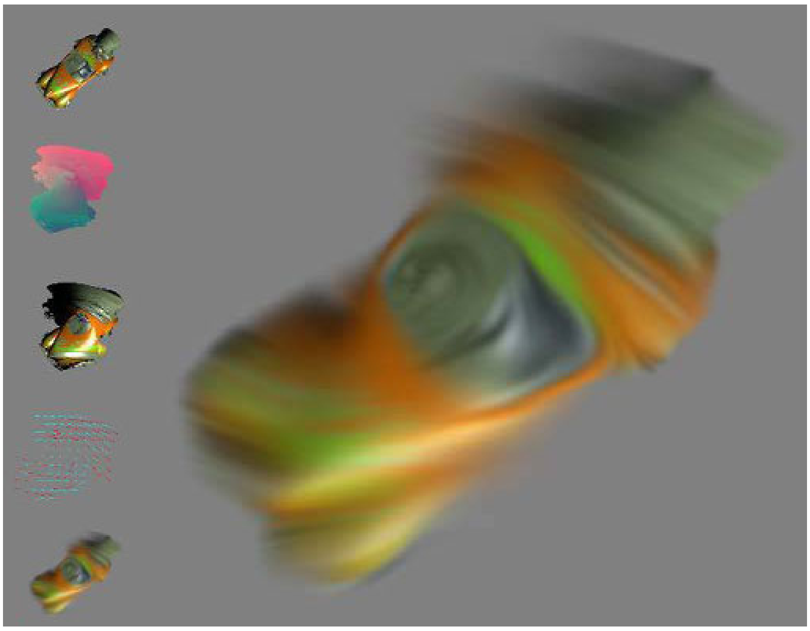

Figure 4.18: Real time motion blurred car using the method of Shimizu et al. [224]. Insets show, from top to bottom, a render of the original geometry, motion vectors, warped geometry, optic flow vector field and final rendered geometry.

avoids leaking artifacts by using an object-ID buffer, transparency and alpha compositing cannot be properly handled. Similar techniques, limited to camera motion and rigid objects [209] or exclusively camera rotation and translation [256] are also used in other next-gen games.

### 4.3.7 *Hybrid methods*

Hybrid methods combine the strengths of individual algorithms to resolve specific aspects of the general motion blur problem. This approach results in algorithms that, relying on mild assumptions, can efficiently be executed in a broad set of scenarios. Their flexibility and the high quality of the results have made them part of the selective group that have been accepted as part of the graphic pipelines of the most demanding production environments. Due to their importance, we will dedicate the following section to explain them in detail.

#### 4.3.7.1 *The REYES architecture*

The first hybrid technique is implemented as part of the REYES rendering architecture [46]. This framework is designed to render geometrically heavy scenes and efficiently implements stochastic point sampling for complex shading functions. In this environment, shading functions are evaluated before visibility is considered, and once non-occluded samples are determined, their values are filtered down according to their spatial relationship with respect to the camera.

Primitives, independently of their type, are subdivided into micropolygons. Considering their screen space dimensions, shading samples $L_l(\omega_{uv}, t_j)$ are calculated and stored in the object's $uv$ space. In this expression $\omega_{uv}$ represents the direction that corresponds to each $uv$ location. The time dimension is discretized using constant steps, and each of the previous samples are also evaluated at those $t_i$ locations.

Stochastic sampling is used to reconstruct the radiance of each image pixel. As such, Equation 4.5 is evaluated at a number of $\omega_j$ and $t_j$ spatio-temporal locations. The value of each $i_d(\omega_j, t_j)$ estimate can be determined using Equation 4.8.

$$i_d(\omega_j, t_j) = \sum_l r(\omega_j, t_j) g_l(\omega_j, t_j) L'_l(\omega_j, t_j) \tag{4.8}$$

Shading samples $L'_l$ are generated using a chain of linear interpolations on cached values. Each $\omega_j$ is mapped to the corresponding $(u, v)$ position in parametric space and its neighbourhood is used to interpolate new values at arbitrary locations. An estimate for time $t_j$ can be interpolated from the corresponding values at the endpoints of the discretized interval $[t_i, t_{i+1}]$. The reconstruction filter $r(\omega_j, t_j)$ accounts for partial occlusion due to transparent and overlapping surfaces as well as reduced coverage due to the motion of the objects.

Note that even if the shading function is spatially and temporally clamped, visibility changes will result in high frequency contents and discontinuities. In cases with extreme variations, the image may suffer from noise that can be alleviated by different means [64, 7].

As seen in Figure 4.19, the shading functions can integrate complex phenomena which can be modeled using shade trees, procedural texturing, image based texture mapping, analytical models or Monte Carlo ray tracing [7].

A lightweight variation of this method may be used in those situations where a full solution cannot be applied. In those cases, a non motion blurred render of the scene is calculated using high quality shading and lighting. A final motion blurred pass is calculated after replacing the original object materials by camera projecting the initial image. This is clearly a compromise that limits the shading information to a snapshot of the areas of the object that are visible in the initial image, but has revealed itself as a powerful tool when the evolution of both shading and geometry is bounded.

#### 4.3.7.2 *Method by Sung, Pearce and Wang*

The visibility function can be fully determined in the early steps of the render by analyzing the geometry and motion of the scene. On the other hand, the evolution of the shading functions is difficult to predict and cannot be fully solved in a preprocessing step. Based on those premises, the method of Sung et al. [237] uses an efficient

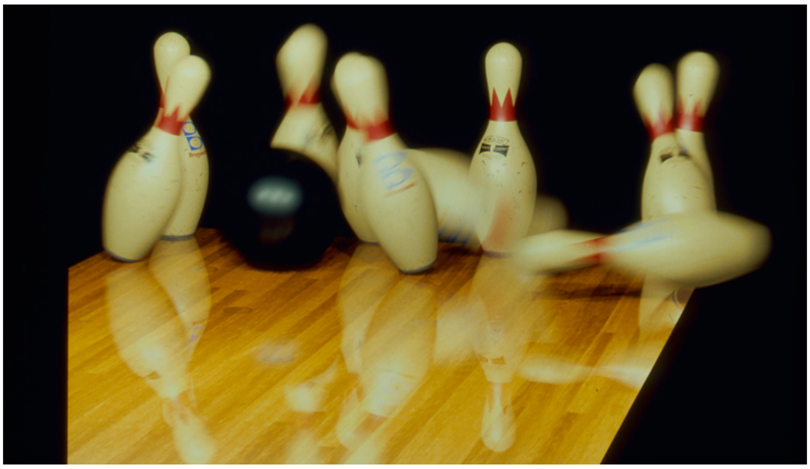

Figure 4.19: Image rendered using the REYES architecture [46]. This architecture can render high quality motion blur. In this image temporal antialiasing has been applied to geometry, shadows, reflections and lighting. ©1989 Thomas Porter, Pixar

analytical solution for visibility and a more flexible stochastic method for subsequent shading computations.

Visibility is determined with an approach inspired by the methods of Catmull [34] and Korein and Badler [123]. These algorithms are used, respectively, to find the areas of the pixel and the ranges of time where visibility remains constant. With the first, pixels and polygons are subdivided until a single micro-polygon can be seen through each $\omega_j$ image space subregion. A modified version of the second algorithm generates contiguous time segments $\tau_l^j$ in which the visibility of an object remains unchanged. Adaptative supersampling using Mitchell's contrast [158] may be triggered to spatially refine each subpixel estimate.

Shading computation relies on stochastic sampling to approximate the radiance of each pixel. The radiance of a polygon, $L_l(\omega_j, t_l^j)$, is assumed to remain unchanged for the duration of each of its constant-visibility time ranges. Adaptative supersampling can further refine the estimates of the shading function and those regions that are subdivided will reuse the visibility of neighboring areas. According to the authors, this is a safe assumption as visibility supersampling has already reduced the differences in the estimations between close samples.

The final pixel value is determined, as in Equation 4.9, by adding the contribution of the $N_j$ sampling areas that result from both refinement steps. Each shading sample $L_l(\omega_j, t_l^j)$ is weighted with a factor $F(\tau_l^j)$ that accounts for both the stochastic probability function and the integrated value of the temporal filter along the whole $\tau_l^j$ interval. The contribution of any visible objects in a given sampling area is also weighted by $H(\omega_j)$, a value that represents the integrated spatial filter.

$$I_{xy} \approx \sum_{j}^{N_j} H(\omega_j) \sum_{l} \sum_{\tau_l^j} F(\tau_l^j) L_l(\omega_j, t_l^j) \tag{4.9}$$

Due to the adaptative nature of the algorithm this method is specially suited for large on-screen motion trails and sharp changes in illumination. The analytical results of visibility computation are used to more efficiently direct the stochastic steps. This contributes to a reduction in the amount of noise from traditional Monte Carlo methods.

### 4.3.8 *Mechanical, optical and sensory inspired models*

How the optics of the camera, shutter geometry and motion, film and sensor influence the final image? How the images captured by a real camera diverge from ones produced with models explained in the previous sections? In this block, we address these points and describe different approaches capable of simulating the internals, limitations and deficiencies of the device.

The geometrical and optical characteristics of cameras can be modeled using different approaches [10]. In most of the cases, pin-hole or thick lens models provide enough accuracy. However they are an oversimplification when more realistic depth of field, zoom, flares due to internal reflections or optical aberrations are required.

As described in section 4.1, the aperture controls the amount and structure of the light arriving at the image plane and determines the amount of static defocus. Among the existing methods [9], distributed ray tracing [45] and postproduction filters [18] are the most widely used to emulate such effects.

If we exclusively account for motion blur, the camera shutter is the main element of interest. The vast majority of the algorithms in section 4.3 are based on a shutter that moves infinitely fast. As a result, the reconstruction filter $r(\omega, t)$ is constant and the motion blur is homogeneous. In general, a mechanical shutter transitions in a finite amount of time, producing uneven exposure. In the context of computer graphics, the influence of shutter shape and its evolution has received little attention [220, 77]. To our knowledge, only optical design software  has implemented accurate models.

Interestingly, shutter simulation has recently received extra attention with the widespread popularity of video cameras and other digital recording devices. Even if professional systems may include a mechanical shutter, consumer devices simulate or replace it with with electronic components. This produces a different family of artifacts. As an example, CCD cameras provide global shutters that expose the whole image simultaneously and the resulting motion blur is similar

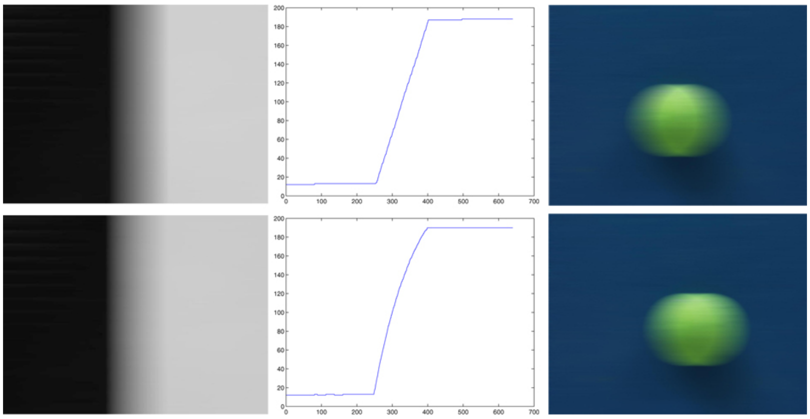

Figure 4.20: Photo-consistent motion blur. First and second columns show the image of an ideal edge in motion and the corresponding simulated intensity profile. Third column shows a tennis ball rendered using two different methods. All images in the first row assume an uniform PSF. Second row uses the model of [139]. Note the differences from the traditional method and the non linear asymmetric shape of the function.

to the one produced by a uniform shutter. On the other hand, CMOS sensors process images using a rolling shutter with an exposure time that is directly proportional to the frame rate. Motion blur is coherent for the pixels of each scanline but shows temporal discontinuities between consecutive lines [156]. Skew, wobble and partial exposure artifacts are typical of those systems [82]. Fields like robotics or artificial vision have developed models to simulate and compensate these effects [183].

Another aspect that is frequently overlooked is how the incoming light affects the imaging media. Radiometry clearly states that the relationship between the calculated scene radiance $L$ and the irradiance at the image plane $E$ is non linear [74]. Using box or gaussian kernels to downsample radiance is an approach that deviates from the real phenomena. Since film and sensor response is highly non linear and produces noise [243, 119], recorded image intensity cannot be assumed to be proportional to the radiance in the image plane. Different algorithms try to mimic more accurate behaviors using high dynamic range imaging and tonemapping techniques [203].

Just a few algorithms have tried to produce photometrically correct motion blur. Lin and Chang describe a method that models the response of the camera based on the capacitor charging process [139]. The algorithm is tuned with a calibration step and its response can be adapted according to the F-stop of the camera. The usual uniform point spread function is replaced with a more realistic filter that enhances accuracy. Figure 4.20 shows the differences between traditional motion blur and the new technique.

Motion blur may also be seen as an effect that needs to be corrected. Pictures taken with long exposure times may contain unexpected motions that can compromise the spatial resolution of the captured image. Artificial motion blur can be recreated by integrating motion compensated snapshots [241].

## 4.4 METHOD COMPARISON

In the previous sections, we have provided a detailed description of most of the algorithms that can integrate motion blur in computer generated images. In this section we compare them, identify their computational requirements and focus on identifying which methods are applicable to each specific situation.

From an operational point of view, an important element to consider is the **type of inputs** that the algorithm can process. Some methods are specific to certain geometrical representations while others are capable of accepting a wider set of primitives. In the first category, the most popular representation is polygonal geometry [81, 35, 46, 237], that may be processed using hardware acceleration both for primitives [275, 240, 83, 109] and textures [4, 145]. Other methods can handle NURBS [192], spheres [123], particles [200] or voxels [85]. Non eulerian fields, such as liquids and gases, can be converted to polygonal representations but they have also been considered without previous transformation [117]. Monte Carlo methods, section 4.3.5, can accept any geometrical description given that it can be point sampled.

Postprocessing algorithms are fairly independent from the type and geometrical complexity of the scene. Motion information can be extracted directly from the images using adhoc heuristics [225] and optical flow methods [19]. In some cases, motion data from the scene is used [224], while in other situations rendered velocity, object-id or depth passes [285, 230] may be used. An in-between approach is used by reprojection algorithms [40], where the original geometry of the scene is needed, but only as a mean to finding novel positions for precalculated shading samples.

An important criteria that helps determining the respective merits of each method is the degree of **quality of the results**. Do we want to produce motion blur that is physically based, photorrealistic or just a cue to the evolution of the scene? In one extreme, geometric replacement use gross shading and geometrical approximations but is capable of producing high frame rates. Common simplifications are based on roughly approximating the original motion [275, 83, 109], non physically based energy distribution [200, 240], artistically driven motion trails [85, 216] or even alpha transparency as a replacement for the light integration process [275, 109]. A number of methods avoid the computation of all but local lighting events [175, 123, 81, 275].

Motion blurred shadows, reflections, transparency or refraction are difficult to achieve with these methods.

A special case is represented by postprocessing and reprojection methods. They both base their computations on prerendered samples whose radiance is the result of completely simulating a given lighting model. View dependent phenomena have to be explicitly handled by, for example, storing intermediate stages of the render.

A small group of algorithms accurately model dynamic geometry and shading. They are frequently based on intensive point sampling which makes them them the most computationally heavy approaches [45, 237, 29, 251, 261, 89]. Recent developments have improved this situation [65, 182, 194]. Extended descriptions and a discussion of their respective sampling approaches have already been exposed in sections 4.3.5.1, 4.3.5.2 and 4.3.7. These can be combined with the methods described in section 4.3.8 so that the influence of the recording device is accurately modeled.

If we consider the **complexity of the motion**, most algorithms approximate intraframe object motion with piecewise linear paths. These paths may be defined at object or vertex level which allows rigid object motion [152, 275, 85] and deforming polygonal meshes [109] respectively. Most methods interpolate motion using a single straight segment built from one or two temporal samples. Some algorithms assume the original motion has to be linear in world space [35], while other will use alternative coordinate systems to enforce more restrictive motions [123, 163, 19].

To our knowledge there is no published algorithm capable of using the original path or equivalent non linear approximations. The nature of the motions and the shutter times involved in the computation (a maximum span of 33 milliseconds at 30 frames per second) makes assumptions like path linearity and constant velocities safe for an ample range of situations. The most flexible extreme is represented by distributed ray tracing [45] and its Monte Carlo based descendants that can generally evaluate both true shading and intricate motion at any point of the problem domain.

Different methods use alternative assumptions on the **number of samples** that are representative of the whole aperture range. Some of them, postproduction and geometric replacement methods, simply evaluate the lighting functions once, usually at the start or middle of the time slot. Image samples can also be based on several time instants that are randomly selected [54], from a predefined pattern [90, 4] or stochastically determined [46]. While the differences can be subtle, a properly selected sampling method can alleviate aliasing artifacts that are perceptually objectionable [50].

In certain situations, the availability of **computational resources** determines which algorithms are applicable to a given scenario. Geometrical replacement methods, section 4.3.3, are usually good candidates to be included in those hardware frameworks where polygonal geometry, texturing and depth culling can be efficiently evaluated. Real time anisotropic filtering allows the use of techniques previously reserved to offline renderers [145]. Gaming frameworks can also provide real time rates by using postprocessing [209, 230, 256]. However they are frequently constrained to certain types of motion. Given the existence of GPU based optical flow algorithms [162, 185], motion field based methods may also be implemented in real time. Monte Carlo methods, due to computational complexity, are usually reserved for batch rendering environments. Only a few methods based on accumulation buffers [90] and frameless rendering [11, 54, 55] have implemented subsets of distribution ray tracing at interactive frame rates. Recent works based on the REYES rendering architecture may also open the door to efficient hybrid methods [288, 68, 16, 20, 101].

Most Monte Carlo and hybrid algorithms are also known for being memory hungry algorithms. They are usually associated with intensive ray traversals which results in incoherent memory accesses and the need to store the whole scene in memory. Acceleration structures such as Kd-trees and BSPs can improve efficiency [258] at the cost of extra memory usage. The need to keep in memory both light paths [251, 261] and samples until the image is reconstructed [89, 65, 182] also contribute to this extra cost.

In summary, and from a strictly practical point of view, there is a tradeoff between the models that can efficiently produce approximate results, and a computationally heavy implementation that accurately calculates the light integration process. For those cases where rendering time and computational resources are not the limiting factors, both hybrid methods and those based on distributed ray tracing are the best candidates. They are capable of producing the most accurate solutions as well as modeling the most complex phenomena. Among all of them, the REYES architecture is probably the one with lower requirements even if, in some cases, it needs to rely on external ray tracers to implement global lighting models [41]. Together with distribution ray tracing, their extensions for photon mapping, lightcuts and metropolis transport and the model of Sung et al. are probably the most flexible.

In those cases where images must be produced at interactive frame rates, geometrical replacement and postprocessing are the primary categories to consider. Every algorithm in these categories are tailored to solve specific subsets of the problem. Real time postprocessing methods are constrained to specific types of camera and object deformation. In any case, there is a clear preference for them in current game architectures [209, 230, 256].

In those cases where a full flagged method is not an option and there is no need for real time solutions, postprocessing methods such as the one of Brostow and Essa [19] and Zheng et al. [285] are capable of producing adequate results that in some cases can be comparable to other more expensive methods.

## 4.5 FUTURE DIRECTIONS

Recent research trends seem to be focused on two diverging directions: finding **improved sampling schemes** to evaluate the rendering equation; and implementing more efficient algorithms based on **hardware acceleration**. Whether the solution comes from focusing computation in the most important areas of the image or redesigning algorithms to make use of parallelism, the main challenge is the efficient use of limited computational resources.

An unexplored approach is based on the complex **relationships between the human visual system and how objects in motion are perceived**. We know the HVS is an incredible biological design even though it is not free of limitations in its spatial [30] and temporal resolutions [115, 217]. Its response is non linear with respect to light wavelength, chromatic and achromatic light [278]. Perceived images are not fully focused [8] and in some circumstances may be noisy [274] or contain aliasing [280]. Also, temporal and spatial integration seem to be an important factor in the perception of images in motion [125, 22, 222].

Knowledge of perceptual mechanisms have been successfully exploited in fields like image quality assessment [51, 199], tonemapping [28] and geometric modeling, rendering and simulation [52, 164, 181, 6]. There is good foundation to think a similar approach can be applied to rendering motion blur.

Computational optimizations can exploit the fact that certain stimuli are not perceived or generate such a low response that can be ignored. Also, existing methods are tuned to produce antialiased images at the image plane sampling rate. Even if this is a reasonable assumption for static images, the limited attentional bandwidth of the HVS makes us think this rate can be relaxed. Also, given the existence of a temporal window of integration, each frame can be rendered taking into account a context that includes the images immediately before and after. This gives opportunities for further improvement.

In previous sections, we also have implicitly accepted that all temporal samples are equally important. In light of HVS' non-linearities this may not be a correct assumption. Determining which integration mechanisms take part, whether they are biased or not and how we can simulate them are important questions that remain unanswered. Algorithms that are aware of the differences between what is perceptually acceptable and what is physically and numerically accurate can also be used to produce motion cues as a substitute for ideal motion blur.

## 4.6 CONCLUSION

This chapter provides an overview of the state-of-the-art in motion blur rendering techniques. We have started our discussion with a description of the physical phenomena and mechanisms involved in the generation of motion blur in film and recording devices.

We have also described a formalization that, in the context of computer generated images, mathematically models the light integration process. It explains the complex interactions that take place in an animated scene based on the objects' geometrical relations and their shading functions. Existing motion blur rendering methods have been categorized according to the approach followed to evaluate this expression. An organization based on analytical, geometrical substitution, texture clamping, Monte Carlo, postproduction, hybrid and physically inspired methods is the result of this effort.

Simulation of motion blur is known to be one of the areas of the rendering pipeline that is heavier in computational resources. We have classified the methods based on their processing requirements and areas of application. Finally, we have briefly introduced a possible area of improvement that, by exploiting the limits of our perceptual system, can provide performance gains. This family of temporally aware methods is an exciting and novel direction that is explored in the following chapters.

# 5 PERCEPTUAL CONSIDERATIONS FOR MOTION BLUR RENDERING

Motion blur is an important cue for the perception of objects in motion. This phenomenon results from the light integration at the imaging device's sensor combined with object or camera motion. This generates a visible streak that follows the trajectory of the movement. It is naturally produced by both film and digital cameras although it has also been leveraged as an artistic resort [187].

Computing motion blur is a requirement for computer generated images where it is used to generate more natural looking images, as well as, to reduce time sampling artifacts such as strobbing, flickering and stuttering. Unlike with real cameras, syntethic images need to explicitly simulate motion blur.

While the methods to produce synthetic motion blur have been studied and classified [170], the perceptual aspects associated with the observation of motion blurred images have not received the same attention. Research has mostly been focused on determining the lower level capabilities of the HVS (human visual system). These works have relied on controlled studies where the stimuli are composed of gratings, gradients and moving dots. There is still the need to determine how these mechanisms work with imagery that is representative of the complexity of real world scenes. We believe this work is one of the first that exposes some insight in this area.

In the next sections we design a series of psychophysical experiments to determine how the HVS reacts to complex stimuli. This study also links the perceptual and practical algorithmic aspects of the problem. We have used path traced images of objects under the influence of complex lighting with the intention of associating standard rendering parameters such as shutter time and antialiasing level to their corresponding perceptual effects.

This problem is inherently multidimensional and the interplay of the different aspects involved is complex. However, this study is targeted to give answers to simple questions such as: When the scene is composed of simple materials, can a render be produced at a reduced cost without perceived quality loss? Do users detect the difference between alternative shutter times? How important are high sampling levels in the computation of motion blur?

The conclusions and recommendations contained in this work may prove useful for the production of photorealistic images. We hope that CG practitioners, render technical directors, shader writers and hobbyists will be able to apply them to minimize render times and

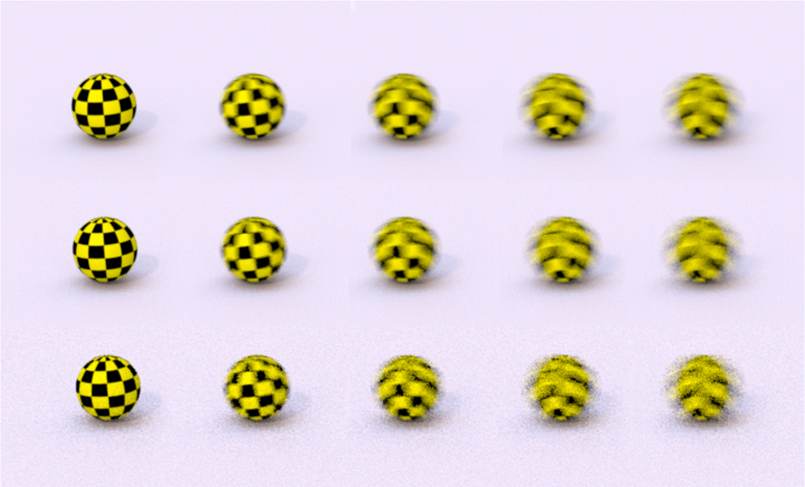

Figure 5.1: In this work, we explore the implications of alternative scene properties and rendering parameters on the perception of motion blurred images. In the figure, a matrix showing a subset of the stimuli used in a series of psychophysical experiments: Columns, from left to right, use shutter times T1 (0% of the frame time, no motion blur), T2 (25%), T3 (50%), T4 (75%) and T5 (100%). Rows, from bottom to top, use antialiasing levels A1 (4 samples per pixel), A3 (256 samples) and A5 (512 samples).

reduce scene complexity without visual quality loss. These results may help settle some ground for the implementation of perceptually inspired motion blur rendering methods. This is especially interesting for the production of feature films and VFXs, where the cost of generating temporally antialiased images may be several times higher with respect to their non-motion-blurred equivalents. Even more, its importance is clear given that render times are usually optimized by following intensive hand-made processes.

## 5.1 PREVIOUS WORK

The literature covering the perception of computer generated stimuli has identified different psychophysical trends. Recent examples include models for the psychophysics of photorealistic materials [249, 211, 105], natural illumination [73, 27, 128, 144], occlusions [124, 282], the ability to differentiate close variations of the stimuli [199, 154] and has also described the influence of object variety [153, 198]. However, none of these focus on the specifics of motion blurred images nor time varying images.

Motion blur rendering has received an important degree of attention. Existing works have mostly focused on the algorithmic aspects of the problem and describe methods capable of performing temporal antialiasing on different frameworks. We refer the reader to the work

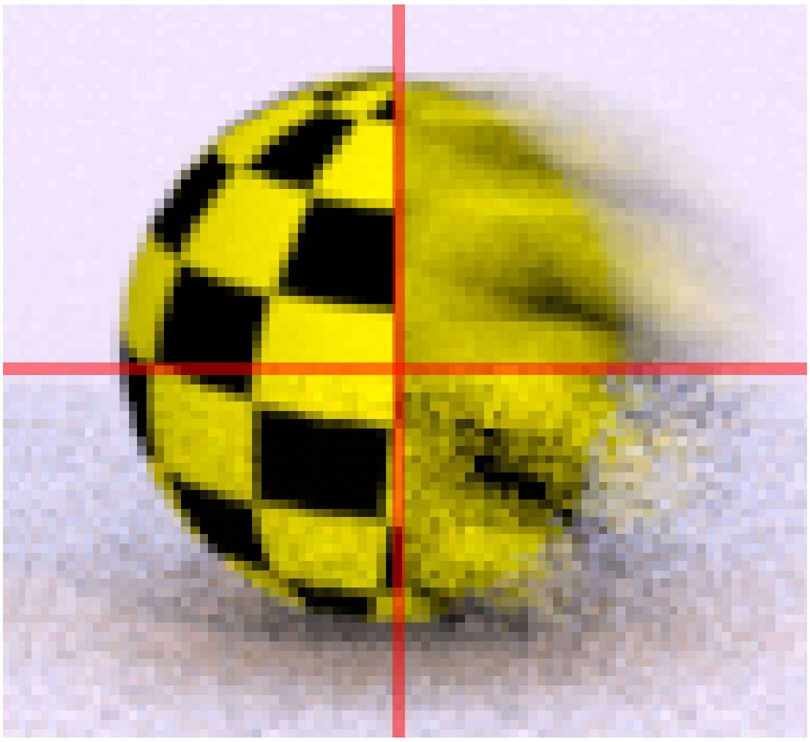

Figure 5.2: Detailed view of the renders at the four corners of figure 5.1. Noisy artifacts are visible in the images computed with lower sampling levels (bottom row), while the snippets at the upper row are clean. Upper left render was computed without motion blur, while the right one uses a temporal window of 100% the frame time.

of Navarro et al. which contains detailed classifications and extensive descriptions of the state-of-the art of the existing techniques [170].

Even if not specific to computer rendered images, the perceptual aspects of temporally varying images have also been described [3] [269] and their corresponding sensitivity functions [134] [208] [255] have been clearly determined. The work of Burr and colleagues is fundamental in this area, being the first ones to describe motion smear, identify the window of temporal integration and quantify the perception of motion blurred stimuli [22] [25]. Even if many of the psychophysical factors have been clearly determined, the literature covering how the HVS reacts to dynamic photorealistic stimuli is still scarce. This chapter focuses on this aspect while exploring the implications of using Monte Carlo rendering algorithms.

Other works have focused on determining how visible certain features are when they are observed in a picture or a rendered image. As such, the corresponding saliency functions have been suggested [6] [142] which, in some cases, are temporally aware [284] [262] [250]. Other studies have proposed alternative perceptual metrics that can determine how different two images are [149] [263] [197]. The temporal component has been considered notably in the context of video processing [270] [279] and quality assurance [266].

### 5.1.1 *Perception of motion blur*

The existence of a temporal window of integration produces an effect that is known as motion smear. During a certain time span, the light hitting the retina is accumulated. As a result, each spatial location contains an image that represents the evolution of the scene and not a discrete snapshot of it. Depending on the stimuli and environmental conditions, the window of integration can span from a few milliseconds to several seconds [269].

Peripheral vision [23] is specialized in the detection of moving objects. Even if it is limited to low spatial frequencies, it is tuned to detect high temporal frequencies. This increases the chances of detecting image flickering, an artifact that is frequently present in low quality rendered images. In contrast, wide fields of view use an alternative visual channel that is specialized for object identification. While it is adapted to prioritize high spatial frequencies, high temporal frequencies are also processed although with reduced efficiency. Temporal integration, increased temporal frequency detection and motion generating faster retinal speeds, may lead to increased perceived blurriness even if this extra blur is not present in the original image.

The opposite is also true: Images may look sharper than they really are. Motion deblurring [195, 24] is triggered for moving objects and in some cases, perceived images may appear less blurry than their static incarnations. This is thought to be a mechanism to increase the resolved amount of detail in the images of moving targets.

The detection of motion itself has been explained using several models that are based on space-time filtering. They focus on the idea of finding moving brightness patterns in the retinal image [3] [268]. Alternative explanations exist for second-order motion generated by variations in contrast and flicker which may be helpful in the presence of occlusions and transparency [72] [95]. Finally, the surrounding environment has also been identified as a cue to detect motion [143]. Our study has been designed to analyze perceived quality as the absence of rendering artifacts. As such, we have chosen stimuli that exceed the threshold for motion detection, contains no object occlusions and contains a single object that can be comfortably tracked by the eyes. These images do not make use of any depth of field, so this source of static image blur is also excluded from the study.

### 5.1.2 *Technical aspects of motion blur perception*

The algorithms associated with motion blur rendering will be briefly described in Section 5.2. A by-product of these approaches is the introduction of certain amount of blur in the images. Since the degree of blurriness of an image is used as a cue for motion detection [22], this has direct implications on how an image is understood.

Some other factors, related to the HVS, rendering pipeline and display devices, are also capable of changing the nature of the image by adding extra blur, giving higher importance to certain temporal frequencies or diverting attentional resources.

Display devices contribute additional blur to the image. With common technologies, such as LCD monitors, the color of each pixel is held for certain amount of time. In this case, images are smoothed by the display's inertia to keep its current state and its relative inefficiency to instantaneous change between different values [184]. Existing models can determine whether a detail in an image will appear sharp or not [267]. The discrete nature of the media produces hold-type blur. The imaging pipeline discretizes the original stimuli so when an object is tracked, the HVS is forced to reconstruct the original image from a reduced set of unconnected snapshots. The lower the number, the bigger the chances the temporal window of integration produces a blurred image that differs from the one that would otherwise be produced from the initial stimuli. This issue has been dealt with using several methods [71] [58]. Both types of blur are intrinsic of the technology and are always present to some degree.

According to Block's Law, for stimulus with similar spatial characteristics, detectability is determined by their energy [79]. Taking into account temporal integration, longer stimulus of reduced intensity may seem equivalent to shorter ones with higher luminance. In the specific case of display devices, several frames are fused together to produce a single unique perception that ultimately is responsible of the perception of continuous motion.

In summary, the perception of moving objects and motion blur is a complex process. Several perceptual mechanisms can interfere with the original images. Apart from the human factors, there is a group of technologically related issues that further complicate any predictions. In the following sections we will concentrate on determining the final appearance of rendered stimuli based on a series of scene and rendering parameters. In some cases, we will see that the explanation for certain results comes hand in hand with the phenomena we just described.

## 5.2 ALGORITHMIC MODEL

The formulations of Sung et al. [237] and Meredith [157] explain the generation of motion blurred CG images with the following expression:

$$I_{xy} = \sum_{l} \int_{\Omega} \int_{\Delta T} r(\omega, t) g_l(\omega, t) L_l(\omega, t) dt d\omega \tag{5.1}$$

In this equation, $I_{xy}$ represents the contents of the image pixel with coordinates $(x, y)$ and $\Omega$ is its corresponding subtended solid angle. Independently of their geometrical representation, the contribution of the primitives in the scene is considered by iterating over each individual object $l$. $g_l(\omega, t)$ is a geometrical term that accounts for occlusions between objects. Its value is 1 if object $l$ is directly visible in the direction $\omega$, 0 otherwise. Shutter shape and efficiency, lens aberrations and film type can influence the final image. The reconstruction filter $r(\omega, t)$ accounts for their overall effect. $L_l(\omega, t)$ represents the radiance of object $l$ without explicitly establishing the method by which is calculated.

In order to account for the complex spatio-temporal relationships taking place in an animated scene, all terms are evaluated at an instantaneous time $t$ over the aperture time $\Delta T$, and over the solid angle $\Omega$. In some cases and depending on the desired filter footprint, $\Omega$ can represent narrower or wider solid angles than the one defined by the pixel.

$$I_{xy} \approx \frac{1}{N_j} \sum_{j}^{N_j} i_d(\omega_j, t_j) \tag{5.2}$$

$$i_d(\omega_j, t_j) = \sum_{l} r(\omega_j, t_j) g_l(\omega_j, t_j) L_l(\omega_j, t_j) \tag{5.3}$$

Equation 5.1 represents the general case for any motion blur algorithm. As far as this study is concerned, rendered images are generated using distribution ray tracing. Equation 5.2 is an approximation that takes into account the sampled nature of this method. Each image pixel is calculated as a sum of $N_j$ discrete point samples. For simplicity, we will consider that motion blur is based on sampling locations in the spatio-temporal domain only. A sample $i_d(\omega_j, t_j)$ accounts for the contribution of each object $l$ as seen in the direction $\omega_j$ at an instantaneous time $t_j$. This value can be calculated by different means, but in general it will be the result of evaluating their respective visibility $g_l$ and radiance $L_l$ functions. These contributions are weighted with the value determined by the filter $r(\omega_j, t_j)$. The color of the pixel may be the result of an average of the samples, but frequently, more sophisticated weighting methods are be applied [50].

## 5.3 DESCRIPTION OF THE STIMULI

The perception and recognition of motion-blurred objects is affected by many elements: The geometry, material and animation of the objects,

| DIMENSION | VALUES |
|---|---|
| Material | Flat (M1), checker (M2), noise (M3) |
| Object speed | Slow (S1), medium (S2), fast (S3) |
| Shutter time (%) | T1 (0), T2 (25), T3 (50), T4 (75), T5 (100) |
| Antialiasing level (samples) | A1 (4), A2 (128), A3 (256), A4 (384), A5 (512) |

Table 5.1: Dimensions and values considered in the study

scene illumination, number of objects in the scene, complexity of the visibility changes, the render settings and even viewing conditions can have an impact on both the resulting images and how they are perceived.

Among these factors, we have selected a subset that, while making the scope of the study tractable, still generates a rich range of cases and visual qualities. We want to focus on the issues associated with scene and rendering technology and, as such, we will analyze object material, object motion, shutter aperture time and antialiasing level. The first two are fully determined by the geometry and animation of the scene. The other two are aspects that are determined by the rendering technology. Far from being specific of certain framework, they are parameters that are representative of a wide range of rendering algorithms.

We must note that the workflows for VFX and feature film pipelines have heavily inspired the selection of the dimensions and the design of the tests. In these environments, rendering technical directors receive scenes whose materials and object animations have been polished by the look development and animation departments and accepted by the director of art. In some cases, shutter settings are also determined by artistic decisions or by the need to match a real camera. Even if we will determine the influence of all the dimensions, we will insist on the algorithm related ones. They offer more opportunities to improve rendering times without significantly changing the original scene.

The dimensions and the values used for each of them are explained in the following parragraphs, as well as listed in Table 5.1:

**Object material:** Three physically plausible material variations with different frequency contents. The intention is evaluating the responses of the participants using shading functions with alternative complexities. Figure 5.3, top row, shows the cases considered: A checker pattern containing abrupt changes in color, a noise pattern containing a moderate amount of high frequencies and small details and a simple flat color material. All of them have been rendered using an approximation of a lambertian surface. Apart from the perceptual implications, the higher the frequencies contained in the pattern, the more challenging it is to antialias, and the higher the chances for the algorithm to produce noisy and strobing artifacts.

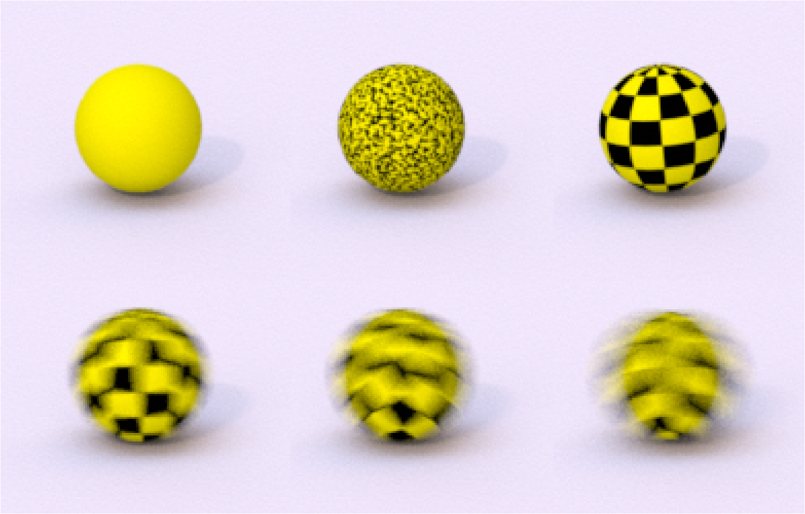

Figure 5.3: Materials and object speeds. Top row: Non motion blurred image rendered with the materials used in the study, from left to right: flat (M1), noise (M3), checker (M2); bottom row, from left to right: slow (S1), medium (S2) and fast animation speeds (S3) rendered using the longest shutter setting (T5).

**Object speed:** Given the camera settings and viewing conditions, we have considered three object speeds that are approximately 5, 10 and 15 degrees of visual angle per second. They provide enough variation to produce images where the object can be easily recognized to other where the sphere is almost reduced to the motion trail. A sample of the corresponding images is shown in Figure 5.3, bottom row.

**Shutter times:** Shutter time simulates the fraction of the frame when the camera accepts incoming light. In Equation 5.1, this parameter has been represented as $\Delta T$. We have considered five values: instantaneous exposure (no motion blur) as well as percentages of 25, 50, 75 and 100% of the frame time.

This parameter modifies the amount of motion blur that the image contains, so it has an impact both on the aesthetics and perceptual responses of the images. It also implies that, for a given number of samples per pixel, longer values create an expanded strata and sparser sampling of the rendering equation. That is, longer shutter times are more challenging to antialias and tend to produce more intense artifacts. Although more accurate models can be considered [235], we have chosen to combine temporal samples using a box filter. Example images are displayed in Figure 5.2, left to right.

**Antialiasing level:** Equation 5.2 represents the antialiasing level as the parameter $N_j$. This number corresponds to the amount of samples that are averaged per pixel. In our case, we have considered five diffent values. The number of rays required has been determined empirically so they generate images ranging from very noisy (4 samples per pixel) to noiseless (255 samples per pixel. Intermediate values have been chosen to cover this range using roughly equidistant steps. The average render times range between 15 seconds to 2 minutes per frame and approximately follow a linear progression. The highest level, that we consider the gold standard, medium and lowest levels are displayed from top to bottom, in Figure 5.2.

Given that temporal integration tends to soften the details in the images, we have decided to ignore the influence of the objects' geometry. Complex object occlusions have also been avoided. As such, this study uses a generic scene showing a sphere of 20cm of diameter rolling over a flat surface. The trajectory is linear at constant speed and parallel to the camera. The camera is static and is placed 3 meters away from the object and 50 cm over the floor. It has been configured to simulate a 50 mm lens.

In order to improve material discrimination, an HDR map provides natural illumination for all the objects in the scene [73]. Arnold engine [128] has been used to produce the rendered images. Its unbiased Monte Carlo radiosity algorithm produces effects such as photorealistic soft shadowing and color bleeding.

A template scene has been modified to use each possible combination of the previous values, resulting in a total of 225 rendered sequences each one 4 seconds long. The images were rendered at a resolution of 640x360 pixels and played at 30 frames per second. The accompanying video shows a subset of the sequences generated for the study.

## 5.4 PERCEPTUAL TESTS: OVERVIEW

Our study comprises a series of psychophysical experiments whose targets are:

- first, finding how influential each of the dimensions described in the previous section are;
- second, determining the most significant perceptual trends within each dimension;
- third: identifying the interplay between different dimensions and how they affect what a human subject sees.

We have designed several tests, based on 2AFC (two alternate forced choice) and matching tasks. Initial tests determine the most generic phenomena and the values that are more representative of each dimension. Further tests build upon them and provide refined results. They are summarized in table 5.2.

Twenty-one participants, sixteen males and five females, aged between 25 and 35, completed the tests. Nine of them had experience in computer graphics and although they were aware of the overall

| TEST | TYPE | DIMENSIONS |
|---|---|---|
| #1 - Broad comparison | 2AFC | All (2 each) |
| #2 - Shutter time / Antialiasing level | Matching/2AFC | Shutter time (5) / Sampling level (5) |
| #3 - Shutter time and antialiasing level | 2AFC | Shutter time (3), sampling level (3) |

Table 5.2: Psychophysical tests completed by the participants in the study. The *Dimensions* column lists the dimensions considered by each test, with the number of values used for each axis between parentheses. The dimensions not listed in a given test are sampled at a single value.

purpose of the study they were naive to the details of it. The remaining twelve had no formal knowledge of computer graphics but had exposure to mainstream graphics technology. All of them self-reported normal or corrected to normal vision. The sequences where displayed in similar dim lighting conditions using a LCD monitor 17.3" diagonal, 1600x900x60Hz, sRGB color space, contrast 1000:1, luminosity 300 $cd/m^2$, gamma 2.2. Participants were placed at roughly 60 cm of the screen. In these conditions each of the sequences subtended 14.4 degrees of visual angle.

## 5.5 TEST #1: BROAD COMPARISON

Given the lack of previous results using photorealistic motion blurred stimuli, the first task is targeted to identifying and gaining an initial knowledge of the most salient trends. As such, it simultaneously consider all the dimensions described in section 5.3.

### 5.5.1 *Description*

In this test, participants were asked to select one out of two videos according to the question *Which sequence has higher quality?* Before starting the task, every subject was briefed about the most important image features that characterize high quality renders. Different example sequences containing noticeable levels of noise, flickering patterns and excessive blurriness were used as a method to identify each feature. In order to ensure these concepts were clearly understood, each participant was requested to perform the same task using pairs comparing the most extreme cases for each algorithmic axis. The answers given during this training were not taken into account in the final results.

Being a preliminary test, each axis is represented by two values that produce visually different renders without making the decisions trivially obvious. These values are: checker (M2) and noise materials (M3), medium (S2) and fast (S3) speeds, medium (T3) and long (T5)

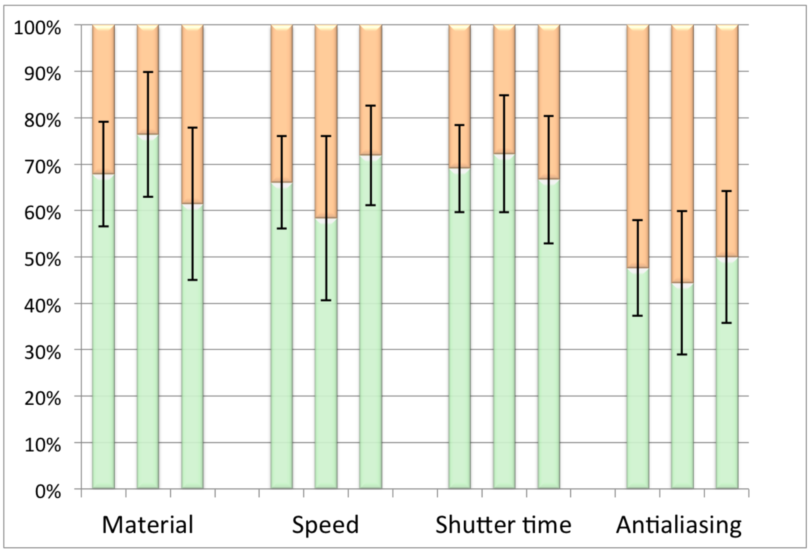

Figure 5.4: Percentage of trials where a particular animation is perceived with higher quality. Blue/orange bars represent the corresponding values for noise/checker material (M3/M2), medium/high speed (S2/S3), medium/long shutter time (T3/T5) and medium/high antialiasing level (A3/A5). Each group of bars shows from left to right, the results for all, experienced and naive users, respectively. Error bars represent the standard error.

shutter times and medium (A3) and high (A5) antialiasing levels. For each trial, the two videos are displayed in sequence one after the other and subjects are allowed to repeat them as many times as required. The pairs are configured so all combinations from a pool of 16 videos are used. The order of the videos in each pair is randomized, so it is the sequence in which the pairs are shown.

### 5.5.2 *Results*

The results show that users perceive higher quality in the renders containing the noise material, medium speed and medium shutter time. A certainty of 95% has been confirmed using a Chi-square test. The answers, represented in Figure 5.4, have different qualitative interpretations:

- The preference for the noise material (67%) seems to be related to the fact that objects whose shading looks too clean or very regular are perceived as unrealistic. Also, the checker material, while not affected by sampling noise, produces more stroboscopic artifacts at every antialiasing level. These results are consistent both for medium and high sampling levels.

- Users manifest a preference for the medium speed in 66% of the situations. The reasons reported were excessive blur, lack of detail and a noticeable level of noise in the images using the fastest speed.

- Medium shutter times are chosen in 70% of the cases as longer values produce slightly unnatural images due to excessive softness.

- More significant is the absence of a clear preference for any of the antialiasing levels (the highest quality setting has received just 5% more votes). However using a visual differences predictor operator [150], the images of individual frames are reported as perceptually different. This suggests that while an accepted mathematical model can detect the differences on static images, the HVS has difficulties finding differences when the temporal dimension is involved.

Segmenting the results according to the users' knowledge of computer graphics has not shown any statistically significant differences. That said, experienced users took notably longer to complete the test with an average of 11:23 minutes versus 8:28 minutes. We attribute this to this prior knowledge preconditioned them to ensure the answer was the correct one. In some cases, and given the subtle image differences, this represented a challenge.

In practical terms the previous results support the idea that certain types of renders can be optimized without visible perceptual degradation, that is, scenes rendered with reasonable levels of quality may be indistinguishable from a gold standard even if the later may have been calculated using more computational resources. This insight will be refined in Sections 5.6 and 5.7 and gives us a good foundation for the remaining tasks of the study.

## 5.6 TEST #2: TOLERANCE TO VARIATIONS IN SHUTTER TIME AND ANTIALIASING LEVEL

The next study focuses on the parameters that can be considered specific of the rendering algorithm: shutter time and antialiasing level. For someone involved in the most practical aspects of the rendering pipeline, these are the only dimensions where he will be allowed to have a direct impact on.

In this section we will determine how efficient observers are when two different stimuli need to be matched, or said in a different way, in which cases two sequences cannot be differentiated even if they have been rendered with distinct settings.

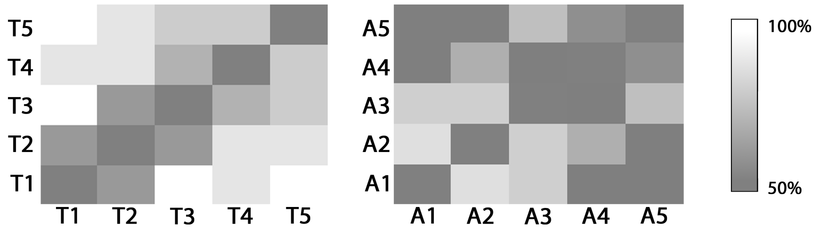

Figure 5.5: Answers to the shutter speed (left) and antialiasing level tasks (right). The gray level in cell $(i, j)$ represents the percentage of answers that were correct when the matching task involved stimuli $i$ and $j$, with darker values meaning less accurate responses. User responses were normalized and are given in terms of a percentage that is equivalent to the response of an imaginary user who provides the same answers. Values T1 to T5 correspond to increasingly longer shutter times, while A1 to A5 represent low to high antialiasing levels.

### 5.6.1 *Description*

The test is composed of two matching tasks. Each task fixes the values for three of the dimensions and use different values for the other one. Fixed dimensions are set to the option that a majority of users preferred in *Test #1*. The free dimension, shutter time for the first task and antialiasing level for the second, uses the five values available.

The set of resulting videos are paired so every possible combination is tested. One of them is randomly used as a reference and shown at the top of the screen. Both videos are then played at the bottom of the screen and the user is requested to select which one matches the reference. Videos are played in sequence and can be repeated as many times as required. Sixteen participants completed each task.

### 5.6.2 *Results*

**Shutter times** can be accurately matched in 86.25% of the cases. A summarized view of the answers is represented in Figure 5.5, left, with brighter areas for those values with the most accurate answers.

Matching is done with high success rate in all the pairs comparing the most differing stimuli. Accuracy is lower in those pairs using videos with less diverse values. That is why darker regions surround the diagonal. We have found that:

- The high number of incorrect matches that is found when comparing pairs using T1-T2-T3 suggests that the strobing artifacts present in these sequences are so objectionable that, in practice, they are visually equivalent.

- Values T4 and T5 seem to be distinguishable. The test in Section 5.5 already raised the importance of excessive blur in the images, so we think the differences in the amount may be a cue that is used to differentiate these sequences.

Matching **sampling levels** seems to be a more challenging task in which the percentage of success drops to 63.75%. This is also supported by an increment in the effort required to complete the taks (an average of 7:49 minutes compared to 6:12 for the previous task).

Figure 5.5, right, shows the results as a matrix. We can observe that:

- The most accurate answers are concentrated in the lowest sampling levels A1, A2 and A3 where the artifacts are more visible.
- Comparisons between higher sampling levels, A4 and A5, produce percentages near chance. For these values, the videos can be considered indistinguishable from a gold standard.

The previous findings suggest that if the HVS is forced to work near its perceptual limits and given a certain level of quality is reached, it becomes less interesting to make further improvements as they become increasingly less noticeable.

It is important to note the lack of accuracy that can be observed in the pairs comparing really low and very high sampling levels.This deviates from the expected trend and the empirical results. We have attributed this singularity to the complexity associated with the task itself. Users informally confirmed that in the shutter time task, the amount of blur and artifacts associated with each level where visible enough to allow an accurate matching. However, we believe that for this task, the lack of strong cues and the importance of visual memory and retention may be factors that need to be considered as a source of bias [57, 247]. This may be the reason behind the reduced accuracy exhibited throughout the test. Section 5.6.3 contains further results on this.

### 5.6.3 *Discounting the effects of memory and matching ability*

In order to cut the previous effects out, we performed an alternative 2AFC subtest with the five sampling levels. Users were requested to select the stimulus they preferred from a pair of alternative sequences. The selection criteria was the perceived quality under similar conditions to the ones explained in Section 5.3. Ten users completed a task composed of twenty pairs of sequences, twice for each possible pairing.

Based on their answers, we perform a significance test following the approach described in previous works [87, 210]. With this method, answers can be clustered in statistically indistinguishable groups based on an analysis of score differences. Two groups can be considered similar when the difference between their scores $R$ is below $\lceil R_c \rceil$. Using a significance level $\alpha$ of 0.01, we compute $R_c$ so that $P(R \geqslant$

$\lceil R_c \rceil) \leqslant \alpha$ [218]. It can be shown [53] that $R_c$ can be obtained from $P(W_{t,\alpha} \geqslant (2R_c - 0.5)/\sqrt{nt})$, with $n$ the number of participants and $t$ the number of samples we compare (20 and 5 in our case). Values of $W_{t,\alpha} \approx 4.604166667$ and $\lceil R_c \rceil = 24$ can be retrieved from Pearson and Hartley's tables [186].

Results show that users consistently prefer high over low sampling levels: A5 was selected over any other level in 59 cases; A4 in 51; A3 in 48; A2 in 34 and A1 in 8. Three groups can be built based on these values: the most voted containing A5, A4 and A3; second most voted grouping A4, A3 and A2; and a final group containing A1. These groups contain stimulus that are perceptually indistinguishable even if users are capable of finding differences between each group. Lower sampling is unequivocally associated with lower quality levels, whereas other stimuli are not so clearly differentiated. In general, these results confirm the capability of the HVS to detect alternative quality levels, and also confirm the suspicion that other factors related to the complexity of the task may affect the performance in the previous matching test.

In general, CG-aware participants show a slightly higher accuracy in their answers: 11% when shutter time is considered and 4% for the sampling level task. Since the videos are not played simultaneously, visual memory and other acquired abilities may be influencing factors. The ultimate intention is determining how effectively spectators perceive motion blurred stimuli and in which cases the lack of quality is noticeable. In the most frequent situations, a single sequence will usually be displayed. The previous tasks can be considered a compromise between this visual experience and the case where several videos are played simultaneously.

With independence with respect to the HVS' capabilities to detect such subtle changes in the individual images of a sequence, there are other factors that we have consciously left outside this study. There are many studies that support the influence of phenomena such as inattentional blindness [147], the attentional spread associated with multiple object tracking [193], change blindness [205] and attentional blink [221]. On the other hand, there is evidence that, in situations where a single element receives all the attentional resources, perception of contrast and spatial frequency is enhanced [140, 1]. As such, the participants in this study have completed the test under conditions where any issues associated to the quality of the images should be easily detected. These studies support our belief that under less constrained viewing conditions, the mechanisms involved in the detection of those artifacts may probably fall shorter and more aggressive simplifications could be possible.

## 5.7 TEST #3: INTERACTIONS BETWEEN SHUTTER TIME AND SAMPLING LEVEL

The objective of this test is uncovering the existence of any interactions between shutter time and antialiasing level that may affect observers' performance to differentiate between alternative stimuli.

Monte Carlo algorithms, namely those inspired on distribution ray tracing, sample the strata defined for all dimensions of the rendering equation simultaneously [253]. Thus the more complex the geometry and shading functions are, the higher the chances these methods require further computation to produce an accurate approximation and avoid aliasing. Using any rendering engine will immediately reveal these underlying correlations.

This section will help support this interpretation based on the results of a new psychophysical test.

### 5.7.1 *Description*

In this test, participants have been requested to complete a 2AFC routine. In this case, the checker material and medium speeds have been fixed in accordance with the preferences manifested in previous tests. Shutter time take the values T3, T4, T5 and antialiasing level use A3, A4 and A5.

Two movies are shown in sequence and observers are requested to select the one with the highest quality. Every pair comes from a pool of 36 tuples containing all possible combinations of the two free parameters. The rest of the conditions and associated explanations are similar as with the previous tests.

### 5.7.2 *Results*

Circular triades have been searched as a method of detecting inconsistencies (i.e. users manifesting preferences that result in contradictions: sequence $X_1$ is preferred over $X_2$, $X_2$ over $X_3$ but also $X_3$ over $X_1$). The coefficient of consistency $\xi$ [116] is found to be 0.6492, which suggests a relatively low number of inconsistencies. The coefficient of agreement $u$, in a theoretical range of $[-1/21, 1]$, is 0.2794, that is, a moderate agreement among users.

The results have been analyzed using a significance test similar to the one detailed in the previous section. In this case, we set the significance level $\alpha$ to 0.01 with $n$ the number of participants and $t$ the number of samples equal to 21 and 9. We compute $W_{t,\alpha} \approx 5.0769231$ and $\lceil R_c \rceil = 36$.

Figure 5.6 shows the resulting groups. Group1 to Group4 have been ordered by decreasing number of votes. Group1, in blue, packages all the images calculated with the shortest shutter time T3 and those com-

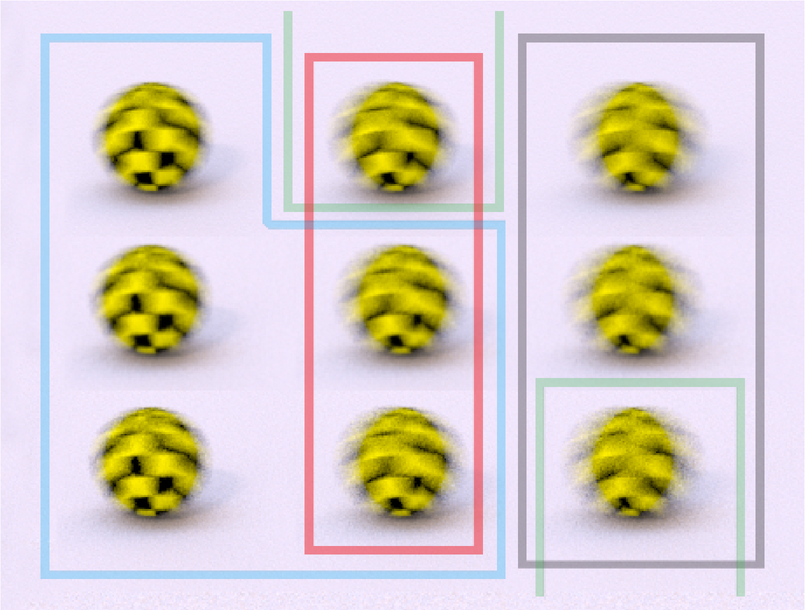

Figure 5.6: Grouping of the scenes rendered with different antialiasing and shutter time settings. From left to right: Shutter time T3, T4, T5; from bottom to top: Sampling rates A3, A4, A5. Group1 to Group4, most to least voted, have been highlighted using blue, red, green and gray respectively.

puted using T4 and the lowest sampling levels. Group2 and Group4, in red and gray, contain all the sequences that have been rendered using shutter settings T4 and T5 respectively.

From the previous partitioning, we can conclude shutter time drives users' answers with the antialiasing level taking a secondary role. The implications are important and twofold:

- first, all the renders computed with any of the settings belonging to a given group are perceptually equivalent to any other render in the same group;
- second, the most expensive renders can be optimized by just selecting the parameters that belong to the same group that require the least computational resources.

In terms of ordering, each sequence has been preferred over any other according to the following number of votes: A4T3, 125; A3T3, 124; A5T3, 119; A5T4, 96; A4T4, 90; A3T4, 78; A5T5, 46, A4T5, 40, A3T5, 38.

All the previous confirms our first findings, suggest the interplay between different dimensions can be useful in the presence of perceptual limits and opens interesting possibilities to reduce computing requirements.

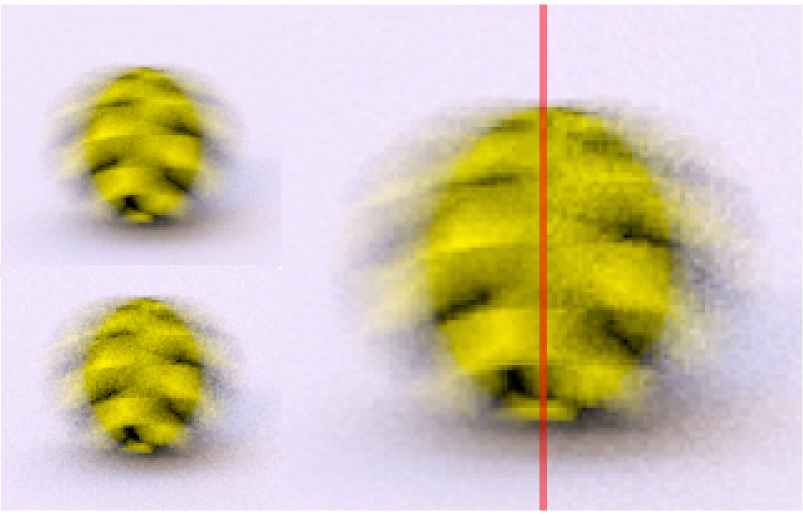

Figure 5.7: A pair of perceptually equivalent renders. Images have been computed using different sampling settings (left,bottom: T5 and A3; left,upper: T5 and A5). Right image shows a zoomed version of the bitmaps. The lower image was rendered in half the time to render compared to the upper one.

## 5.8 CONCLUSIONS AND FUTURE DIRECTIONS

The existing knowledge on the perception of motion blurred images was based on stimuli that was not representative of natural world scenes. We have expanded the results to those situations where the stimuli is based on simple photorealistic images.

We have found evidence that psychophysical limits, in line with the ones found with simpler stimuli, are applicable and take an important role on perceptual tasks. We believe there is also a spread of the attentional resources to complete more demanding tasks, as such; the tolerance to noise and high frequency artifacts is notable.

This insight comes from different psychophysical experiments that are based on high level properties of the 3D scenes and parameters of the render algorithm. These are object material, object speed, the shutter time simulated by the virtual camera and the antialiasing levels applied by the algorithm. The results suggest that, in some cases, images can be produced using aggressive simplifications without degradation of the perceived quality.

In a practical situation, these results can provide methods to drastically reduce the render times and the computational resources required. As an example, Figure 5.8 shows two pairs of images, each one using the same object rendered using alternative settings. In each pair, the image that was rendered with the highest sampling level took double the time to render. Even if these images are noticeably different when compared side by side, they are perceptually indistinguishable when played at the viewing conditions considered in the study.

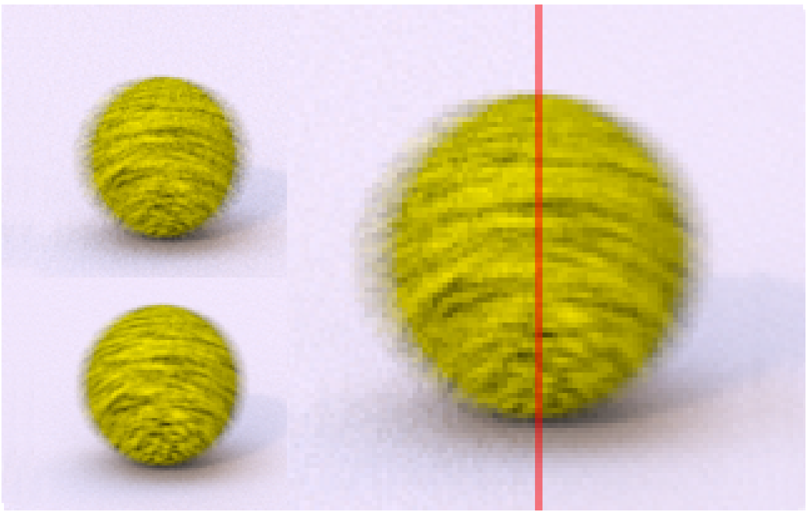

Figure 5.8: A second pair of perceptually equivalent renders. Images have been rendered using alternative sampling and shutter time settings (bottom: T3 and A5; upper: T4 and A3). Right image shows a zoomed version of the original images. In this case, the upper image was rendered in half the time.

There is still work to be completed to to provide more detail for the material and object speed dimensions. The results can also be extended to a wider set of scenarios covered by the rendering equation. Elements such as geometrical occlusion, evolving shading, level of detail or morphing geometry may also be explored as they are known to produce a shift in the appearance of the objects. This may potentially lead to formal models explaining the existing perceptual limitations as well as an expansion of concepts such as visual equivalence [197].

# 6 INFLUENCE OF STEREO VISION ON MOTION BLUR PERCEPTION

In the previous chapter we described a set of perceptual tests that uncovered aspects on the ability of the HVS to differentiate between motion blurred stimuli. A possible use of this knowledge is the optimization of rendering processes that involve temporal sampling. In many cases, these processes are quite expensive and any reductions of the computation times are always welcome, even more if they do not impose any severe quality losses. For monoscopic rendering, we found time savings of one half were easily achievable.

In this chapter we will extend those findings to the rendering of stereoscopic movies. We will focus on the perceptual aspects of the problem and will identify which characteristics are the most salient for a human observer. As with the single-view experience, we will suggest different approaches that will help reducing the requirement of rendering processes.

Our initial assumption is when observing stereoscopic image sequences, the HVS' limitations are more prominent and as a consequence, it may be even more resilient to rendering artifacts. As before, we will take advantage of any findings to simplify image computation. Time savings are expected to be significant knowing that, in the absence of rendering optimizations, a stereo pair takes twice the time to render than a single view.

Stereoscopic media has increasingly higher demand in the industry where it is one of the current most notable trends. Our conclusions may have a direct impact on production pipelines and can potentially reduce the cost of content production. That is why we consider it is an interesting area to research.

## 6.1 PREVIOUS WORK

Chapters 4 and 5 already introduced many references on motion blur rendering algorithms and the aspects associated to the perception of motion blurred stimuli. In this section we will extend the previous to the works that are specific of stereoscopic rendering and HVS' responses to stereo pairs.

### 6.1.1 *Stereo vision*

The brain is capable of extracting relative depth information from a single monoscopic view. However, the most powerful cues are con-

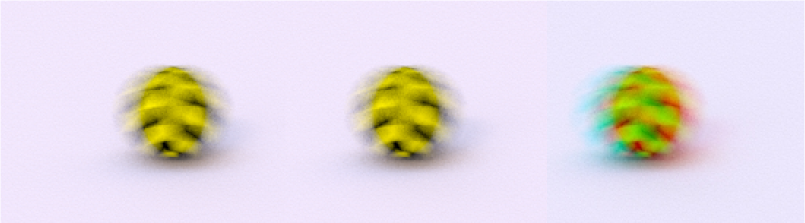

Figure 6.1: An example stereo pair used in the study. From left to right, the left view, right view and a red-blue anaglyph computed from both views.

tained in the combination of the left and right eye views of a scene and the disparity between both. This allows extracting an accurate description of the depth of the scene. Stereopsis is the process by which a viewer infers depth information based on the contents of the images retrieved using binocular vision.

In nature, stereo vision comes from using two images of the same object, each one produced from a different observation point: the position of the eyes. Once acquired, both images are compared to find correspondences. After finding those matching points, the distance to a given image detail can be inferred. Most information comes from two matching cues: accomodation and vergence. The first one defines where both eyes are focused, while the second describes the angle between the focus point and the eyes. Different theories have been proposed on how the information is understood, the internal processes involved as well as how images are finally fused [214] but they mostly coincide in the fact that there is an underlying triangulation process that works out the world position of each feature.

The brain is flexible enought to generate 3D information using incomplete matches and in the absence of clear features. That is the case of random dot stereograms [108], anaglyphs whose left and right images have differing luminances and crominances [146], as well as images with missing correspondences [5]. To some extent, we take advantage of these abilities, as the stimuli we use is noisy and this noise does not match between views. This is a standard situation in the industry, where both eyes are generally rendered using independent processes.

### 6.1.2 *Displays and visual confort*

Interest in stereo content has grown, faded out and re-emerged several times in the last few centuries [290]. Many methods have been envisaged to deliver the content to the spectators. Most of them, independently of the technology they rely on, are based on a surface where two different images, one for each eye, are displayed side-by-side, simultaneously or in rapid sequence.

The most recent state of the art devices use screens and projectors. In the case of field-sequencial stereo, or active stereo, the images for each eye are briefly displayed one after the other. This sequence is synchronized with the occlusion of each of the eyes in alternation. Consumer products are already capable of showing images at 120Hz which allows a frequency rate of 60Hz for each eye. In most cases this is above the flicker-free threshold. Passive alternatives project both images simultaneously and a pair of glasses with special optical properties (polarized, tuned to certain spectral bands, ...) let the correct image reach the eye. .

Using flat images requires to break the correspondence between accommodation and vergence: Eyes are required to always focus on the surface of the screen while vergence, as in nature, is determined by the the point of interest and the differences between the images. This may result in fatigue after reduced viewing times [98] [264].

With independence of this issue, display technologies are associated to different sets of artifacts. In some cases, the problem is a suboptimal temporal interpolation method [58] or the excess of motion blur due to the sample-and-hold effect and the limited capacity of the device to quickly change the value of a pixel [246], the crosstalk and ghosting effect between eyes or even the reduction in the luminance imposed by stereo glasses and projection surfaces [248]. The work of Boev et al. contains an overview of many of them [15].

### 6.1.3 *Stereo artifacts and image quality*

The literature covering rendering artifacts is vast and in many cases, rendering advances have been focused on eliminating or reducing their importance. It is not the target of this chapter to make a detailed discussion, but interested readers can find a complete compendium of the references on motion blur rendering and its associated limitations from the cites in chapter 4.

Apart from the difficulties associated with single-image computation, stereo rendering poses a extra set of challenges. In the simplest case, a stereo render can be considered the aggregation of two independent images. However, there is plenty of evidence to support the fact that the overall experience greatly depends on the relation of both images. Compression artifacts have already been studied in this context, with results showing that more agressive compression parameters lead to similar depth perception even if image quality and eye comfort may be reduced [219]. The same study suggest there are uncovered relations between compression ratios and coding asymmetry. Current standards are already taking advantage of them to increase compression ratios [254].

Some other studies suggest that when stereo pairs mix different resolutions for each eye, the perceived quality is more closely related to the image with the highest quality [233]. This effect, known as binocular suppression has recently been used to optimize the generation of images [21] . Even more, frequencies found in just one of the views may be perceived in the resulting stereo experience [234]. The connections to our case are pretty direct, since alternative rendering settings produce changes in the frequency contents and intensity of the defects in the images.

The effects of image blur in stereo pairs has also been studied. In general, blur can be considered one of the most important cues in the perception of depth [121]. Campisio suggest a series of objetive metrics that estimate the impact of image blur [31]. In some cases, mismatches between the depth suggested by image blur and binocular disparity may lead to disconfort due to conflicting depth perceptions [133].

Sampling artifacts can also induce defects in the perception of depth as a result of inaccuracies in the size, position and interocular disparity of the views of an object [188]. Compression algorithms may also be the source of similar problems [32].

There are studies focused on correcting 3D image artifacts due to display devices, camera settings and image processing techniques [155] While related to the previous, our study assume artifacts are an intrinsic feature of the images that does not need to be reduced but leveraged. In contrast, our interest is determining the degree of perceived quality between different levels of sampling artifacts. In the case of motion blurred stimuli, the contours of the objects are semi-transparent and rapidly evolving so the consequences of sampling artifacts may differ.

In summary, there are many indications that image artifacts can be tolerable to some extent. We already found evidence that under certain conditions, they may be indistinguishable. This is the main idea behind our research.

### 6.1.4 *Rendering optimizations*

The research community is aware of the fact that, for a stereo render, a naive algorithm that independently renders each eye simply duplicates computation times. In many cases, the operations required for each eye are strikingly similar or near equivalent. Chapter 4 already contains many algorithms that can produce motion blurred stereo renders by applying the same process to each of the points of view of the scene.

More efficient algorithms are based on reprojection and postproduction, but they cannot offer a complete solution for the generation of accurate stereo renders. The information contained in a single frame is not enough to generate the view for the other eye [39]. Also, view dependent phenomena such as reflections and refractions cannot be

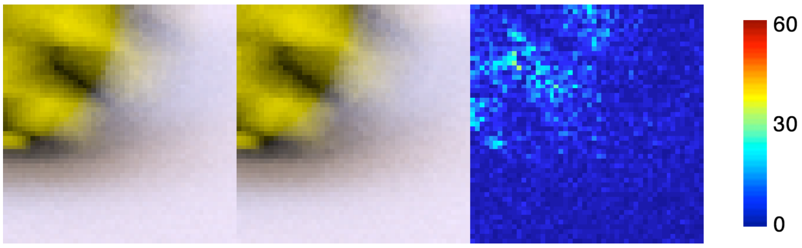

Figure 6.2: Zoom of a stereo pair and its absolute pixel differences. Images correspond to the object moving at high speed, rendered using long shutter time and high number of samples.

accurately reused [171]. In some cases, GPU acceleration and geometry shaders can be used to reduce the impact on render times [56].

There are several techniques that use the coherence between stereo images. Taking advantage of depth information computed at render time and the images previously computed, Didyk suggested a method that can halve the computation times to produce a stereo pair [59].

The family of algorithms for novel view generation and image based rendering may also help alleviating the burden of the renders.They can use images and recorded footage to generate a tridimensional representation of the scene, and produce new views of it. Many of them are based on the Lumigraph [80] and Light-Field rendering [136]. Some of these techniques have been expanded to work with video sequences [289]. .

Even if there are many related fields that have been studied intensely, we believe this is the first time that research has focused simultaneously on stereo vision, realistic scene rendering, motion blur and the effects of image artifacts on human perception. We shall see that the practical oportunities as numerous and our findings may have a direct application to areas such as production rendering of stereoscopic feature films and visual effects.

## 6.2 DESCRIPTION OF THE STIMULI

The studies in this chapter use stimuli that is based on the same scenes described in section 5.3. The rendering method is also similar and it is based on a template scene that has been modified according to the values of each of the dimensions described in table 5.1.

As with our previous study, we have focused on the two main dimensions where a technical artist will be allowed to work on: antialiasing level, that is number of samples per pixel, and shutter time. The relative influence of each one and the relations between them will be studied using a series of user trials that will be described in Section 6.3.

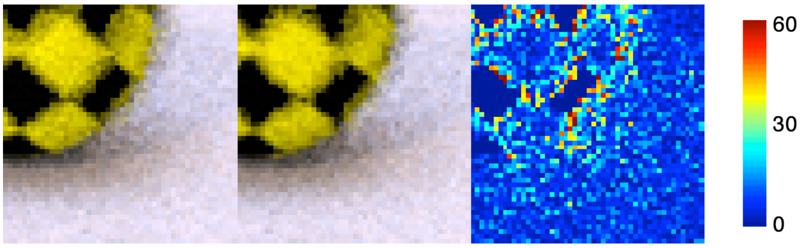

Figure 6.3: Zoom of a stereo pair and their absolute pixel differences. In this image, a high speed object has been rendered using long shutter time and low number of samples. Comparing this image with Figure 6.2 shows significantly more visible differences.

### 6.2.1 *Stereo pairs*

Each of the resulting sequences has been rendered using a stereo camera rig. This setup uses three cameras, one in the middle that matches the one used in the previous chapter and two extra side cameras representing the eyes. The last two are separated using a standard interocular distance (6.5 centimetres). All three cameras use the same focus point, which matches the one in the monoscopic renders. The central camera is used only to align with the original one, and only the images produced by the stereo cameras are used in the tests.

As before, the output is composed of 225 sequences, rendered using the Arnold renderer, a commercial implementation of the Monte Carlo algorithm. Each of the eye-frames are composided side-by-side resulting in a stereo pair of 1280x360 pixels that is displayed at 30 frames per second. One of the frames is shown in Figure 6.1.

Stereo vision is achieved using a NVidia Vision kit. It integrates shutter glasses to alternatively hide the left and right images displayed on a LCD monitor. The experiments are written in Matlab using the Psychotoolbox library [17]. This library provides a simple interface to configure both the stereo display and the glasses as well as very accurate synchronization between both.

### 6.2.2 *Monte Carlo, noise and stereo*

Monte Carlo renderers sample each pixel using pseudo-random sequences that, in general, are not eye dependent. In case the sampling level is lower than required, these methods tend to produce noise. For single-eye sequences, this produces temporal artifacs in the form of flickering.

As it is frequently done in the industry, each image in our stereo pairs is rendered independently. Stereo jobs are prepared so each image can be rendered in parallel by a different machine, with independence of how the render is performed for the other eye. The pattern of noise that is produced, even if static for a single eye, may not match between different eyes. This results in inter-ocular disparity which, in a extreme case, can make stereo fusion difficult and prevent stereopsis.

In the current scene, the most problematic areas are produced by shadows, global illumination, texture sampling and motion blur. The area that may show the most visible issues is the motion trail, which represents a significant percentage of the image pixels the bigger the faster the motion. Speed increases the chances of detecting issues in the image. Figures 6.2 and 6.3 shows examples of images rendered with different sampling levels. Each eye is compared side-by-side and their pixel differences shown in false color. The images show that the absolute differences are much more intense in the low quality render which is results in more intense interocular disparity.

## 6.3 perceptual tests: overview

As in the case of monoscopic images, we have designed a set of tests to determine the influence of each of the rendering parameters in the perception of motion blurred stimuli. From previous knowledge and in the absence of clear assumptions, we can anticipate two outcomes based on two extremes:

- The HVS is less efficient detecting rendering artifacts when the stimuli is composed of stereo pairs. So renders can be accelerated using aggressive assumptions without visual degradation.

- On the other hand, stereo fusion may impose severe requirements between the images in the stereo pair. Rendering artifacts may violate those requirements and generate disconfort or prevent stereopsis. So renders may require increased quality and as a consequence, extended rendering times.

We will focus our efforts on four different tests, that mimic the ones that thrown the most salient results with monoscopic images. In particular, see Table 6.1, we will perform all the 2AFC tests and exclude the matching tasks.

Depending on the task, between 18 and 32 participants completed the tests. The proportion of males to females is roughly 3:1. Ages ranged between 20 and 40 years old, with an average of 28. Half of them had working experience in computer graphics. All of them self-reported normal or corrected to normal vision.

The sequences where displayed in similar dim lighting conditions using a LCD monitor 17.3" diagonal, 1680x1050x120Hz, sRGB color space, contrast 20000:1, luminosity 300 $cd/m^2$, gamma 2.2. NVidia 3D Vision shutter glasses were used to generate stereo perception. Participants were placed at roughly 60 cm of the screen. In these

| TEST | TYPE | DIMENSIONS |
|---|---|---|
| #1 - Broad comparison | 2AFC | All (2 each) |
| #2 - Shutter time / Antialiasing level | 2AFC | Shutter time (5) / Sampling level (5) |
| #3 - Shutter time and antialiasing level | 2AFC | Shutter time (3), sampling level (3) |

Table 6.1: Psychophysical tests completed by the participants in the study. The *Dimensions* column lists the dimensions considered by each test, with the number of values used for each axis between parentheses. The dimensions not listed in a given test are sampled at a single value.

conditions each of the sequences subtended 14.4 degrees of visual angle.

## 6.4 TEST #1: BROAD COMPARISON

In the previous chapter we found that users preferred the noise material, medium speed and medium shutter time over any other alternatives. Participants supported their decision based on the absence of visible artifacts, a moderate amount of bluriness and the increased realism of the image.

In this section we repeat the same test, this time using stereoscopic stimuli. As before, participants were asked to select one out of two videos according to the question *Which sequence has higher quality?* Prior to starting the task, every subject was briefed about the most important image features that characterize high quality renders. They also had the opportunity of practising with pairs of stimuli whose differences in quality were so obvious that the answers were straightforward.

Figure 6.4 shows a graphical representation of the answers, classified by the type of participant. Figure 6.5 compares the statistics found for the same test performed with monoscopic and stereoscopic images. In all cases, a certainty of 95% has been confirmed using a Chi-square test.

A detailed review of the answers shows the following facts:

- Preference for the noise material is higher than for the checker material. This result is equivalent to the one shown for monoscopic stimuli (a difference of less than a 2%) and we attribute it to similar causes: the existence of stroboscopic artifacts. Even if these issues exist for both materials, the checker material contains bigger patches with flat color that tend to make them more visible.

- Stereo speed preferences are balanced. Note that the percentages with respect to the monoscopic test differ in more than 15% and error tolerances are slightly higher. However, considering pure statistical results, both are within similar ranges. Even if the

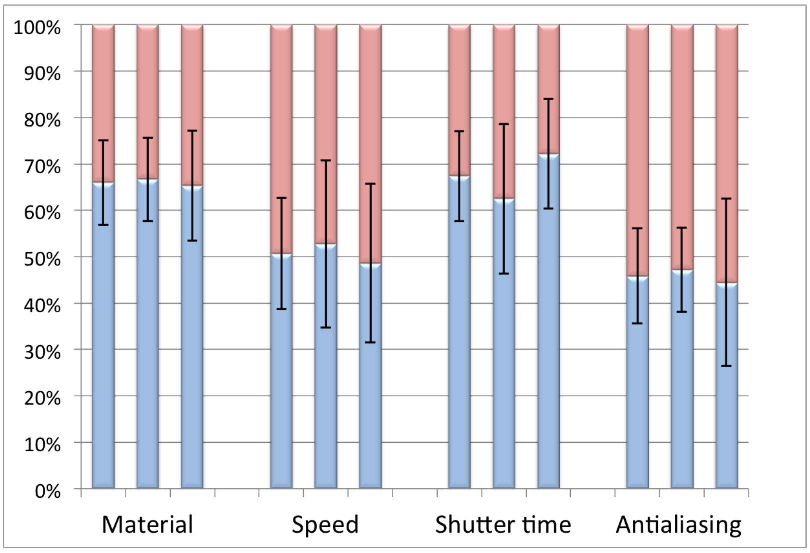

Figure 6.4: Percentage of trials where a particular animation is perceived with higher quality. Blue/red bars represent the corresponding values for noise/checker material (M3/M2), medium/high speed (S2/S3), medium/long shutter time (T3/T5) and medium/high antialiasing level (A3/A5). Each group of bars shows from left to right, the results for all, experienced and naive users, respectively. Error bars represent the percentage of error.

quantitative evidence cannot point to one of the speeds, informal observation of the images using the stereoscopic setup lead us to think the extra cues associated with stereo vision (parallax, image disparity, increased perception of depth and volume, ...) may be aiding the HVS to understand the scene. Also, the excess of noise that was reported for the fastest speed in single eye tests seem to be less noticeable.

- Shutter times is the result where preference seems to be more defined. As with previous results, the excessive softness reported for the fastest motion seem to be reason behind it.

- Also, antialiasing levels are equally preferable. As with the monoscopic stimuli, the visual differences predictor shows the noise between the aligned stereo pairs are above JND (just-noticeable-differences) as well when the same eye is compared between different sampling levels.

- Even if the images for each eye are generated with two independent render processes and the patterns and associated artifacts do not correspond pixel by pixel, see Figure 6.2 and 6.3 for examples, participants did not report any issues and were capable of completing stereo fusion without noticeable discomfort.

All of the previous points makes us think that attentive resources may be diverted toward generating stereopsis. These results seem to

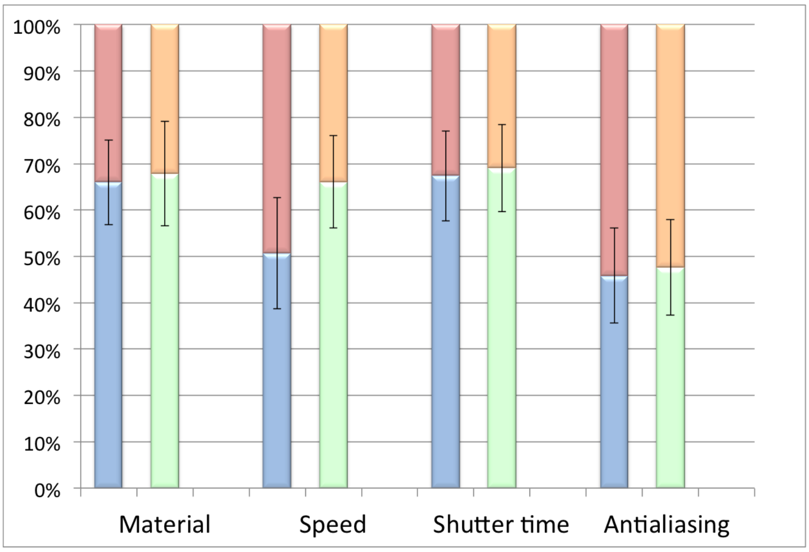

Figure 6.5: Comparison of the answers for mono and stereo stimuli. Bars represent the percentage of trials where a particular animation is perceived with higher quality. Green/red and orange/blue bars represent the corresponding values for noise/checker material (M3/M2), medium/high speed (S2/S3), medium/long shutter time (T3/T5) and medium/high antialiasing level (A3/A5). Each group of bars shows from left to right, the results for stereoscopic and monoscopic stimuli, respectivelly. Error bars represent the percentage of error.

suggest that cognitive tasks such as completing the fusion between the two images may take precedence over other tasks like noise and artifact detection. This is an interesting finding that will need to be confirmed in the next sections.

## 6.5 TEST #2: TOLERANCE TO VARIATIONS IN SHUTTER TIME AND ANTIALIASING LEVEL

In this section we will determine the observers' relative capability to detect changes in image quality when the rendering method is modifies the value of a single of the dimensions. From the four parameters studied in the previous section, this exercise focuses on shutter time and antialiasing level. These are the settings that are determined by the algorithm and not the aesthetics of the scene. While they have a direct impact on the quality of the images, they do not essentially change their contents.

As a secondary target, we will try to identify the common characteristics of the groups of movies that produce similar perceptual effects. This provides opportunities to optimize rendering jobs, and in case the alternative parameters are selected from the same perceptual group, without loss of quality.

### 6.5.1 *Description*

In the previous chapter, we determined that certain aspects such as visual memory and user professional background may affect the performance of the observers to match video sequences. A secondary test was required to discount the effects of these elements (see section 5.6.3 for a complete description). The current test aims to perform an exercise under similar conditions, that is, reducing the influence of the HVS' abilities that are not directly related to the detection of motion blur and rendering artifacts. As a consequence and building on our previous findings, we have decided to focus on tests designed around 2AFC tasks.

This test is composed of two different tasks, each one fixing the values for three of the dimensions and varying the remaining one. Fixed dimensions use the preferred setting as from *Test #1*. In those axis where there is no clear preference, the same setting as in the monoscopic trial is used. The free dimension, shutter time for the first task and antialiasing level for the second, uses the five values available, T1 to T5 and A1 to A5.

Videos are paired so each possible combination is used. Pairs are shuffled and within each pair, videos are displayed in random order. This ensures learning effects do not alter the answers. The videos of each pair are played in sequence and participants can repeat them as many times as required. Users are requested to select the video they consider has the higher quality. Sixteen users completed each task and each one completed two full tests.

After gathering the answers, we have performed a statistical significance test with similar characteristics as the one described in section 5.6.3. This method arranges answers in statistically indistinguishable groups based on an analysis of score differences. Like with the previous tests, we set a significance level $\alpha$ of 0.01 and a value of $W_{t,\alpha} \approx 4.604166667$. In this case, $\lceil R_c \rceil = 30$, determined from $n = 32$ participants and $t = 5$ video samples. Since both tests use the same $n$ and $t$ values, the previous coefficients are valid for the analysis of shutter time and antialiasing level.

### 6.5.2 *Results*

Table 6.2 shows the number of times each video has been selected over the other video from the same pair. After ordering the sequences by decreasing preference, we have performed a statistical analysis to determine the number and contents of the perceptually equivalent groups. Two videos are considered statistically equivalent when the difference in their scores is lower than the value $\lceil R_c \rceil$, 30 in this case.

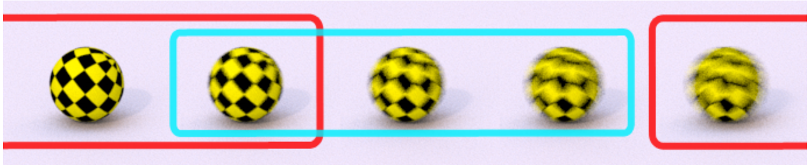

Figure 6.6: Snippets of the stimuli used in test #2 (shutter time). From left to right, T1 to T5. Images are clustered in perceptually equivalent groups. In blue, most voted sequences. In red, the images with the lowest perceived image quality.

| SHUTTER TIME | # VOTES | SAMPLING LEVEL | # VOTES |
|---|---|---|---|
| T1 (0%) | 34 | A1 (4 spp) | 48 |
| T2 (25%) | 60 | A2 (128 spp) | 63 |
| T3 (50%) | 85 | A3 (256 spp) | 77 |
| T4 (75%) | 87 | A4 (384 spp) | 68 |
| T5 (100%) | 54 | A5 (512 spp) | 64 |

Table 6.2: Number of votes per sequence. The table displays the number of times a given sequence has been selected over other sequence for the variations of shutter time and sampling levels. Between parenthesis the percentage of the frame time the shutter time remains opened and the number of samples per pixel.

### 6.5.2.1 *Shutter times*

Under those conditions, two different perceptual groups emerge when different shutter times are considered. The most voted one, surrounded by a blue line in figure 6.6, contains values T4, T3 and T2, while the less-voted group, in red, comprises values T2, T5 and T1. The internal order of each group corresponds the relative preference measured in the tests. Each group contains stimuli that are perceptually indistinguishable.

The following conclusions can be extracted:

- High quality results are associated with middle shutter time values (25, 50 and 75% of the frame time). On one hand, excesive bluriness has already pointed out as a signal of low quality and lack of realism. On the other hand, a complete absence of motion blur tends to produce strobing artifacts. These values are a compromise between both extremes and seem to support our previous findings for monoscopic stimuli.
- Lower quality results are represented by values T2, T5 and T1, that is, the most extreme shutter times. Artifacts associated with those values may be so perceivable that users can easily determine their existence. This is also supported by the results for the matching task in section 5.6.2. In that case, artifacts were

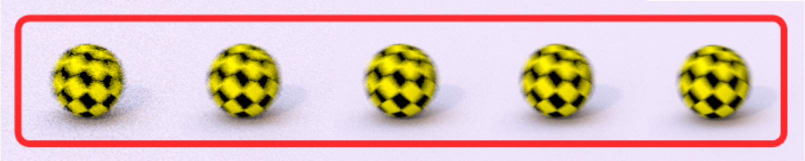

Figure 6.7: Stimuli used in test #2 (sampling level). From left to right A1 to A5. Images are clustered according to users answers yielding a single perceptually equivalent group.

used as a cue to differentiate between sequences which may also be the current case.

- Even if strobing and excessive blur are issues with different visual appearance, users seem to equally weight them.
- Both groups overlap at value T2. The difference in votes are high enough to generate different arrangements, but at the same time differences are not so detectable to detach each one . This may indicate the HVS is working near its perceptual limits.

The integrity and quality of the results has been checked using a circular triades analysis. This method can determine the existence of contradictions in the scores and users answers. In this case, the coefficient of consistency $\xi$ is 0.7813 in the $[0, 1]$ range which corresponds to a relatively low number of inconsistencies. The coefficient of agreement $u$ is 0.144 in the range $[-1/31, 1]$. This range represents increasingly higher degrees of agreement. This value represents a moderate agreement among users, which gives room to think users arrive to similar conclusions but there is still certain variations in their answers.

### 6.5.2.2 *Antialiasing level*

A analagous test was performed based on different antialiasing levels. These values correspond to sampling levels of 4 and 512 samples/pixel for A1 and A5, and roughly equidistant values for the rest.

The most salient conclusion is the lack of any clear preference for a single value or a group of them. Every sampling level receives similar number of votes which may be considered a strong indication of the inability of the HVS to differentiate between different stimuli.

Back then, when monoscopic sequences were considered, matching tasks showed significant inconsistencies. This forced us to complete a set 2AFC task to discard memory or learning effects. The second test clearly showed the HVS is capable of determining which sequences have higher quality. In that case, we found three overlapping groups (low to high quality: A1, A2/A3/A4 and A3/A4/A5, more details in section 5.6.3).

In the case of stereoscopic stimuli, the HVS seem to be incapable of finding any differences. This results in a single group containing all the previous sampling levels. A reduction in the consistency of the answers results in a coefficient $\xi$ of 0.5062 in the $[0, 1]$ range. Agreement between participants is also lower, $u$ is 0.0177 in the range $[-1/31, 1]$.

In summary, the uniformity in the number of votes added to lower consistency and agreement levels seem like strong indicators of the reduced ability to detect noisy artifacts when stereo pairs are used.

This is an important result, as for the cases considered, all the render modes tested are equivalent to a gold standard. This is true with independence of how expensive they are to produce. The same conclusions cannot be taken for monoscopic vision where, even if allowing some flexibility, the perceptual limits are clearly defined.

## 6.6 TEST #3: INTERACTIONS BETWEEN SHUTTER TIME AND SAMPLING LEVEL

The two previous sections have evaluated the capacity of the HVS to differentiate changes in the rendering parameters when modified one at a time. This section will consider modifying both of them simultaneously. As we will see, the following test will reveal a series of interesting interactions between both axis.

In more general terms, this test more closely resembles practical situations where different rendering settings can be balanced so the time required to produce a sequence is reduced. The immediate result is a set of applicable directions on how to produce stimuli that is perceptually equivalent.

### 6.6.1 *Description*

In this test, participants have been requested to complete a 2AFC routine. In this case, the checker material and medium speeds are fixed in accordance with the preferences manifested in previous tests. Shutter time takes the values T3, T4, T5 and antialiasing level use A3, A4 and A5, which produces a set of nine different video sequences.

Movies are paired together and each pair is shown in sequence. Every pair comes from a pool of 36 tuples containing all possible combinations of the two free parameters. Observers are requested to select the one with the highest quality. The rest of the conditions and associated explanations are similar as with the previous tests.

### 6.6.2 *Results*

Results have been clustered according to the number of votes received by each sequence. In this case, significance tests are completed using values of 0.01, 18 and 9 for $\alpha$ the significance level , $n$ number of partic-

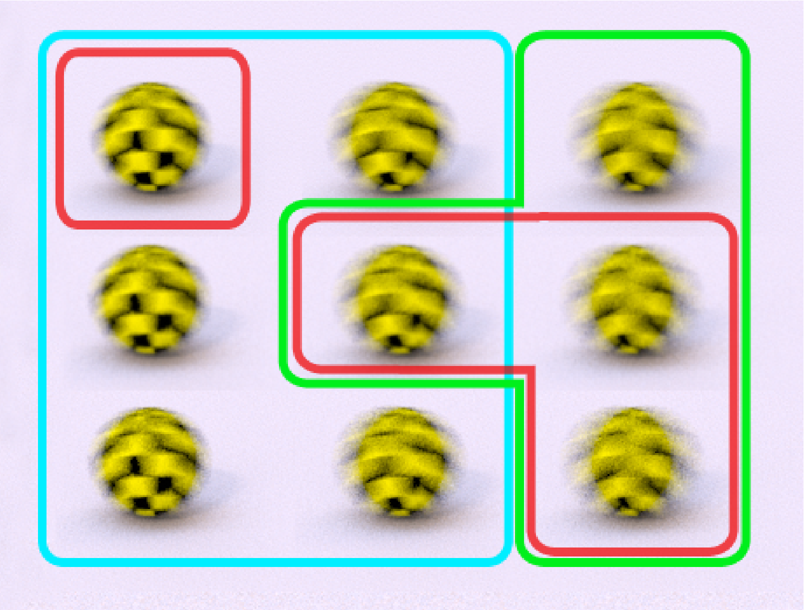

Figure 6.8: Grouping of the scenes rendered with different antialiasing and shutter time settings. From left to right: Shutter time T3, T4, T5; from bottom to top: Sampling rates A3, A4, A5. Group1 to Group3, most to least voted, have been highlighted using blue, red and green respectively.

ipants and t the size of the stimuli set respectively. The corresponding clustering factors are $W_{t,\alpha} \approx 5.0769231$ and $\lceil R_c \rceil = 33$.

Results are graphically shown in Figure 6.8. The most salient conclusions are:

- There are three perceptually equivalent groups. Groups have been ordered by decreasing number of votes from Group1 (blue) to Group2 (red) and finally Group3 (green).
- The most voted group corresponds to the lower and medium shutter times. This closely matches the results found for the monoscopic experience.
- The two less voted groups correspond to the medium and longest shutter times. These two groups are almost a complete overlap. .
- Most of the groups and their corresponding boundaries are associated with changes in shutter levels more frequently than antialiasing levels. This is a result similar to the one found for monoscopic perception and strongly suggest the HVS is more tolerant to noise than it is to strobbing artifacts or excesive blur.
- There are significant overlaps in the region that corresponds to the medium shutter times. This seem to suggest the HVS cannot determine the relative quality of the images when this setting is used, and depending on the pair that is compared to, it can be seen as high quality or lower quality.

- With minimal exceptions, the HVS seem to be incapable of differentiating between sampling levels. This is a result that was previously found for the study in section 6.5.2.

In terms of ordering, each sequence has been preferred over any other according to the following number of votes: A4T3, 94; A5T4, 90; A5T3, 85; A3T4, 83; A3T3, 80; A4T4, 75; A4T5, 49; A5T5, 49; A3T5, 43.

An analysis of circular triades throw results that are inbetween the ones found when sampling levels and shutter times are considered individually. While for antialiasing levels and shutter times the coefficient of agreement $u$ are 0.0177 and 0.144, combined stimuli offers a value of 0.1137 in the range $[-1/35, 1]$. The same occurs for the coefficient of consistency $\xi$: 0.5062 and 0.7813 for antialiasing level and shutter times, compared to 0.6074 for the combined stimuli. This may indicate that there is an underlying progression in the perceptual difficulty associated with the three tests, that ranges from the simplest one, ie. shutter time to the most challenging one, ie. sampling level. Based on the uncertainty and the degree of agreement of the answers, combined stimuli represents a middle point between both.

The final remarks are quite similar, and in some cases go further the ones found for monoscopic perception, that is:

- The HVS is capable of determining which renders have higher quality respect to the ones that have been computed with reduced accuracy. As with the previous cases, shutter time seem to drive users' answers. Sampling levels seem to take a secondary role, to the point of not finding any differences between stimuli rendered using alternative antialiasing levels.

- Any renders computed with any settings included in a given group are considered equivalent by a human observer.

- The most expensive renders can be optimized by just selecting the parameters that belong to the same group that require the least computational resources. In that sense, there are many alternative settings that are equivalent to the gold standards found in the top row of Figure 6.8.

## 6.7 CONCLUSIONS AND FUTURE DIRECTIONS

In this section we have expanded the previous knowledge on the perception of motion blurred images with the use of stereo stimuli. In this case, we have used stereo pairs and performed several tests whose results can be compared to our previous findings for monoscopic renders. As before, we have used photorealistic images using similar scenes, so the respective conclusions can be analysed from a common perspective.

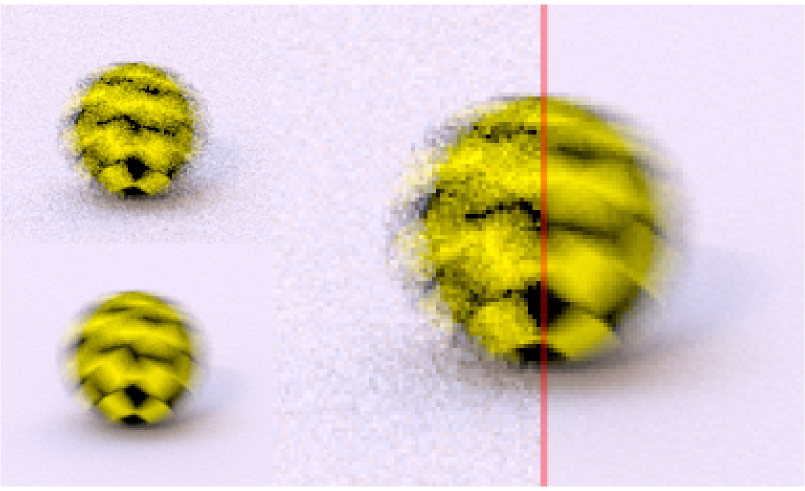

Figure 6.9: Side by side comparison of images from two stereo pairs computed with two extreme sampling levels. Top-left, image computed with low sampling level and long shutter times (T5A1); Bottom-left, image computed with high sampling level and long shutter time (T5A5); Left: Zoom of the two images.

The bigger picture is the psychophysical limits seem to be in accordance in both cases. Our results indicate that the HVS seems to be more sensitive to changes in shutter times than sampling levels. However, a remarkable difference is that, for the cases and scenes studied, the HVS seem to be incapable of differentiating between different sampling levels if stereo stimuli is used.

The studies relied on stimuli that ranged from images that can be considered a gold standard to very noisy sequences. Results suggest more intensely than with monoscopic stimuli, that stereo images can be produced using aggressive simplifications without degradation of the perceived quality.

As an example, Figure 6.9 shows a pair of rendered images that use different shutter times. Even if they are easily distinguisable, they cannot be differentiated when observed using a stereoscopic capable displaying device . In this case, both images generate a high quality perceptual experience, rendering times where comparable, but the left image is noticeably less noisy. When used Monte Carlo rendering, this result can be used to allocate computational complexity independently of the final aspect of the images. For example, given this tolerance to noise, render can focus on sampling object materials while using less resources to their evolution over time.

## 6.8 FINAL REMAKS

Chapters 5 and 6 have provided clear directions on the most influential elements and the interplay between different aspects involved in the rendering of temporally varying scenes. Apart from the empirical results themselves, these observations can be directly applied on demanding fields such as visual effects and film production. Moreover, they support recent industry trends such as stereoscopic rendering.

We believe we have delineated a set of first steps designed to reduce the computing requirements of motion blur rendering processes without altering the perceived quality of the results. Equivalent approaches have already been successfully implemented as part of several rendering algorithms that ignored the time dimension.

We have found evidences that may suggest the method that is used to deliver the images to the spectator introduces new opportunities to play with the balance between a variety of render settings. In many cases, the resulting images continue to be perceived with similar degrees of quality.

Even if our research has been bound to a single scene and a few parameter variations, the results look encouraging. Extending our research to general scenery as well as a more diverse set of rendering settings seem like an interesting line to follow. These extensions may open new possibilities to apply rendering optimizations. These are exciting directions that need to be explored as the benefits for the rendering of motion blurred images can be important.

# Part IV

# CONCLUSIONS

The finishing part of this thesis contains a summary of the techniques presented in the previous chapters. This discussion details their main contributions and the impact in their respective areas of application.

The chapter includes a series of conclusions and final remarks and anticipate future directions that may lead to extensions of this research.

# 7 CONCLUSIONS

This thesis began raising concerns on a trend that seems to be pervasive in the field of computer graphics and photo-realistic rendering: new techniques frequently come associated with a sharp increase in the computational resources needed to calculate the images. With these methods, advance is based on using increasingly more complex mathematical and algorithmic models with enhanced accuracy.

With this issue in mind, our research has focused on finding alternative methods to reduce the requirements of a set of well known methods. All of them are examples of algorithms commonly used by the visual effects, the film industry and the graphics communities.

We have relied on two alternative mindsets:

- The first one is based on extending and improving the methods to increase their efficiency. Results include two different real time algorithms to render participating media and compute screen space antialiasing.
- The second approach tries to reduce the cost of rendering by completely avoiding the computation that is not perceivable by human observers. In this case, we have provided insights to leverage the response of the HVS respect to certain image characteristics and artifacts that can be found in monoscopic and stereo stimuli.

These two general methods and their individual contributions are detailed in the following sections.

## 7.1 VOLUMETRIC RENDERING

In the first block of the thesis, Part i, we have presented a framework capable of rendering inhomogeneous participating media at interactive frame rates. The main contribution of the work is extending the traditional ray marching method to be applicable at interactive rates with simultaneous use of advanced lighting.

Our approach is based on modeling the incident light field as a constellation of point lights computed from a HDR environment while the media is represented as a volume of voxels. Both approaches have a widespread use and there are many algorithms that can generate compatible data sets.

As a second contribution, we have modified the rendering equation to take advantage of the nature of the participating media and have defined a series of assumptions to simplify this model. The resulting mathematical representation is still capable of generating high quality images but it is more efficient to evaluate.

The third contribution is the use of two novel optimization techniques, a view-dependent validity pass and a view-independent distance map, that introduces time reductions of up to 8 times respect to a brute force implementation.

In summary, our technique addresses two of the main issues that bind the efficiency of GPU implementations: the complexity of per-fragment calculations that is reduced using a simplified mathematical representation; and the memory intensive media traversals that are optimized using changes in the algorithm.

Our technique can be extended in several ways. It currently neglects the use of multiple scattering transmission. With the existing methods, this can only achieved at the cost of considerable performance drops or using precomputation. Also, while creating visually acceptable images, assuming an isotropic phase function makes our method less accurate and excludes the use of certain types of media.

There is a wide range of optimization techniques that rely on alternative space classification structures and compression algorithms. They can potentially alleviate the memory footprint and the overhead imposed by the current voxel based representation. Using these methods may compensate for the use of more complex physically-based lighting models.

## 7.2 SCREEN SPACE ANTI-ALIASING

In Part ii, we have presented an algorithm designed to anti-alias pre-rendered images. The method is inspired in the original Morphological Antialiasing by Reshetov and takes advantage of the parallelism that is available in GPU based architectures.

This technique extends existing state-of-the-art techniques in quality and, in some cases, in efficiency. The method can be executed at a moderate cost and it is easy to include in an existing pipeline.

The technique has received significant attention and has been reviewed in several international specialized publications. Different implementations have been integrated into several rendering engines and commercial games. It has also given birth to several recent extensions and improvements.

## 7.3 MOTION BLUR RENDERING

Part iii starts with an in-depth review of the state-of-the-art on motion blur rendering techniques. More than fifty techniques, ranging from the earliest methods from the 70's to the most recent publications, are described and matched up in reference to their strengths and limitations.

One of the main contributions is a taxonomy of methods based on different formulations of a general mathematical representation. This expression models the light integration process that naturally takes place on film and sensors and it is the basis of the simulation rendering methods perform.

The report finishes with a systematic comparison of the methods based on their individual assumptions and provides a set of guidelines to help identifying which methods are more suitable for different practical scenarios.

A second section takes a different approach and analyses the responses of the HVS to motion blurred stimuli coming from a Monte Carlo renderer. To our knowledge, this is the first time a perceptual study has been completed using photo-realistic sequences containing temporal antialiasing. Our findings are specific to a single scene and a set of parameter variations, however, we have been able to extract certain clues that may indicate which features are more important for a human observer.

The study supports our initial intuition: certain psychophysical limits may play an important role with respect to the visual impact of image artifacts. The net effect is a reduction of the importance of issues such as excessive blur, flickering and strobbing. Moreover, the degree of tolerance to noise and high frequency artifacts is, in the cases considered by us, notable.

These results suggest that images may be produced using aggressive algorithmic simplifications without degradation of the perceived quality. We have identified several situations where a lower quality sequence is indistinguishable from a gold standard. In practical terms, this means that a faster and a less accurate render may be used instead of a more expensive one.

As important as these findings it is the fact that variations in shutter time tend to be more visible than alternative sampling levels. Given that a minimum level of quality has been reached, observers seem to be more inclined to detect changes in scenes using different exposure ranges than in scenes computed using alternative sampling levels.

This is an interesting result, that fits in the design of current rendering pipelines: Shutter times are usually fixed on the basis of given image aesthetics or camera settings, while sampling level is an algorithmic setting that is adjusted to balance quality level and cost. This has a direct practical implication meaning that a production crew in charge of the optimization of renders can leverage the lower sensitivity to the number of samples per pixel and reduce the time taken to compute each frame at the same time the perceived quality is kept constant.

Feature film and VFX studios are continuously demanding improvements in the way stereoscopic images are produced. The last part of this thesis follows this trend and extends the previous results to stereo rendering of motion blurred scenes.

Using a similar set of perceptual tests, this time performed with the aid of consumer grade shutter glasses, we have found the HVS may be more elastic to strobbing artifacts and noise. All the sampling settings considered fall in the same perceptually equivalent group. As before, rendering optimizations may be designed on this basis. Even more, the gains can be significant since stereo sequences are much more expensive to compute than single eye renders.

Using two independent Monte Carlo renders for each stereo pair produces artifacts that are different for each of the eyes. This generates interocular disparity which in extreme cases can impede stereoscopic fusion. Participants have confirmed stereopsis occurs without increasing fatigue. At least for the stimuli used in the study, it is safe to compute images independently and algorithms are not required to embed any type of sampling coherence.

In summary, the results suggest that limitations of the HVS may help reducing the requirements associated with high quality renders. Also, the method used to deliver the images to the spectator introduces new opportunities to play with the balance between a variety of render settings.

## 7.4 FINAL REMARKS

This thesis has introduced new algorithms to efficiently solve two interesting problems: computation of photorealistic images of participating media and postprocessing of antialiased images.

We believe we have also depicted the first steps to designing more efficient methods to render motion blur on the basis of taking advantage of the characteristics of the human visual system. These conclusions are applicable for standard sequences and stereoscopic footage.

Even if the results are promising, there is plenty of room for exploration. There are many aspects that still need to be studied, with may lead to opportunities to improve current pipelines. Specifically, the perceptual aspects of image rendering provide exciting directions that need to be analysed as the benefits for the rendering of realistic images may be significant.

# 8 CONCLUSIONES

Esta tesis comenzó poniendo de manifiesto una tendencia que parece estar generalizada en el campo de los gráficos por ordenador y el render foto-realista: las nuevas técnicas vienen frecuentemente asociadas con un incremento importante en los recursos computacionales necesarios para calcular las imágenes. Con estos métodos, los avances vienen dados por la utilización de modelos matemáticos y algorítmicos de complejidad creciente.

Con este problema como motivación, nuestra investigación se ha centrado en encontrar métodos alternativos para reducir los requerimientos de un conjunto de algoritmos bien conocidos. Todos ellos son ejemplos de algoritmos utilizados con frecuencia en el campo de los efectos especiales, la industria del cine y la comunidad de render.

Hemos utilizado dos aproximaciones alternativas:

- La primera basada en extender y mejorar algoritmos existentes de forma que su eficiencia se vea incrementada. Los resultados incluyen dos métodos nuevos para el render de medios participativos y la generación de anti-aliasing en espacio de pantalla.
- La segunda trata de reducir el coste de los renders en base a evitar aquellos cálculos que generan mejoras no percibidas por un observador humano. En este caso, describimos varios resultados que permiten utilizar la respuesta del sistema visual humano a ciertas características y problemas asociadas con imágenes que contienen motion blur. Estos resultados han sido definidos tanto para imágenes monoscópicas como estereoscópicas.

Estos dos métodos y sus contribuciones correspondientes se detallan en las secciones siguientes.

## 8.1 RENDER VOLUMÉTRICO

En el primer bloque de la tesis, Parte i, hemos presentado un framework capaz de calcular imágenes a partir de medios participativos no homogéneos en tiempos interactivos. La contribución principal del trabajo es la extensión del método tradicional de ray-marching para que pueda ser aplicado en tiempos interactivos con uso simultaneo de iluminación avanzada.

Nuestra solución se basa en modelar el campo de luz incidente como una constelación de luces puntuales generadas a partir de un mapa de entorno HDR, mientras que el volumen esta representado por una matriz de vóxeles. Ambos enfoques tienen un amplio uso y existen multitud de algoritmos que son capaces de generar conjuntos de datos compatibles.

Como segunda contribución, hemos modificado la ecuación de render para tomar ventaja de la naturaleza de los medios participativos y hemos definido una serie de suposiciones para simplificar el modelo. La representación matemática resultante es capaz de dar lugar a imágenes de alta calidad mientras que puede ser evaluada mas eficientemente.

La tercera contribución es el uso de dos optimizaciones no descritas anteriormente: un pase de validez que es dependiente del punto de vista y un mapa de distancias independiente del punto de vista. Ambos introducen reducciones de tiempo de hasta 8 veces con respecto a una implementación de fuerza bruta.

En resumen, nuestra técnica ataca dos de los problemas que limitan la eficiencia de las implementaciones sobre GPU: la complejidad de los cálculos por fragmento que es reducida utilizando una expresión matemática simplificada; y los recorridos del volumen que son intensivos en accesos a memoria y que son optimizados utilizando cambios en el algoritmo.

Nuestra técnica puede ser extendida de varias formas. Actualmente ignora los cálculos relativos a multiple-scattering. Con los métodos actuales, esto solo puede conseguirse con impactos significativos en el rendimiento o bien, utilizando pre-cálculos. También, mientras se es capaz de generar imágenes visualmente aceptables, suponer que el medio tiene una función de fase isotrópica, hace que nuestro método sea menos exacto y elimina la posibilidad de reproducir ciertos tipos de volúmenes.

Hay un amplio rango de optimizaciones que se basan en estructuras de organización espacial y métodos de compresión. Estos serian potencialmente capaces de limitar el impacto en memoria y la sobrecarga impuesta por la representación basada en vóxeles. Las ganancias derivadas de utilizar estos métodos pueden compensar el coste asociado con el uso de modelos de iluminación basados en física.

## 8.2 ANTIASING EN ESPACIO DE PANTALLA

En Parte ii, hemos presentado un algoritmo diseñado para calcular antialiasing en imágenes pre-calculadas. El método esta inspirado en el antialiasing morfológico de Reshetov y utiliza el paralelismo disponible en las arquitecturas GPU. Esta técnica extiende los métodos actuales en calidad y, en algunos casos, en eficiencia. El método puede ser ejecutado con un coste moderado y es fácilmente integrable dentro de un pipeline existente. La técnica ha recibido una cantidad de atención significativa y ha sido revisada en varias publicaciones internacionales especializadas. Varias implementaciones han sido integradas dentro de conocidos motores de render y juegos comerciales. Además, ha dado lugar a varias extensiones y mejoras.

## 8.3 cálculo de motion blur

Parte iii comienza con una revisión en profundidad del estado del arte en métodos de render de motion blur. Mas de cincuenta técnicas desde los métodos descritos en los 70 hasta las publicaciones mas recientes, son descritas y comparadas en referencia a sus puntos fuertes y limitaciones.

Una de las contribuciones mas sobresalientes es una taxonomía de métodos basados en diferentes formulaciones de una expresión matemática general. Esta expresión modela el proceso de integración de la luz que sucede de forma natural en película y sensores fotográficos y que es la base de la simulación incluida en los métodos de render.

El informe termina con una comparación sistematizada de los métodos . Estos son comparados de acuerdo a sus correspondientes suposiciones y, de acuerdo a ellas, se proponen una serie de métodos para identificar que soluciones son las mas adecuadas para cada escenario.

Una segunda sección toma una postura diferente y analiza las respuestas del sistema visual humano a estímulos que contienen motion blur calculado mediante el algoritmo de render de Monte Carlo. Según hemos podido comprobar, esta es la primera vez que un estudio perceptual ha sido completado utilizando secuencias fotorealistas la cuales incluyen antialiasing temporal. Nuestros resultados son específicos de una sola escena y un conjunto de variaciones de parámetros de render, de todas formas, hemos sido capaces de extraer ciertos indicios que pueden identificar cuales son las características mas importantes para un observador humano.

El estudio apoya nuestra intuición inicial: ciertos limites psicofísicos tienen un rol significativo con respecto al impacto visual de los errores de render. El efecto general es una reducción de la importancia de problemas como blur excesivo, flickering y efectos estroboscópicos. Mas aun, el grado de tolerancia al ruido y las altas frecuencias es, para los casos considerados, notable. Estos resultados sugieren que es posible calcular imágenes utilizando simplificaciones algorítmicas agresivas sin degradación de la calidad percibida. Hemos identificado varias situaciones donde una secuencia de baja calidad es indistinguible de una referencia de alta calidad. En términos prácticos, esto quiere decir que un render mas rápido y menos detallado puede ser utilizado en vez de uno mas caro de calcular.

Tan importante como estos resultados es el hecho de que variaciones en la velocidad de obturación tienden a ser mas visibles que niveles de sampling alternativos. Una vez se ha alcanzado el nivel de calidad mínimo, observadores parecen mas inclinados a detectar cambios en escenas con diferentes rangos de exposición que en escenas calculadas con niveles de sampling alternativo.

Este es un resultado interesante, que se ajusta al diseño de los pipelines de render actuales: las velocidades de obturación están normalmente definidas por los parámetros de la cámara o bien por consideraciones estéticas, mientras que el nivel de sampling se ajusta para balancear el nivel de calidad con el coste. Esto tiene implicaciones practicas directas, ya que un equipo de producción encargado de optimizar renders puede tomar ventaja de la menor sensibilidad al número de samples por pixel y reducir el tiempo requerido para calcular cada frame al mismo tiempo que la calidad de las imágenes permanece, perceptualmente, constante.

Los estudios de cine y efectos especiales requieren constantemente nuevas mejoras. Entre otras, los métodos para calcular imágenes estereoscópicas pueden verse favorablemente afectadas por nuevas alternativas. La última parte de esta tesis reconoce esta demanda y extiende los resultados anteriormente descritos a render estéreo de escenas con motion blur.

Utilizando un conjunto de test perceptuales similares a los anteriores, esta vez con la ayuda de gafas estéreo activas, hemos encontrado que el sistema visual humano puede ser incluso mas elástico al ruido y problemas estroboscópicos. Todos los niveles de sampling considerados caen en la misma categoría perceptual. Como se ha comentado, las optimizaciones de render pueden ser diseñada de acuerdo a esto. Mas aun, las ganancias serán probablemente superiores dado que el render de secuencias estéreo es mucho mas caro de calcular que aquellas para un solo ojo.

Utilizar dos renders Monte Carlo independientes para cada par estéreo produce errores que son diferentes para cada ojo. Esto genera disparidad interocular, que en casos extremos, puede impedir la fusión estéreo. Los participantes en el estudio han confirmado que la estereopsis tiene lugar sin incrementar la fatiga. Al menos para los estímulos incluidos en el estudio, es seguro calcular imágenes independientemente y por lo tanto, los algoritmos no necesariamente deben incluir ningún tipo de coherencia en el sampling.

En resumen, los resultados sugieren que las limitaciones del sistema visual humano pueden ayudar a reducir los requerimientos asociados con el render de escenas de alta calidad. Además, el método utilizado para presentar las imágenes al espectador introduce nuevas oportunidades para modificar el balance entre diferentes parámetros de render.

## 8.4 NOTAS FINALES

Esta tesis ha introducido nuevos algoritmos que son capaces de resolver dos problemas interesantes: el render de imágenes fotorealistas de medios participativos y el post-procesado de imágenes con antialiasing.

También creemos que hemos definido los primeros pasos que permiten diseñar métodos de render de motion blur mas eficientes en base a tomar ventaja de las características del sistema visual humano. Estas conclusiones son aplicables tanto para secuencias estándar como para las estereoscópicas.

Incluso si los resultados son prometedores, hay suficientes áreas para la exploración. Hay bastantes aspectos que aun necesitan ser estudiados, que en algunos casos pueden poner al alcance nuevas oportunidades para mejorar los pipelines actuales. Específicamente, los aspectos perceptuales proveen direcciones fascinantes que necesitan ser analizadas ya que los beneficios para el render imágenes realistas pueden ser significativos.

# BIBLIOGRAPHY

[1] J. Abrams, A. Barbot, and M. Carrasco. Voluntary attention increases perceived spatial frequency. *Atten Percept Psychophys*, 72(6):1510–21, 2010. 91.

[2] Ansel Adams. *The Camera. The Ansel Adams Photography series I*. Little, Brown and Company, 1980. 41.

[3] Edward H. Adelson and James R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, Feb 1985. 79 and 80.

[4] Tomas Akenine-Möller, Jacob Munkberg, and Jon Hasselgren. Stochastic rasterization using time-continuous triangles. In *GH '07: Proc. of the 22nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 7–16. Eurographics Association, 2007. 43, 61, 71, and 72.

[5] B. L. Anderson and K. Nakayama. Toward a general theory of stereopsis: binocular matching, occluding contours, and fusion. *Psychological review*, 101 (3), 07 1994. 98.

[6] Oscar Ansón, Veronica Sundstedt, Diego Gutiérrez, and Alan Chalmers. Efficient selective rendering of participating media. In *Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, APGV '06, pages 135–142. ACM, 2006. 74 and 79.

[7] A.A. Apodaca and L. Gritz. *Advanced RenderMan: Creating CGI for Motion Pictures*. Morgan Kaufmann Publishers, 2000. 67.

[8] D. A. Atchison, G. Smith, and N. Efron. The effect of pupil size on visual acuity in uncorrected and corrected myopia. *Am J Optom Physiol Opt*, 56:315–323, May 1979. 74.

[9] Brian A. Barsky and Todd J. Kosloff. Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers*, pages 999–1010. World Scientific and Engineering Academy and Society (WSEAS), 2008. 43 and 69.

[10] Brian A. Barsky, Daniel R. Horn, Stanley A. Klein, Jeffrey A. Pang, and Meng Yu. Camera models and optical systems used in computer graphics: Part i, object-based techniques. In *Proceedings of the 2003 international conference on Computational science and its applications: PartIII*, ICCSA'03, pages 246–255. Springer-Verlag, 2003. 43 and 69.

[11] Gary Bishop, Henry Fuchs, Leonard McMillan, and Ellen J. Scher Zagier. Frameless rendering: double buffering considered harmful. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 175–176. ACM, 1994. 43, 60, and 73.

[12] Michael J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Comput. Vis. Image Underst.*, 63 (1):75–104, 1996. 64.

[13] Philippe Blasi, Bertrand Le Saëc, and Christophe Schlick. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum (Eurographics '93)*, 12(3):201–210, 1993. 11.

[14] James F. Blinn. Light reflection functions for simulation of clouds and dusty surfaces. *SIGGRAPH Comput. Graph.*, 16(3):21–29, 1982. 8.

[15] Atanas Boev, Danilo Hollosi, Atanas Gotchev, and Karen Egiazarian. Classification and simulation of stereoscopic artifacts in mobile 3dtv content. *Proceedings of SPIE*, 7237, 2009. 99.

[16] Solomon Boulos, Edward Luong, Kayvon Fatahalian, Henry Moreton, and Pat Hanrahan. Space-time hierarchical occlusion culling for micropolygon rendering with motion blur. In *Proceedings of the Conference on High Performance Graphics*, HPG '10, pages 11–18. Eurographics Association, 2010. 73.

[17] D. H. Brainard. The Psychophysics Toolbox. *Spatial vision*, 10(4):433–436, 1997. 102.

[18] Ron Brinkmann. *The art and science of digital compositing*. Morgan Kaufmann Publishers, 1999. 69.

[19] Gabriel J. Brostow and Irfan Essa. Image-based motion blur for stop motion animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 561–566. ACM, 2001. 43, 57, 64, 71, 72, and 73.

[20] J. S. Brunhaver, K. Fatahalian, and P. Hanrahan. Hardware implementation of micropolygon rasterization with motion and defocus blur. In *Proceedings of the Conference on High Performance Graphics*, HPG '10, pages 1–9. Eurographics Association, 2010. 73.

[21] Abdullah Bulbul, Zeynep Cipiloglu, and Tolga Capin. Technical section: A perceptual approach for stereoscopic rendering optimization. *Comput. Graph.*, 34(2):145–157, April 2010. 100.

[22] D. Burr. Motion smear. *Nature*, 284:164–165, Mar 1980. 43, 74, 79, and 81.

[23] D. C. Burr. Temporal Summation of Moving Images by the Human Visual System. *Royal Society of London Proceedings Series B*, 211:321–339, March 1981. 80.

[24] D. C. Burr and M. J. Morgan. Motion Deblurring in Human Vision. *Royal Society of London Proceedings Series B*, 264:431–436, March 1997. 80.

[25] D. C. Burr, J. Ross, and M. C. Morrone. Seeing objects in motion. *Proc. R. Soc. Lond., B, Biol. Sci.*, 227:249–265, Mar 1986. 79.

[26] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 263–270. ACM, 1993. 57 and 65.

[27] Martin Čadík. Human perception and computer graphics. Technical Report DC-PSR-2004-04, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, 2004. 78.

[28] Martin Čadík, Michael Wimmer, Laszlo Neumann, and Alessandro Artusi. Evaluation of HDR tone mapping methods using essential perceptual attributes. *Computers & Graphics*, 32(3):330–349, June 2008. 74.

[29] Mike Cammarano and Henrik Wann Jensen. Time dependent photon mapping. In *Proceedings of the 13th Eurographics workshop on Rendering*, EGRW '02, pages 135–144. Eurographics Association, 2002. 43, 56, 57, and 72.

[30] F. W. Campbell and D. G. Green. Optical and retinal factors affecting visual resolution. *J. Physiol. (Lond.)*, 181:576–593, Dec 1965. 74.

[31] P. Campisi, P. L. Callet, and E. Marini. Stereoscopic images quality assessment. In *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, 2007. 100.

[32] Michela Cancellaro, Veronica Palma, and Alessandro Neri. Stereo video artifacts introduced by a distributed coding approach. In *Fifth International Workshop on Video Processing and Quality Metrics for Consumer Electronics - VPQM*, 2010. 100.

[33] E. Catmull and R. Rom. A class of local interpolating splines. *Computer aided geometric design. Academic Press.*, pages 317–326, 1974. 26.

[34] Edwin Catmull. A hidden-surface algorithm with anti-aliasing. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '78, pages 6–11. ACM, 1978. 68.

[35] Edwin Catmull. An analytic visible surface algorithm for independent pixel processing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 109–115. ACM, 1984. 43, 47, 48, 71, and 72.

[36] Eva Cerezo, Frederic Pérez, Xavier Pueyo, Francisco J. Serón, and François X. Sillion. A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328, 2005. 11.

[37] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum. Plenoptic sampling. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 307–318. ACM Press/Addison-Wesley Publishing Co., 2000. 56.

[38] S. Chandrasekhar. *Radiative transfer.* Oxford, Clarendon Press, 1950. 7, 8, and 10.

[39] Shenchang Eric Chen. Quicktime vr: an image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 29–38. ACM, 1995. 63 and 100.

[40] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 279–288. ACM, 1993. 43, 63, and 71.

[41] P. H. Christensen, J. Fong, D. M. Laur, and D. Batali. Ray tracing for the movie 'Cars'. *Symposium on Interactive Ray Tracing*, 0:1–6, 2006. 73.

[42] J. Cohen and P. Debevec. The lightgen HDRShop plugin. http://www.hdrshop.com/main-pages/plugins.html., 2001. 9, 10, and 17.

[43] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, 1982. 53.

[44] Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1):51–72, 1986. 55.

[45] Robert L. Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '84, pages 137–145. ACM, 1984. 43, 53, 54, 69, and 72.

[46] Robert L. Cook, Loren Carpenter, and Edwin Catmull. The reyes image rendering architecture. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 95–102. ACM, 1987. 26, 43, 66, 68, 71, and 72.

[47] F. C. Crow. A comparison of antialiasing techniques. *IEEE Comput. Graph. Appl.*, 1(1):40–48, 1981. 54.

[48] Franklin C. Crow. The aliasing problem in computer-generated shaded images. *Commun. ACM*, 20:799–805, November 1977. 26.

[49] Nicolas Cuntz and Andreas Kolb. Fast hierarchical 3d distance transforms on the GPU. In *Eurographics 07 (Short Presentations)*, September 2007. 13.

[50] Frank Dachille and Arie Kaufman. High degree temporal antialiasing. In *Proceedings of the Computer Animation*, CA '00, pages 49–56. IEEE Computer Society, 2000. 55, 72, and 82.

[51] Scott J. Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. volume 1666, pages 2–15. SPIE, 1992. 74.

[52] Cyrille Damez, Kirill Dmitriev, and Karol Myszkowski. State of the art in global illumination for interactive applications and high-quality animations. *Computer Graphics Forum*, 22(1):55–77, march 2003. 74.

[53] H. David. *The method of paired comparisons*. Charles Griffin and Company, 1963. 91.

[54] Abhinav Dayal, Benjamin Watson, and David Luebke. Improving frameless rendering by focusing on change. In *ACM SIGGRAPH 2002 conference abstracts and applications*, SIGGRAPH '02, pages 201–201. ACM, 2002. 43, 60, 72, and 73.

[55] Abhinav Dayal, Cliff Woolley, Benjamin Watson, and David Luebke. Adaptive frameless rendering. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05. ACM, 2005. 43, 59, 60, and 73.

[56] François De Sorbier, Vincent Nozick, and Venceslas Biri. Accelerated stereoscopic rendering using GPU. In *16th International Conference in Central Europe on Computer Graphics Visualization and Computer Vision2008 WSCG08*, volume 32, pages 239–244, 2008. 101.

[57] Robert Desimone, Leonardo Chelazzi, Earl K. Miller, and John Duncan. Neuronal mechanisms of visual attention. *Early vision and beyond*, pages 219–226, 1995. 90.

[58] Piotr Didyk, Elmar Eisemann, Tobias Ritschel, Karol Myszkowski, and Hans-Peter Seidel. Perceptually motivated realtime temporal upsampling of 3D content for high refresh rate displays. *Computer Graphics Forum*, 29(2):713–722, 2010. 81 and 99.

[59] Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Adaptive image-space stereo view synthesis. In *Vision, Modeling and Visualization Workshop*, pages 299–306, 2010. 101.

[60] Mark A. Z. Dippé and Erling Henry Wold. Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.*, 19(3):69–78, 1985. 55 and 56.

[61] W. Donnelly. *Per-Pixel Displacement Mapping with Distance Functions*, volume GPU Gems 2, Programming Techniques for High-Performance Graphics and General-Purpose Computation, chapter 8, pages 123–136. Addison-Wesley Professional, 2005. 13.

[62] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. *ACM Trans. Graph.*, 24(3): 1115–1126, 2005. 56.

[63] Philip Dutré, Kavita Bala, and Philippe Bekaert. *Advanced Global Illumination*. A. K. Peters, Ltd., 2006. 55.

[64] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann, third edition, 2003. 26 and 67.

[65] Kevin Egan, Yu-Ting Tseng, Nicolas Holzschuch, Frédo Durand, and Ravi Ramamoorthi. Frequency Analysis and Sheared Reconstruction for Rendering Motion Blur. *SIGGRAPH (ACM Transactions on Graphics)*, 2009. 43, 45, 56, 72, and 73.

[66] Klaus Engel and Thomas Ertl. Interactive high-quality volume rendering with flexible consumer graphics hardware. In *Eurographics State of the Art of Report*, 2002. 9 and 14.

[67] Alireza Entezari, Randy Scoggins, Torsten Möller, and Raghu Machiraju. Shading for fourier volume rendering. In *VVS '02: Proceedings of the 2002 IEEE symposium on Volume visualization and graphics*, pages 131–138. IEEE Press, 2002. 8.

[68] Kayvon Fatahalian, Edward Luong, Solomon Boulos, Kurt Akeley, William R. Mark, and Pat Hanrahan. Data-parallel rasterization of micropolygons with defocus and motion blur. In *Proceedings of the Conference on High Performance Graphics 2009*, HPG '09, pages 59–68. ACM, 2009. 73.

[69] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 15–22. ACM, 2001. 61.

[70] Eliot A. Feibush, Marc Levoy, and Robert L. Cook. Synthetic texturing using digital filters. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '80, pages 294–301. ACM, 1980. 47 and 48.

[71] X. F. Feng. LCD motion blur analysis, perception, and reduction using synchronized backlight flashing. In *Society of PhotoOptical Instrumentation Engineers Conference Series*, volume 6057, pages 213–226, February 2006. 81.

[72] D. J. Fleet and K. Langley. Computational analysis of non-fourier motion. *Vision Research*, 34(22):3057–3079, 1994. 80.

[73] Roland W. Fleming, Ron O. Dror, and Edward H. Adelson. Real-world illumination and the perception of surface reflectance properties. *Journal of Vision*, 3: 347–368, 2003. 78 and 85.

[74] David Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, May 2003. 70.

[75] Joe Geigel and F. Kenton Musgrave. A model for simulating the photographic development process on digital images. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 135–142. ACM Press/Addison-Wesley Publishing Co., 1997. 43.

[76] Andrew S. Glassner. Spacetime ray tracing for animation. *IEEE Comput. Graph. Appl.*, 8(2):60–70, 1988. 43 and 49.

[77] Andrew S. Glassner. An open and shut case. *IEEE Comput. Graph. Appl.*, 19(3): 82–92, 1999. 43 and 69.

[78] Nolan Goodnight, Rui Wang, Cliff Woolley, and Greg Humphreys. Interactive time-dependent tone mapping using programmable graphics hardware. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 26–37. Eurographics Association, 2003. 17.

[79] Andrei Gorea and Christopher W. Tyler. New look at bloch's law for contrast. *J. Opt. Soc. Am. A*, 3(1):52–61, Jan 1986. 81.

[80] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 43–54. ACM, 1996. 101.

[81] Charles W. Grant. Integrated analytic spatial and temporal anti-aliasing for polyhedra in 4-space. *SIGGRAPH Comput. Graph.*, 19(3):79–84, 1985. 43, 47, and 71.

[82] Barry Green. Sensor artifacts and cmos rolling shutter. http://www.dvxuser.com/jason/CMOS-CCD/, Accessed December 2010, . 70.

[83] Simon Green. Stupid opengl shader tricks. http://developer.nvidia.com/page/documentation.html, Accessed December 2010, . 43, 50, and 71.

[84] R. Grzeszczuk, C. Henn, and R. Yagel. Advanced geometry techniques for ray casting volumes. In *ACM SIGGRAPH 1998 Course Notes, Course 4*, 1998. 13.

[85] Xin Guan and Klaus Mueller. Point based surface rendering with motion blur. In *Symposium on Point Based Graphics*, SPBG '04. Eurographics Association, 2004. 43, 51, 52, 61, 71, and 72.

[86] Stefan Guthe, Michael Wand, Julius Gonser, and Wolfgang Straßer. Interactive rendering of large volume data sets. In *IEEE Visualization*, 2002. 9.

[87] Diego Gutiérrez, Francisco J. Serón, Jorge Lopez-Moreno, Marıia P. Sanchez, Jorge Fandós, and Erik Reinhard. Depicting procedural caustics in single images. In *ACM SIGGRAPH Asia 2008 papers*, SIGGRAPH Asia '08, pages 120:1–120:9. ACM, 2008. 90.

[88] Diego Gutiérrez, Wojciech Jarosz, Craig Donner, and Srinivasa G. Narasimhan. Scattering. In *ACM SIGGRAPH 2009 Courses*, SIGGRAPH 2009, pages 21:1–21:397. ACM, August 2009. 8.

[89] Toshiya Hachisuka, Wojciech Jarosz, Richard Peter Weistroffer, Kevin Dale, Greg Humphreys, Matthias Zwicker, and Henrik Wann Jensen. Multidimensional adaptive sampling and reconstruction for ray tracing. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 33:1–33:10. ACM, 2008. 43, 55, 72, and 73.

[90] Paul Haeberli and Kurt Akeley. The accumulation buffer: hardware support for high-quality rendering. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '90, pages 309–318. ACM, 1990. 27, 43, 58, 59, 72, and 73.

[91] M. J. Harris and A. Lastra. Real-time cloud rendering. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3), pages 76–84. Blackwell Publishing, 2001. 8.

[92] Vlastimil Havran, Cyrille Damez, Karol Myszkowski, and Hans-Peter Seidel. An efficient spatio-temporal architecture for animation rendering. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 106–117. Eurographics Association, 2003. 43 and 57.

[93] Tim Hawkins, Per Einarsson, and Paul Debevec. Acquisition of time-varying participating media. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 812–815. ACM Press, 2005. 7 and 17.

[94] Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications*, 6:56–67, 1986. 59.

[95] Jay Hegde, Thomas Albright, and Gene Stoner. Second order motion conveys depth order information. *Journal of Vision*, 4(10):1–1, 2004. 80.

[96] Kyle Hegeman, Michael Ashikhmin, and Simon Premoze. A lighting model for general participating media. In *SI3D*, pages 117–124, 2005. 8.

[97] Simon Heinzle, Johanna Wolf, Yoshihiro Kanamori, Tim Weyrich, Tomoyuki Nishita, and Markus Gross. Motion blur for EWA surface splatting. In *Computer Graphics Forum (Proc. Eurographics '10)*, 2010. 51.

[98] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks. Vergence-accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):33, 2008. 99.

[99] Nate Hoffman. Morphological antialiasing in God of War III. communities.us.playstation.com, Accessed November 2011. 29.

[100] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. Technical report, Massachusetts Institute of Technology, 1980. 65.

[101] Qiming Hou, Hao Qin, Wenyao Li, Baining Guo, and Kun Zhou. Micropolygon ray tracing with defocus and motion blur. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 64:1–64:10. ACM, 2010. 73.

[102] A. Ishimaru. *Wave Propagation and Scattering in Random Media*. Academic Press, New York, U.S.A., 1978. 7, 8, and 10.

[103] Henrik Wann Jensen. Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96*, pages 21–30. Springer-Verlag, 1996. 56.

[104] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 311–320. ACM Press, 1998. 8.

[105] Jorge Jiménez, Veronica Sundstedt, and Diego Gutiérrez. Screen-space perceptual rendering of human skin. *ACM Transactions on Applied Perception*, 6(4): 23:1–23:15, 2009. 78.

[106] Jorge Jiménez, Diego Gutiérrez, Jason Yang, Alexander Reshetov, Pete Demoreuille, Tobias Berghoff, Cedric Perthuis, Henry Yu, Morgan McGuire, Timothy Lottes, Hugh Malan, Emil Persson, Dmitry Andreev, and Tiago Sousa. Filtering approaches for real-time anti-aliasing. In *ACM SIGGRAPH Courses*, 2011. xi.

[107] Jorge Jiménez, Belen Masía, Jose I. Echevarría, Fernando Navarro, and Diego Gutiérrez. *GPU Pro 2*, chapter Practical Morphological Anti-Aliasing. AK Peters Ltd., 2011.

[108] David G. Jones and Jitendra Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699 – 708, 1992. 2nd European Conference on Computer Vision. 98.

[109] N. Jones and J. Keyser. Real-time geometric motion blur for a deforming polygonal mesh. In *Proceedings of the Computer Graphics International 2005*, pages 14–18. IEEE Computer Society, 2005. 43, 50, 71, and 72.

[110] James T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 143–150. ACM, 1986. 44 and 55.

[111] James T. Kajiya and Brian P Von Herzen. Ray tracing volume densities. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 165–174. ACM, 1984. 8.

[112] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods. Vol. 1: basics*. Wiley-Interscience, 1986. 9.

[113] Ladislav Kavan, Steven Collins, Jiri Zara, and Carol O'Sullivan. Geometric skinning with approximate dual quaternion blending. In *ACM Trans. Graph.*, volume 27, page 105, New York, NY, USA, 2008. ACM Press. xiv.

[114] Timothy L. Kay and James T. Kajiya. Ray tracing complex scenes. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 269–278. ACM Press, 1986. 13.

[115] D. H. Kelley. Spatio-temporal frequency chracteristics of color-vision mechanisms. *J Opt Soc Am*, 64:983–990, Jul 1974. 74.

[116] M. G. Kendall and B. Babington-Smith. On the method of paired comparisons. *Biometrica*, 31(3/4):324–345, 1940. 92.

[117] Doyub Kim and Hyeong-Seok Ko. Eulerian motion blur. In *Eurographics Workshop on Natural Phenomena*, EGWNP '07, pages 39–46. ACM, 2007. 43, 60, 61, and 71.

[118] Thomas Klein, Magnus Strengert, Simon Stegmaier, and Thomas Ertl. Exploiting frame-to-frame coherence for accelerating high-quality volume raycasting on graphics hardware. In *IEEE Visualization*, page 29, 2005. 9.

[119] Kodak. CCD image sensor noise sources. Application note, Kodak, January 2005. 70.

[120] Thomas Kollig and Alexander Keller. Efficient illumination by high dynamic range images. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*, pages 45–50, 2003. 10 and 17.

[121] F. Kooi and A. Toet. Visual comfort of binocular and 3d displays. *Displays*, 25 (2-3):99–108, 2004. 100.

[122] Rusty Koonce. Deferred shading in tabula rasa. In *GPU Gems 3*, pages 429–457. Addison Wesley, 2007. 27 and 28.

[123] Jonathan Korein and Norman Badler. Temporal anti-aliasing in computer generated animation. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '83, pages 377–388. ACM, 1983. 43, 46, 47, 58, 68, 71, and 72.

[124] Oscar Kozlowski and Jan Kautz. Is accurate occlusion of glossy reflections necessary. In *Proceedings of the 4th symposium on Applied perception in graphics and visualization*, APGV '07, pages 91–98. ACM, 2007. 78.

[125] J. Krauskopf and J. D. Mollon. The independence of the temporal integration properties of individual chromatic mechanisms in the human eye. *J. Physiol. (Lond.)*, 219:611–623, Dec 1971. 74.

[126] J. Kruger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 38. IEEE Computer Society, 2003. 9 and 13.

[127] D. Kundur and D. Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3):43–64, May 1996. 43.

[128] Jaroslav Křivánek, Marcos Fajardo, Per H. Christensen, Eric Tabellion, Michael Bunnell, David Larsson, Anton Kaplanyan, Bruno Levy, and Richard Hao Zhang. Global illumination across industries. In *ACM SIGGRAPH Course Notes*, 2010. 78 and 85.

[129] Eric Lafortune. *Mathematical models and Monte Carlo algorithms for physically based rendering*. Ph.D. thesis, Katholieke Universiteit Leuven, February 1996. 53.

[130] Eric P. Lafortune and Yves D. Willems. Bi-directional path tracing. In *Proc. of third international conference on computational graphics and visualization techniques (COMPUGRAPHICS '93)*, pages 145–153, 1993. 56.

[131] Eric P. Lafortune and Yves D. Willems. Rendering participating media with bidirectional path tracing. In *Proceedings of the eurographics workshop on Rendering techniques '96*, pages 91–100. Springer-Verlag, 1996. 8.

[132] Eric LaMar, Bernd Hamann, and Kenneth I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 355–361. IEEE Computer Society Press, 1999. 9.

[133] Marc T. M. Lambooij, Wijnand A. IJsselsteijn, and Ingrid Heynderickx. Visual discomfort in stereoscopic displays: a review. *Proceedings of SPIE*, 6490(1584), 2007. 100.

[134] H. De Lange. Research into the dynamic nature of the human fovea cortex systems with intermittent and modulated light. i. attenuation characteristics with white and colored light. *J. Opt. Soc. Am.*, 48(11):777–783, Nov 1958. 79.

[135] Mark E. Lee, Richard A. Redner, and Samuel P. Uselton. Statistically optimized sampling for distributed ray tracing. *SIGGRAPH Comput. Graph.*, 19(3):61–68, 1985. 55.

[136] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 31–42. ACM, 1996. 101.

[137] Bruno Levy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. Least squares conformal maps for automatic texture atlas generation. In ACM, editor, *ACM SIGGRAPH conference proceedings*, Jul 2002. xiv.

[138] Wei Li, Klaus Mueller, and Arie Kaufman. Empty space skipping and occlusion clipping for texture-based volume rendering. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 42. IEEE Computer Society, 2003. 9.

[139] H.Y. Lin and C.H. Chang. Photo-consistent motion blur modeling for realistic image synthesis. In *Pacific-Rim Symposium on Image and Video Technology*, pages 1273–1282, 2006. 43 and 70.

[140] T. Liu, J. Abrams, and M. Carrasco. Voluntary attention enhances contrast appearance. *Psychol Sci*, 2009. 91.

[141] Tom Lokovic and Eric Veach. Deep shadow maps. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 385–392. ACM Press/Addison-Wesley Publishing Co., 2000. 61 and 62.

[142] Peter Longhurst, Kurt Debattista, and Alan Chalmers. A GPU based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, AFRIGRAPH '06, pages 21–29. ACM, 2006. 79.

[143] J. M. Loomis and K. Nakayama. A velocity analogue of brightness contrast. *Perception*, 2(4):425–428, 1973. 80.

[144] Jorge Lopez-Moreno, Veronica Sundstedt, Francisco Sangorrin, and Diego Gutiérrez. Measuring the perception of light inconsistencies. In *Symposium on Applied Perception in Graphics and Visualization (APGV)*. ACM Press, 2010. 78.

[145] Jörn Loviscach. Motion blur for textures by means of anisotropic filtering. In *Eurographics Symposium on Rendering*, pages 7–14, 2005. 43, 52, 71, and 73.

[146] C. Lu and D. H. Fender. The interaction of color and luminance in stereoscipic vision. *Investigative Opthalmology*, 11(6):482–490, 1972. 98.

[147] A. Mack and I. Rock. *Inattentional blindness*. MIT Press, 1998. 91.

[148] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings Fifth Berkeley Symposium on Math. Stat. and Prob*, volume 1, pages 281–297. L. M. LeCam and J. Neyman. University of California Press, 1967. 9 and 10.

[149] Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Visible difference predicator for high dynamic range images. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2763–2769, 2004. 79.

[150] Rafał Mantiuk, Scott Daly, Karol Myszkowski, and Hans-Peter Seidel. Predicting visible differences in high dynamic range images - model and its calibration. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly, editors, *Human Vision and Electronic Imaging X, IS&TSPIE's 17th Annual Symposium on Electronic Imaging (2005)*, volume 5666, pages 204–214, 2005. 88.

[151] Nelson L. Max. Atmospheric illumination and shadows. *SIGGRAPH Comput. Graph.*, 20(4):117–124, 1986. 8.

[152] Nelson L. Max and Douglas M. Lerner. A two-and-a-half-d motion-blur algorithm. *SIGGRAPH Comput. Graph.*, 19(3):85–93, 1985. 43, 44, 63, and 72.

[153] Rachel McDonnell, Michéal Larkin, Simon Dobbyn, Steven Collins, and Carol O'Sullivan. Clone attack! perception of crowd variety. *ACM Trans. Graph.*, 27: 26:1–26:8, August 2008. 78.

[154] Ann McNamara. Exploring visual and automatic measures of perceptual fidelity in real and simulated imagery. *ACM Trans. Appl. Percept.*, 3:217–238, July 2006. 78.

[155] Lydia M. J. Meesters, Wijnand A. IJsselsteijn, and Piter J. H. Seuntiens. A survey of perceptual evaluations and requirements of three-dimensional tv. *IEEE Trans. Circuits Syst. Video Techn.*, pages 381–391, 2004. 100.

[156] Marci Meingast, Christopher Geyer, and Shankar Sastry. Geometric models of rolling-shutter cameras. *Omnidirectional vision, Camera networks, and non-classical cameras*, 2005. 70.

[157] Ryan Meredith-Jones. Point sampling algorithms for simulating motion blur. Master's thesis, University of Toronto, 2000. 41, 44, and 81.

[158] Don P. Mitchell. Generating antialiased images at low sampling densities. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 65–72. ACM, 1987. 55 and 68.

[159] Don P. Mitchell. Spectrally optimal sampling for distribution ray tracing. *SIGGRAPH Comput. Graph.*, 25(4):157–164, 1991. 44.

[160] Don P. Mitchell. Consequences of stratified sampling in graphics. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 277–280. ACM, 1996. 55.

[161] Don P. Mitchell and Arun N. Netravali. Reconstruction filters in computer-graphics. *SIGGRAPH Comput. Graph.*, 22(4):221–228, 1988. 26 and 55.

[162] Yoshiki Mizukami and Katsumi Tadamura. Optical flow computation on compute unified device architecture. In *Proceedings of the 14th International Conference on Image Analysis and Processing*, ICIAP '07, pages 179–184. IEEE Computer Society, 2007. 73.

[163] Klaus Mueller, Torsten Möller, J. Edward Swan II, Roger Crawfis, Naeem Shareef, and Roni Yagel. Splatting errors and antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):178–191, 1998. 43, 51, and 72.

[164] Karol Myszkowski. Perception-based global illumination, rendering, and animation techniques. In *Proceedings of the 18th spring conference on Computer graphics*, SCCG '02, pages 13–24. ACM, 2002. 74.

[165] S. Narasimhan and S. Nayar. Shedding light on the weather. In *Proceedings of CVPR*, June 2003. 8.

[166] Fernando Navarro. *Game Development Tools*, chapter Taming the Beast: Managing Complexity in Game Build Pipelines. AK Peters Ltd./CRC Press, 2011.

[167] Fernando Navarro, Susana Castillo, Francisco J. Serón, and Diego Gutiérrez. Motion blur perception and stereo rendering. To be submitted to ACM Transactions on Applied Perception.

[168] Fernando Navarro, Diego Gutiérrez, and Francisco J. Serón. Interactive HDR lighting of dynamic participating media. *Vis. Comput.*, 25(4):339–347, February 2009.

[169] Fernando Navarro, Susana Castillo, Francisco J. Serón, and Diego Gutiérrez. Perceptual considerations for motion blur rendering. *ACM Trans. Appl. Percept.*, 8(3):20:1–20:15, August 2011.

[170] Fernando Navarro, Francisco J. Serón, and Diego Gutiérrez. Motion blur rendering: State of the art. *Computer Graphics Forum*, 30(1):3–26, 2011. 27, 77, and 79.

[171] Diego Nehab, Pedro V. Sander, Jason Lawrence, Natalya Tatarchuk, and John R. Isidoro. Accelerating real-time shading with reverse reprojection caching. In *Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 25–35. Eurographics Association, 2007. 43, 57, and 101.

[172] Ren Ng, Ravi Ramamoorthi, and Pat Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.*, 22(3):376–381, 2003. 9 and 10.

[173] Duc Quang Nguyen, Ronald Fedkiw, and Henrik Wann Jensen. Physically based modeling and animation of fire. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '02, pages 721–728. ACM, 2002. 61.

[174] Hubert Nguyen. *GPU Gems 3*. Addison-Wesley Professional, first edition, 2007. 28.

[175] Alan Norton, Alyn P. Rockwood, and Philip T. Skolmoski. Clamping: A method of antialiasing textured surfaces by bandwidth limiting in object space. *SIGGRAPH Comput. Graph.*, 16(3):1–8, 1982. 26, 43, 52, and 71.

[176] NVidiaCorporation. Cg language specification. http://developer.download.nvidia.com/cg/Cg_1.5/1.5.0/0019/Cg_Specification.pdf, accessed April 2012, . 12.

[177] NVidiaCorporation. Framebuffer object extension. http://oss.sgi.com/projects/ogl-sample/registry/EXT/framebuffer_object.txt, accessed April 2012, . 17.

[178] Marc Olano, Jon Cohen, Mark Mine, and Gary Bishop. Combatting rendering latency. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, I3D '95, pages 19–ff. ACM, 1995. 60.

[179] Alan V. Oppenheim and Alan S. Willsky. *Signals & systems*. Prentice-Hall, Inc., 1983. 25.

[180] Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23 (3):488–495, 2004. 10 and 17.

[181] Carol O'Sullivan, Sarah Howlett, Yann Morvan, Rachel McDonnell, and Keith O'Conor. Perceptually Adaptive Graphics. In *STAR-Proc. of Eurographics 2004*, State of the Art Reports, pages 141–164. INRIA and the Eurographics Association, 2004. 74.

[182] Ryan S. Overbeck, Craig Donner, and Ravi Ramamoorthi. Adaptive wavelet rendering. *ACM Trans. Graph.*, 28:140:1–140:12, December 2009. 43, 56, 72, and 73.

[183] Dennis Pachur, Tim Laue, and Thomas Röfer. *Real-Time Simulation of Motion-Based Camera Disturbances, RoboCup 2008: Robot Soccer World Cup XII*, pages 591–601. Springer-Verlag, 2009. 43 and 70.

[184] Hao Pan, Xiao-Fan Feng, and Scott Daly. LCD motion blur modeling and analysis. In *International Conference on Image Processing*, pages 21–24, 2005. 81.

[185] K. Pauwels and M. M. Van Hulle. Realtime phase-based optical flow on the GPU. In *CVGPU08*, pages 1–8, 2008. 73.

[186] E. S. Pearson and H. O. Hartley. *Biometrika Tables for Statisticians, 3rd ed.* Cambridge University Press, 1966. 91.

[187] Bryan Peterson. *Understanding Shutter Speed: Creative Action and Low-Light Photography Beyond 1/125 Second*. Amphoto Books; 1st edition, 2008. 41 and 77.

[188] Jonathan D. Pfautz. Sampling artifacts in perspective and stereo displays. volume 4297, pages 54–62. SPIE, 2001. 100.

[189] Matt Pharr and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., 2004. 55.

[190] Michael Potmesil and Indranil Chakravarty. A lens and aperture camera model for synthetic image generation. In *Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '81, pages 297–305. ACM, 1981. 62.

[191] Michael Potmesil and Indranil Chakravarty. Modeling motion blur in computer-generated images. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '83, pages 389–399. ACM, 1983. 43, 44, 62, and 63.

[192] Martin Preston. Temporal issues in the rendering of non-uniform rational B-Spline surfaces. M.Sc. thesis, University of Manchester, 1992. 43, 60, and 71.

[193] Zenon W. Pylyshyn and Ron W. Storm. Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spatial Vision*, 3(3):179–197, 1988. 91.

[194] J. Ragan-Kelley, J. Lehtinen, J. Chen, M. Doggett, and F Durand. Decoupled sampling for real-time graphics pipelines. Technical report, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory Technical Report Series, 2010. 43, 58, and 72.

[195] V. S. Ramachandran, V. M. Rao, and T. R. Vidyasagar. Sharpness constancy during movement perception (short note). *Perception*, 3(1):97–98, 1974. 80.

[196] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001. 9.

[197] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. Visual equivalence: towards a new standard for image fidelity. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07. ACM, 2007. 79 and 95.

[198] Ganesh Ramanarayanan, Kavita Bala, and James A. Ferwerda. Perception of complex aggregates. *ACM Trans. Graph.*, 27:60:1–60:10, August 2008. 78.

[199] Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '99, pages 73–82, 1999. 74 and 78.

[200] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans. Graph.*, 2(2):91–108, 1983. 43, 48, 49, and 71.

[201] William T. Reeves, David H. Salesin, and Robert L. Cook. Rendering antialiased shadows with depth maps. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 283–291. ACM, 1987. 61.

[202] Matthew Regan and Ronald Pose. Priority rendering with a virtual reality address recalculation pipeline. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, pages 155–162. ACM, 1994. 60.

[203] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec. *High Dynamic Range Imaging - Acquisition, Display and Image-based Lighting*. Morgan Kaufman Publishers, 2006. 9, 43, and 70.

[204] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 267–276. ACM Press, 2002. 17.

[205] R. A. Rensink, J. K. O'Regan, and J. J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368–373, 1997. 91.

[206] Alexander Reshetov. Morphological antialiasing. *Proceedings of High Performance Graphics*, pages 109–116, 2009. 25 and 28.

[207] Nico Ritsche. Real-time shell space rendering of volumetric geometry. In *GRAPHITE '06: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 265–274. ACM Press, 2006. 13.

[208] J. G. Robson. Spatial and temporal contrast-sensitivity functions of the visual system. *J. Opt. Soc. Am.*, 56(8):1141–1142, Aug 1966. 79.

[209] Gilberto Rosado. *GPU Gems 3 - Motion Blur as a Post-Processing Effect*, chapter 27, pages 575–583. Addison-Wesley, 2007. 43, 66, and 73.

[210] Michael Rubinstein, Diego Gutiérrez, Olga Sorkine, and Ariel Shamir. A comparative study of image retargeting. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA '10, pages 160:1–160:10. ACM, 2010. 90.

[211] Holly Rushmeier. The perception of simulated materials. In *ACM SIGGRAPH 2008 classes*, SIGGRAPH '08, pages 7:1–7:12. ACM, 2008. 78.

[212] Holly E. Rushmeier and Kenneth E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 293–302. ACM, 1987. 8.

[213] Pedro Sander and John Isidoro. Computation culling with explicit early-z and dynamic flow control. In *SIGGRAPH '05: ACM SIGGRAPH 2005 GPU Shading and Rendering Course*. ACM, 2005. 59.

[214] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, April 2002. 98.

[215] Ellen J. Scher Zagier. A human's eye view: motion blur and frameless rendering. *Crossroads*, 3:8–12, May 1997. 43 and 60.

[216] Johannes Schmid, Robert Sumner, Huw Bowles, and Markus Gross. Programmable motion effects. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 29(3): 57:1–57:9, 2010. 43, 50, 51, and 71.

[217] Robert Sekuler and Randolph Blake. *Perception*. Boston: McGraw-Hill, 5th edition, 2006. 74.

[218] I. Setyawan and R. L. Lagendijk. L.: Human perception of geometric distortions in images. In *SPIE, Security, Steganography and Watermarking of Multimedia Contents*, volume IV, pages 272–280, 2004. 91.

[219] Pieter Seuntiens, Lydia Meesters, and Wijnand Ijsselsteijn. Perceived quality of compressed stereoscopic images: Effects of symmetric and asymmetric jpeg coding and camera separation. *ACM Trans. Appl. Percept.*, 3(2):95–109, April 2006. 99.

[220] Roland V. Shack. The influence of image motion and shutter operation on the photographic transfer function. *Appl. Opt.*, 3(10):1171–1181, 1964. 43, 44, 62, and 69.

[221] K.L. Shapiro, J.E. Raymond, and K.M. Arnell. The attentional blink. *Trends in Cognitive Sciences*, 1(8):291 – 296, 1997. 91.

[222] R. Shapley and C Enroth-Cugell. Visual adaptation and retinal gain controls. *Progress in Retinal Research*, 3:263–346, January 1984. ISSN 0278-4327. 74.

[223] Lin Shi and Yizhou Yu. Controllable smoke animation with guiding objects. *ACM Trans. Graph.*, 24(1):140–164, 2005. 17.

[224] Clement Shimizu, Amit Shesh, and Baoquan Chen. Hardware-accelerated-motion-blur-generation. Technical report, University of Minnesota Computer Science Department, 2003. 43, 65, 66, and 71.

[225] Mikio Shinya. Spatial anti-aliasing for animation sequences with spatio-temporal filtering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 289–296. ACM, 1993. 43, 44, 64, and 71.

[226] Oles Shishkovtsov. Deferred shading in S.T.A.L.K.E.R. In *GPU Gems 2*, pages 143–166. Addison Wesley, 2005. 27 and 28.

[227] Maryann Simmons and Carlo H. Séquin. Tapestry: A dynamic mesh-based display representation for interactive rendering. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pages 329–340. Springer-Verlag, 2000. 58.

[228] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 527–536. ACM, 2002. 8.

[229] Tiago Sousa. Vegetation procedural animation and shading in crysis. In *GPU Gems 3*, pages 373–385. Addison Wesley, 2007. 28.

[230] Tiago Sousa. Crysis next-gen effects. In *Game Developers Conference 2008*, 2008. 43, 65, 71, and 73.

[231] Jos Stam. Stochastic rendering of density fields. In *Proceedings of Graphics Interface '94*, pages 51–58, 1994. 8.

[232] Jos Stam. Multiple Scattering as a Diffusion Process. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 41–50. Springer-Verlag, 1995. 8.

[233] Lew B. Stelmach, Wa James Tam, Daniel V. Meegan, and André Vincent. Stereo image quality: effects of mixed spatio-temporal resolution. *IEEE Trans. Circuits Syst. Video Techn.*, 10(2):188–193, 2000. 100.

[234] Lew B. Stelmach, Wa James Tam, Daniel V. Meegan, André Vincent, and Philip Corriveau. Human perception of mismatched stereoscopic 3d inputs. In *ICIP*, 2000. 100.

[235] Ian Stephenson. Improving motion blur: Shutter efficiency and temporal sampling. *Journal of Graphics, GPU, and Game Tools*, 12(1):9–15, 2007. 84.

[236] Bo Sun, Ravi Ramamoorthi, Srinivasa G. Narasimhan, and Shree K. Nayar. A practical analytic single scattering model for real time rendering. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1040–1049. ACM Press, 2005. 9.

[237] K. Sung, A. Pearce, and C. Wang. Spatial-temporal antialiasing. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):144–153, 2002. 41, 43, 44, 67, 71, 72, and 81.

[238] László Szirmay-Kalos, György Antal, and Balázs Benedek. Global illumination animation with random radiance representation. In *Proceedings of the 14th Eurographics workshop on Rendering*, EGRW '03, pages 64–73. Eurographics Association, 2003. 58.

[239] László Szirmay-Kalos, Mateu Sbert, and Tamás Umenhoffer. Real-time multiple scattering in participating media with illumination networks. In *Eurographics Symposium on Rendering*, pages 277–282, 2005. 8.

[240] Natalya Tatarchuk, Chris Brennan, and John R. Isidoro. *Motion Blur Using Geometry and Shading Distortion In ShaderX2: Shader Programming Tips and Tricks with DirectX 9.0*. Wordware, 2003. 43, 50, 51, and 71.

[241] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis. Synthetic shutter speed imaging. *Computer Graphics Forum*, 26(3):591–598, 2007. 43 and 71.

[242] Nicolas Thibieroz. Deferred shading with multisampling anti-aliasing in directx 10. In *ShaderX*$^7$, pages 225–242. Charles River Media, 2009. 27 and 28.

[243] Hollis N. Todd and Richard D. Zakia. *Photographic sensitometry; the study of tone reproduction*. Morgan and Morgan Hastings-on-Hudson, N.Y., 1st edition, 1969. 70.

[244] Parag Tole, Fabio Pellacini, Bruce Walter, and Donald P. Greenberg. Interactive global illumination in dynamic scenes. *ACM Trans. Graph.*, 21(3):537–546, 2002. 58.

[245] Dani Tost, Sergi Grau, Maria Ferre, and Anna Puig. Ray-casting time-varying volume data sets with frame-to-frame coherence. In *Proceedings of SPIE*, 2006. 9.

[246] Sylvain Tourancheau, Kjell Brunnström, Borje Andrén, and Patrick Le Callet. LCD motion blur estimation using different measurement methods. *Journal of the Society for Information Display*, 17(3):239–249, 2009. 99.

[247] Anne Treisman and Sharon Sato. Conjunction search revisited. In *Journal of Experimental Psychology: Human Perception and Performance*, pages 459–478, 1990. 90.

[248] Jeroen van Baar, Steven Poulakos, Wojciech Jarosz, Derek Nowrouzezahrai, Rasmus Tamstorf, and Markus Gross. Perceptually-based compensation of light pollution in display systems. In *Proceedings of the 2011 ACM Symposium on Applied Perception in Graphics and Visualization*. ACM, August 2011. 99.

[249] Peter Vangorp, Jurgen Laurijssen, and Philip Dutré. The influence of shape on the perception of material reflectance. *ACM Trans. Graph.*, 26, July 2007. 78.

[250] Peter Vangorp, Timothy Condon, James Ferwerda, Kavita Bala, Roeland Schoukens, and Philip Dutré. Visual equivalence in dynamic scenes. Technical report, Katholieke Universiteit Leuven, 2009. 79.

[251] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 43, 55, 56, 72, and 73.

[252] E. Veach and L Guibas. Bidirectional estimators for light transport. In *Eurographics Rendering Workshop*, pages 147–162. Springer-Verlag, June 1994. 56.

[253] Eric Veach. *Robust monte carlo methods for light transport simulation*. PhD thesis, 1998. Adviser-Guibas, Leonidas J. 53 and 92.

[254] Anthony Vetro, Thomas Wiegand, and Gary J Sullivan. Overview of the stereo and multiview video coding extensions of the h . 264 / mpeg-4 avc standard. *Proceedings of the IEEE*, 99(4):626–642, 2011. 99.

[255] Veijo Virsu, Jyrki Rovamo, Pentti Laurinen, and Risto Nasasnen. Temporal contrast sensitivity and cortical magnification. *Vision Research*, 22(9):1211 – 1217, 1982. 79.

[256] Alex Vlachos. Post processing in the orange box. In *Game Developer's Conference 2008*, 2008. 43, 66, and 73.

[257] J. Vollrath, D. Weiskopf, and T. Ertl. A generic software framework for the GPU volume rendering pipeline. In *VMV '05: Proceedings of Vision, Modeling, and Visualization Conference*, 2005. 8.

[258] Ingo Wald, William R. Mark, Johannes Günther, Solomon Boulos, Thiago Ize, Warren Hunt, Steven G. Parker, and Peter Shirley. State of the art in ray tracing animated scenes. In *Eurographics 2007 State of the Art Reports*, 2007. 73.

[259] Bruce Walter, George Drettakis, and Steven Parker. Interactive rendering using the render cache. In *Rendering Techniques (Proc. of the Eurographics Workshop on Rendering)*, volume 10, pages 235–246. Springer-Verlag/Wien, Jun 1999. 58.

[260] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: a scalable approach to illumination. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 1098–1107. ACM, 2005. 56.

[261] Bruce Walter, Adam Arbree, Kavita Bala, and Donald P. Greenberg. Multidimensional lightcuts. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 1081–1088. ACM, 2006. 43, 56, 58, 72, and 73.

[262] Zhou Wang and Qiang Li. Video quality assessment using a statistical model of human visual speed perception. *J. Opt. Soc. Am. A*, 24(12):B61–B69, Dec 2007. 79.

[263] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13(1):1–14, 2004. 79.

[264] John P. Wann and Mark Mon-Williams. Health issues with virtual reality displays: what we do know and what we don't. *SIGGRAPH Comput. Graph.*, 31 (2):53–57, May 1997. 99.

[265] Henrik Wann Jensen, James Arvo, Philip Dutré, Alexander Keller, Art Oven, Matt Pharr, and Peter Shirley. Monte carlo ray tracing. In *SIGGRAPH 2003 Course Notes #44*, 2003. 53.

[266] A. B. Watson, Q. J. Hu, J. F. McGowan, and J. B. Mulligan. Design and performance of a digital video quality metric. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3644, pages 168–174, May 1999. 80.

[267] Andrew B. Watson. The spatial standard observer: A human vision model for display inspection. *SID Symposium Digest of Technical Papers*, 37(1):1312–1315, 2006. 81.

[268] Andrew B. Watson and Albert J. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am. A*, 2(2):322–341, Feb 1985. 80.

[269] Andrew B. Watson, Albert J. Ahumada Jr., and Joyce E. Farrell. Window of visibility: a psychophysical theory of fidelity in time-sampled visual motion displays. *J. Opt. Soc. Am. A*, 3(3):300–307, Mar 1986. 79 and 80.

[270] Chih wei Tang. Spatiotemporal visual considerations for video coding. *IEEE Transactions on Multimedia*, 9:231–238, 2007. 80.

[271] Rudiger Westermann and Bernd Sevenich. Accelerated volume ray-casting using texture mapping. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 271–278. IEEE Computer Society, 2001. 9.

[272] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980. 53, 54, and 62.

[273] Lance Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274, 1978. 53 and 61.

[274] Lance Williams. Pyramidal parametrics. *SIGGRAPH Comput. Graph.*, 17(3):1–11, 1983. 74.

[275] Matthias M. Wloka and Robert C. Zeleznik. Interactive real-time motion blur. *The Visual Computer*, 12:273–295, 1996. 43, 49, 71, and 72.

[276] Cliff Woolley, David Luebke, Benjamin Watson, and Abhinav Dayal. Interruptible rendering. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, I3D '03, pages 143–151. ACM, 2003. 60.

[277] Xiaolin Wu. An efficient antialiasing technique. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '91, pages 143–152. ACM, 1991. 26.

[278] Gunter Wyszecki and W. Stiles. *Color Science: Concepts and methods, Quantitative data and Formulae*. John Wiley and Sons, Inc., 2nd edition edition, 1982. 74.

[279] Jun Xia, Yue Shi, Kees Teunissen, and Ingrid Heynderickx. Perceivable artifacts in compressed video and their relation to video quality. *Image Commun.*, 24: 548–556, August 2009. 80.

[280] J. I. Yellott. Spectral analysis of spatial sampling by photoreceptors: topological disorder prevents aliasing. *Vision Res.*, 22:1205–1210, 1982. 74.

[281] Peter Young. Coverage sampling antialiasing. `developer.nvidia.com/csaa-coverage-sampling-antialiasing`, Retrieved January 2012. 27.

[282] Insu Yu, Andrew Cox, Min H. Kim, Tobias Ritschel, Thorsten Grosch, Carsten Dachsbacher, and Jan Kautz. Perceptual influence of approximate visibility in indirect illumination. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, APGV '09. ACM, 2009. 78.

[283] Ellen J. S. Zagier. Defining and refining frameless rendering. Technical report, University of North Carolina at Chapel Hill, 1996. 60.

[284] Yun Zhai and Mubarak Shah. Visual attention detection in video sequences using spatiotemporal cues. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA '06, pages 815–824. ACM, 2006. 79.

[285] Y. Zheng, H. Köstler, N. Thürey, and U. Rüde. Enhanced Motion Blur Calculation with Optical Flow. *Proceedings of Vision, Modeling and Visualization 2006*, pages 253–260, Nov 2006. 43, 65, 71, and 73.

[286] Kun Zhou, Qiming Hou, Minmin Gong, John Snyder, Baining Guo, and Heung-Yeung Shum. Fogshop: Real-time design and rendering of inhomogeneous, single-scattering media. Technical report, Microsoft Research, 2007. 9.

[287] Kun Zhou, Zhong Ren, Stephen Lin, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Real-time smoke rendering using compensated ray marching. Technical report, Microsoft Research, 2007. 8.

[288] Kun Zhou, Qiming Hou, Zhong Ren, Minmin Gong, Xin Sun, and Baining Guo. Renderants: interactive reyes rendering on GPUs. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*, pages 1–11. ACM, 2009. 73.

[289] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Trans. Graph.*, 23(3):600–608, August 2004. 101.

[290] Ray Zone. *3D Filmmakers: Conversations with Creators of Stereoscopic Motion Pictures*. Scarecrow Press, 2005. 98.

[291] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pages 371–378. ACM, 2001. 51.