Audio and Visual Rendering with Perceptual Foundations

Nicolas Bonneel

UNIVERSITY OF NICE - SOPHIA ANTIPOLIS

DOCTORAL SCHOOL STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

P H D T H E S I S

to obtain the title of

PhD of Science

of the University of Nice - Sophia Antipolis

Specialty : COMPUTER SCIENCE

Defended by

Nicolas BONNEEL

Audio and Visual Rendering with Perceptual Foundations

Thesis Advisor: George DRETTAKIS

prepared at INRIA Sophia Antipolis, REVES Team

defended on September 15, 2009

Jury :

Reviewers :	Kavita BALA	-	Cornell University
	Bernard PEROCHE	-	LIRIS - R3AM
Advisor :	George DRETTAKIS	-	INRIA - REVES
President :	François SILLION	-	INRIA - ARTIS
Examinators :	Frédo DURAND	-	MIT - CSAIL
	Mathias PAULIN	-	IRIT - VORTEX
	Olivier WARUSFEL	-	IRCAM

Acknowledgments

I particularly acknowledge my supervisor George Drettakis for the hundreds (thousands?) of hours spent with me during my PhD, his great ideas and the cool supervision work he did.

I also want to acknowledge my main collaborators: Michiel van de Panne, who gave great ideas for the second part of the thesis, and who is very cool as well, and Frédo Durand, who hosted me at MIT-CSAIL for a month, initiating the successful 'hair project' (Part 2, Chapter 6). I also thank my co-authors, in particular Sylvain Paris, Sylvain Lefebvre, Clara Suied, Nicolas Tsingos, and Isabelle Viaud-Delmon, as well as our modelers. In particular, Fernanda Andrade-Cabral who did a great job modeling most scenes in a rush during deadlines. A big thank you to Monique who put up with me for almost 3 years in the same office, and managed to drive our European project CROSSMOD at the same time, as a project assistant. By the way, I also thank all the CROSSMOD team, involving ISTI CNR-Pisa, UNIBRIS-Bristol, CNRS/IRCAM-Paris, VUT-Vienna, FAU-Erlangen, for this fruitful collaboration. I acknowledge the reviewers and members of the jury, for their time spent on my thesis and for their interesting feedback.

I also thank the rest of the lab, and in particular Marcio and David, for the great moments spent at INRIA during my PhD, and our team assistant Sophie who helped me a lot during this stay. I finally thank my family and (girl-)friends (!) who supported me all this time. Particularly Lucie during my sleepless nights in the lab, and my father who encouraged me doing Computer Graphics early on.

Contents

Introduction

In recent years, computer generated complex audiovisual scenes have become more and more present in our everyday life, mainly when watching animation movies or movies with digital effects, and when playing 3D games. Indeed, since the first entirely synthetic 3D movie *Toy Story* in 1995, the level of realism and complexity of synthetic scenes in films has never ceased to increase. At the same time, very realistic 3D games have been released (e.g., *Crysis*, *NBA 2K7*, *Call of Duty 5*, *Fallout 3*) and encounter great success. Also, with the development of the Internet, complex virtual worlds are currently emerging allowing users to share the same large virtual environment (e.g., *Second Life*, *Google Earth*). These recent developments have a number of important consequences. In particular the increasing complexity of these scenes makes them:

- *Hard to design:* The authoring of very complex virtual scenes is a long, tedious and costly task. For example, the design of the movie *WALL-E* required up to 50 animators as well as the creation of 2400 sounds effects for the environment [Disney 2009]. Similarly, the recent videogame Crysis contains 1Gb of texture data and 85,000 shaders [InCrysis 2009].

- *Hard to render:* Realistic rendering of highly complex scenes is difficult. The realtime constraint of games currently only allows limited realism in complex environments; in contrast film makers spend a large amount of computation time for rendering (about 6 hours per frame at Pixar [Pixar 2009]). In addition, the technical complexity of these systems is very high. For example, the same game Crysis contains a million lines of code [InCrysis 2009].

The increasing complexity of virtual environments, and the increasing demand for highly realistic rendering introduces a number of challenging research problems. In this thesis we concentrate on the two issues we mentioned previously: content creation, and audiovisual rendering. We will address the first using real world data such as photographs, which already contain a large amount of information; the goal is to allow non expert artists (or casual users) to create rich content. We also address the issue of audiovisual rendering by exploiting the limitations of human audiovisual perception to simplify computation.

Computer graphics applications rely on the use of rendering models and real world external data to produce images and sounds which will be perceived by the end user. Real world data can be used directly or can be used to infer parameters of a model. For example, textures can be extracted from photographs and directly applied to 3D models created by artists. However, more expensive and more complex setups are commonly used (such as light stages [Debevec *et al.* 2000, Paris *et al.* 2008, Matusik *et al.* 2003] or 3D scanners) to obtain a faithful reconstruction of our real world. Using these techniques allows the realistic reproduction of the real world, but lacks the flexibility of creating novel, entirely synthetic, scenes. Laser scanners are very expensive, and light-stage setups, in addition to their cost, are very complex and specialized systems, consisting in thousands of lights in a very large dome. Although these setups are used in the film industry, or for very high-end games, they are neither appropriate nor affordable for typical low-end games or for casual users.

As a last step of the content creation process, it is interesting to note that the final re-

sult is **perceived** by an observer. This means that most of the real physically measurable information will by discarded by the human observer due to the limits of our perception [Ramanarayanan *et al.* 2007].

In the first part of this thesis (Chapters 2 to 5, published as [Moeck *et al.* 2007, Bonneel *et al.* 2008, Bonneel *et al.* 2010, Grelaud *et al.* 2009]), we thus study the possibilities for algorithmic improvements in visual rendering and in the generation of 3D sounds taking into account both audio and visual perception combined. For example, a well-known perceptual effect due to the combination of audio and visual perception (or *crossmodal* perception) is ventriloquism [Hairston *et al.* 2003]: a sound does not need to be played at the exact location of its visual representation to be associated with it, and a small shift of the sound is not perceived at all. This tolerance is commonly used by ventriloquists to make their puppet speak, and one of our initial motivations was to use such effects to improve algorithms.

In the second part (Chapter 6, published as [Bonneel *et al.* 2009a], and Chapter 7, submitted for reviews [Bonneel *et al.* 2009b]), we automatically learn the visual appearance of a photograph in order to produce images using its style. We have concentrated on two examples. In the first case, we learn hair appearance using a statistical approach to produce plausible hair renderings. Hair rendering is a difficult topic in itself due to multiple scattering of light in the hair. Our chosen inverse problem is thus even more difficult, and we solve it by finding an appropriate perceptual metric in conjunction with state of the art rendering and reflectance model. In our second example, we learn the style of a photograph to provide a tool for fast creation and rendering of a sketched 3D scene using texture synthesis. A user can then rapidly create a 3D "casual model", and have it quickly rendered with the style of the chosen photograph.

We illustrate the components of our work in Fig.1. The first part uses the knowledge in human perception to improve audio and visual algorithms. In Part 2, Chapter 6, hair appearance is obtained from photographs and is then used in a rendering algorithm, with perceptual validation. Finally, in Part 2, Chapter 7, we use a photograph to produce renderings in the style of the photo.

Audio-Visual Crossmodal Algorithms using Perception

The use of perception in computer graphics to improve algorithms has become more and more common in recent years. Indeed, with the increasing complexity of today's virtual environments, we should strive to only render what the user can perceive and discard information which will not be seen or heard. In previous work, this has been done mainly to accelerate graphics in (visual) algorithms [O'Sullivan *et al.* 2004, Luebke *et al.* 2002]. For example, in [Ramasubramanian *et al.* 1999], visual perception is used to control the sampling of a path tracer through the use of a Visual Difference Predictor resulting in significant speedup. In our work [Drettakis *et al.* 2007], not presented in this thesis, the Visual Difference Predictor is used to interactively control levels of detail of complex scenes, using spatial and contrast masking. Less work has been done in the audio community for interactive 3D sound rendering. An interesting case is the work of [Tsingos *et al.* 2004], where

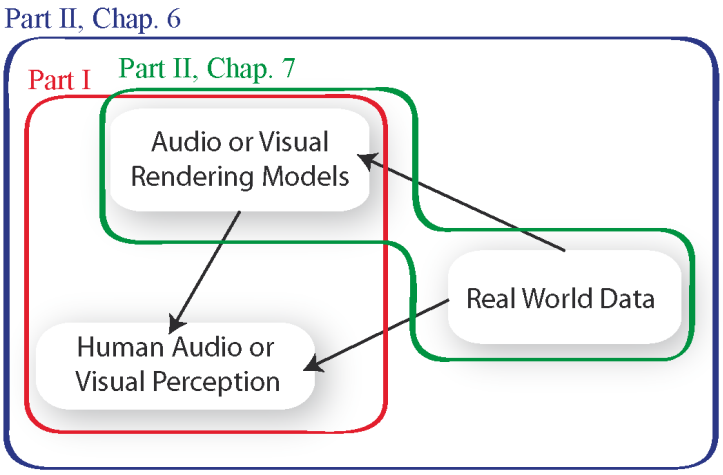


Figure 1: *The three main components defining audiovisual algorithms. A human perceives the real world and the results of rendering algorithms. However, we can also use real world data to directly improve rendering algorithms.*

audio perception is used to handle hundreds of sound sources in a virtual environment.

It is interesting to note that, in the context of virtual environments, most of the prior work relies on the use of perception in a single modality at any given time. However, a human being is multimodal by nature, relying on multiple senses at the same time to make a decision. The ventriloquism effect discussed above [Hairston *et al.* 2003] is such an example. Another interesting case is temporal window tolerance (or tolerance in asynchrony) [Guski & Troje 2003]. Previous work also shows that a dim sound could influence the perceived light intensity, contrast or threshold [Stein *et al.* 1996, Odgaard *et al.* 2003, Lippert *et al.* 2007, Bolognini *et al.* 2005, Vroomen *et al.* 2000] or that a dim light could change the auditory threshold [Lovelace *et al.* 2003]. Other work shows that perceived visual quality can be influenced by sound [Storms & Zyda 2000]. A key intuition that motivated work for the first part of this thesis, is that we could actually use the mutual influence of several modalities to improve performance of algorithms. This intuition is supported by the literature in the neuroscience community which states that humans perform faster [Kinchla 1974] when multiple senses are excited at the same time, and by preliminary work in the computer graphics/sound community [Tsingos *et al.* 2004]. In particular, we focused on the audio and the visual modalities, and we will call an algorithm using both combined, a *crossmodal* algorithm. Also, during this thesis, we will use the word 'render' to refer both to audio rendering (the generation of an audio stream) and visual rendering (the generation of images), depending on the context. In most case we deal with interactive rendering.

We will first show how human spatial tolerance between a sound and its visual representative can be used in a crossmodal clustering algorithm for 3D audio rendering ([Moeck *et al.* 2007], Chapter 2). In particular, it allows us to group nearby sound sources depending on whether they are visible or not. This first contribution follows the

work of [Tsingos *et al.* 2004] where they present an initial pilot study of the influence of visuals on audio quality. Using a perceptually based audio engine where auditory masking is used to speed up computation by removing inaudible sound sources, and where sound sources are clustered together, the goal is to determine an algorithm which uses the audiovisual spatial tolerance to cluster sound sources in a perceptually meaningful way. However, more natural scenarii with numerous sounding events occur when objects are colliding and thus generate sounds of impacts. We thus introduce the major contribution of this part: an efficient way to generate hundreds of collision sounds at the same time ([Bonneel *et al.* 2008], Chapter 3). This chapter presents a new way to use the sparsity of modal sounds in the frequency domain to efficiently render them, and uses human tolerance in asynchrony between the visual impact event and its generated sound in a scheduling algorithm. These two chapters make use of the spatio-temporal integration windows, we mentioned above.

Realistic materials are now commonly used in audio and in visual rendering, through physically based audio simulation and physical measurements of material visual properties (Bidirectional Reflectance Distribution Function, BRDF). Also, material perception has been well studied for visuals [Rushmeier 2008, Vangorp *et al.* 2007] and audio [Klatzky *et al.* 2000] separately. We thus present an experimental study on the crossmodal perception of materials when varying visual quality and audio quality simultaneously. This work shows that a given material can be well depicted with high quality visuals and lower quality audio ([Bonneel *et al.* 2010], Chapter 4), or lower quality visuals and high quality audio. The key intuition is that the cost of visual rendering is much higher than that of audio rendering. Reducing visual quality while increasing audio quality is thus preferable to reduce the overall visual rendering cost.

Finally, we conclude this part by merging our crossmodal contributions into a complete framework used in a internally developed game, and presenting practical usage of the results of our crossmodal material perception study in a crossmodal level-of-detail selection algorithm ([Grelaud *et al.* 2009], Chapter 5). This last contribution demonstrates the practical interest of using the crossmodal algorithms developed in the first part of this thesis.

Visual Rendering using a single Photograph

Designing virtual environments of natural scenes traditionally involves talented, highly trained artists and realistic rendering models. However, artists are not always available at design time, and can be very expensive. For example, a typical game costs millions of euros to produce. Use of artists is not appropriate when the end user wants to cheaply create his own art (such as in game avatar customization, e.g., Second Life, or for casual art); in addition, such users are usually not particularly skilled. This is also the case when the content creator is an engineer or technician and not an artist, which is the case in many applications such as urban planning, architectural design etc. Similarly, some applications cannot afford the use of realistic (and complex) rendering algorithms. This can be the case of lightweight devices such as PDAs or mobile phones, which do not have the

computational power of today's high-end computers, and which are also used for game applications. Lightweight rendering is also important for prototyping applications, where a fast preview of the scene is needed. The key intuition in the second part of this thesis is that a huge amount of information is already present in the natural world, and in particular in photographs. The use of digital cameras is becoming more and more common, which facilitates the retrieval of a huge amount of information. Although some work exists to create virtual environments from photographs in computer graphics [Snavely *et al.* 2006] and computer vision [Hartley & Zisserman 2004], they mainly focus on creating digital representations of the real world. In many cases, a user may want to be inspired by a photograph while creating his own environment.

The second part of this thesis thus treats the problem of using photographs to give a (visual) rendering a given appearance. We cast this as an inverse problem solved with machine learning in our first contribution, and we use a texture synthesis method in our second result.

Our first example of improving rendering using photographs is for the rendering of hair. In the context of the avatar customization scenario mentioned above, recent work shows that hair appearance is the main feature modified by users [Ducheneaut *et al.* 2009]. Our first contribution thus consists in the retrieval of hair appearance (the reflectance and small scale noise of the hair) from a single flash photograph. For this, we use a database of features extracted from pre-rendered images with carefully sampled appearance parameters, and find the best match between the photograph and the database ([Bonneel *et al.* 2009a], Chapter 6). This solution is appropriate where very expensive setups are not available and where the high dimensionality of the problem and the high hair rendering cost makes it impractical for a user to use manual searching to obtain a desired appearance.

The second example is in the context of "casual modeling", i.e., allowing naive users to create 3D content and CG renderings rapidly. We describe how a high quality rendering of a roughly modeled 3D scene can be achieved through the use of an example photograph and a guided texture synthesis approach. Specifically, using our solution, a user can draw a 3D scene in about 30 seconds using rough proxy geometries and obtain a realistic natural rendering based on a photograph of the desired style ([Bonneel *et al.* 2009b], submitted for reviews, Chapter 7). We first infer the missing details of the sketched scene from a detailed segmentation and then infer the colors from the photograph. This can be used for rapid prototyping of 3D scenes by non artists, or for lightweight games when the computational power needed to solve an expensive rendering model is not available. We believe that such an approach is a promising direction for fast content creation in the near future.

Structure of the thesis

This thesis adopts the following structure. In the first part, a shared previous work chapter on sound rendering and perception is presented (Chapter 1), and our four main crossmodal contributions follow (Chapters 2 to 5). In the second part, we present two results related to the use of a single photograph to infer the style of a rendering (Chapters 6 and 7). Related previous work is presented separately in these chapters.

Part I

Perceptually based Audio-visual rendering

Preface

In this part, we present our contributions on crossmodal experiments using virtual reality and their practical use in algorithms. By observing that a human is multimodal by nature, and examining the previous work performed in neuroscience and for unimodal perceptual audio and visual rendering algorithms, we develop crossmodal audio-visual algorithms.

This part is organized as follows. We first describe the common previous work related to this part in Chapter 1, which mainly relates to audio rendering (for recorded and modal sounds), and perception for both sounds and graphics. We then present our four main contributions: an audio-visual clustering algorithm for sound spatialization (Chapter 2), the fast frequency domain generation of impact sounds using crossmodal simultaneity perception (Chapter 3), a perceptual experiment on the evaluation of the quality of audiovisual materials (Chapter 4), and a combined crossmodal pipeline demonstrating the practical interest of our crossmodal algorithms (Chapter 5).

CHAPTER 1

Previous work

An extensive review of the literature on audio and visual rendering as well as perception is far beyond the scope of a single thesis. In this section, we have chosen a small selection of work very closely related to our projects. In particular, we will describe research on audio rendering and spatialization of large-scale environments, including both recorded sounds and sounds generated on the fly such as impact sounds. We finally describe crossmodal perceptual results mainly found in the neuroscience literature since we will make use of these to further improve our algorithms.

1.1 Audio rendering

1.1.1 Audio rendering of recorded sounds

Rendering spatialized sound for 3D virtual environments has been a subject of research for many years. These include techniques permitting real-time rendering of sound reflections [Funkhouser *et al.* 2004, Funkhouser *et al.* 1999, Lokki *et al.* 2002], mainly for pre-recorded sounds.

We mainly describe the work in [Tsingos *et al.* 2004] which we use as a basis for many of our results. In order to render and spatialize multiple sound sources simultaneously, they first cull inaudible sound sources. To do this, they first precompute the energy in each frame of all sound sources in the scene individually. Then, at runtime, for each frame of the simulation, sounds are ordered by decreasing energy and their energy is greedily accumulated. The accumulation stops when the auditory masking threshold is reached, and the remaining sound sources are not played at all since they are inaudible. In practice, the auditory masking threshold is updated each time a sound source is accumulated, since the human hearing threshold depends on the overall audio level of the environment.

The remaining sound sources are then clustered together (Fig.1.1): depending on their angular position relative to the listener, and their distance to the listener, as well as their loudness, a clustering step is performed using the Hochbaum-Shmoys algorithm [Hochbaum & Schmoys 1985] or using a recursive cluster splitting [Moeck *et al.* 2007]. In each cluster, the sounds are *pre-mixed*, and only clusters need to be spatialized.

The spatialization for headphones, consists in applying two personalized filters, one for each ear, which depends on the angular location of the object. These filters are called Head Related Transfer Functions (HRTF) and can be measured [IRCAM 2009] by placing small microphones in each ear of a listener and recording impulse responses of chirp signal (a sound sweeping all frequencies). HRTFs can also be simulated using the Kirchoff approx-

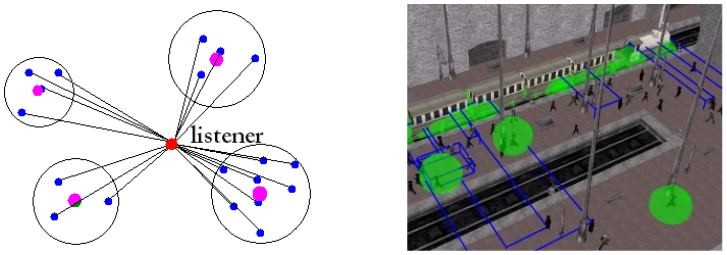


Figure 1.1: *Left: Resulting clusters from [Tsingos et al. 2004, Moeck et al. 2007]. Sound sources are in blue, the listener in red, and the cluster representative in magenta. Right: An application of the clustering in a real demo [Tsingos et al. 2004]. The sounds are dynamically clustered (blue boxes) depending on the listener position. Each cluster is pre-mixed, and are then spatialized at its center (green sphere).*

imation [Tsingos *et al.* 2007] or other boundary element methods [Katz 2001] to compute the sound scattering on a 3D head model [Dellepiane *et al.* 2008]. Other methods do exist, and a good overview of these methods can be found in [Larcher 2001]. Spatialization can also be done for many loudspeakers [Larcher 2001].

In [Tsingos *et al.* 2004], the audio processing is done in the Fourier domain by precomputing the short time Fourier Transform of each sound in a precomputation step. This allows for efficient HRTF spatialization by performing the time domain convolution as a product in the frequency domain.

In [Moeck *et al.* 2007], we further included a perceptual pre-mixing in clusters based on [Tsingos 2005], using the sparseness of the audio signal in the Fourier domain to provide scalable or progressive rendering of complex mixtures of sounds. As a result, audio spatialization of several thousands of sound sources can be handled via clustering.

One drawback related to precomputed metadata, such as per-frame sound energy, is that sounds synthesized in real time, such as modal sounds, cannot be directly supported.

1.1.2 Audio rendering of impact sounds

While computer games typically employ recorded sounds, physically based synthesis of impact sounds [van den Doel & Pai 1998, O'Brien *et al.* 2002] often provides much better results. Various techniques have been developed to optimize this approach, notably recursive evaluations [van den Doel & Pai 2003] and mode-culling [Raghuvanshi & Lin 2006] which is very effective in reducing the computational overhead.

In what follows we will use the term *impact sound* to designate a sound generated as a consequence of an event reported by the physics engine (impact, contact etc.); we assume that this sound will be synthesized on-the-fly.

Modal representation of sounds

A common way to efficiently generate contact sounds physically is "modal synthesis", to produce a "modal sound". Such a sound is produced by decomposing the initial object

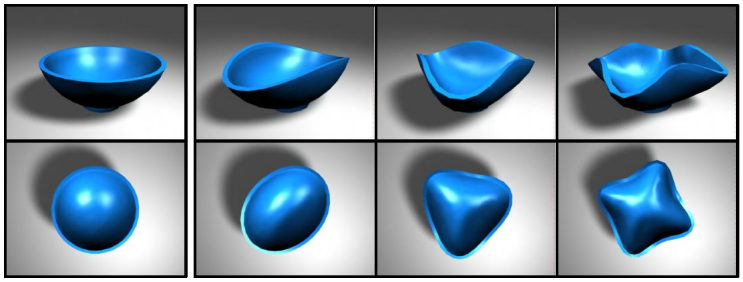


Figure 1.2: *A top and side view of a bowl, with 3 of its vibrational modes [O'Brien et al. 2002].*

into several vibrational modes (Fig.1.2) in a precomputation step and exciting these modes at runtime when the object collides with a surface.

Using the approach proposed by [O'Brien *et al.* 2002], one can compute these modes as follows. The goal is to decouple the linear elasticity equation:

$$\nabla \cdot \sigma + F = \rho \ddot{u}$$

with u the shape of the deformed object, with body forces F and density ρ. σ is the stress tensor, and we will assume the relationship given by *Hooke's law*: $\sigma = c : \varepsilon$, where ε is the strain tensor, c is a stiffness tensor which only depends on the material, and : designates the tensor product. ε and σ are represented by 3x3 matrices varying at each point in 3D space, and c is a constant 3x3x3x3 tensor. This relationship holds for small deformations which is the case when objects only vibrate. For isotropic materials, this relationship can be expressed in a simpler way:

$$\sigma(u) = \lambda \operatorname{tr} \varepsilon(u) I + 2\mu\varepsilon(u) \tag{1.1}$$

Where I is the identity matrix, tr the trace operator, and λ and μ are Lamé coefficients. Tabulated values for λ and μ are given in [O'Brien *et al.* 2002].

Using a finite element discretization, and adding a damping term, we can obtain a linear system of the following form [O'Brien *et al.* 2002]:

$$Ku + C\dot{u} + M\ddot{u} = f$$

where K, C and M are the stiffness, damping and mass matrices. This can be obtained by assembling small 12x12 matrices at each tetrahedron of a tetrahedralization of the mesh (we can make use of the freely available TetGen [Si 2003] to generate a relatively good quality tetrahedral meshing of the object with Delaunay triangulation). Specifically, using linear basis elements, and noting $p_{[1]}$,$p_{[2]}$,$p_{[3]}$ and $p_{[4]}$ the object space 3D coordinates (at rest position) of the 4 vertices (called nodes) of each tetrahedron, we obtain the basis elements β [O'Brien & Hodgins 1999] by :

$$\beta = \begin{bmatrix} p_{[1]} & p_{[2]} & p_{[3]} & p_{[4]} \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1}$$

The elementary matrices (representing the mutual influence of node $[i]$ and node $[j]$) and vectors (evaluated at each node $[i]$) are computed for each tetrahedron of volume vol. Noting a and b one of the x,y,z component of each computed value, and $\delta_{a,b}$ the Kronecker delta, they are formulated by:

$$f_{[i]a} = -\frac{vol}{2} \sum_{j=1}^{4} p_{[j]a} \sum_{k=1}^{3} \sum_{l=1}^{3} \beta_{j,l} \beta_{i,k} \sigma_{k,l} \tag{1.2}$$

$$k_{[ij]ab} = -\frac{vol}{2} \left(\lambda \beta_{i,a} \beta_{j,b} + \mu \beta_{i,b} \beta_{j,a} + \mu \sum_{k=1}^{3} \beta_{i,k} \beta_{j,k} \delta_{a,b} \right)$$

$$m_{[ij]ab} = \frac{\rho vol}{20} (1 + \delta_{i,j}) \delta a, b$$

To produce a damping of the oscillations, a stiffness damping term can be used, which replaces Equation 1.1 by:

$$\sigma(u) = \lambda \operatorname{tr} \varepsilon(u + \alpha_1 \dot{u}) I + 2\mu \varepsilon(u + \alpha_1 \dot{u})$$

where α_1 represents a stiffness damping parameter. An inertial damping coefficient α_2 is also added, thus leading to the Rayleigh damping formulation:

$$C = \alpha_1 K + \alpha_2 M$$

where tabulated values of α_1 and α_2 are found in [O'Brien *et al.* 2002]. This leads to :

$$K(u + \alpha_1 \dot{u}) + M(\alpha_2 \dot{u} + \ddot{u}) = f$$

To decouple the above linear system, we then compute a Cholesky factorization $M = LL^t$. We perform an eigen value decomposition of the matrix $L^{-1} K L^{-t} = V \Omega V$, where V is the matrix of the eigen vectors and Ω the diagonal matrix of the eigen values. The mode frequency ω_i and decay α_i are respectively the imaginary (in rad/s) and the real part of these eigen values. Note that the 6 lowest eigen modes represents the rigid transformations and should not be used since they do not make the object vibrate, and that usually a small subset of these modes is necessary to produce a sound (and in particular, no modes below 20Hz and above 20kHz will be heard). These values only depend on the geometry and material of the objects, and not on the current position in space. These values can thus be precomputed.

Also, at each impact reported by a rigid physic simulation engine (e.g., PhysX [1]), all modes are given an amplitude. By noting $g = V^t L^{-1} f$, and assuming no coupling between air and the surface of the object, the mode amplitudes are given by $a_i = \frac{2\Delta t g_i}{w_i}$.

However, a far field approximation of the surface-air coupling can be modeled by multiplying each mode amplitude (or in a more efficient way, the columns of the precomputed matrix $V^t L^{-1}$) by the sum over all surface triangular elements of centroid c:

$$I = \frac{\rho \omega^2}{4\pi r} \sum_c (\vec{n_c} \cdot \vec{u_c}) Area_c$$

[1] http://www.ageia.com

where $\vec{n_c}$ is the normal of the triangle, $Area_c$ its area, and $\vec{u_c}$ the displacement of the mode at the centroid, which is given by the rows of matrix $V^t\,L^{-1}$. ρ is the air density and r the distance to the object (which can be factored out). This approximation is the Cremer far field approximation which can be found in [James *et al.* 2006]. However, a more accurate radiation factor is also provided in [James *et al.* 2006].

Other methods for computing the modes do exist. For example, in [Raghuvanshi & Lin 2006] a spring-mass system at the surface of the object is presented, and analytic solutions for simple cases are shown in [van den Doel *et al.* 2004]. Modes can also be extracted from recordings and measurements: several sounds are recorded by striking different locations on the object. Modes and gains are then fitted for each impact location [Pai *et al.* 2001].

Note that only the amplitude of each mode is computed at runtime, depending on the position of the impact on the object, and all frequencies and decays are precomputed offline, depending only on the object's material and geometry. This makes modal sounds very attractive for efficiently generating impact sounds. They also require a small amount of memory storage.

Modal sound synthesis

The acoustic response of an object to an impulse is then given by:

$$s(t) = \sum_k a_k e^{-\alpha_k t} \sin(\omega_k t), \tag{1.3}$$

where $s(t)$ is the time-domain representation of the signal (see Fig.1.3), ω_k is the angular frequency and α_k is the decay rate of mode k; a_k is the amplitude of the mode, which is calculated on the fly (see above). However, the force applied to an object during an impact is rarely stricly impulsive, and smoother force profiles are usually preferred, such as Gaussian profiles. This is handled by convolving Equation 1.3 by this profile, often resulting in a low-pass filter.

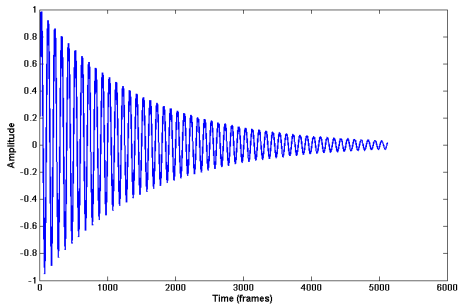


Figure 1.3: *A single mode at 440Hz rapidly decaying (30 s^{-1})*

.

Equation 1.3, with the force profile convolution, can be efficiently implemented using a recursive formulation [van den Doel & Pai 2003] which makes modal synthesis attractive to represent contact sounds, both in terms of speed and memory. Given an amplitude, a

mode frequency and decay, and a force profile, the computation of the resulting sound only requires 5 floating point operations per sample per mode.

In particular, assuming a vector s containing the mode values sampled at the sampling rate S_R, they use the following recursive formulation:

$$s(t) = 2R\cos(\theta)\, s(t-1) - R^2\, s(t-2) + a_k R \sin(\theta) F(m-1)$$

with:

$$R = e^{-\alpha_k / S_R}$$

$$\theta = \omega_k / S_R$$

This handles a force profile F to allow for smooth events (a soft impact, using a Gaussian F for example), or rolling sounds (a noisy F). By precomputing R^2, $2R\cos(\theta)$, $a_k R \sin(\theta)$ for each mode, this makes the generation of modal sounds very efficient.

We recall that the only quantities which must be computed at run-time are the gains a_k since they depend on the contact position on the objects, the applied force, and the listening position.

Modal sounds for complex soundscapes

There has also been some work on modal sound synthesis for complex scenes. In [van den Doel *et al.* 2004] a method is presented handling hundreds of impact sounds. Although their frequency masking approach was validated by a user study [van den Doel *et al.* 2002], the mode culling algorithm considers each mode independently, removing those below audible threshold.

[Raghuvanshi & Lin 2006] proposed a method based on mode pruning which they call mode compression and sound sorting by mode amplitude. However, they base their modes compression on a perceptual experiment which studies the frequency discrimination of consecutively played frequencies, although they use it as a way to remove nearby modes which are played at the same time. This is not exactly the same scenario, since in the last case, beating due to nearby frequencies is removed. They also use a scalable modes mixing step. However, using a recursive time domain formulation of modes, recursion coefficients can only be obtained by computing the entire sound at previous frames thus possibly defeating the purpose of scalability. No perceptual validation of the approximation was finally presented.

For both, the granularity of progressive modal synthesis is the mode; in the examples they show, a few thousand modes are synthesized in real time.

1.2 Audio-visual perception

We first present a short overview on unimodal audio *or* visual perception used for computed graphics algorithms. This overview helps to introduce our crossmodal work and is thus very brief. More details are given in each chapter, when appropriate.

We then review the literature on audio-visual perception related to our projects, mainly published in the neuroscience community. However, neurosciences describe reproducible experiments in highly restricted setups in order to study brain or neural mechanisms. It is thus unclear how these results generalize to more complex (or "ecological") scenes such as the ones encountered in virtual environments. In our work, although we will be inspired by neuroscience results, we will re-perform experiments in virtual environments to validate these intuitions. Please see [Spence & Driver 2004] for an extensive review on crossmodal results in the neuroscience literature.

We finally review material perception literature related to our own work.

1.2.1 Unimodal preliminaries

In recent years there have been many efforts to exploit perception to reduce computation for interactive virtual environments, ultimately with the goal to "render only what you can perceive". A survey of the early work in this domain can be found in [Luebke *et al.* 2002] and [O'Sullivan *et al.* 2004]. Examples of such work in graphics include use of frequency based raytracing ([Bolin & Meyer 1995], Fig.1.4), visual differences predictors for raytracing acceleration (e.g., [Ramasubramanian *et al.* 1999, Myszkowski 1998]), or perceptually based level-of-detail (LOD) control [Luebke & Hallen 2001, Williams *et al.* 2003, Drettakis *et al.* 2007]. These algorithms using visual perception outperform bruteforce methods which compute information which will not be perceived at all. However, the cost of predicting the eyes' response to visual stimuli using VDP is generally high, which make these algorithms interesting for very complex scenes [Drettakis *et al.* 2007] or to accelerate very slow rendering algorithms [Bolin & Meyer 1995, Ramasubramanian *et al.* 1999]. They also allow graceful degradations using LODs or progressive rendering in a perceptually meaningful way.

Although much effort has been made to use perception for graphics applications, less work has been done for interactive audio rendering. Tsingos et al. use perception to optimize masking and clustering ([Tsingos *et al.* 2004]), as discussed above.

Conversely, in our work we will address the perceptual audio-visual rendering using both modalities at the same time rather than performing a separate treatment for visuals and for sounds.

1.2.2 Spatio-temporal integration windows

Temporal integration window

Neuroscience extensively studied the perception of simultaneity between a visual event and a corresponding audio event, typically using beeping loudspeakers and flashing LEDs. These studies led to different delays for human's tolerance in asynchrony. Also, during the synthesis of impact sounds in virtual environments, modal sounds are usually computed at the exact moment of the impact. Our goal is to use this tolerance in asynchrony in order to delay the introduction of new impact sounds when the computer is already overloaded.

Different physical and neural delays in the transmission of signals can result in "contamination" of temporal congruency. This results in a tolerance in the asynchrony be-

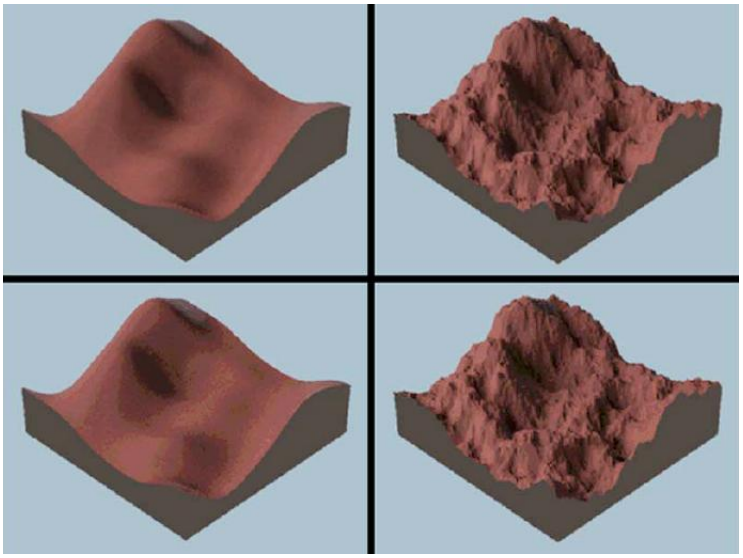


Figure 1.4: *Using visual perception to accelerate a raytracer. Depending on the image spatial frequency content (left/right) our visual system is more or less tolerant to quantization artifacts. Bottom row is quantized to 4 bits, and fewer artifacts are visible at high frequencies (right). [Bolin & Meyer 1995]*

.

tween the signals coming from different senses, in particular between auditive and visual signals. For example [Fujisaki *et al.* 2004] have shown that brain recalibrates in the presence of a fixed audio-visual time lag presented for several minutes, thus shifting the subjective simultaneity toward this time lag. Therefore, the brain needs to compensate for temporal lags to recalibrate audiovisual simultaneity. For this reason, it is difficult to establish a time window during which perception of synchrony is guaranteed, since it depends both on the nature of the event (moving or not) and its position in space (distance and direction) [Alais & Carlile 2005]. Some studies report that delaying a sound may actually improve perception of synchrony with respect to visuals [Begault 1999]. One study [Guski & Troje 2003] (among others [Sekuler *et al.* 1997, Sugita & Suzuki 2003]), reports that a temporal window of 200 ms represents the tolerance of our perception for a sound event to be considered the consequence of the visual event. We will therefore adopt this value as a threshold for our temporal scheduling algorithm.

Spatial integration window.

While the primary application of 3D audio rendering techniques is simulation and gaming, no spatial audio rendering work to date evaluates the influence of combined visual and audio restitution on the required quality of the simulation. However, a vast amount of literature in neurosciences suggest that cross-modal effects, such as ventriloquism, might significantly affect 3D audio perception [Hairston *et al.* 2003, Alais & Burr 2004]. This effect tells us that in presence of visual cues, the location of a sound source is perceived as shifted toward the visual cue, up to a certain threshold of spatial congruency. Above this threshold, there is a conflict between the perceived sound location and its visual rep-

resentation and the ventriloquism effect no longer occurs. The spatial window (or angular threshold) of this effect seems to depend on several factors (e.g., temporal synchronicity between the two channels and perceptual unity of the bimodal event) and can vary from a few degrees [Lewald *et al.* 2001] up to 15° [Hairston *et al.* 2003].

Also, in [Fouad *et al.* 1997], the visual gaze of the listener is used in the prioritization of sound rendering and [Tsingos *et al.* 2004] presents an initial pilot study of the influence of visuals on the perceived sound quality. Although [Tsingos *et al.* 2004] shows that perceived quality is degraded when visuals are added, no further investigation was proposed.

1.2.3 Material perception

Rendering plausible materials interactively in virtual environments (VE) is a challenging task ([Brainard *et al.* 2008]). Improving material perception in this context requires the study of the influence of both visual quality and audio quality on the perception of materials.

Material perception has received significant attention in recent years in computer graphics. Notably, [Vangorp *et al.* 2007] studies the effect of geometry shape and lighting on perception of material reflectance (Figure 1.5). They design an experiment showing that the shape of the objects affects the impression of the material, and that the sphere often used for picking materials in 3D softwares is not the best choice. [Fleming *et al.* 2003] studied the influence of the illumination on the perception of materials, and reported that materials were best depicted with real world illumination (Figure 1.6). This draws the attention to be ported on the illumination and geometry to be used when designing experiments on material recognition (Chapter 4).

In [Ramanarayanan *et al.* 2007], the concept of visual equivalence is introduced, based on material properties, geometry and illumination. They provide a key definition for visual equivalence: two images are visually equivalent if the object shape and material are judged to be the same in both images, and if, in a side-by-side comparison, a person is unable to tell which image is the reference. This definition differs with respect to previous low-level image quality metrics focusing on pixel by pixel differences, allowing for a higher level comparison.

A more complete exposition on material perception from renderings can be found in [Rushmeier 2008]. However, these works only focused on visual cues and do not address other modalities.

Similarly, auditory cues for material perception have also been used experimentally. In particular, material classification has been studied by [Giordano & McAdams 2006] where subjects had to determine the material an object was made from (wood, plexiglass, steel or glass), by striking real physical objects. They show that two main categories of material were correctly classified (wood and plexiglass vs. steel and glass). We will also use these categories in order to use well distinguished material classes when designing an experimental protocol on materials (Chapter 4).

Perception of material depending on contact sounds is also studied in [Klatzky *et al.* 2000]. In one experiment, subjects were asked to rate similarity of the material of two sounding objects using audio only. A second study asked participants

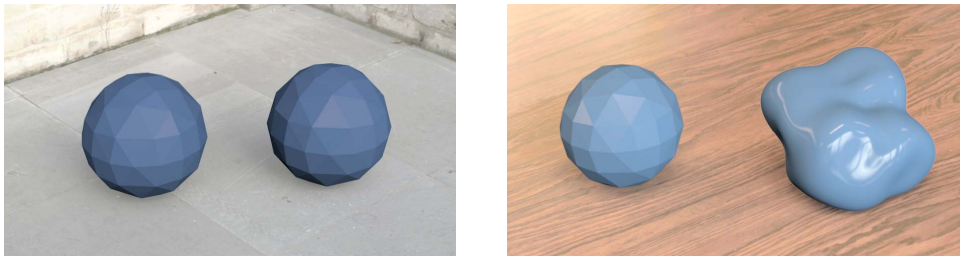


Figure 1.5: *The left image shows tessellated spheres with two different materials, yet they are perceived as made from the same material. The right image show objects with the same material, yet their appearance is very different [Vangorp et al. 2007]*

.

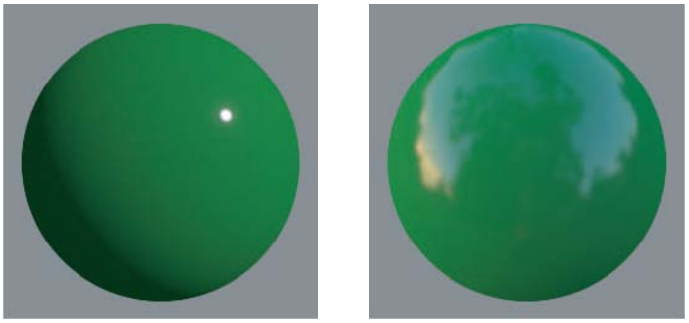


Figure 1.6: *Two spheres rendered with the same material. The left image uses a single point light source whereas the right image uses a captured environment map. Most observers report that material quality is better depicted in the right image [Fleming et al. 2003]*

.

to classify the sounds into groups of materials. Whereas they do not include visual cues in the material perception, we have been inspired by some aspects of their experimental methodology.

While we are unaware of work on audio-visual material perception, there has been work on combining haptics and audio for material perception (e.g., [Guest *et al.* 2002]). However, we consider the haptic and visual modalities to be very different, and will not review this literature here.

Nonetheless, earlier work [Storms & Zyda 2000] has found some improvement in overall perception of visual image quality in the presence of better sound. This experimental study of static images and sounds showed that the perceived quality of a high quality visual display evaluated alone was enhanced when coupled with high quality sound. The study further showed that the perceived quality of a low quality auditory display evaluated alone was reduced when coupled with a high quality visual display. Visual degradations were varied by resampling images or adding noise, while audio degradation was varied by changing sampling rates or by adding Gaussian noise. In other previous work, Mastoropoulou et al. have studied the effect of sound on rendering animations (e.g., [Mastoropoulou *et al.* 2005]); while this work does study the joint effect of sound

and graphics on quality perception, it does not treat the case of materials which is the focus of our study (Chapter 4).

CHAPTER 2

Progressive Perceptual Audio Rendering of Complex Scenes

Contents

Figure 2.1: *A subject performing our crossmodal perceptual experiment on the workbench. The participant is asked to judge the audio quality of VR scenes with different cluster distributions. We show that in the audio-visual condition, more clusters are needed in the viewing frustum.*

The contributions in this chapter were published in the *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* [Moeck *et al.* 2007].

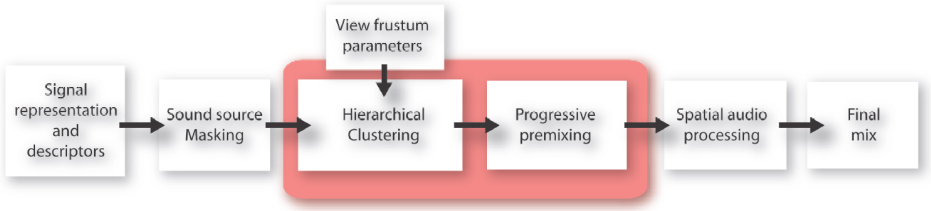


Figure 2.2: *Overview of our overall sound rendering pipeline. In particular, we introduce a sound source clustering metric that better handles visible sources.*

2.1 Introduction

Spatialized audio rendering is a very important factor for the realism of interactive virtual environments, such as those used in computer games, virtual reality, or driving/flight simulators, etc. The complexity and realism of the scenes used in these applications has increased dramatically over the last few years.

Recent research has proposed solutions to the computational limitations due to the handling of numerous sound sources, both in audio rendering and spatialization. Perceptual masking with sound source clustering [Tsingos *et al.* 2004], or other clustering methods [Herder 1999, Wand & Straßer 2004] do resolve some of the issues. However, the clustering algorithms proposed to date for sound spatialization are either restricted to static scenes, or add an unacceptable computation overhead due to a quadratic step in cluster construction when the number of sources is large. In addition, the cost of per source computation, sometimes called premixing, can quickly become a bottleneck, again for complex soundscapes.

Also, virtual environments rarely consist only in sounding objects, but also display their visual 3D representation. Although much effort has been made in either visual or audio perception in virtual environments, very little work has been done considering both the audio *and* the visuals at the same time. In [Tsingos *et al.* 2004], a preliminary study is reported but is inconclusive. However, being able to use both audio and visual information should allow for better quality soundscapes. In particular, audio-visual spatial tolerance has been extensively studied in neuroscience, exhibiting a spatial "integration window". This should improve the quality of audio clustering algorithms, taking into account visual information.

The contributions presented in this chapter have been published as part of [Moeck *et al.* 2007], which also resolved the high premixing cost issue and proposed a recursive clustering algorithm. In [Moeck *et al.* 2007], a perceptual validation of the premixing strategy is also provided. In this chapter, in the context of crossmodal algorithms, we will only present the investigation of crossmodal perceptual issues related to *clustering*, based on *pilot user studies* we conducted. In particular, we investigate the influence of visuals on audio clustering for audio-visual scenes, and propose a modified clustering metric taking into account the indication that it is probably better to have more sources in the view frustum.

2.2 Cross-modal effects for sound scene simplification

The previous use of perception for audio rendering does not consider visual information of corresponding sound sources. Intuitively, it would seem that such interaction of visual and audio rendering should be taken into account, and play a role in the choice of metrics used in the audio clustering algorithm. A first attempt was presented in [Tsingos *et al.* 2004], but was inconclusive presumably due to the difficulties with speech stimuli, which are generally considered to be a special case.

Research in ventriloquism (see Section 1.2.2), could imply that we should be more tolerant to localization errors for sound rendering when we have accompanying visuals. If this were the case, we could change the weighting terms in the clustering algorithm to create fewer clusters for sound sources in the visible frustum. However, a counter argument would be that in the presence of visuals, we are more sensitive to localization, and we should favor more clusters in the viewing frustum.

Our goal was to see whether we could provide some insight into this question with a pilot perceptual study. The next step was to develop and test an improved audio clustering algorithm based on the indications obtained experimentally.

2.2.1 Experimental setup and methodology

We chose the following experimental setup to provide some insight on whether we need more clusters in the visible frustum or not.

The subjects are presented with a scene composed of 10 animated - but not moving - objects emitting “ecologically valid” sounds, i.e., a moo-ing sound for the cow, a helicopter sound, etc. (Figure 2.3).

We have two main conditions: audio only (i.e., no visuals) (condition *A*) and audio-visual (*AV*). Within each main condition we have a control condition, in which sources follow a uniform angular distribution, and the condition we test, where the proportion of clusters in the visible frustum and outside the visible frustum is varied.

We ran our test with 6 subjects (male, aged 23-45, with normal or corrected to normal vision, reporting normal hearing). All were naive about the experiment. Five of them had no experience in audio. Prior to the test, subjects were familiarized with isolated sound effects and their corresponding visual representation.

The subject stands 1 meter away from a 136 x 102 cm screen (Barco Baron Workbench), with an optical headtracking device (ART) and active stereo glasses (see Figure 2.1). The field of view in this large screen experiment is approximately 70 °.

Headphones are used for audio output and our system uses binaural rendering [Blauert 1997, Møller 1992] using the LISTEN HRTF database [IRCAM 2009]. Our subjects were not part of the database. Hence, they performed a “point and click" pre-test to select the best HRTF over a subset of 6 HRTFs selected to be “most representative” similar to [Sarlat *et al.* 2006]. The marks attributed for the test are given with a joystick.

The *A* condition was presented first for three candidates, while *AV* condition was presented first for the other three. No significant effect of ordering was observed.

To achieve the desired effect, objects are placed in a circle around the observer; 5 are placed in the viewing frustum and 5 outside. For both control and main conditions, four

Figure 2.3: *An example view of the experimental setup for the audio-visual pilot user study.*

configurations are used randomly, by varying the proportion of clusters. Condition 1/4 has one cluster in the view frustum and 4 outside, 2/3, has 2 in the view frustum and 3 outside, etc. A uniform distribution of clusters corresponds to condition 1/4, with only 1 cluster in the frustum. Each condition is repeated 15 times with randomized object positions; these repetitions are randomized to avoid ordering effects.

We used the ITU-recommended *triple stimulus, double blind with hidden reference* technique [Grewin 1993, International Telecom. Union 1994]: 2 versions of the scene were presented (“A” and “B”) and a given reference scene which corresponds to unclustered sound rendering. One of the 2 scenes was always the same as the reference (a *hidden reference*) and the other one corresponds to one of our clustering configurations. For each condition, the subject was presented with a screen with three rectangles (“A", “R" and “B"), shown in Fig. 2.3. The subjects were given a gamepad, and were instructed to switch between “A", “B” and “R” using three buttons on the pad, which were highlighted depending on the version being rendered. The subjects were asked to compare the quality of the approximations (“A” or “B”) compared to the reference. They were asked to perform a “*quality judgment paying particular attention to the localization of sounds*" for the 2 test scenes, and instructed to attribute one of 4 levels of evaluation “No difference", “Slightly different”, “Different” and "Clearly different" from the reference, which were indicated in rectangles next to the letter indicating the scene version (see Fig. 2.3).

2.2.2 Analysis and results

We attributed a mark for each evaluation (from 0 to 3). As suggested by this ITU-R standard protocol, we only kept the difference between the test sample and the hidden reference. We also normalized the data by dividing each mark by the mean score of the user (the average of all marks of the candidate over all his tests).

There was no significant difference between the *A* and *AV* conditions regarding the re-

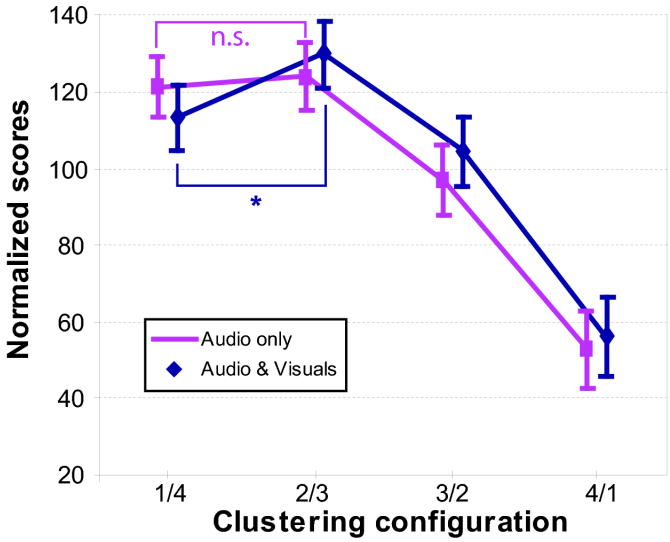


Figure 2.4: *Mean values and 95% confidence intervals (N=6) in A and AV conditions as a function of the number of clusters inside/outside the view frustrum. For AV, the 2/3 configuration gives the best quality scores, which is not the case in the A condition. The "*" underlines that quality judgements in 1/4 and 2/3 cluster configurations for AV are significantly different ($p<0.05$), while the same comparison is non significant (n.s.) in the A condition.*

spective scores of each cluster configuration. However, the difference of quality ratings between configurations was not similar in the two conditions. In condition *A*, 1/4 and 2/3 configurations lead to a similar quality evaluation (see Figure 2.4). In condition *AV*, the best quality is perceived in configuration 2/3. While 2/3 and 1/4 configurations are not perceived differently in condition *A* (Wilcoxon test, N=90, T=640.5, Z=0.21, p=0,83), the quality scores of 2/3 configuration are higher than those of 1/4 configuration in condition *AV* (Wilcoxon test, N=90, T=306.5, Z=2.56, p=0.01). The low perceived quality of the 1/4 configuration can be explained by the loss of accuracy in the spatialization outside the viewing frustum: although spatialization is much improved for visible objects, it is significantly degraded for invisible ones since only one cluster represents most of the sounding objects.

Overall, we consider the above results as a significant indication that, when we use the audio clustering algorithm with visual representation of the sound sources, it is better to have two clusters in the view frustum, compared to a uniform angular distribution. This is indicated by the results for the 2/3 configuration, which is statistically different from all the other configurations in the AV condition. We expect this effect to be particularly true for scenes where there are visible sound sources in the periphery of the view frustum.

Figure 2.5: *Two frames from the walkthrough to test the new audio-visual criterion.*

2.2.3 An audio-visual metric for clustering

Given the above observation, we developed a new weight in the clustering metric which encourages more clusters in the view frustum. We modify the cost-function of the clustering algorithm presented in [Tsingos *et al.* 2004] by adding the following weighting term:

$$1 + \alpha \left(\frac{\cos\theta_s - \cos\theta_f}{1 - \cos\theta_f} \right)^n \tag{2.1}$$

where θ_s is the angle between the view direction and the direction of the sound source relative to the observer, θ_f is the angular half-width of the view frustum and α controls the amplitude and n decay-rate of this visual improvement factor.

2.3 Implementation and Results

To test the new audio-visual criterion, we constructed a variant of the street scene (see Figure 2.5) presented in [Moeck *et al.* 2007] and an appropriate path, in which the positive effect of this criterion is clearly audible. For this test, we used $\alpha = 10$ and $n = 1.5$, which proved to be satisfactory. In this scene, the user follows a path and stops in a given location[1]. We have 132 sources in the scene and target budget of 8 clusters. By switching between the reference, and the approximations with and without the audio-visual metric, we can clearly hear the improvement when more clusters are used in the view frustum. In particular, the car on the right has a siren whose sound is audibly displaced towards the center with the audio-only metric.

2.4 Discussion and Conclusion

We presented a cross-modal perceptual study aimed at determining possible influence of the visuals on the required quality for audio clustering. Although one could expect ventriloquism to allow for rendering simplifications for visible sources, our study suggests that

[1] See the paper's video available at http://www-sop.inria.fr/reves/Basilic/2007/MBTDVA07/MBTDVA07.avi

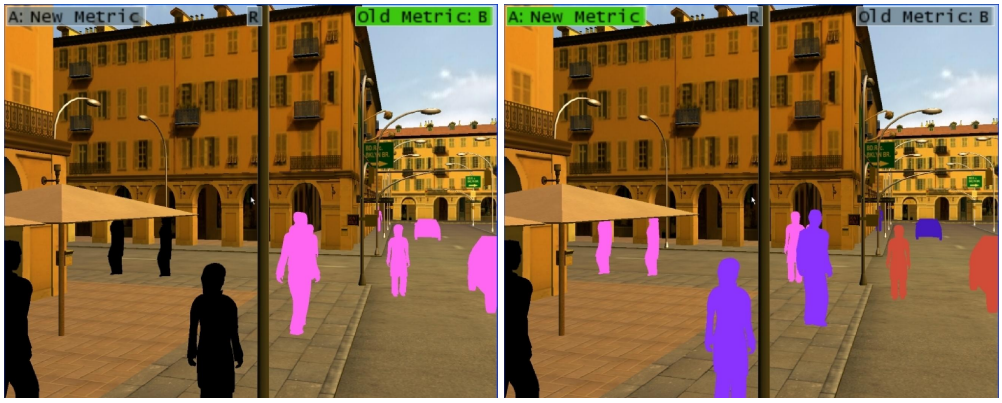


Figure 2.6: *Left: the clusters without the audio-visual metric. Right: the clusters with our new metric. We clearly see that the new metric separates the sources appropriately.*

more clusters might actually be required in this case. A possible explanation for this is that, in a complex scene, clustering is likely to simplify auditory localization cues beyond common ventriloquism thresholds. As a consequence, we introduced a new metric to augment the importance of sources inside the view frustum. We demonstrated an example where, with a large number of sound sources outside the view frustum, it leads to improved results.

In the future, it would be interesting to experiment with auditory saliency metrics to drive clustering and evaluate our algorithms on various combinations of A/V displays, for example, , 5.1 surround or Wave Field Synthesis (WFS) setups. Also, the influence of ventriloquism on these algorithms merits further study.

We also believe that authoring is now becoming a fundamental problem for complex soundscapes. Indeed, authoring complex sounding environments with recorded sounds remains a tedious task. Additional complexity can arise from procedurally synthesized sounds. The most commonly used procedural sounds are impact sounds, generated from objects collisions. In this context, it is interesting to note that humans are tolerant to the asynchrony between an impact sound and the corresponding visual event. This can be seen as a complementary perceptual phenomenon to the spatial ventriloquism effect. Adapting our algorithms to handle combinations of sample-based and impact sounds, and using our audio-visual *temporal* tolerance is the topic of the following chapter.

CHAPTER 3

Fast Modal Sounds with Scalable Frequency-Domain Synthesis

Contents

The contributions in this chapter have been published in the special issue of *ACM Transactions on Graphics*, volume 27, number 3, Proceedings of *SIGGRAPH* [Bonneel *et al.* 2008].

3.1 Introduction

In the previous chapter we studied the *spatial* tolerance between a sound and its visual representation. In order to investigate the possible benefits of using perceptual *temporal* tolerance to asynchrony, we used sounds which react to particular events. In particular, these sounds are produced by the impact of colliding objects, and thus provide an adequate basis for studying and using the temporal tolerance between the visual event of the impact

and the expected corresponding impact sound. One of our major motivations was to treat natural environments consisting of hundreds of simultaneous colliding objects at the same time. However, these environments are too complex for current algorithms, since they cannot generate the sounds in realtime. We thus first developed an improved impact sound generation algorithm, including an improvement based on crossmodal perception, and ran a perceptual study to evaluate our new algorithm.

The rich content of today's interactive simulators and video games includes physical simulation, typically provided by efficient physics engines, and 3D sound rendering, which greatly increases our sense of presence in the virtual world [Larsson *et al.* 2002]. Physical simulations are a major source of audio events: e.g., debris from explosions or impacts from collisions (Fig. 3.1). In recent work several methods have been proposed to physically simulate these audio events notably using *modal synthesis* [O'Brien *et al.* 2002, van den Doel & Pai 2003, James *et al.* 2006]. Such simulations result in a much richer virtual experience compared to simple recorded sounds due to the added variety and improved audio-visual coherence. The user's audiovisual experience in interactive 3D applications is greatly enhanced when large numbers of such audio events are simulated.

Previous modal sound approaches perform recursive synthesis in the *time domain* [van den Doel & Pai 2003]. Recent interactive methods progressively reduce computational load by reducing the number of *modes* in the simulation [Raghuvanshi & Lin 2006, van den Doel *et al.* 2004]. Their computational overload however is still too high to handle environments with large numbers of impacts, especially given the limited budget allocated to sound processing, as is typically the case in game engines.

Interactive audiovisual applications also contain many recorded sounds. Recent advances in interactive 3D sound rendering use *frequency-domain* approaches, effecting perceptually validated progressive processing at the level of Fourier Transform coefficients [Tsingos 2005]. For faster interactive rendering, perceptually based auditory masking and sound-source clustering can be used [Tsingos *et al.* 2004, Moeck *et al.* 2007]. These algorithms enable the use of high-quality effects such as Head-Related Transfer Function (HRTF) spatialization, but are limited to pre-recorded sounds. While the provision of a common perceptually based pipeline for both recorded and synthesized sounds would be beneficial, it is not directly obvious how modal synthesis can be efficiently adapted to benefit from such solutions. In the particular case of contact sounds, a frequency-domain representation of the signal must be computed on-the-fly, since events causing the sounds are triggered and controlled in real-time using the output of the physics engine.

Our solution to the above problems is based on a fundamental intuition: modal sounds have an inherently sparse representation in the frequency domain. We can thus perform frequency-domain modal synthesis by fast summation of a small number of Fourier coefficients (see Fig. 3.1). To do this, we introduce an efficient approximation to the short-time Fourier Transform (STFT) for modes. Compared to time-domain modal synthesis [van den Doel & Pai 2003], we observe 5-8 times speedup in our test scenes, with slight degradation in quality. Quality is further degraded for sounds with faster decays and high frequencies.

In addition to the inherent speed-up, we can integrate modal and recorded sounds into

a common pipeline, with fine-grain scalable processing as well as auditory masking and sound clustering. To compute our STFT we use a constant exponential approximation; we also reconstruct with an Hann window to simplify integration in the full pipeline. However, these approximations reduce the quality of the onset of the impact sound or "attack". We propose a method to preserve the attacks (see Sect. 3.3.3), which can be directly used for modal sounds; using it in the combined pipeline is slightly more involved. We use the Hann window since it allows better reconstruction with a small number of FFT coefficients compared to a rectangular window. A rectangular window is better at preserving the attacks, but results in ringing. Our attack-preserving approach starts with a rectangular subwindow, followed by appropriate Hann subwindows for correct overlap with subsequent frames.

In contrast to typical usage of pre-recorded ambient sounds, physics-driven impact sounds often create peaks of computation load, for example the numerous impacts of debris just after an explosion. We exploit results from human perception to perform temporal scheduling, thus smoothing out these computational peaks. We also performed a perceptually based user evaluation both for quality and temporal scheduling. In summary, we present the following contributions:

- A fast frequency-domain modal synthesis algorithm, leveraging the sparsity of Fourier transforms of modal sounds.
- A full, perceptually based interactive audio rendering pipeline with scalable processing, auditory masking and sound source clustering, for both recorded and modal sounds.
- A temporal scheduling approach based on research in perception, which smooths out computational peaks due to the sudden occurrence of a large number of impact sounds.
- A pilot perceptual user study to evaluate our algorithms and pipeline.

We have implemented our complete pipeline; we present interactive rendering results in Sect. 3.6 for scenes such as those shown in Fig. 3.1 and Fig. 3.6.

3.2 Our Approach

Overview The basic intuition behind our work is the fact that modal sounds have a sparse frequency domain representation. We will show some numerical evidence of this sparsity with examples, and then present our fast frequency-domain modal synthesis algorithm. To achieve this we introduce our efficient STFT approximation for modes, based on singular distributions. We then discuss our full perceptual pipeline including scalable processing, auditory masking and sound source clustering. We introduce a fast energy estimator for modes, used both for masking and appropriate budget allocation. We next present our temporal scheduling approach which smooths out computation peaks due to abrupt increases in the number of impacts. After discussing our implementation and results, we describe

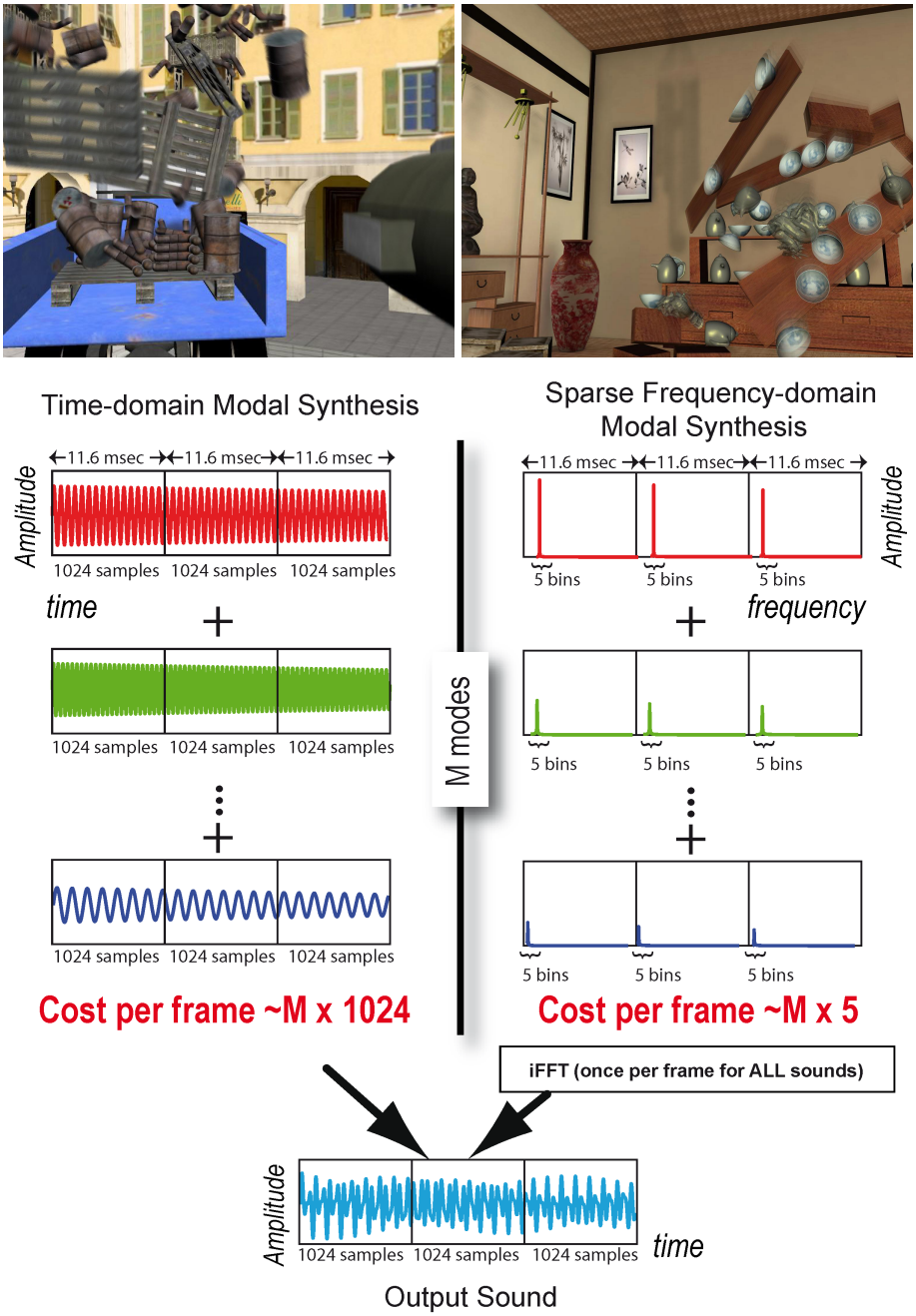


Figure 3.1: Frequency-domain fast mode summation: *Top: Frames of some of the test scenes our method can render in real time. Bottom: (Left) Time-domain modal synthesis requires summing all modes at every sample. (Right) Our frequency-domain modal synthesis exploits the inherent sparsity of the modes' discrete Fourier transforms to obtain lower costs per frame.*

our pilot perceptual user study, allowing us to evaluate the overall quality of our approximations and the perception of asynchrony. Analysis of our experimental results gives an

indication of the perceptual validity of our approach.

Fourier-domain mode mixing Traditional time-domain modal synthesis computes Eq. 1.3 for each sample in time. For frequency-domain synthesis we use the discrete Fourier transform (FFT) of the signal (we show how to obtain this in Sect. 3.3.1). If we use a 1024-sample FFT we will obtain 512 complex coefficients or *bins* representing the signal of a given audio frame (since our signals are real-valued we will only consider positive frequencies). For each such frame, we add the coefficients of each mode in the frequency domain, and then perform an inverse FFT (see Fig. 3.1) once per frame, after all sounds have been added together. The inverse FFT represents a negligible overhead, with a cost of 0.023 ms using an unoptimized implementation [Press *et al.* 1992]. If the number of coefficients contributed by each mode is much less than 512, frequency-domain mixing will be more efficient than an equivalent time-domain approach. However, this will also result in lossy reconstruction, requiring overlapping frames to be blended to avoid possible artifacts in the time-domain. Such artifacts will be caused by discontinuities at frame boundaries resulting in very noticeable clicks. To avoid these artifacts, a *window* function is used, typically a Hann window. Numerous other options are available in standard signal processing literature [Oppenheim *et al.* 1999]. Our method shares some similarities with the work of [Rodet & Depalle 1992] which uses inverse FFTs for additive synthesis.

In what follows we assume that our audio frames overlap by a 50% factor, bringing-in and reconstructing only 512 new time-domain samples at each processing frame using a 1024-sample FFT.

A Hann window is expressed in the time domain as:

$$H(t) = 0.5\left(1 - \cos(\frac{2\pi t}{T})\right)$$

with t the time variable and T the length of the window.

We implemented the Hann window as a product of two square roots of a Hann window, one in frequency to synthesize modes using few Fourier coefficients and the other in time to blend overlapping frames [Zölzer 2002]. At 44.1kHz, we thus process and reconstruct our signal using $512/44100 = 11$ms-long frames.

3.3 Efficient Fourier-Domain Modal Synthesis

We provide some numerical evidence of our intuition, that most of the energy of a modal sound is restricted to a few FFT bins around the mode's frequency. We constructed a small test scene, containing 12 objects with different masses and material properties. The scene is shown in Fig. 3.6. We computed the energy with the signal reconstructed using all 512 bins, then progressively reconstruct with a small number of bins distributed symmetrically around the mode's frequency, and measured the error. We compute percent error averaged over all modes in the scene, for 1 bin (containing the mode's frequency), then 3 bins (i.e., together with the 2 neighboring bins on each side), then both these together with the 2 next bins, etc. Using a single bin, we have 52.7% error in the reconstructed energy; with 3 bins the error drops to 4.7% and with 5 bins the error is at 1.1%. We thus assume that bins

are sorted by decreasing energy in this manner, which is useful for our scalable processing stage (see Sect. 3.4).

This property means that we should be able to reconstruct modal sounds by mixing a very small number of frequency bins, without significant numerical error; however, we need a way to compute the STFT of modes efficiently.

One possible alternative would be to precompute and store the FFTs of each mode and then weight them by their amplitude at runtime. However, this approach would suffer from an unacceptably high memory overhead and would thus be impractical. The STFT of a mode sampled at 44.1kHz requires the storage of 86 frames of 512 complex values, representing 352 Kbytes per mode per second. A typical scene of two thousand modes would thus require 688 Mb.

In what follows we use a formulation based on singular distributions or generalized functions [Hormander 1983], allowing us to develop an efficient approximation of the STFTs of modes.

3.3.1 A Fast Short-time FFT Approximation for Modes

We define $m_k(t)$ as follows for notational convenience:

$$m_k(t) = e^{-\alpha_k t} \sin(\omega_k t) \tag{3.1}$$

We want to estimate the short-time Fourier transform over a given time-frame of a mode $m(t)$ (Eq. 3.1), weighted by a windowing function that we will denote $H(t)$ (e.g., a Hann window). We thus proceed to calculate the short time transform $s(\lambda)$ where λ is the frequency, and t_0 is the offset of the window:

$$s(\lambda) = \mathcal{F}_\lambda\{\ m(t+t_0)H(t)\ \} \tag{3.2}$$

The Fourier transform $\mathcal{F}_\lambda\{\ f(t)\ \}$ that we used corresponds to the definition:

$$\mathcal{F}_\lambda\{\ f(t)\ \} = \int_{-\infty}^{\infty} f(t)\ e^{-i\ \lambda\ t} dt \tag{3.3}$$

Note that the product in the time domain corresponds to the convolution in the frequency domain (see Eq. A.4 in the Appendix). We can use a polynomial expansion (Taylor series) of the exponential function:

$$e^{\alpha(t+t_0)} = e^{\alpha t_0} \sum_{n=0}^{\infty} c_n (\alpha t)^n \tag{3.4}$$

where $c_n = 1/n!$. Next, the expression for the Fourier transform of a power term is a distribution given by:

$$\mathcal{F}_\lambda\{\ t^n\ \} = 2\pi i^n \delta^{(n)}(\lambda) \tag{3.5}$$

where δ is the Dirac distribution, and $\delta^{(n)}$ its n'th derivative. From Eqs. 3.4 and 3.5, we have the expression for the Fourier transform of the exponential:

$$\mathcal{F}_\lambda\Big\{\ e^{\alpha(t+t_0)}\ \Big\} = e^{\alpha t_0} \sum_{n=0}^{\infty} c_n \alpha^n 2\pi i^n \delta^{(n)}(\lambda) \tag{3.6}$$

The Fourier transform of a sine wave is a distribution given by:

$$\mathcal{F}_\lambda\{\ \sin(\omega(t+t_0))\ \} = i\pi\left(e^{-i\omega t_0}\delta(\lambda{+}\omega) - e^{i\omega t_0}\delta(\lambda{-}\omega)\right) \tag{3.7}$$

We also know that δ is the neutral element of the convolution (see Eq. A.2 in the Appendix). Moreover, we can convolve the Fourier Transforms of the exponential and the sine since they both have compact support. From Eqs. 3.6 and 3.7, we finally have:

$$\mathcal{F}_\lambda\{\ m(t+t_0)\ \} = \pi e^{\alpha t_0}\sum_{n=0}^{\infty} c_n\alpha^n i^{n+1}\cdot\left(e^{-i\omega t_0}\delta^{(n)}(\lambda{+}\omega) - e^{i\omega t_0}\delta^{(n)}(\lambda{-}\omega)\right) \tag{3.8}$$

Convolution of Eq. 3.8 with a windowing function H leads to the desired short time Fourier transform of a mode. Using the properties of distributions (Eq. A.2, A.3 in the Appendix), and Eq. 3.8, we have:

$$s(\lambda) = \frac{1}{2}e^{\alpha t_0}\sum_{n=0}^{\infty} c_n\alpha^n i^{n+1}\left(e^{-i\omega t_0}\mathcal{F}_\lambda(H)^{(n)}(\lambda{+}\omega) - e^{i\omega t_0}\mathcal{F}_\lambda(H)^{(n)}(\lambda{-}\omega)\right) \tag{3.9}$$

$\mathcal{F}_\lambda(H)^{(n)}(\lambda{+}\omega)$ is the n-th derivative of the (complex) Fourier transform of the window H, taken at the value $(\lambda{+}\omega)$.

Note that Eq. 3.8 is still a distribution, since we did not constrain the mode to be computed only for positive times, and the mode itself is not square-integrable for negative times. However, this distribution has compact support which makes the convolution of Eq. 3.9 possible [Hormander 1983].

For computational efficiency, we truncate the infinite sum of Eq. 3.9, and approximate it by retaining only the first term. The final expression of our approximation to the mode STFT is thus:

$$s(\lambda) \approx \frac{1}{2}e^{\alpha t_0}c_0 i\left(e^{-i\omega t_0}\mathcal{F}_\lambda(H)(\lambda{+}\omega) - e^{i\omega t_0}\mathcal{F}_\lambda(H)(\lambda{-}\omega)\right) \tag{3.10}$$

Instead of $c_0 = 1$ (Eq. 3.4), we take c_0 to be the value of the exponential minimizing $\int_{t_0}^{t_0+\Delta t}(e^{-\alpha t} - c_0)^2 dt$, where Δt is the duration of a frame, resulting in a better piecewise constant approximation:

$$c_0 = \frac{e^{-\alpha t_0} - e^{-\alpha(t_0+\Delta t)}}{\alpha\,\Delta t}. \tag{3.11}$$

This single term formula is computationally efficient since the Fourier transform of the window can be precomputed and tabulated, which is the only memory requirement. Moreover both complex exponentials are conjugate of each other meaning that we only need to compute one sine and one cosine.

3.3.2 Speedup and Numerical Validation

Consider a scene requiring the synthesis of M modes. Using a standard recursive time-domain solution [van den Doel & Pai 2003], and assuming 512-sample frames, the cost of the frame is $M \times 512 \times C_{mt}$. The cost C_{mt} of evaluating a mode in the time domain is 5 or 6 multiplies and adds, using the recursive formulation of Eq. 6 in [van den Doel & Pai 2003].

In our approach, assuming an average of B bins per mode, the cost of a frame is $M \times B \times C_{STFT}$ plus the cost (once per frame) of the inverse FFT. The cost C_{STFT} of evaluating Eq. 3.10, is about 25 operations. With a value of $B = 3$ (it is often lower in practice), we have a potential theoretical speedup factor of 30-40 times. If we take into account the fact that we have a 50% overlap due to windowing, this theoretical speedup factor drops to 15-20 times.

We have used our previous test scene to measure the speedup of our approach in practice, compared to [van den Doel & Pai 2003]. When using $B = 3$ bins per mode, we found an average speedup of about 8, and with $B = 5$ bins per mode about 5. This reduction compared to the theoretical speedup is probably due to compiler and optimization issues of the two different algorithms.

Finally, we examine the error of our approximation for a single sound. We tested two different windows, a Hann window with 50% overlap and a Rectangular window with 10% inter-frame blending. In Fig. 3.2 we show 3 frames, with the reference [van den Doel & Pai 2003] in red and our approximation in blue, taking $B = 5$ bins per mode.

Taken over a sequence including three impacts (a single pipe in the Magnet scene, see Sect. 3.6), for a total duration of about 1 sec., the average overall error for the Rectangular window is 15% with 5 bins, and 21% with 3 bins (it is 8% if we use all 512 bins). This error is mainly due to small ringing artifacts and possibly to our constant exponential approximation, which can be seen at frame ends (see Fig. 3.2). Using the Hann window, we have 35-36% error for both 512 and 5 bins, and 36% with 3 bins. This would indicate that the error is mainly due to the choice of window. As can be seen in the graph (Fig. 3.2 (right)) the error with the Hann window is almost entirely in the first frame, at the onset, or "attack", of the sound for the first 512 samples (i.e., 11 ms at 44.1kHz). The overall quality of the signal is thus preserved in most frames; in contrast, the ringing due to the rectangular window can result in audible artifacts. For this reason, and to be compatible with the pipeline of [Moeck *et al.* 2007], we chose to use the Hann window.

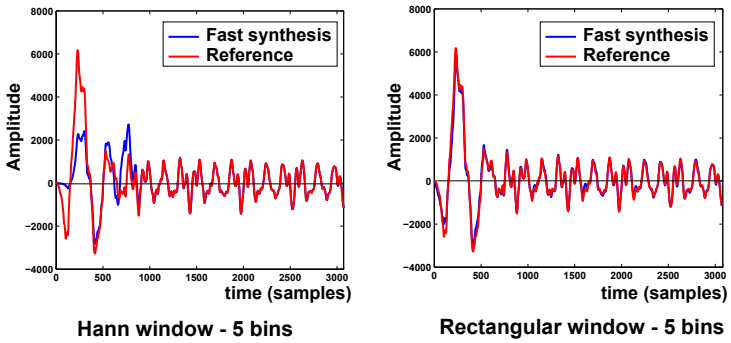


Figure 3.2: *Comparison of Reference with Hann window and with Rectangular window reconstruction using a 1024-tap FFT.*

3.3.3 Limitations for the "Attacks" of Impact Sounds

Contrary to time domain modal synthesis approaches such as [van den Doel & Pai 2003, Raghuvanshi & Lin 2006], the use of the Hann window as well as the constant exponential approximation (CEA) degrades the onset or "attack" for high frequency modes. This attack is typically contained in the first few frames, depending on decay rate.

To study the individual effect of each of the CEA and the Hann window in our reconstruction process, we computed a time-domain solution using the CEA for the example of a falling box (Fig. 3.3(left)) and a time domain solution using a Hann window to reconstruct the signal (Fig. 3.3(right)). We plot the time-domain reference [van den Doel & Pai 2003] in red and the approximation in blue. As we can see, most of the error in the first 7msec is due to the Hann window whereas the CEA error remains lower. The effect of these approximations is the suppression of the "crispness" of the attacks of the impact sounds.

Use of the Hann window and the CEA as described previously has the benefit of allowing seamless integration between recorded and impact sounds, as described next in Sect. 3.4. In complex soundscapes containing many recorded sounds, this approximation may be acceptable. However, in other cases the crispness of the attacks of the modal sounds can be important.

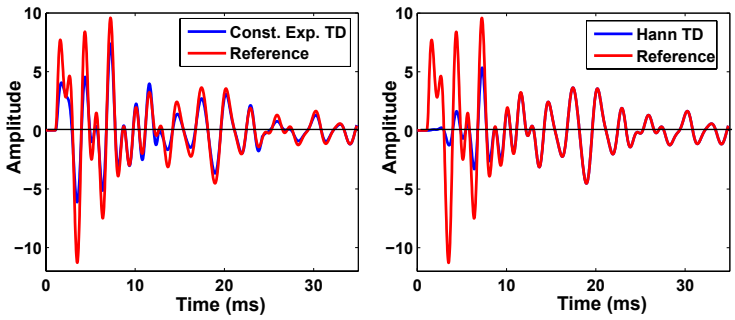


Figure 3.3: *Comparison of Constant Exponential Approximation in the time domain (TD) and Hann window reconstruction in the TD, with the reference for the sharp sound of a falling box.*

To better preserve the attacks of the impacts sounds, we treat them in a separate buffer and split the first 1024-sample frame into four subframes. Each subframe has a corresponding CEA and with a specialized window. In what follows, we assume that all contact sounds start at the beginning of a frame.

We design a windowing scheme satisfying four constraints: 1) avoid "ramping up" to avoid smoothing the attack, 2) end with a 512 sample square root of Hann window to blend with the buffer for all frames other than the attack, 3) achieve perfect reconstruction, i.e., all windows sum to one, 4) require a minimal number of bins overall, i.e., use Hann windows which minimize the number of bins required for reconstruction [Oppenheim *et al.* 1999].

The first subframe is synthesized using a rectangular window for the first 128 samples (constraint 1) followed by half of a Hann window ("semi-Hann" from now on) for the next 128 samples and zeros in the remaining 768 samples; this is shown in blue in Fig. 3.4.

The next two subframes use full 256-sample Hann windows, starting at samples 128 and 256 respectively (red and green in Fig. 3.4). The last subframe is composed of a semi-Hann window from samples 384 to 512 and a square root of a semi-Hann window for the last 512 samples for correct overlap with the non-attack frames, thus satisfying constraint 2 (black in Fig. 3.4). All windows sum to 1 (constraint 3), and Hann windows are used everywhere except for the first 128 samples (constraint 4). These four buffers are summed before performing the inverse FFT, replacing the original 1024 sample frame by the new combined frame. We use 15 bins in the first subframe.

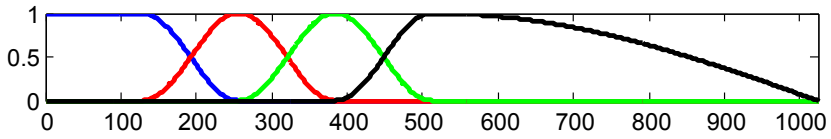


Figure 3.4: *Four sub-windows to better preserve the sound attack.*

The increase in computational cost is negligible, since the additional operations are only performed in the first frame of each mode: for the Oriental scene example shown in Figure 3.1[1], the additional cost of the mixed window for attacks is 1.2%. However, the integration of this method with recorded sounds is somewhat more involved; we discuss this in Sect. 3.8.

3.4 A Full Perceptually Based Scalable Pipeline for Modal and Recorded Sounds

An inherent advantage of frequency domain processing is that it allows fine-grain scalable audio processing at the level of an FFT bin. In [Tsingos 2005], such an approach was proposed to perform equalization and mixing, reverberation processing and spatialization on prerecorded sounds. Signals are prioritized at runtime and a number of frequency bins are allocated to each source, thus respecting a predefined budget of operations. This importance sampling strategy is driven by the energy of each source at each frame and used to determine the cut-off point in the list of STFT coefficients.

Given our fast frequency-domain processing described above, we can also use such an approach. In addition to the fast STFT synthesis, we also require an estimation of energy, both of the entire impact sound, and of each individual mode. In [Tsingos 2005] sounds were pre-recorded, and the FFT bins were precomputed and pre-sorted by decreasing energy.

For our scalable processing approach, we first fix an overall mixing budget, e.g., 10,000 bins to be mixed per audio frame. At each frame we compute the energy E_s of each impact sound over the frame and allocate a budget of frequency bins per sound proportional to its energy. We compute the energy E_m of each mode for the entire duration of the sound once, at the time of each impact, and we use this energy weighted by the mode's squared

[1] See also the paper's video available at http://www-sop.inria.fr/reves/Basilic/2008/BDTVJ08/FastModalSounds.avi

amplitude to proportionally allocate bins to modes within a sound. After experimenting with several values, we assign 5 bins to the 3 modes with highest energy, 3 bins for the next 6 and 1 bin for all remaining modes. The use of a single bin to represent a mode can create ringing artefacts which result in the perception of a ghost mode. However, in our case we found that most of the energy of the impact sounds was located in the first few modes. We thus only used a single bin to represent low energy modes, so that ringing artefacts are not perceived. In contrast, using 3 bins allows good reconstruction but requires three times more computation, which is not desirable for low energy modes. The use of 5 bins per mode, with our overlap-add technique, allows almost perfect reconstruction which is needed for the highest energy and audible modes. We summarize these steps in the following pseudo-code.

```
1. PerImpactProcessing(ImpactSound S) // at impact notification
2.   foreach mode of S
3.     compute total energy E_m
4.   Sort modes of S by decreasing E_m
5.   Compute total energy of S for cutoff
6.   Schedule S for processing
```

```
1. ScalableAudioProcessing() // called at each audio frame
2.   foreach sound S
3.     Compute E_s
4.     Allocate FFT bin budget based on E_s
5.     Modes m_1, m_2, m_3 get 5 bins
6.     Modes m_4 − m_9 get 3 bins
7.     1 bin to remaining modes until end of budget
8.   endfor
```

3.4.1 Efficient Energy Estimation

To allocate the computation budget for each impact sound, we need to compute the energy, E_s, of a modal sound s in a given frame, i.e., from time t to time $t + \Delta t$:

$$E_s = \int_t^{t+\Delta t} s^2(x)dx. \tag{3.12}$$

From Eq. 1.3 and 3.1, we express E_s as:

$$E_s = < s, s > = \sum_{i=0}^{M}\sum_{j=0}^{M} a_i a_j < m_i, m_j > . \tag{3.13}$$

For a given frame, the scalar product $< m_i, m_j >$ has an analytic expression (see Eq. A.7 in the additional material[2]). Because this scalar product is symmetric, we only have to compute half of the required operations.

In our experiments, we observed that most of the energy of an impact sound is usually concentrated in a small number N of modes (typically 3). To identify the N modes with highest energy, we compute the total energy, E_m, of each mode as:

[2]As we will see in Chapter 5, a simpler and more efficient formula for this same expression is given in Eq. A.8. We will also propose an even more efficient *approximation* in the same chapter.

$$E_m = \int_0^\infty \left(\sin(\omega x)e^{-\alpha x}\right)^2 dx = \frac{1}{4}\frac{\omega^2}{\alpha(\alpha^2+\omega^2)} \tag{3.14}$$

After computing the E_m's for each mode, we weight them by the square of the mode's amplitude. We then sort the modes by decreasing weighted energy. To evaluate Eq. 3.13 we only consider the N modes with highest energy. We re-use the result of this sort for budget allocation.

We also compute the total energy for a sound, which is used to determine its duration, typically when 99% of the energy has been played back. We use Eq. 3.13 and an expression for the total energy (rather than over a frame), given in the additional material (Eq. A.6).

Numerical Validation We use the test scene presented in Sect. 3.3.1 (Fig. 3.6) to perform numerical tests with appropriate values for N. We evaluated the average error of the estimated energy, compared to a full computation. Using 3 modes, for all objects in this scene, the error is less than 9%; for 5 modes it falls to 4.9%.

3.4.2 A Complete Combined Audio Pipeline

In addition to frequency-domain scalable processing, we can also use the perceptual masking and sound-source clustering approaches developed in [Tsingos *et al.* 2004, Moeck *et al.* 2007]. We can thus mix pre-recorded sounds, for which the STFT and energy have been precomputed, with our frequency domain representation for modal sounds and perform global budget allocation for all sounds. As a result, masking between sounds is taken into account, and we can cluster the surviving unmasked sources, thus optimizing the time for per-sound source operations such as spatialization. In previous work, the masking power of a sound also depends on a tonality indicator describing whether the signal is closer to a tone or a noise, noisier signals being stronger maskers. We computed tonality values using a spectral flatness measure [Tsingos *et al.* 2004] for several modal sounds and obtained an average of 0.7. We use this constant value for all modal sounds in our masking pipeline.

3.5 Temporal Scheduling

One problem with systems simulating impact sounds is that a large number of events may happen in a very short time interval (debris from an explosion, a collapsing pile of objects, etc.), typically during a single frame of the physics simulation. As a result, all sounds will be triggered simultaneously resulting in a large peak in system load and possible artifacts in the audio ("cracks") or lower audio quality overall. Our idea is to spread out the peaks over time, exploiting results on audio-visual human perception.

As mentioned in Sect. 1.2.2, there has been extensive study of audiovisual asynchrony in neuroscience which indicates that the brain is able to compensate for the different delays between an auditory and a visual event in causal inference. To exploit this property, we introduce a scheduling step at the beginning of the treatment of each audio frame. In particular, we maintain a list of sound events proposed by the physics engine (which we call

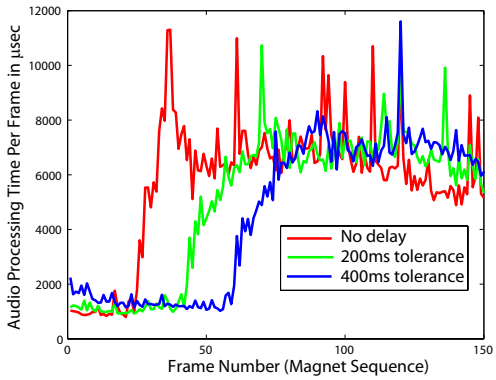


Figure 3.5: *Effect of temporal scheduling; computational peaks are delayed, the slope of increase in computation time is smoothed out and the duration of peaks is reduced. Data for the Magnet sequence, starting just before the group of objects hits the floor.*

TempSoundsList) and a list of sounds currently processed by the audio engine (*CurrSoundsList*). At the beginning of each audio frame, we traverse *TempSoundsList* and add up to 20 new sounds to *CurrSoundsList* if one of the following is true:

- *CurrSoundsList* contains less than 50 sounds
- Sound s has been in *TempSoundsList* for more than T ms.

The values 20 and 50 were empirically chosen after experimentation. We use our perceptual tolerance to audio-visual asynchrony by manipulating the threshold value T for each sound. In particular we set T to a desired value, for example 200 ms corresponding to the results of [Guski & Troje 2003]. We then further modulate T for sounds which are outside the visible frustum; T is increased progressively as the direction of the sound is further from the center of the field of view. For sounds completely behind the viewer the delay T is set to a maximum of 0.5 seconds. Temporal scheduling only occurs for impact sounds, and not for recorded sounds such as a gun firing which are time-critical and can have a remote effect.

Our approach reduces the number and density of computational peaks over time. Although some peaks still occur, they tend to be smaller and/or sharper, i.e., occur over a single frame (see Fig. 3.5). Since our interactive system uses buffered audio output, it can sustain such sparse peaks over a single frame, while it could not sustain such a computational overload over several consecutive frames.

3.6 Implementation and Results

Our system is built on the *Ogre3D*[3] game engine, and uses the *PhysX*[4] real-time physics engine simulator. Throughout this chapter , we use our own (re-)implementations

[3] http://www.ogre3d.org
[4] http://www.ageia.com

of [van den Doel & Pai 2003], [O'Brien *et al.* 2002] and [Raghuvanshi & Lin 2006]. In our implementation of [O'Brien *et al.* 2002] we replaced the computationally intensive force f in Eq.1.2 by the force value reported by PhysX at the nearest surface node to the impact point. For [van den Doel & Pai 2003], we use $T = 512$ samples at 44.1KHz; the size of the impact filter with a force profile of $\cos(2\pi t/T)$ is $T = 0.37ms$ or 16 samples (Eq.17 of that paper).

For objects generating impact sounds, we precompute modes using the method of O'Brien et al. [O'Brien *et al.* 2002]. Sound radiation amplitudes of each mode are estimated with a far-field radiation model (Eq. 15, [James *et al.* 2006]).

Audio processing was performed using our in-house audio engine, with appropriate links between the graphics and audio. The audio engine is described in detail in [Moeck *et al.* 2007]. In total we run three separate threads: one for each of the physics engine, graphics and audio. All timings are reported on a dual-processor, dual-core Xeon running at 2.3Ghz.

Figure 3.6: *From left to right, snapshots of the large Magnet scene, the Boxes scene, the test scene used for numerical validation and our prototype rolling demo.*

3.6.1 Interactive Sessions Using the Pipeline

We have constructed four main scenes for demonstration purposes, which we refer to as "Oriental", "Magnet", "Truck" and "Boxes"; we show snapshots of each in Fig. 3.1 (Truck and Oriental) and 3.6. The goal was to construct environments which are similar to typical simulators or games settings, and which include a large number of impact sounds, as well as several prerecorded sounds. All sounds were processed in the Fourier-domain at 44.1kHz using 1024-tap FFTs and 50% overlap-add reconstruction with Hann windowing. The "Oriental" and "Box" scenes contain modal sounds only and thus use the attack preserving approach (Sect. 3.3.3). Hence, our audio thread runs at 83Hz; we then output reconstructed audio frames of 512 samples. The physics thread updates object motion at 140Hz, and the video thread runs at between 30-60Hz depending on the scene and the rendering quality (shadows, etc.).

The Magnet scene contains prerecorded industrial machinery and closing door sounds, while the Truck scene contains traffic and helicopter sounds. Basic scene statistics are given in Table 3.1, for the demo versions of the scenes shown in Figures 3.1 and 3.6, or in the video - see footnote 1.

Scene	O	T	P	M_i	M/o
Oriental	173	730K	0	665	214
Boxes	200	200K	0	678	376
Magnet	110	300K	16	971	164
Truck	214	600K	15	268	221

Table 3.1: *Basic statistics for example scenes. O: number of objects simulated by the physics engine producing contact sounds, T: total number of triangles in the scene, P: number of pre-recorded sounds in the scene and M_i: maximum number of impact* sounds played *per frame. M/o: average number of modes/object.*

Scene	Total	Mixing	Energy	Masking	Clustering
Magnet	3.2	1.3	0.6	1.0	0.3
Truck	2.7	1.5	0.9	0.3	0.1

Table 3.2: *Cost in milliseconds of each stage of our full pipeline.*

3.6.2 Quality and Performance

We performed two comparisons for our fast modal synthesis: the first was with the "standard" time-domain (TD) method of [van den Doel & Pai 2003] using recursive evaluation, and the second with the mode-culling time-domain approach of [Raghuvanshi & Lin 2006], which is the fastest TD method to date. We used the "Oriental" scene for this comparison, containing only modal sounds.

Comparison to "standard" TD synthesis In terms of quality, we tested examples of our frequency-domain synthesis with 3 and 5 bins per mode, together with a time-domain reference. The quality for 5 bins is very close to the reference. The observed speedup was 5-8 times, compared to [van den Doel & Pai 2003].

Comparison to mode-culling TD synthesis To compare to mode-culling, we apply the mode truncation and quality scaling stages of [Raghuvanshi & Lin 2006] at each audio frame. We then perform fast frequency domain synthesis for the modes which have not been culled. For the same mode budget our frequency processing allows a speedup of 4-8 times; the difference in speedup with the "standard" TD synthesis is due to implementation issues. The quality of the two approaches is slightly different for the same mode budget, but in both cases subjectively gives satisfactory results.

Full Combined Pipeline The above comparisons are restricted to modal sounds only. We also present results for other scenes, augmented with recorded sounds. These are rendered using our full pipeline, at low overall budgets but with satisfactory quality.

We present statistics for our approach in Table 3.2, using a budget of 8000 bins. First we indicate the cost (in milliseconds) of each component of our new combined pipeline: mixing, energy computation, masking and clustering, as well as the total cost. As we can see there is a non-negligible overhead of the pipeline stages; however the benefit of being

able to globally allocate budget across modal and recorded sounds, and of course all the perceptually based accelerations, justifies this cost.

The number of sounds at each frame over the course of our interactive test sequences varied between 195 and 970. If no masking or culling were applied there would be between 30,000 to 100,000 modes to be played per audio frame on average in these sequences. We use 15,000 to 20,000 frequency bins in all interactive sessions. The percentage of prerecorded sounds masked was around 50% on average and that of impact sounds was around 30%.

3.7 Pilot Perceptual Evaluation

Despite previous experimental studies for perceptually based audio rendering for *pre-recorded* sounds [Moeck *et al.* 2007, Tsingos *et al.* 2004], and the original neuroscience experiments for asynchrony [Guski & Troje 2003], we consider it imperative to conduct our own pilot study, since our context is very different. We have two conditions in our experiment: the goal of the first condition is to evaluate the overall audio quality of our approximations while that of the second is to evaluate our temporal scheduling.

3.7.1 Experiment Setup and Procedure

In our experiment we used the Magnet and Truck (Sect. 3.6) environments, but with fewer objects, to avoid making the task too hard for the subjects. We used two 6 second pre-recorded paths for each of the two scenes. To ensure that all the stimuli represent the exact same sequence of events and to allow the presentation of a reference in real-time, we synchronize all threads and store the output audio and video frames of our application to disk. Evidently any other delay in the system has to be taken into account. We choose the parameters of our simulation to be such that we do not perceive “cracks" in the audio when running interactively with the same budget settings. Video sequences are then played back during the study. For the reference sequences, all contact sounds are computed in the time-domain and no perceptual processing is applied when mixing with the recorded sounds in the frequency domain. We use non-individualized binaural rendering using Head Related Transfer Functions (HRTFs) chosen from the *Listen* database [IRCAM 2009].

The interface is a MUSHRA-like [International Telecom. Union 2003] slider panel (see Figure 3.7), in which the user can choose between a reference and five different approximations (A, B, C, D, E), each with a different budget of frequency bins. The subject uses a slider to rate the quality of each stimulus. One of the five stimuli is a hidden reference. The radio button above each slider allows the subject to (re)start the corresponding sequence. For the synchronization condition, we choose one budget which has a good rating in the quality condition (typically C_3 in Table 3.3), and delay the audio relative to graphics by a variable threshold T. The budgets and thresholds used are shown in Table 3.3. A tick box is added under each sound for synchrony judgment.

The subject listens to the audio with headphones and is instructed to attend to both visuals and audio. There are 8 panels, corresponding to all the conditions; stimuli are presented in random order. Detailed instructions are given on screen to the user, who is

Figure 3.7: *The MUSHRA-like [International Telecom. Union 2003] interface used in our validation experiment.*

	Budget FFT bins				Audio Delay (ms)			
Scene	C_1	C_2	C_3	C_4	T_1	T_2	T_3	T_4
Magnet	700	1.5K	2.5K	4K	0	120	200	400
Truck	1K	2K	4K	8K	0	120	200	400

Table 3.3: *Budget and delay values used for the perceptual experiments. C_i and T_i are the budget and delay conditions used.*

asked to rate the quality and tick if asynchrony between audio and visuals is detected. Rating of each panel is limited to 3 minutes, at which point rating is disabled.

3.7.2 Analysis of the Experiments

We ran the experiment with 21 subjects who were members of our research institutes, and were all naive about the goal of our experiments. We show the average quality ratings and the percent perceived asynchrony averages for the experiment in Table 3.4.

As we can see, for the Magnet scene, the budget of 4,000 bins was sufficient to give quality ratings of 83-85% very close to the hidden reference rated at 84-89%. An analysis of variance (ANOVA) [Howell 1992] with repeated measures on quality ratings shows a main effect of budget on perceived quality ($F(4,20)=84.8$, $p<0.00001$). For the Truck scene the quality rating for 8,000 bins was lower. This is possibly due to the fact that the

	Percent Perceived Quality					% Perceived Asynchrony			
Scene	C_1	C_2	C_3	C_4	Ref	T_1	T_2	T_3	T_4
Magnet1	51.1	64.7	78.0	83.1	84.9	0	24	48	71
Magnet2	48.8	70.1	76.5	85.2	88.9	10	5	0	10
Truck1	26.3	41.8	54.2	66.0	87.3	24	24	14	38
Truck2	24.8	28.6	42.7	66.0	89.0	14	14	38	29

Table 3.4: *Results of our experiments: average quality and percent perceived asynchrony for the 2 scenes and 2 paths.*

recorded sounds require a significant part of the frequency bin budget, and as a result lower the overall perceived quality.

In terms of asynchrony, the results have high variance. However, it is clear that audio-visual asynchrony was perceived less than 25% of the time, for delays under 200msec.

Overall, we consider these results to be a satisfactory indication that our approximations work well both in terms of progressive processing and for our temporal scheduling algorithm. In particular, there is a strong indication that increasing the budget does result in perceptually improved sounds, and that only a small percentage of users perceive asynchrony with temporal scheduling with delays less than 200ms.

3.8 Discussion and Conclusions

We have presented a new frequency-domain approach to modal sound rendering, which exploits sparseness of the Fourier Transform of modal sounds, leading to a 4-8 speedup compared to time-domain approaches [van den Doel & Pai 2003, Raghuvanshi & Lin 2006], with slight quality degradation. Furthermore, our approach allows us to introduce a combined perceptual audio pipeline, treating both prerecorded and on-the-fly impact sounds, and exploiting scalable processing, auditory masking, and clustering of sound sources. We used crossmodal results on perception of audiovisual synchronization to smooth out computational peaks which are frequently caused by impact sounds, and we performed a pilot perceptual study to evaluate our combined pipeline.

Use of the Hann window allows direct integration of modal and recorded sounds (see Sect. 3.4), but leads to lower quality attacks of impact sounds. We have developed a solution to this problem, splitting the treatment of the first frame of each attack into four subframes with appropriate windows. This solution can be easily used in scenes exclusively containing modal sounds. For the pipeline combining recorded and modal sounds, and in particular for clustering, we would need a separate buffer for attacks in each cluster thus performing twice as much post-processing (HRTF processing, etc.). The rest of the pipeline would remain unchanged. This issue will be addressed in Chapter 5.

We developed an initial solution for rolling sounds in our pipeline, using a noise-like sound profile, similar in spirit to [van den Doel *et al.* 2001]. To simulate the continuous rolling excitation, we precompute a noise profile in the Fourier domain, and perform dynamic low-pass filtering based on velocity. Convolution with the excitation is a simple

product in the Fourier domain, making our approach efficient. A first example is shown in Figure 3.6. Nonetheless, a general solution to continuous excitation in our framework requires mixing delayed copies of past frames, incurring additional costs. We expect masking and progressive processing to limit this overhead, similar to the reverberation in [Tsingos 2005].

Another limitation is the overhead of our pipeline which is not negligible. For practical usage of real-time audio rendering, such as game engines, we believe that the benefits outweigh this drawback. In addition to the perceptually based accelerations, we believe that the ability to treat recorded and synthesized sounds in a unified manner is very important for practical applications such as games.

Currently, when reducing the budget very aggressively the energy computation can become a dominant cost. It is possible to precompute a restricted version of the energy, if we assume that forces are always applied in the normal direction. This is the case for recording-based systems (e.g., [van den Doel & Pai 1998]). However, this reduces the flexibility of the system. We address this final issue by approximating the energy in Chapter 5.

CHAPTER 4

Bimodal perception of audio-visual material properties for virtual environments

Contents

The contributions in this chapter have been accepted for publication in the journal *ACM Transactions on Applied Perception* [Bonneel *et al.* 2010].

4.1 Introduction

Interactive audio-visual virtual environments are now commonplace, ranging from computer games with high-quality graphics and audio, to virtual environments used for training, car and flight simulation, rehabilitation, therapy etc. In such environments, synthetic objects have audio-visual material properties, which are often based on physical measurements of real objects. Realistic, high-quality rendering of these materials is a central element for the overall realism [Vangorp *et al.* 2007] and the sense of immersion offered by such virtual environments: This is true both for graphics and audio. Real-time or interactive performance is central to such systems. One way these systems handle the ever-increasing

complexity of the graphics and the sounds is to use *level-of-detail* (LOD) rendering. This approach consists in rendering lower quality versions of entities in the virtual environment, which require lower computation time. As a result, more complex environments can be rendered. In what follows, we will use the general term *material* to mean the audio-visual material properties which are physically measurable. The goal of our study is twofold. First we ask whether audio and graphics mutually interact in the perception of material rendering quality, and in particular when independently varying the LOD for both audio and graphics in an interactive rendering context. Second, if such an interaction exists, we want to see whether it can be exploited to improve overall interactive system performance. Therefore, we hope both to identify a perceptual effect of the influence of audio and graphics on material perception *and* to achieve algorithmic gain.

Given the interactive audio-visual context of our work, we will concentrate on choices of stimuli which are feasible in the context of such systems. This inevitably leads to the use of approximations to create LOD for both audio and graphics, so that practical algorithmic benefit can be achieved. In the virtual environments of this study audio is rendered in realtime and graphics rendering runs at 29 frames per second.

We have designed an experiment to evaluate whether there is a mutual influence of audio and graphics on the perception of materials. Since we are interested in optimizing the perception of material quality in an interactive rendering setting, we chose to perform a material similarity experiment (see [Klatzky *et al.* 2000] for a similar experiment on material perception of contact sounds for audio only).

Stimuli vary along two dimensions: graphics LOD and audio LOD. Participants are asked to compare these to a hidden audio-visual reference. This reference is rendered at the highest possible quality given the constraints of the interactive system and stimuli are synthetic objects falling onto a table. It is called "hidden" since the participant is not aware that this stimulus is a reference stimulus. However, this stimulus is indeed shown and not hidden to the participant. Audio for contact sounds is provided by using modal synthesis [van den Doel & Pai 2003], and LOD result from choosing a progressively larger number of modes. Graphics are rendered using an environment map and Bidirectional Reflectance Distribution Functions (BRDF) [Cook & Torrance 1982]. A BRDF describes how a material reflects light, and can be measured from real materials [Matusik *et al.* 2003]. To provide visual LOD, we project the BRDF onto a Spherical Harmonic basis [Kautz *et al.* 2002], and increase the number of coefficients to obtain progressively better visual quality. Increasing the number of modes or spherical harmonic coefficients in our LOD improves the mathematical approximation, i.e., the error of the approximations with respect to the high-quality reference diminishes.

Results of the present experiment show that this also results in better perceived quality for each of audio and visuals independently. Most interestingly, we also show that for this context there is a mutual influence of sound and graphics on the perception of material similarity. In particular, if we interpret the similarity rating to a high-quality reference as a measure of quality, material quality is judged to be higher when sound LOD is higher. We highlight how this result can be directly used to significantly improve overall rendering performance in an interactive audio-visual system. To our knowledge, this study is the first to demonstrate a combined effect of graphics and audio on a task related to material

perception.

4.2 Methods

4.2.1 Participants

Ten volunteers (7 men) from 23 to 46 years old (mean age 30.8 years, Standard deviation: 7.7 years) participated in the experiment. All had normal or corrected to normal vision and all reported normal hearing. All were naive to the purpose of the experiment.

4.2.2 Stimuli

We next present a detailed description of both visual and auditory stimuli as presented to the participants, and the corresponding LOD mechanisms used to create the stimuli.

4.2.2.1 Visual LOD

In order to make our method applicable for realtime rendering, we interactively render realistic materials with measured BRDF. This rendering can then be used to generate visual stimuli for the experiment and is also usable in the context of interactive audio-visual applications (computer games, virtual environments, audiovisual simulations etc.). Using realtime rendering in the experiment simplifies the potential application of the results in interactive systems, since the conditions are the same. To achieve realtime rendering in complex environments, various approximation have been proposed which include infinite light sources [Ramamoorthi & Hanrahan 2002, Kautz *et al.* 2002], static viewpoint, [Ben-Artzi *et al.* 2006] and/or static geometry [Sloan *et al.* 2002, Kristensen *et al.* 2005]. We assume infinite light sources through the use of an environment map which gives the illumination from distant sources (e.g., a panoramic photograph of the sky). This approximation is reasonable since the environment map was captured at the true location of the object and the motion of the object is not large compared to the size of the environment.

A commonly used method to interactively render measured BRDFs with environment maps is the projection of the BRDF or visibility and the environment map into a set of basis functions [Kautz *et al.* 2002]. This is performed by computing the scalar product of the BRDF and each basis element. Rendering is performed by computing the dot product of these coefficients. Choices of the basis functions include Spherical Harmonics (SH) [Ramamoorthi & Hanrahan 2002, Kautz *et al.* 2002, Green 2003, Kristensen *et al.* 2005, Sloan *et al.* 2002], Wavelets [Ng *et al.* 2003], Zonal harmonics [Sloan *et al.* 2005] or any other orthogonal basis.

We have chosen BRDF rendering with SH projection because it allows a relatively smooth increase of material quality when increasing the number of coefficients (i.e., number of basis functions used in the calculation). Additionally, the increase in number of

coefficients is directly related to the specularity (or glossiness) of the material: higher degree SH basis functions correspond to higher frequencies and thus well represent glossier materials or lighting.

In what follows, we will assume (θ_i, ϕ_i) to be the incoming light direction, (θ_o, ϕ_o) the outgoing view direction, Y_k the k'th basis function, f_k the projection of the function onto a SH basis function, and N the number of SH bands.

Spherical Harmonics form a basis of spherical functions. A BRDF can thus be approximated by the sum of N^2 of these function bases as:

$$f(\theta_i, \phi_i, \theta_o, \phi_o) \cos\theta_i = \sum_{k=1}^{N^2} Y_k(\theta_i, \phi_i) f_k(\theta_o, \phi_o)$$

The environment map can also be decomposed into N^2 SH :

$$L(\theta_i, \phi_i) = \sum_{k=1}^{N^2} Y_k(\theta_i, \phi_i) L_k,$$

where L_k is the coefficient of radiance for the kth basis function. Because of SH orthogonality, rendering a point x consists in computing the dot product to find the outgoing radiance $L(x)$:

$$L(x) = \sum_{k=1}^{N^2} f_k(\theta_o, \phi_o) L_k$$

Note that, following standard practice, we included the $\cos\theta_i$ term which takes into account the attenuation due to incident angle into the BRDF without loss of generality.

In practice, the coordinate system (CS) of the environment map (world CS) has to be aligned with the BRDF CS (local CS). Thus, to efficiently render the scene, we precomputed an environment map with SH rotations into a $128 * 128 * N^2$ cubemap containing multiple rotations of the environment map's SH, and a $128 * 128 * N^2$ cubemap for the BRDF containing SH for multiple outgoing directions. This method is similar to [Kautz *et al.* 2002], except for SH rotations which are all precomputed and tabulated for efficiency. We did not take visibility into account since our stimuli consisted of a single object falling with no occlusion of light coming from the environment map. Self occlusion with respect to the environment map was also negligible.

For visual rendering, previous studies [Fleming *et al.* 2003] show that using natural outdoors illumination of the object can aid in material perception. We thus chose a configuration (outdoor summer scene, with occlusion of the sky by trees) where light had relatively high frequencies to avoid impairing material appearance by having a "too diffuse" look. We acquired a High Dynamic Range (HDR) environment map and integrated the stimuli into a HDR photo consistent with the environment map, with a method similar to [Debevec 1998]. Shadows were computed with a Variance Shadow Map [Donnelly & Lauritzen 2006].

We also chose glossy materials to be able to get a sufficient number of levels of detail when increasing the number of SH basis functions. Lambertian surfaces are already very well approximated with 3 SH bands [Ramamoorthi & Hanrahan 2001a, Ramamoorthi & Hanrahan 2001b].

Rendering was performed using deferred shading to floating point render targets (High Dynamic Range rendering) and Reinhard et al's global tonemapping operator [Reinhard *et al.* 2002] was applied to account for low dynamic light intensity range of monitors and human eye sensitivity. Interactive rendering (29 frames per second (fps)) was achieved for up to 12 SH bands.

The visual rendering time was kept constant between LODs by adding idle loops in order to slow down low visual LODs to avoid subjects being disturbed by varying framerate.

4.2.2.2 Sound LOD

Contact sounds of rigid objects can be realistically generated in several ways. The context of interactive audio-visual rendering precludes the use of physical models such as the one used in [McAdams *et al.* 2004] applicable for bars only or recorded sounds in [Giordano & McAdams 2006]. In contrast, as we have seen in previous chapters, tetrahedral finite elements methods provide an accurate simulation of object deformations [O'Brien *et al.* 2002], for complex object shapes such as those used in computer games. The method is used to solve the linear elasticity problem of objects of general shapes, under small deformations (Hooke's law) which is suitable for vibrating objects. This approach results in a set of vibrational modes which are excited with a force at each contact. Each mode results in an audio stream, given as a sine wave of the modes' frequency modulated by an exponential decay and a constant amplitude. In this way, computing a contact sound $s(t)$ over time t consists in computing a sum of N modes:

$$s(t) = \sum_{n=1}^{N} a_n e^{-\alpha_n t} \sin(\omega_n t)$$

where a_n is the mode amplitude which is computed in realtime, α_n is the decay (in seconds^{-1}) which indicates how long the sound of mode will last, and ω_n is the frequency (in radians per second). The simulation was performed using filled solid objects, and sound radiation amplitudes of each mode were estimated using a far-field radiation model (see Eq. 15 in [James *et al.* 2006]).

Varying the sound LOD consists in varying the number of excited modes N, or *mode culling* [Raghuvanshi & Lin 2006]. In our case, we order modes by energy similarly to Chapter 3. We found that this ordering provided good quality sounds, in particular when small numbers of modes are used. A pilot experiment was performed for a given set of mode budgets, and the best sounding values were selected. This pilot experiment also guided our choice of sorting by energy compared to other possible orderings (e.g., by amplitude [van den Doel *et al.* 2002]).

4.2.2.3 Comparison of audio and visual stimuli

Both SH and Modal synthesis refer to a projection of a scalar field (the directional reflectance, the incident radiance and the displacement of each node of the tetrahedral mesh) into a set of functional basis. The common point of these bases is that they both refer to the eigenvalues of a Laplacian operator either over a sphere (which gives Spherical Harmonics)

or over the mesh (which gives vibrational modes). Thus, they both lead to the same type of reconstruction errors: fewer basis functions results in a smoother reconstructed function, if basis functions are sorted by their frequency (mode frequency or SH band). The combined choice of these two methods for audio and visual is thus consistent.

4.2.2.4 Objects, materials and LOD choices

Shapes were carefully chosen to facilitate material recognition. We use two objects identified by the study of [Vangorp *et al.* 2007]: the Bunny and the Dragon (see Fig. 4.1, 4.2). According to this study, both of these shapes convey accurate perception of the material of the objects.

We further make use of the study proposed by [Giordano & McAdams 2006] (see section 1.2.3) to determine the materials used in our experiment. In particular, we used Gold (similar to steel in the steel/glass material category of that study) and plastic (similar to plexiglass in the plexiglass/wood category).

Visual rendering was performed using measured BRDFs “gold-metallic-paint3" and “specular-green-phenolic" from the database of [Matusik *et al.* 2003].

We selected five different levels of visual quality, and five levels of sound quality. They were chosen so that perceptual degradations were as close as possible to uniformly distributed; given the discrete nature of the BRDF LOD, the choices were however very limited. The validity of the LOD choices is discussed in section 4.1. Some of the visual stimuli can be seen in the first two rows of Fig. 4.1 and 4.2. We did try to find a smoother, perceptually-based way of automatically choosing these LOD, but given that stimuli are animated sequences, and the limited choice of detail levels mentioned above, this was not possible.

The LOD used in the experiment correspond to budgets given in table 4.3. Budgets represent the number of modes mixed for sound, or the number of SH bands for graphics.

Given the interactive rendering context, the highest quality or “reference" solution is still an approximation. To verify how far these are from the “ground truth" we computed static offline references by sampling the rendering integral over the environment map. The use of SH stored in a spherical parametrization leads to distortion near the singularity, and given that we can handle at most 12 bands in realtime, the reference renderings are not exactly the same as the 12 bands stimuli which serve as references in the experiment. No particular treatment has been applied to limit ringing since it would result in a reduction of high frequencies (low pass filtering) which is undesirable for glossy materials. As noted by [Kautz *et al.* 2002], ringing is also masked by bumpy complex models.

Overall, as can be seen by comparing the middle and last rows of Fig. 4.1 and 4.2, the differences between our highest quality interactive rendering and the offline reference are overall acceptable, although our rendering has more contrast due to the approximations described above. This comparison is provided to give some evidence that the highest quality interactive rendering is close to the true offline reference. The use of more Spherical Harmonic bands slows down the visual rendering and results in desynchronization between audio and visuals. Since we target interactive applications (see Sec.4.4.4), we decided to only use interactively rendered stimuli for testing. However, since we do not show a true

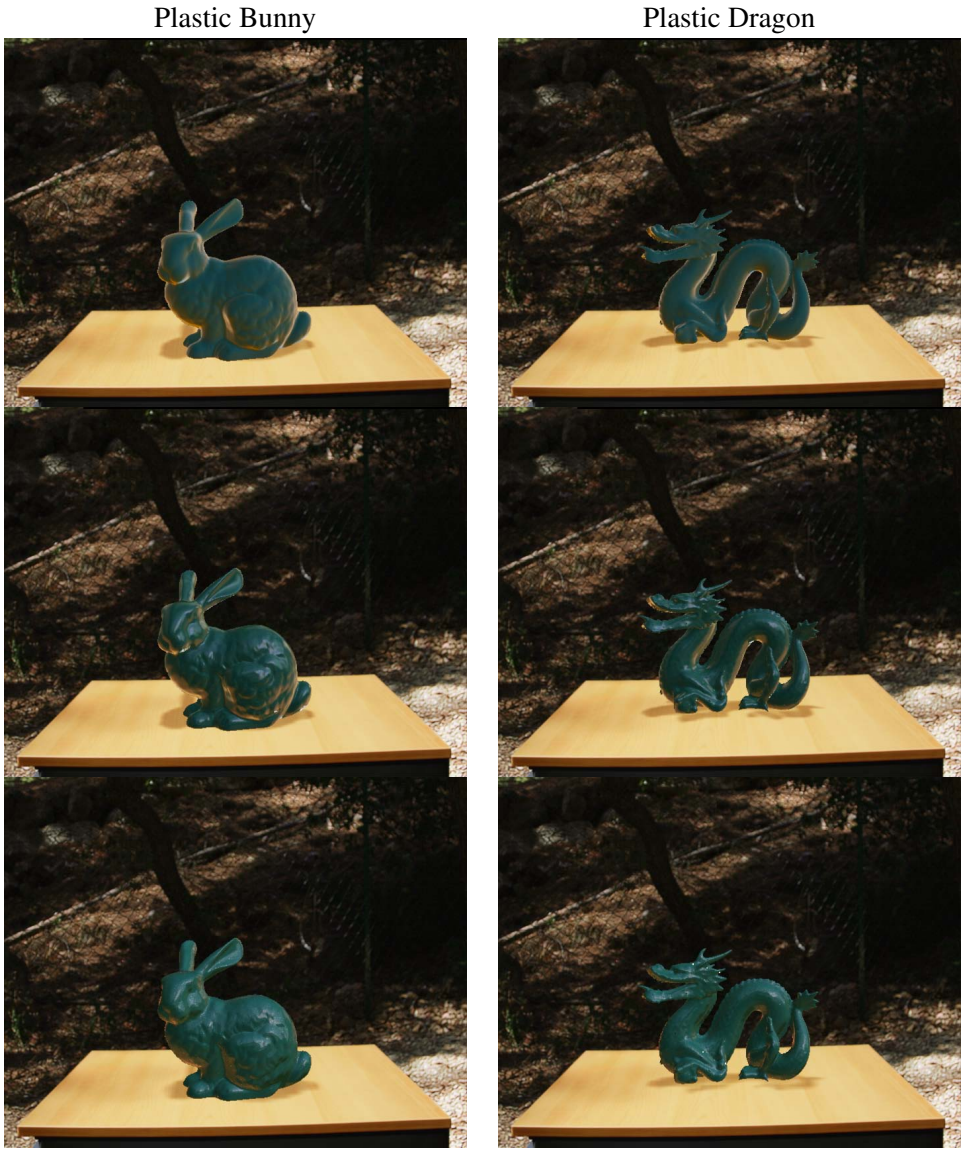


Figure 4.1: *First row: lowest visual LOD. Second row: highest visual LOD. Both were rendered interactively in the experiment. The last row shows the offline reference rendering of each object for Plastic.*

reference to participants, we believe that our current highest quality renderings convey a plausible representation of plastic and gold materials. Consequently, we believe that higher quality interactive renderings should not impact the results of our present study.

4.2.3 Procedure

In each trial, two sequences are shown to the participant. In each sequence an object falls onto a table and bounces twice, producing audible bounce sounds. One of the two se-

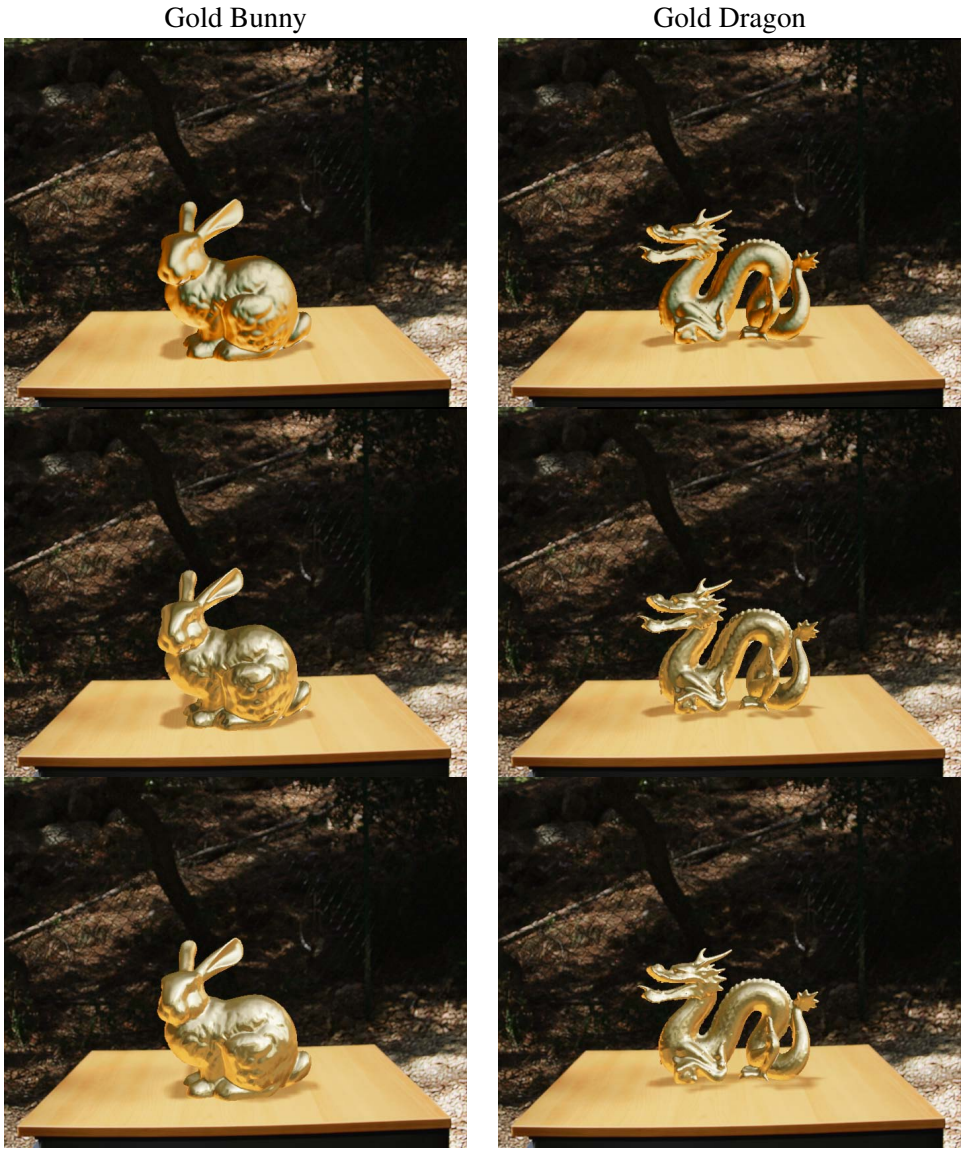


Figure 4.2: *First row: lowest visual LOD. Second row: highest visual LOD. Both were rendered interactively in the experiment. The last row shows the offline reference rendering of each object for Gold.*

quences is a reference (i.e., highest quality) rendering both in audio and graphics. The participant is unaware of this. The object falls for 0.5 seconds, while the total length of the sound varies from 0.5 and 1.5 seconds starting at the time of the impact. The duration of the sound is of course shorter for plastic and longer for gold, and also depends on the object shape. Participants were asked to rate on a scale from 0 to 100 the perceived similarity of the materials of the two falling objects “A" and “B". Since the subjects rate similarity to a high-quality reference rendering of the material, this can also be seen as an implicit ma-

	Bunny				Dragon			
	Gold		Plastic		Gold		Plastic	
LOD	BRDF	Sound	BRDF	Sound	BRDF	Sound	BRDF	Sound
1	3	8	2	4	3	8	2	17
2	4	20	3	23	4	26	3	34
3	5	28	4	34	5	39	4	62
4	9	81	7	58	9	109	7	103
5	12	409	12	233	12	439	12	346

Figure 4.3: *LOD used for the experiment. BRDF represents the number of SH bands, while Sound represent the number of modes.*

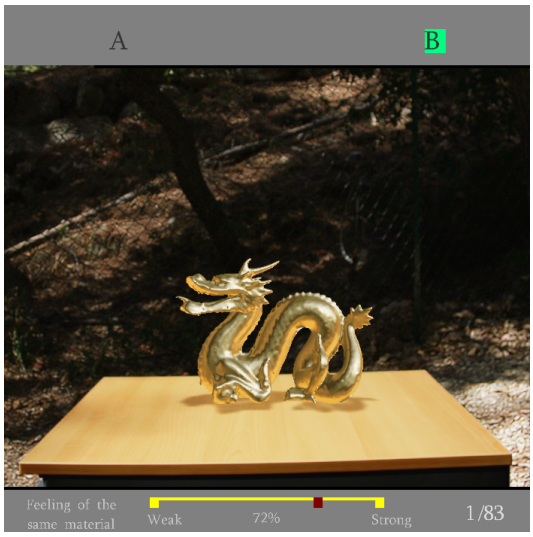


Figure 4.4: *Screenshot of the user interface. Note that the 83 trials indicated in the lower right corner include 8 training trials.*

terial quality test. This point will be discussed in section 4.4.4. Printed instructions were given before starting the experiment. An initial pair was presented separately to the participant showing the worst quality audio *and* visual with the highest quality audio *and* visual. Participants were told that this pair should be considered as the most different pair and they were explicitly told that this pair represented the lowest score (i.e., the "weakest feeling of the same material"). Participants were asked to attend to both modalities simultaneously. They were also asked not to pay attention to the shadows and the motion of the object itself. Each trial was completed in 8 seconds on average. A short training was performed at the beginning of the experiment, consisting in 8 trials.

The experiment is naturally divided into four blocks based on the combination of *Object* and *Material*. Participants performed the blocks in counterbalanced order. For each of these blocks, two main parameters vary: Sound LOD (the quality, or LOD of the con-

tact sounds of the falling objects was controlled by varying the number of modes used for modal synthesis) and BRDF LOD (the quality of the material rendering was controlled by varying the number of SH coefficients used for each LOD). Each trial was repeated three times. The order of the stimuli presentation was pseudo randomized. For each block and each trial we measure the similarity rating.

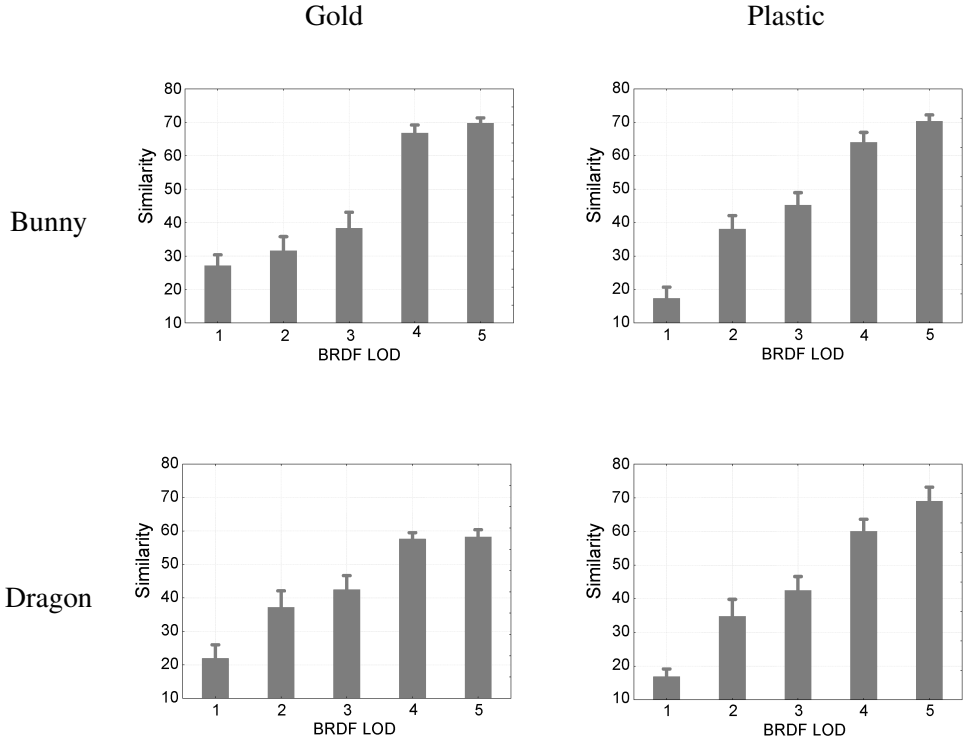


Figure 4.5: *Average mean ratings of material similarity depending on visual LOD for each object and material. Error bars represent the Standard Error of Mean (SEM). The similarity rating increased with the number of SH bands for graphics. This result confirms that visual LOD were well chosen for each object and material.*

4.2.4 Apparatus

Audio was rendered on headphones and spatialized with stereo panning in front of the participant. The visual algorithms were implemented in a game-oriented rendering engine (Ogre3D), with a high quality graphics card (GeForce 8800GTX). Screen resolution was 1600x1200 on a 20.1 inch screen (DELL 2007FP), and the rendering ran at about 29 fps in a 700x700 screen (700x561 being devoted to the stimuli, the rest for the interface, see Fig. 4.4). Responses were given on a standard keyboard. Two keys were selected to switch between stimuli with the letters “A" and “B" being highlighted respectively on the top left or right of the interface (Fig. 4.4). Rating was performed on a finely discretized scale from

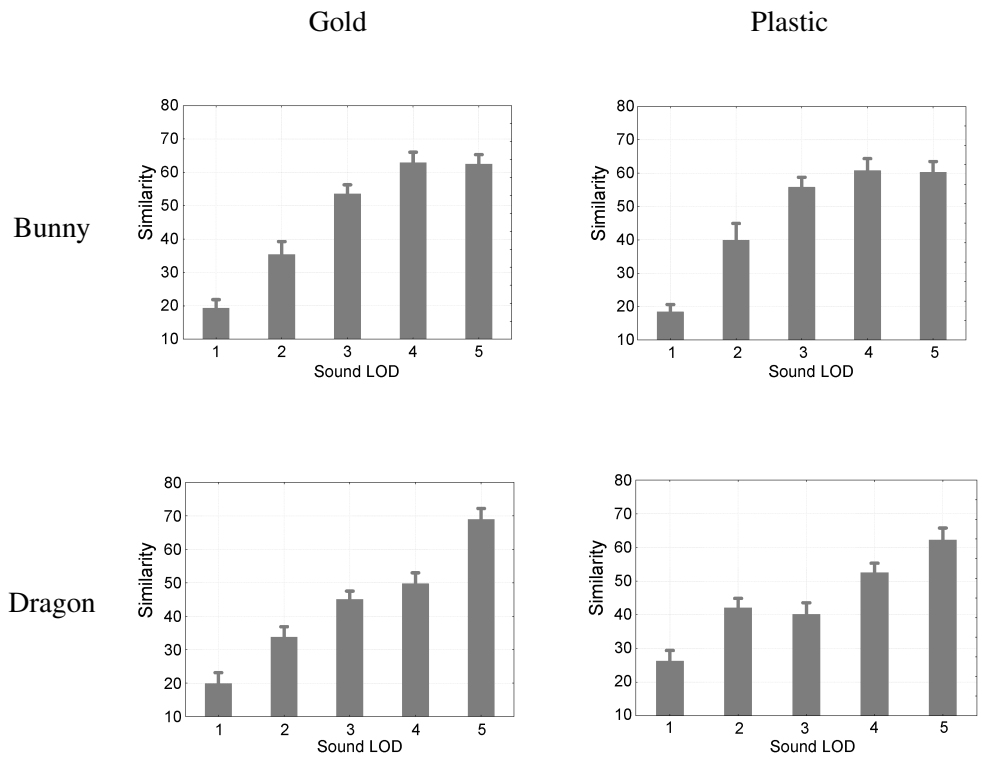


Figure 4.6: *Mean ratings and SEM of material differences depending on sound LOD for each object and material. When increasing the number of modes for the sound, we indeed observe a better perception of material quality.*

0 to 100: the cursor could be moved by 0.5% using the left and right arrows. The Return key was used to validate the choice and go to the next trial. Ratings were recorded, as well as the number of times each of the two stimuli was played (hidden reference and degraded LOD - see Procedure, Sec. 4.2.3).

4.3 Results

4.3.1 Similarity ratings

For each experimental block we performed a 2-way-repeated-measure analysis of variance on the similarity ratings. The two independent factors considered were BRDF LOD and Sound LOD (each with five levels). To account for violations of the sphericity assumption, p-values were adjusted using the Huynh-Feldt correction. $p < 0.05$ was considered to be statistically significant.

A first global analysis (a repeated-measures ANOVA with Object, Material, BRDF LOD, Sound LOD and Repetition as within-subjects factors) showed that there was no significant main effect of Repetition, and no significant interaction between Repetition and

other factors. This result indicates that participants performed the task well and were stable in their judgment across one experimental block. We thus performed all furthering analysis on the mean rating over the three repetitions.

4.3.1.1 BRDF and Sound

For each material and object, the ANOVA revealed a significant main effect of BRDF LOD (Gold Bunny: $F_{4,36} = 72.03$; $\varepsilon = 0.87$; $p < 0.0001$, Plastic Bunny: $F_{4,36} = 128.85$; $\varepsilon = 0.79$; $p < 0.0001$, Gold Dragon: $F_{4,36} = 22.40$; $\varepsilon = 0.52$; $p < 0.0001$, Plastic Dragon: $F_{4,36} = 38.21$; $\varepsilon = 0.53$; $p < 0.0001$). These results show that an increase in the quality of the BRDF gives improved ratings of similarity with the reference (see Fig. 4.5).

Similarly, the ANOVA revealed a significant main effect of Sound LOD (Gold Bunny: $F_{4,36} = 144.82$; $\varepsilon = 0.59$; $p < 0.0001$, Plastic Bunny: $F_{4,36} = 55.80$; $\varepsilon = 0.83$; $p < 0.0001$, Gold Dragon: $F_{4,36} = 62.95$; $\varepsilon = 0.54$; $p < 0.0001$, Plastic Dragon: $F_{4,36} = 44.95$; $\varepsilon = 0.52$; $p < 0.0001$). In a manner similar to BRDF LOD, increasing the LOD of the modal synthesis improves the similarity rating with respect to the reference (see Fig. 4.6).

We performed post-hoc analyses (Bonferroni) comparing all pairs of BRDF LOD and all pairs of Sound LOD. All were significantly different ($p < 0.02$) except for the following cases: for the Gold Bunny, BRDF LOD 1 and 2 ($p = 1.0$) and BRDF 2 and 3 ($p = 0.51$), and for its sound LOD, levels 4 and 5 ($p = 1.0$); for the Plastic Bunny, BRDF LOD 2 and 3 ($p = 0.10$), and 4 and 5 ($p = 0.22$) and for its sound LOD, levels 3 and 4 ($p = 1.0$), 3 and 5 ($p = 1.0$) and 4 and 5 ($p = 1.0$); for the Gold Dragon, BRDF LOD 2 and 3 ($p = 1.0$) and 4 and 5 ($p = 1.0$), and for its sound LOD, levels 3 and 4 ($p = 1.0$); for the Plastic Dragon, BRDF 2 and 3 ($p = 1.0$) and 4 and 5 ($p = 1.0$), and for its sound LOD, levels 2 and 3 ($p = 1.0$).

4.3.1.2 Interaction between BRDF and Sound

The most interesting aspect for this work is the *interaction* between *BRDF LOD* and *Sound LOD*. The ANOVAs also revealed that for each material and object, a significant interaction between BRDF LOD and sound LOD exists (Gold Bunny: $F_{16,144} = 14.95$; $\varepsilon = 0.32$; $p < 0.0001$, Plastic Bunny: $F_{16,144} = 17.11$; $\varepsilon = 0.45$; $p < 0.0001$, Gold Dragon: $F_{16,144} = 9.14$; $\varepsilon = 0.32$; $p < 0.0001$, Plastic Dragon: $F_{16,144} = 5.12$; $\varepsilon = 0.70$; $p < 0.0001$); see Fig. 4.7. This indicates that the quality of sound and the quality of BRDF rendering mutually interact on the judgment of similarity and that the final material quality judgement is not simply a combined weighting of visual and audio quality.

4.3.1.3 Other Interactions

As discussed earlier (Sect. 4.2.2.4), *BRDF LOD* and *Sound LOD* did not take the same values between the two different materials and the two objects. As a consequence, the previous ANOVAs were conducted on each material and object to identify the effect of these specific LOD on similarity ratings.

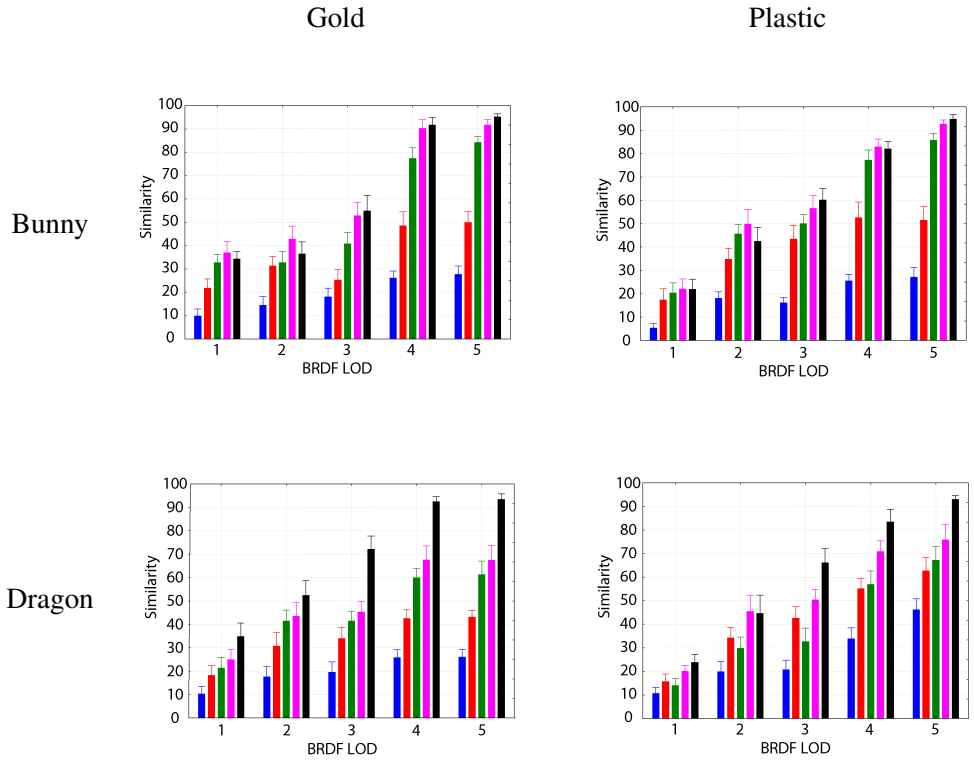


Figure 4.7: *Interaction between BRDF and sound: Mean similarity ratings and SEM of the different BRDF LOD and Sound LOD, for the two different objects and the two different materials. Blue, red, green, pink and black bars represent increasing sound LOD while the main horizontal axis represents increasing BRDF LOD. When varying sound quality, greater perceived differences can be seen at high BRDF quality rather than at low BRDF quality.*

We are also interested in exploring the potential differences between the two objects or the two materials in more detail. To do this, a repeated-measures ANOVA including *Object, Material, BRDF LOD* and *Sound LOD* as within-subjects factors was also performed. This ANOVA revealed, as could be expected from the preceding analysis, a main effect of *BRDF LOD* ($F_{4,36} = 89.57$; $\varepsilon = 0.56$; $p < 0.0001$), *Sound LOD* ($F_{4,36} = 154.34$; $\varepsilon = 0.70$; $p < 0.0001$), and an interaction effect between *BRDF LOD* and *Sound LOD* ($F_{16,144} = 29.46$; $\varepsilon = 0.44$; $p < 0.0001$). It also revealed an interaction effect between *Material* and *BRDF LOD* ($F_{4,36} = 6.53$; $\varepsilon = 0.61$; $p < 0.0005$), (see Fig. 4.8), *Material* and *Sound LOD* ($F_{4,36} = 4.86$; $\varepsilon = 0.71$; $p < 0.003$), and *Object* and *Sound LOD* ($F_{4,36} = 14.75$; $\varepsilon = 0.90$; $p < 0.0001$). No significant main effects of *Material* or *Object* were shown.

Interaction between *Material* and *Sound LOD* as well as *Object* and *Sound* was due to very small differences between similarity ratings for Gold or Plastic.

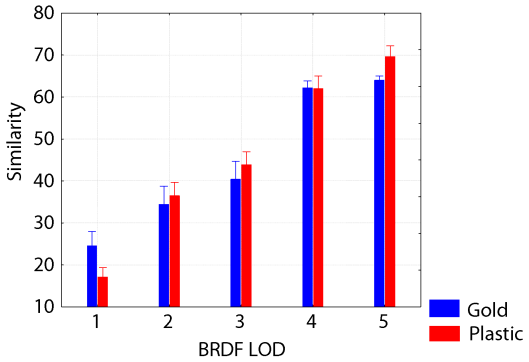


Figure 4.8: *Interaction between BRDF and Material. Mean similarity ratings and SEM of the BRDF LOD per material. We were obliged to choose different BRDF LODs for each material to accommodate the differences of the perceived materials. Given the discrete nature of the BRDF LODs we see that the choice of LOD results in similar similarity ratings.*

4.4 Discussion

This experiment demonstrated an interaction between *BRDF* and *Sound LOD*, which has significant algorithmic consequences. We also discuss the validity of stimuli and some potential avenues for generalization of this work.

4.4.1 Stimuli Validation

Both visual and audio LOD have been manually chosen in order to be as perceptually uniform as possible, when taken independently, with careful inspection. This section describes a validation of this manual choice.

A first validation of the choice of stimuli can be performed by observing perceived material similarity with the reference for increasing *BRDF LOD* alone and increasing *Sound LOD* alone; this is shown in Figures 4.5 and 4.6. With the exception of sound levels 2 to 3 for the Plastic Dragon, material quality was overall rated as increasing when the quality of one modality alone increases. This is a strong indication that the choice of stimuli (see Sect. 4.2.2) is valid and allows us to have confidence in our results. However, although increasing, material quality does not seem to be perceived as linearly correlated with the sound LOD for the Bunny (Figure 4.6), for the highest two LODs. This may suggest that differences in the last two audio LODs that have been perceived in the manual selection phase have not been detected by the participants, or that the perceived difference does not impact the perceived quality of the material. In terms of material perception, the audio LODs do not reach the same uniformity as in the manual selection process which only took into account audio quality without any material consideration. This result could indi-

cate that the number of modes can be reduced significantly without affecting the perceived quality of the material, although possibly affecting the perceived quality of the sound itself.

Although there was no significant difference between *all* LOD for graphics as well as for sound, there is a clear tendency (see Fig. 4.5 and Fig. 4.6): similarity increases as LOD increases. This confirms that our choice for each LOD was reasonable.

As noted previously, we chose different LOD for the different materials. In Fig.4.3 we see the different choices for visual and audio LOD for each of the two materials, and in Fig.4.8 their respective ratings. If we had chosen the same visual LOD for both Gold and Plastic, ratings would significantly differ from Gold to Plastic. Recall that the choice of SH bands is discrete, and thus no intermediate choice was possible.

Figure 4.9: *The Gold Dragon with the third visual quality (A) and the best visual quality (B - visual hidden reference). When using the highest audio quality the approximation A was rated as being very similar to the hidden audiovisual reference (72 on a scale of 100). In contrast, the best visual quality (B) when seen with Sound LOD equal to 3 (intermediate), was actually judged to be* less similar *(61 on a scale of 100) than the audiovisual reference.*

4.4.2 BRDF SH Rendering

Another interesting observation is the lack of perceived differences between BRDF LOD 4 and 5 (see Fig. 4.5). This means that we could easily render 9 SH bands (i.e., 81 coefficients) for Gold or 7 (i.e., 49 coefficients) for Plastic instead of 12 bands (144 coefficients) without perceivable difference in material similarity. For comparison, the rendering time for 12 bands is 17.8ms (without additional cost), whereas it is 6.6ms for 9 bands and 1.2ms for 7 bands. Previous work on Spherical Harmonics lighting observes that 3 SH bands were enough to render Lambertian surfaces ([Ramamoorthi & Hanrahan 2001a, Ramamoorthi & Hanrahan 2001b]) given the fast decay of SH coefficients for the Lambertian term. In our context, our experimentation indicates that 7 (plastic) or 9 (gold) bands could be enough to render glossy materials like metallic BRDFs, giving the impression of an unchanged material compared to the reference. However, it does not preclude visible pixel-per-pixel differences.

4.4.3 Interaction between Sound and Visual Quality

The most interesting result is the interaction between BRDF and sound LOD in perceived quality. If we interpret similarity to the reference as a measure of quality, we see that, for the same BRDF LOD, material quality is judged to be higher when the sound LOD is higher. This can be seen in the different-colored bars for each *BRDF LOD* in Fig. 4.7.

As an example consider the BRDF LOD 3 with the Sound LOD 5 in all objects and materials (see Fig. 4.7). Material similarity compared to the reference is overall rated at the same level as the BRDF LOD 5 with Sound LOD 2. This clearly indicates that the sound LOD can perceptively counteract a low number of SH bands (see also for example Fig. 4.9).

This result is important since the cost of rendering better quality sound is typically much lower than the cost of better quality BRDF rendering. This is because, computing the dot product for BRDF rendering requires $\mathcal{O}(N^2P)$ operations where N is the number of SH bands (representing N^2 SH basis functions) and P is the number of pixels drawn on screen, whereas the sound requires $\mathcal{O}(M)$ operations where M is the number of modes. Considering the quadratic increase in computational cost for BRDF rendering compared to the linear cost in modal sound rendering, it is more beneficial to reduce graphics quality while increasing audio quality for the same global perceived material difference to the reference.

To get a feeling for the practical implications of this result, the computation time of the third sound LOD is about 0.21ms and for the highest quality 1.95ms. For BRDFs, the computation time (performed on GPU) for the third quality is about 0.5ms (with an additional 16.7ms of constant cost for soft shadows, deferred shading pass and rotations) whereas it is 17.8ms (in addition to the 16.7ms constant cost) for the highest BRDF quality. In this particular case, we have a gain of 15.56ms per frame if we choose our BRDF and sound LOD based on the results of our study. Another way to see this is that the frame rate (assuming this BRDF rendering to be the only cost), would increase from 30fps to 60fps. This gives a very strong indication of the utility of our results.

Besides the very promising algorithmic consequences of our findings, we believe that the actual effect of audio-visual interaction on material perception we have shown could be very promising in a more general setting. Given the interactive rendering context of our work, and the consequent constraints, we were in some ways limited (discrete levels of detail, some parameters which are only loosely related to physical quantities etc). Nonetheless, to our knowledge this is the first study which shows interaction of audio and graphics in a material perception task. We thus are hopeful that our finding will be a starting point for more general perceptual research, in which the constraints of interactive rendering will not be required. This could allow the use of parameters such as decay for sound synthesis or a continuous visual level of detail parameter, and lead to wider, more perceptually-motivated results.

4.4.4 Algorithmic Generalization

Evidently, our study is only a first step in determining the combined influence of sound and visual rendering quality on perceiving material similarity, and in particular similarity to a "gold standard" reference. Our study only examined a limited setting with two objects and two materials, although the choice of materials corresponds to hopefully representative classes of material properties. Our approach is thus still limited to two dimensions, evaluating the perceived quality based on 2 parameters: BRDF and audio LOD, with a sparse sampling of geometries (dragon and bunny) and a sparse sampling of materials (gold and plastic). In future work, it would be interesting future to sample the space of geometries and materials more accurately, and determine a fully general four dimensional mapping of the perceived quality depending on BRDF, audio, geometry and material properties.

Clearly the similarity to best-quality is not a direct quality judgment. However, in the context of interactive rendering for example, the end result is essentially the same. If we assume that the best-quality result is sufficient for a given interactive application, clearly it is beneficial to spend significantly lower computational resources with the same perceived result. On the other hand, under no circumstances should our results be taken as an accurate predictor of quality for all conditions, for example if users carefully examine a video sequence to detect image or audio artifacts.

More extensive studies of different material types and object geometries should be undertaken, including more objects made of several different materials. Material and geometry, together with illumination, were the parameters used to study *visual equivalence* in computer graphics [Ramanarayanan *et al.* 2007]; an interesting extension of our study could be to use a similar set of values. We believe that extensions to our results could have a significant potential and utility in an algorithmic context, when managing audio-visual rendering budgets with a global approach. For example, we have applied our results in an algorithmic context in [Grelaud *et al.* 2009].

4.5 Conclusion

Our goal was to determine whether the combined quality levels of visual and sound rendering influence the perception of material, and in particular in the context of interactive systems. The constraints of interactive rendering led us to choose Spherical Harmonic-based levels of detail for BRDF rendering, and a mode-culling contact sound synthesis approach. We designed a study in which subjects compare the similarity of interactive sequences with a given audio-visual reference (i.e., high-quality sound and graphics).

The results of our study show that, for the cases we examined, better quality sound improves the perceived similarity of a lower-quality visual approximation to the reference. This result has direct applicability in rendering systems as we will show in Chapter 5, since increasing the visual level of detail is often much more costly than increasing the audio level of detail. The examples provided show potential for significant computation time savings, for the same, or even better perceived material quality.

To our knowledge, our study is the first to demonstrate interaction between audio and graphics in a task related to perception of glossy materials. The use of lower visual quality

stimuli would result in a more diffuse aspect, similar to matte plastic. Our findings in terms of material perception are thus conditioned by the choice of our stimuli. This was validated in section 4.4.1. Given our motivation for interactive audio-visual rendering, we were necessarily constrained in our choices of stimuli and the extent of our setup. Nonetheless, we are hopeful that our initial study, which indicates the existence of a potentially cross-modal audiovisual effect on material recognition, will inspire more perceptually oriented studies in a more general context.

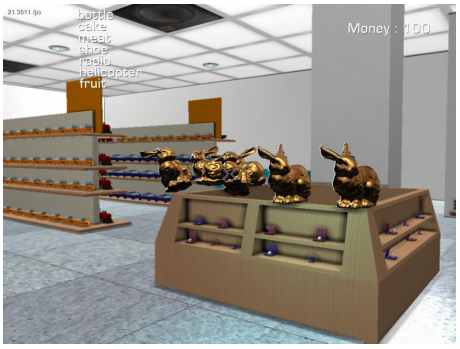
In the following chapter, we will see how this experimental study can result in a computational benefit. In particular, we can automatically choose the tradeoff between audio and visual quality of a material in a realtime rendering context in order to obtain the highest perceived material quality with a constraint on the audiovisual computational rendering cost.

CHAPTER 5

Efficient and Practical Audio-Visual Rendering for Games using Crossmodal Perception

Contents

Using our approach, we can render sounds for larger number of impact sounds in real time.



We use crossmodal perception to jointly select the level of detail for sound and graphics.

Figure 5.1: *Illustration of our results.*

The contributions in this chapter have been published at the *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* [Grelaud *et al.* 2009].

5.1 Introduction

To conclude the first part of this thesis, we present a unified pipeline for crossmodal algorithms.

Our previous use of synthesized impact sounds in a scalable pipeline (see Chapter 3) provides an efficient way to generate impact sounds produced by physical events. Also, we used a crossmodal scheduling algorithm based on our perception of asynchrony to delay the introduction of new impact sounds when the processor is overloaded (see Chapter 3, Section 3.5), and presented a way to take into account audiovisual spatial tolerance in a clustering algorithm (see Chapter 2).

However, the combination of impact and pre-recorded sounds involves a non-negligible overhead due to energy computation, thus mitigating the potential gains from using the techniques of audio clustering, masking and scalable processing developed in [Tsingos *et al.* 2004, Moeck *et al.* 2007]. Indeed, masking effects potentially allow the culling of large fraction of sound sources which will not be computed. In the case of impact sounds which are costly to compute, such masking would allow a large speed-up.

While crossmodal perception has been successfully used to improve audio rendering, to our knowledge there is no previous method which uses crossmodal perception simultaneously for graphics and audio rendering, and in particular, level-of-detail selection. This could be beneficial since our experience of virtual environments is inherently bimodal, particularly if there is mutual influence of sound and visuals on perception which we can exploit algorithmically. We reported such mutual influence for our perception of materials in Chapter 4.

In this chapter we address the two issues discussed above: We present a new fast approximation to energy estimation for impact sounds, allowing us to fully exploit the benefits of the scalable processing pipeline of [Moeck *et al.* 2007]. We also introduce a joint audio-visual level-of-detail selection algorithm based on our study on material perception (see Chapter 4). We also show how to integrate high-quality "attacks" (e.g., the onset of an impact sound) into the full audio processing pipeline, and present a full crossmodal audio-visual processing pipeline which is suitable for games. We have integrated our results in an experimental, but quite complete, game engine [Chiu 2008], putting together all of these contributions and showing their applicability to computer games.

5.2 Efficient Energy Computation for Impact Sounds

The use of a combined audio rendering pipeline for both recorded and impact sounds has a high potential for benefit, since we get significant speed benefits using masking and clustering [Tsingos *et al.* 2004], and we have a smooth quality/cost tradeoff using scalable processing [Moeck *et al.* 2007]. An initial approach integrating this pipeline with impact sounds was presented in Chapter 3. However, the full potential benefit of this combined approach is hindered by the relatively high cost of the computation of impact sounds energy, which is required both for masking and scalable processing.

For recorded sounds, the energy is precomputed for each sound file to avoid on the

fly computations. In the case of impact sounds, this energy cannot be precomputed since sounds are generated on-the-fly during the simulation. Online energy computation based on the values in the current audio buffer would be costly and pointless, since the goal is to avoid computing the audio signal if it is masked by other sounds. We thus require a quick estimate of energy without actually computing the impact sound.

5.2.1 Energy Computations for Masking and Scalable Processing

We first briefly summarize the energy computations required to use impact sounds with the combined pipeline. Computations occur at two instants in time: at *impact* and at *each frame*. In Chapter 3, at each impact the total energy of each mode was efficiently computed. At each frame, the energy of the sound over the frame was then estimated using scalar products of a subset of the modes of the sound. This approximation was shown to work well for impact sound processing but still required much computational effort per frame.

The solution we show here, together with the integration of attack processing in clustering (Sect. 5.4) allows the full and efficient integration of high-quality impact sounds into a practical audio processing pipeline.

5.2.2 An Efficient Energy Approximation for Impact Sounds

We assume that the power (energy per unit time, that we will call "instant energy") of an impact sound decreases exponentially:

$$E(t) = Ae^{-\alpha t} \tag{5.1}$$

Thus if we know the parameters A and α of this exponential, we can easily compute an approximation of the energy in a given frame, by analytically integrating over the desired interval. The two unknown parameters A and α satisfy two equations concerning the energy:

$$E_{Tot} = \int_0^{\infty} Ae^{-\alpha t} dt \tag{5.2}$$

$$E_{Part} = \int_0^{T} Ae^{-\alpha t} dt \tag{5.3}$$

Thus, given the total energy E_{Tot} of the sound and a partial energy E_{Part}, we are able to exactly determine parameters A and α.

These parameters are thus given by the following equations:

$$\alpha = -\frac{1}{T}\log(1 - \frac{E_{Part}}{E_{Tot}}) \tag{5.4}$$

$$A = \alpha\, E_{Tot} \tag{5.5}$$

The energy E_s for each frame is computed by integrating Eq. 5.1, which represents a negligible computation cost per frame:

$$E_s = -\frac{1}{\alpha}.(E(t + \Delta t) - E(t)) \tag{5.6}$$

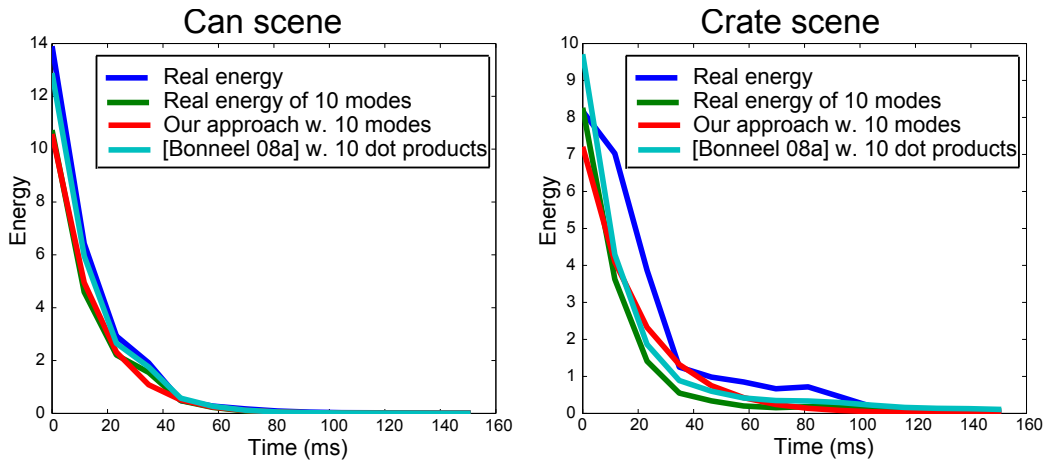


Figure 5.2: *Plot of the instant energies computed with our approach (red), our previous energy computation (Chapter 3) (cyan), and the reference (blue), over the length of an impact sound on two scenes. Also shown the reference energy computed with 10 modes, simulating what our approach computes. Note that our approximation is much more efficient. Left: "Cans" sequence, right "Crates" sequence.*

However, computing these values require the knowledge of a partial energy E_{Part} for the system to be solved. This can be achieved efficiently, also by computing scalar products. We found that the scalar product S of two modes m_1 and m_2 taken from 0 to T can be easily computed via the expression of the scalar product Q of those two modes taken from 0 to infinity:

$$\begin{aligned} S &= < m_1, m_2 > \\ &= \int_0^T e^{-a_1 t} \sin(\omega_1 t) e^{-a_2 t} \sin(\omega_2 t) dt \\ &= (1 - e^{-T(a_1+a_2)}) Q \end{aligned} \tag{5.7}$$

with

$$\begin{aligned} Q &= \int_0^\infty e^{-a_1 t} \sin(\omega_1 t) e^{-a_2 t} \sin(\omega_2 t) dt \\ &= \frac{2(a_1+a_2)\omega_1\omega_2}{((a_1+a_2)^2 + (\omega_1+\omega_2)^2)((a_1+a_2)^2 + (\omega_1-\omega_2)^2)} \end{aligned} \tag{5.8}$$

The partial energy E_{Part} is computed by summing the pairwise scalar product of a subset of highest energy modes, similar to the computation of the total sound energy E_{Tot} previously used. A typical value for T is 10ms. This optimization process is performed only once per impact, and does not have to be repeated per frame. Also, since Q is already computed to get the total energy of the sound, the only additional per-impact cost is the computation of an exponential function. Only the simple Eq. (5.6) has to be computed per frame for each impact sound, which represents a negligible cost. Our new approximation thus allows faster computation of the instant sound energy which is used for masking and budget allocation.

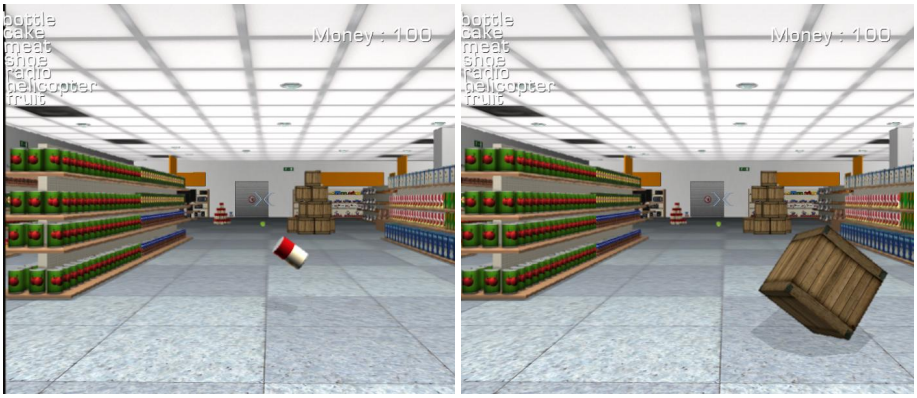


Figure 5.3: *Two frames from the test sequence used for the fast energy estimation evaluation (see text).*

5.2.3 Numerical Evaluation and Speedup

We performed a numerical evaluation of our energy estimation approach on two sequences. These scenes are respectively a can and a crate falling on the ground (see Fig. 5.3).

We computed the exact energy E_s of each impact sound in each frame using all the modes, and we plotted this compared to our approximation in Fig. 5.2, over 86 frames for both sequences. As we can see, our approximation is overall accurate, with an average L1 relative error of 24% for "Cans" and 27% for "Crates". Each can had a mesh of 415 elements, and used 113 modes; and for each crate there are 497 elements and 362 modes.

If we use the approximation given in Chapter 3, the average cost of the energy is 0.2 ms per frame for the "Cans" sequence and 0.2 ms per frame for "Crates". In contrast, our approximation requires about 1μs and 0.8μs respectively for each sequence, corresponding to a speedup of 200 and 250 times. In addition, for sounds with large numbers of elements, which are required for sounds with fast decaying, high frequency modes, a higher number of modes is required for the approximation. Given the low cost of our approach, we can thus significantly improve the energy approximation without significant overhead in this case.

5.3 Crossmodal Audio-visual LOD Selection

We develop a crossmodal audio-visual level-of-detail selection algorithm based on the perceptual study performed in Chapter 4. We use the main result of that study, concerning the mutual influence of sound and graphics on how users perceive material. While this approach only applies when objects are actually sounding, i.e., during impacts, this is when a large number of sound events occur, resulting in a significant additional load to the system. As a result, there is strong potential for gain using our selection method. Indeed, since audio rendering is much cheaper than spherical harmonic (SH) lighting, we can take advantage of the crossmodal influence of audio and visuals on the perceived quality, by reducing visual LOD and increasing audio LOD. According to the study, in this case the

perceived similarity with a reference rendering will remain very high, despite the significantly reduced computational cost.

In the experiment, the LOD used are the number of bands of the spherical harmonics for the rendering of the BRDF using an environment map, and the number of modes used for modal sound synthesis. Also, two materials were used (gold and plastic), and two different objects were studied (bunny and dragon model). For the case of the golden bunny, we can plot the results as shown in Fig. 5.4. In the context of an audio-visual 3D rendering

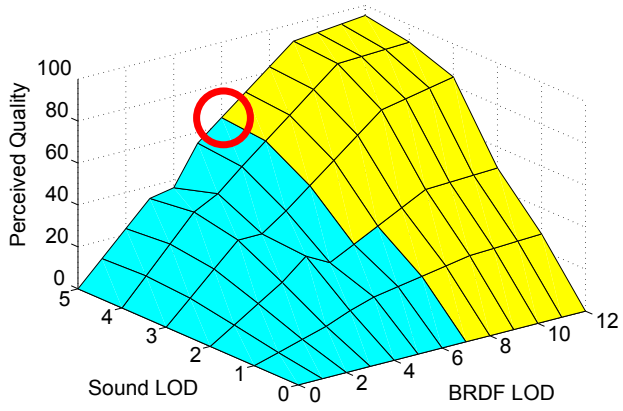


Figure 5.4: *Material rating depending on sound and visual LOD, and the 1.4ms threshold cost (threshold for P = 1000 shaded pixels – the size of the object on the screen). Golden bunny.*

engine, we can interpret and use these results in several ways. One option would be to fix the "target quality" and then minimize the cost or computation budget used to render both the graphics and the sound. A second option is to allocate a fixed budget for both graphics and sound, and maximize the perceived quality. We have chosen the latter option, since BRDF rendering with a large number of spherical harmonic bands can be very costly; in the presence of several such objects even the joint LOD with minimum cost predicted using this model could be unacceptably high.

5.3.1 Crossmodal Audio Visual LOD Metric

We perform an optimization step to determine the appropriate graphics and visual LODs once per frame. The constraint is determined by the actual CPU/GPU cost of rendering audio modes and shaded pixels. For graphics, this is the cost of the pixel shader used to query the SH textures and compute the dot product lighting.

The complexity of rendering audio modes is linear in the number of modes, while the cost of rendering the SH lighting is quadratic in the number of spherical harmonic bands,

and linear in the number of displayed pixels. We thus use the following cost estimation:

$$C_{AV} = C_M\ M + C_S\ S^2\ P, \tag{5.9}$$

where M is the number of modes, S is the number of spherical harmonic bands, P is the number of shaded pixels, and C_M and C_S are the costs of rendering respectively one audio mode and one band of one SH-shaded pixel. The values for C_M and C_S can be measured once for every hardware setup. In our case $C_M = 5.23\mu s/mode$, $C_S = 0.0368\mu s$ per SH coefficient and per pixel. Note that rendering S bands requires computing S^2 such coefficients. These values were measured with a GeForce8800 GTX, and a 2.3GHz Intel Xeon CPU. To efficiently determine the number of shaded pixels, we use an early-depth pass, deferred shading and occlusion queries. The above expression shows the quadratic increase in SH cost and the linear cost in the number of modes.

The target cost C_T is a user-defined parameter typically depending on the hardware setup. We share this parameter across all objects, i.e., if we have N objects, the budget of C_T/N ms is assigned to each object in the scene. The audio-visual perceived quality is determined by an interpolation of the tabulated values given in Chapter 4, Fig.4.7. We evaluate the cost function C_{AV} for each combination of 13 SH bands (0 to 12) and 6 mode budgets given in Chapter 4, Fig. 4.3. We choose the combination which results in the highest "perceived quality" as determined by the values reported in our previous study. As an example, for a target budget C_T/N of 1.4ms, used to render an object occupying 1000 pixels on the screen, we can visualize the operation LOD choice operation in Fig. 5.4. In this example, we will choose 6 SH bands and level 5 for the modes. We perform this operation for each object, and select the number of SH bands and the number of modes resulting in the highest quality, while respecting the target cost.

With this approach we determine a budget for impact sounds and spherical harmonics. The audio LOD is used for the current audio frame, and the spherical harmonic LOD is used in the next visual frame. Visual blending is performed in the pixel shader by progressively adding terms in the dot product (without splitting an SH band) during 0.5ms.

Evidently this LOD selection is valid only when the objects are actually making a sound, in this case when impacts occur. When the objects are not sounding, we use a fixed LOD for graphics, i.e., 5 spherical harmonic bands. The choice of 5 bands is also based on the results presented in Chapter 4, since we can see that perceived quality with this approximation is not much different from the perceived quality obtained with the highest visual LOD, for all sound LODs. The crossmodal LOD selection mechanism is applied at the first audio frame of an impact, and maintained for a short period of time (typically 7 seconds).

5.4 A General Crossmodal Audiovisual Pipeline

The fast energy estimation and the crossmodal audiovisual LOD selection pave the way for us to introduce a general crossmodal rendering pipeline, in which we use crossmodal perception for combined graphics and sound resource management.

Attack processing with Clusters In Chapter 3, special processing is performed to provide high-quality rendering of the "attacks" or onset of impact sounds. This involves a specially designed windowing operation in the first frame of each attack. This approach was however not integrated into the full pipeline.

The inclusion of high-quality attack processing in the pipeline is important to allow high-quality audio rendering. In each cluster, we have a set of recorded sounds and a set of impact sounds. Using the improved attack processing presented in Chapter 3, Section 3.3.3, we need to have one frequency domain buffer for attacks, and one for both recorded and impact sounds. Budget allocation for scalable processing is performed only in the latter buffer. Windowing operations are performed separately in each buffer, and an inverse FFT is computed for each *impact sound* buffer in each cluster. Recorded sound buffers are processed together, resulting in a single inverse FFT. Attack buffers and the recorded/impact sound buffers are then correctly windowed in the time domain to produce the final sound. The additional overhead of this approach is thus one inverse FFT per audio channel.

General Crossmodal AV Pipeline We have implemented a complete perceptually based audio-visual rendering pipeline. We have also included the crossmodal audio clustering metric (see Chapter 2), which assigns more audio clusters to the viewing frustum, the crossmodal scheduling approach presented in Chapter 3, Section 3.5, which significantly improves performance, and the audio-visual LOD selection introduced here.

At each frame, both for audio and graphics, we evaluate the parameters used for each of these crossmodal algorithms, and set the appropriate values for clustering, impact sound scheduling and AV LOD selection.

5.5 Results

We have implemented the general AV pipeline into a home-grown "production level" game engine developed in our institutions [Chiu 2008]. All the examples shown in this chapter are taken from this engine.

Energy approximation and attack integration The combination of the integration of attacks in the pipeline and the energy approximation now make it possible to have more impact sounds with higher quality in complex game-like settings. We developed an example game play (Fig.5.6), in which we have 7 recorded sounds (toys, stereos, door, cashier sounds etc.) and a large number of impact sounds, with a peak of 712 impacts just after the piles of boxes and cans are knocked over.

We first show the sequence using the method of Chapter 3[1]. The cost of the previous energy estimation method results in low quality audio processing, which is clearly audible. If we compare to a similar sequence (we do not have exact path replay in our implementation), using our energy approximation we see that the quality achievable by the scalable processing is significantly improved.

[1] See the paper's video available at http://www-sop.inria.fr/reves/Basilic/2009/GBWAD09/GBWAD09.avi

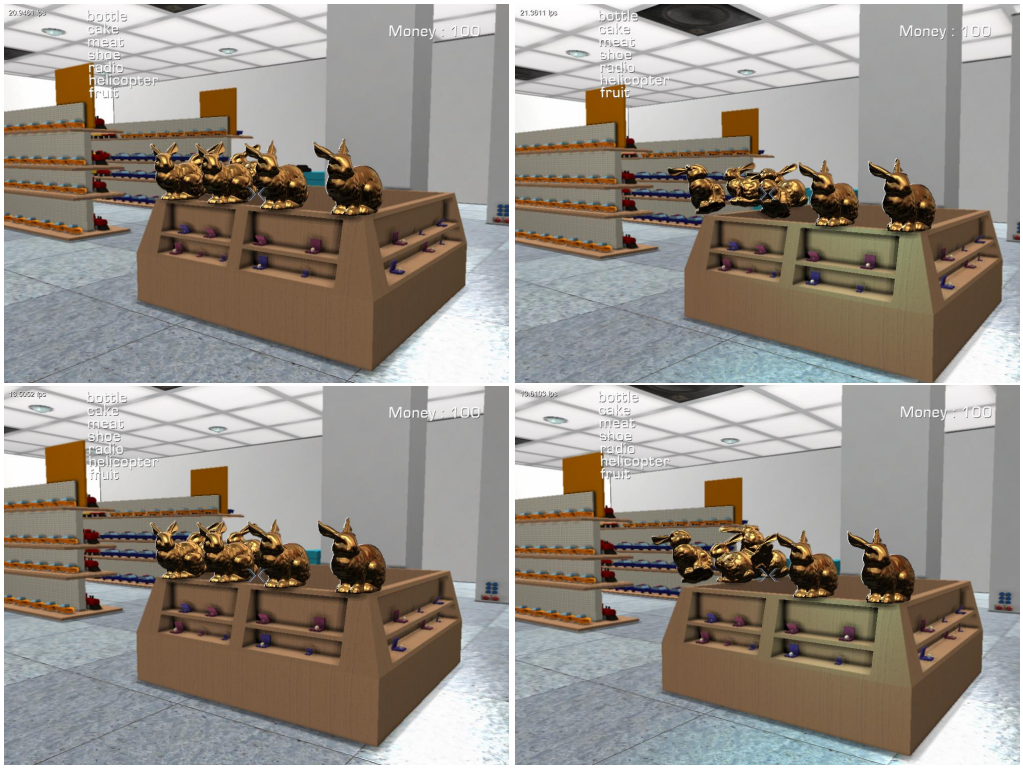


Figure 5.5: *(a)-(b) Two frames of our test sequence running at 26 fps using crossmodal LOD manager. (c)-(d) Two similar frames using highest LOD running at 18 fps.*

Figure 5.6: *A frame from our game scenario.*

We next show the same sequence first without attack processing in the clusters and using the attacks, both with the energy approximation. As we can hear, the quality of the impact sounds is audibly better using the attack processing approach; thanks to the integration with clustering, the overhead is minimal. Rendering this scene without clustering, given the number of sounds, would be impossible in real time.

We provide a detailed breakdown of the costs of the different stages for the above sequence, using both the energy approximation and the integration of attacks in clusters (see Table 5.1). In the scenes we tested, we achieve masking of around 60% of the contact sounds.

Stage	Time (μs)
Masking	3.339
Clustering	4.652
Impact sound synthesis	2.759
Energy	0.824
Scalable mixing	12.600

Table 5.1: *Time in ms for each stage of the unified pipeline*

Crossmodal Material LOD We have implemented our crossmodal LOD selection approach in the same game engine [Chiu 2008]. In the accompanying video we show a sequence containing two piles of golden bunnies which are accidentally knocked over. In Fig. 5.5 we illustrate the choice of visual LODs for two frames of the sequence. We used a budget of 100 ms shared by all SH-shaded objects. Note that this budget will be shared by the GPU for SH lighting computations and the CPU for modes generation, both running in parallel. This budget does not reflect the exact running time of the simulation but is used as an indication.

For comparison, we computed an appropriate fixed audio-visual level of detail, corresponding to the perceived similarity to the reference rendering, as predicted by the perceptual study. For example, if the average choice of LOD in the sequence represents 90% similarity to the reference, we will choose 90% of the spherical harmonic bands and the modes used for the reference.

On average across the entire sequence, the model described in Chapter 4 predicts that we have a quality level of 90.6%, i.e., the rating of perceived similarity to the high quality rendering. The computation in this sequence is 44% faster; on average we achieved 26 fps using our approach and 18 fps using the equivalent quality with fixed LOD.

Full crossmodal approach There are three main differences with the work presented in Chapter 3: we integrate high-quality attack processing with clustering, the fast energy approximation, and the crossmodal LOD. We integrated these effects with the previous crossmodal metrics (clustering and scheduling) to provide a unified crossmodal pipeline.

The final sequence of the video shows all the elements working together. We selectively turn on and off some of the features, illustrating the various benefits obtained.

5.6 Discussion and Conclusion

We have presented a complete perceptually inspired crossmodal audio-visual rendering pipeline. We introduced an approximate energy estimation method for contact sounds,

and showed how to integrate high-quality attacks with clustering. We also presented a crossmodal LOD selection approach, which is to our knowledge the first algorithm which jointly chooses audio and graphics levels of detail, based on crossmodal perception. We believe that the integrated method presented here offers a practical and useful pipeline for audio visual rendering, which can be useful for games or other similar applications.

Our crossmodal LOD selection approach uses spherical harmonic rendering, since we base our algorithm on the study presented in Chapter 4 which used this approach. We expect that similar results can be obtained with other approaches for environment map rendering (e.g., zonal harmonics [Sloan *et al.* 2005] or sampling techniques [Agarwal *et al.* 2003]).

The potential for computation gains can be significant, especially during events which result in large numbers of impacts. Nonetheless, the work reported here is only a first step. In particular, we need to determine how well the results given in Chapter 4 generalize to other kinds of materials and objects, and to determine how perceived quality ratings are affected by the more realistic game settings shown here. Our intuition is that perceived quality should actually be higher in the realistic setting compared to the experimental condition. We also expect that a generic material model, based for example on some general material categories, could be applied in a more general setting. One way to do this would be to sample a perceptually uniform material space such as [Pellacini *et al.* 2000] and interpolate the results for any material. Also, the environment map used for this game setting was different from the one used in the experiment from Chapter 4, since this new environment is an indoors environment, contrary to the previous experiment. It has been shown that the environment lighting influences the perception of materials ([Fleming *et al.* 2003]), and a further BRDF prediction could include this parameter. In our case, an indoor environment makes the golden bunnies appear a bit more diffuse than with a higher frequency environment.

Part II

Visual rendering using a single photograph

Preface

In this part, we present two ways to use photographs in order to improve visual rendering algorithms. Photographs contain a large amount of information, and can be used to infer its style to a given rendering. This can be done either by finding parameters of a physical model given the photograph, or by directly using the photograph as input to a rendering algorithm.

Contrary to the first part, the previous work is not shared across our two contributions, since their overlap is small. We will thus present the two previous work sections separately, in each chapter.

We will first show how to use a single photograph of hair in order to convey the same appearance to a virtual hairstyle (Chapter 6). We will then show how to use a single photograph to allow for the rapid modeling and rendering of a 3D scene, using the style of the photograph (Chapter 7).

CHAPTER 6

Single Photo Estimation of Hair Appearance

Contents

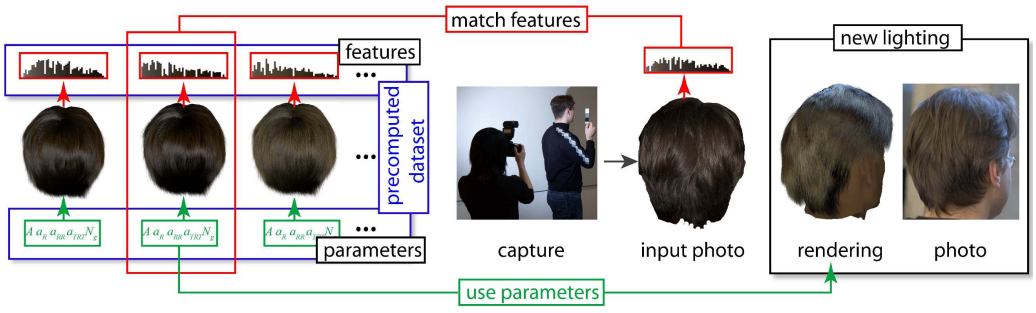


Figure 6.1: *Appearance Capture. Using a flash photograph, we estimate a set of hair appearance parameters which reproduce the overall likeness of the observed hair. We compute image features and a corresponding distance to match the input photo with a dataset of prerendered images with known appearance parameters. The estimated parameters are taken from the best-matching image, and can be used to render images in new conditions.*

The contributions in this chapter have been accepted for publication in the special issue of *Computer Graphics Forum* for the *Eurographics Symposium on Rendering* [Bonneel *et al.* 2009a].

6.1 Introduction

Recent advances improve the rendering of realistic hair using advanced models of scattering [Moon *et al.* 2008, Zinke *et al.* 2008, Zinke 2007, Moon & Marschner 2006, Marschner *et al.* 2003]. However, they involve numerous parameters, and matching the hair appearance of a given person is difficult even when the geometry is given. This process is time-consuming and error-prone even for trained artists [Mihashi *et al.* 2003]. An alternative is to use image-based rendering, but current hair-capture methods rely on complex hardware with many cameras and tens of lights, and produce models with very large storage requirements [Paris *et al.* 2008].

Our primary goal is to develop a simple, low-cost way to 'close the loop' between modern hair rendering and hair as it is readily observed in the real world or in photos. We abandon the notion of pixel-accurate reconstruction of a given photo and instead take a statistical and perceptual modeling approach. We introduce a method that enables the estimation of hair appearance with a very lightweight structured lighting setup: a single photograph taken with a flash on the camera. We use only approximate knowledge of the hair geometry. The simplicity of the setup is of particular importance for the user-driven creation of avatars and characters in games and interactive simulations, which we envisage as a prime application area. Recently, hairstyle and hair color have been reported to be among the most important features for avatar personalization [Ducheneaut *et al.* 2009].

The acquisition of hair geometry is beyond the scope of this chapter and we assume the existence of a small set of macroscopic models that define 3D hair-strand geometries for distinct classes of hairstyle, e.g., long curly hair, straight layered hair, etc. These models (see Fig. 6.2) are modeled by artists or reuse existing captured hair geometry [Paris *et al.* 2008]. We distinguish this *hairstyle model* from the remaining parameters, which form an independently-defined *hair appearance model*. Our hair appearance model consists of absorption, reflectance [Marschner *et al.* 2003], and geometric noise [Yu 2001] parameters, which we seek to estimate. A pair of melanin absorption parameters determines the overall hair color, and three lobe width parameters define the visual appearance of the specular highlights, glints, and transmissivity effects.

In this work the best-matching hairstyle model is manually selected; we shall see that this choice can impact parameter estimation (§6). It is also possible to assign the same hair appearance to different hairstyle models, although best likeness of a photo is still achieved by rendering with an appropriate match in both the hairstyle and appearance.

We also define a *geometric noise* parameter that can have a significant impact on the appearance of hair. Taken together, these parameters define a six dimensional hair appearance parameter, $\mathcal{P}$. For rendering, we use the fast multiple-scattering technique of Zinke et al. [Zinke *et al.* 2008]. We chose this method for efficiency; others could be used instead.

Given an approximate geometry, we wish to match the appearance of the rendering

to the appearance of the input flash photo, using the available free parameters in the appearance model (see Fig. 6.1). We first define a feature that captures the aggregate visual properties of hair, together with a metric on this feature. We propose the use of a luminosity-weighted color distribution in Lab color-space as an appropriate image feature, $\mathcal{F} = f(I)$, and an earth-mover's distance, $\delta(f(I_A), f(I_B))$, as a metric between two images in this feature space. Although our method does not produce physically validated parameter estimations, it generally achieves visually plausible results by working within the degrees of freedom available in the rendering model. We validate our image feature and the related metric using a perceptual evaluation.

To match hair appearance, we compute a reference dataset by sampling the parameter space and rendering images of our representative geometric models. For each reference rendering, we compute our feature, which provides us with a large dataset of $(\mathcal{P}, \mathcal{F})$ tuples, i.e., images for which we know both the parameters and the feature. Given the computed feature for an input photo, we search for the closest feature match in the dataset and return the associated model parameters. The entire process is illustrated in Fig. 6.1.

Contributions: We present a new approach to estimate hair appearance using a single photo; we achieve this by recasting the process as an image retrieval problem. To do this, we first introduce a hair appearance model with two new components: a melanin-based hair pigmentation model that reproduces the natural subspace of hair absorptions, and a geometry noise parameter. We then introduce an image feature and distance metric for matching hair appearance, and do a perceptual evaluation of these. Lastly, we test the single-photo hair appearance estimation on a set of 64 photos and do robustness evaluations.

6.2 Related Work

Modeling the geometry and appearance of human hair is challenging. Overviews of the progress that has been made on these problems over the past decade can be found in [Ward *et al.* 2007] and [Bertails 2006]. There is also considerable knowledge about the biology of human hair, including its microstructure, density, growth, response to humidity, and pigmentation [Halal & Schoon 2001, L'Oréal 2008]. Hair rendering models have evolved considerably, with increasingly sophisticated modeling of the complex light paths that occur in hair [Kajiya & Kay 1989, Marschner *et al.* 2003, Moon & Marschner 2006, Zinke 2007, Zinke *et al.* 2008, Moon *et al.* 2008]. However, setting the various required model parameters remains an unaddressed problem and motivates our work. The intricate geometry of hair and complex light diffusion precludes the use of general appearance analysis methods such as those proposed for BRDF estimation.

There have been several notable efforts to model aspects of hair from images. The work of Grabli et al [Grabli *et al.* 2002] infers the ambient, diffuse, and specular colors of a wig. However, real hair fibers are known to have a more complex reflectance than synthetic materials [Marschner *et al.* 2003] and thus it is unclear how well this method generalizes. Paris et al. [Paris *et al.* 2004] capture the hair geometry from a single, controlled light source and a video camera but the reflectance is not retrieved in this process. Wei et al. [Wei *et al.* 2005] improve this method with an approach that works under ambient

lighting. They also map photographic data on the hair geometry to model the hair appearance. These mapped colors are fixed and do not vary with the lighting environment nor the view direction, thereby limiting the rendering to reproducing the captured conditions. Paris et al [Paris *et al.* 2008] describe an image-based rendering method that yields faithful matches to photographs. This technique requires a large amount of data which would be unwieldy for many applications and requires a complex capture setup.

More broadly related are interfaces that help guide users in parameter selection [Marks *et al.* 1997], the determination of specular lobe shape from image statistics for linear light sources [Gardner *et al.* 2003], skin BRDF models based on pigment concentrations [Donner *et al.* 2008], and the use of multiscale statistics for estimating BTF parameters [Ngan & Durand 2006].

6.3 Synthetic Appearance Model

We first discuss the rendering model used and then describe a novel color absorption model based on human hair pigmentation. Lastly, fine-scale geometric distributions can also have a significant visual impact on the appearance of hair. To this end, we define a global *geometry noise* parameter.

6.3.1 Rendering

For rendering, we use the forward scattering map of Zinke et al.[Zinke *et al.* 2008] because both efficiency and quality of the rendering are of concern in our setting. In this paragraph we summarize the rendering details for completeness.

We use the scattering model presented by Marschner et al [Marschner *et al.* 2003]. We split the scattering parameters into two groups: free model parameters and fixed parameters. For the latter, we found that they have the least influence on the overall rendered hair appearance or can be assumed to have typical values [Marschner *et al.* 2003]. Our fixed-value parameters are: cuticle angle ($-3°$); eccentricity (0.9); caustic power (0.4); caustic width ($1.5°$); fading range for caustic (0.3); index of refraction (1.55); density [Zinke *et al.* 2008] (0.7); and hair radius (120 μm). The free parameters are the absorption coefficients, A_r, A_g, A_b, and the R, TT, and TRT lobe widths, $\alpha_R, \alpha_{TT}, \alpha_{TRT}$, which intuitively correspond to the "color" and "shininess" of the hair. In our tests, hair radius has been increased compared to values given in the literature to account for the sparsity of our hair models.

To efficiently render images using [Zinke *et al.* 2008], we use a CPU implementation of the raytracing-based forward scattering map method and parallelize it to exploit multicore architectures. We perform the final rendering step directly on the GPU. With this, 640×480 images are rendered at a rate of 1.2–1.9 images per minute. We choose this over the faster GPU-based algorithm because of the higher quality we obtained from the forward scattering map.

Figure 6.2 shows the hairstyle geometries that we use: three are modeled using Maya, and four are captured data [HPD 2008]. The hairstyles each have 80,000–120,000 hairs.

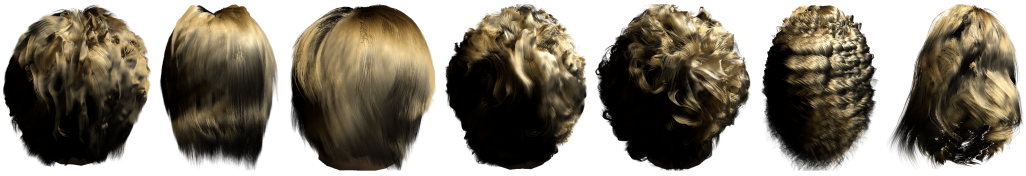


Figure 6.2: *Hairstyle geometries used for datasets (without noise), from left to right: straight short hair, straight long hair, straight clumpy hair, wavy hair, tangled hair, long curly hair, very long straight hair*

6.3.2 Melanin Model

In order to reduce the number of parameters for hair appearance, we exploit knowledge of natural human hair pigment absorption to reduce the number of parameters in the original Marschner model [Marschner *et al.* 2003]. Our reparametrization can also greatly facilitate manual hair appearance specification by reducing the number of degrees of freedom and ensuring that the absorption is realistic.

Natural hair color is largely due to wavelength-dependent absorption within hair fibers. Hair pigmentation is composed of two kinds of melanin pigments: *eumelanin* and *pheomelanin* [L'Oréal 2008, Tobin 2008], and the spectral absorption of these two pigments is known [Sarna & Swartz 1988] (Fig.6.3).

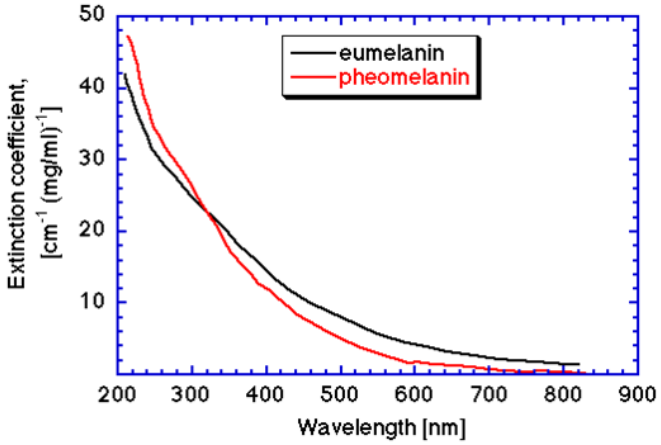


Figure 6.3: *Eumelanin and pheomelanin extinction coefficients with respect to wavelength [Sarna & Swartz 1988].*

The absorption can thus be modeled as a linear combination of the concentration of these melanins i.e., $A(\lambda) = a_1(\lambda)m_1 + a_2(\lambda)m_2$, where m_1, m_2 are the eumelanin and pheomelanin concentrations and $a_1(\lambda)$, $a_2(\lambda)$ are their spectral absorption curves. We point sample the absorption curves for red, green and blue wavelengths, (575, 535, and 445 nm, respectively), and normalize with respect to the r concentration of the melanins, yielding: $A_r = m_1 + m_2$, $A_g = 1.3036m_1 + 1.6390m_2$, $A_b = 2.2272m_1 + 3.8370m_2$.

This defines a 2D subspace in the 3D absorption parameter space without reducing the desired expressivity of the model. Because $m_1, m_2 \in [0, \infty]$, it will be convenient to instead represent the melanin concentrations by $\hat{m}_1 = e^{-km_1}$ and $\hat{m}_2 = e^{-km_2}$, where k is experimentally determined (§6.4.2). This gives finite ranges, $\hat{m}_1, \hat{m}_2 \in [0, 1]$. We shall exploit the melanin model during appearance estimation, where it will help achieve better sampling of the absorption parameters by eliminating unnatural hair colors from consideration, such as green or blue hair.

6.3.3 Geometry Noise

We distinguish between two different levels of geometry variation. The macroscopic geometry or hairstyle is chosen by the user from a fixed set of models (Figure 6.2) and captures aspects such as the curliness and length of the hair. However, a direct application of rendering techniques [Marschner *et al.* 2003, Zinke *et al.* 2008] to many modeled or captured hair geometries can yield unrealistic results. Figure 6.4 (left) shows an example of this for captured geometry [Paris *et al.* 2008]. This can be rectified using an additional geometry variation based on small-scale noise. We add noise in a manner similar to Yu [Yu 2001]. The perturbation applied to a given vertex v on a hair strand is given by

$$\Delta v = N_g \, A \, V(\alpha \sin(2\pi\alpha p) + \frac{1}{2} e^{3\alpha} - 1) \tag{6.1}$$

where $\alpha \in [0, 1]$ is the normalized curvilinear abscissa , V is a unit-length random direction, $p \in [0, 8]$ is a random frequency, N_g is a noise amplitude, and A is a hairstyle-specific scaling factor that is required to deal with the given modeling scale of any hairstyle. Fig. 6.4(right) shows the changed appearance with the noise model. Without any geometric noise, the hair can have a noticeable unrealistic plastic-like appearance.

Figure 6.4: *Impact of geometric noise on hair appearance. Left: Without geometry noise. Right: With geometry noise.*

The final set of free parameters in our appearance model is given by $\mathcal{P}$: $\{\hat{m}_1, \hat{m}_2, \alpha_R, \alpha_{TT}, \alpha_{TRT}, N_g\}$.

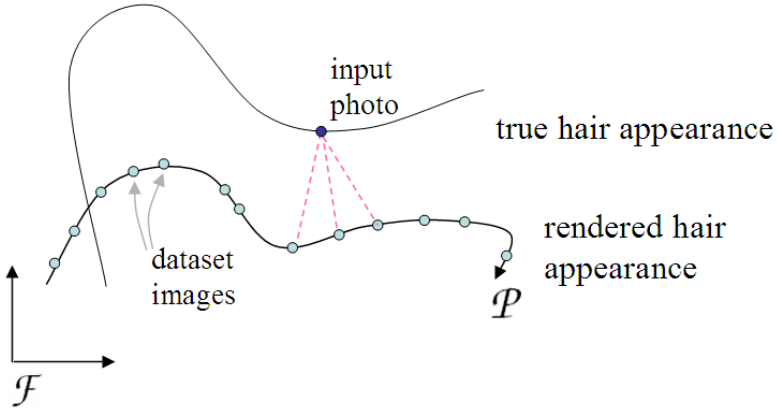


Figure 6.5: *An abstract view of the appearance estimation problem. $\mathcal{F}$ defines the space of features used to represent the appearance. Rendered images lie on a manifold parameterized by the model parameters, $\mathcal{P}$.*

6.4 Appearance Estimation

Our method relies on a reference dataset of rendered images sampling the parameter space and we seek to find the reference image whose hair appearance is most similar to the input photograph. A schematic illustration of the appearance estimation problem is shown in Figure 6.5. Both the true hair appearance and the rendered hair appearance can be seen as forming manifolds in a suitably-chosen feature space, $\mathcal{F}$, although we only know the underlying parameterization of the rendered appearance. An input photo may not in general lie on the manifold of images achievable by the rendering, which may in part be attributed to a limited expressivity of the rendering model. Given an input photo, we propose a novel method to estimate its underlying parameters, $\mathcal{P}$, by posing this as a search problem. We search for the point on the rendered manifold which is most similar in appearance to the input photo, and then return its associated parameters.

6.4.1 Feature Selection and Distance Metric

Image features for matching hair appearance should ideally be easy to compute, induce a perceptually-meaningful metric, work across a large variety of hair appearances, and be somewhat invariant to the specific geometry of the hair. We propose the use of luminance-reweighted color distributions in Lab color space as our feature, and the use of Earth-mover's distance (EMD) as a distance metric between features. These choices are inspired by the use of color distributions and EMD for Image Retrieval [Rubner *et al.* 2000]. A modification we propose in the context of the hair appearance estimation is to use a luminance-based reweighting of the color distributions, which we found to yield improved estimates.

The image color distributions are represented by color-clusters in the Lab color space. We use k-means clustering with 50 clusters and initialized using k-means++ [Arthur & Vassilvitskii 2007]. To achieve fast clustering, a subsampled set of 1 in every 50 pixels is used during the k-means iterations. The final nearest-neighbor assignment is done for all pixels in the image to obtain a pixel count for each cluster. Each

cluster's pixel count n_i is reweighted by the cluster luminance L_i, $n'_i = Ln_i$. Finally, the resulting cluster pixel counts $\{n'_i\}$ are normalized, i.e., $\hat{n'_i} = n'_i/N$, where $N = \sum_i n'_i$. The final image feature is then a set of tuples, $F = \{(C_i, \hat{n'_i})\}$, where C_i is the Lab color of cluster i, i.e., the cluster center. Computing the color clusters for an image takes ~600 *ms*.

The EMD metric minimizes the work required to turn one distribution into another, and is symmetric, i.e., $\delta(\mathcal{F}_a, \mathcal{F}_b) = \delta(\mathcal{F}_b, \mathcal{F}_a)$. The metric is computed by solving a transport problem, where the 'mass' in the source distribution bins needs to be transported with minimal cost to the target distribution bins to exactly fill them. In our problem, the source and target bins correspond to the clusters of $\mathcal{F}_a$ and $\mathcal{F}_b$, and mass corresponds to normalized cluster counts, $\hat{n'}$. As a distance, the EMD returns the minimal total work required, as defined by $\delta = \sum_{ij} d_{ij} f_{ij}$, where d_{ij} is the *ground distance* between source bin i and target bin j and f_{ij} is the flow (mass) carried between these two bins. We use Euclidean distance in the Lab space as our ground distance. The EMD optimization is posed as a linear programming problem and is solved using a streamlined version of the simplex method [Rubner *et al.* 2000].

6.4.2 Synthetic Dataset

The synthetic dataset consists of a large set of precomputed tuples, $D : \{(\mathcal{F}_i, \mathcal{P}_i)\}$, with one tuple per rendered image. The final datasets are compact in practice because there is no need to retain the images once its features have been computed.

We compute multiple datasets, D^k, one for each modeled hairstyle, k, and using 4000–5000 sample points per dataset. We use a simple sampling strategy given that the real distributions of the model parameters for human hair are unknown. We draw random samples using uniform distributions U. The Marschner lobe parameters are sampled using $\alpha_R, \alpha_{TT}, \alpha_{TRT} \sim U[2°, 20°]$. This represents an extension of the typical values given in [Marschner *et al.* 2003]. However, we note that these typical value ranges are not necessarily strictly respected in prior art. For example, [Marschner *et al.* 2003] uses $\alpha_R = 8°$ and $\alpha_{TT} = 6°$, yielding an α_R to α_{TT} ratio of 1.33 instead of the recommended ratio of 2.0, and [Zinke *et al.* 2008] uses $\alpha_R = 8°$ and $\alpha_{TT} = 10°$ for a ratio of 0.8. Establishing accurate ranges or prior likelihoods for these parameters would require physically-validated measurements for a large set of examples. Noise is sampled using $N_g \sim U[0.3, 1]$.

For sampling melanin concentrations, we use an informed strategy that samples the space of final observed hair colors in an approximately uniform fashion. We build on our observation that the mean color of rendered hair is approximately linearly correlated with $exp(-kA_i)$, with $k \approx 6.3$. We sample m_1 and m_2 so as to maintain uniform sampling of the dominant red component as much as possible. We use $\hat{m}_1 \sim U[0, 1]$, where $\hat{m}_1 = exp(-km_1)$ and $\hat{m}_2 \sim U[0, 1]$, where $\hat{m}_2 = exp(-km_2)$. From this, the absorptions can be expressed as $A_i = -\ln(\hat{m}_1^{p_i} \hat{m}_2^{q_i})$. To keep the correlation as much as possible for the dominant red component, we set $p_r = 1$, $q_r = 1$, $p_g = 1.3036$, $q_g = 1.6390$, $p_b = 2.2272$ and $q_b = 3.8370$, as determined by the linear melanin combination model described earlier (§6.3.2). Our sampling for absorptions allows for a broad range of $[0, \infty]$. This results in values outside the typical range of $[0.2, \infty]$ given in [Marschner *et al.* 2003] for

the absorptions; however values as small as 0.03 are found in [Moon & Marschner 2006].

Our acquisition configuration is only loosely specified, namely a photo of the back of the head with a flash near to the lens. To increase the robustness of our technique to variation in light and camera direction, we include random perturbations of these factors in our dataset. For each dataset image, we add random offsets in $[-2.5°, 2.5°]$ to the viewing direction and lighting direction.

The rendering illumination is the same across all dataset images and all hairstyles and thus gives a self-consistent default exposure for the renderings. In order to achieve compatible exposures with the gray-card calibrated photos, we apply an illumination scaling parameter, s, to the rendered images. The value of s is determined using a one-time cross-validation. Specifically, we sample s in the set of bracket of exposures {0.6, 0.8, 1.0, 1.2, 1.4} and we keep the value yielding the best overall match results ($s = 1.4$ in our case) for a small set of test photos. We noted that values of $s \leq 1.0$ were unable to reproduce lighter hair colors. Lastly, we also compute an image mask so that pixels that are alpha-blended with the background can be excluded from the feature computation. All images are rendered against a blue background. All pixels having color components $b > g$ and $b > r$ are excluded from the mask, and this is further followed by a 3×3 image erosion operation.

6.4.3 Photo Preprocessing

We take a flash photo of the back of the head together with a reference 18% gray card, taken indoors with a short exposure, e.g., $1/200s$, in order to minimize the impact of indirect lighting. Input photographs are first downsampled to 640×480 using filtering based on bicubic interpolants to match the resolution of the database. They are then processed for white balance using the color of the imaged gray card. We currently use Photoshop for this step, as well as for segmenting the hair in the photograph. Lastly, we scale image luminance to achieve a 12% on-screen luminance for the imaged gray card as commonly done by photographers.

6.4.4 Parameter Estimation

Given a preprocessed input photograph, its features $\mathcal{F}$ are computed and the best-matching hairstyle, i.e., choice of dataset, is manually selected. A simple linear search is then used to select the nearest neighbor, i.e., $j^* = \arg\min_j \delta(\mathcal{F}, \mathcal{F}_j)$, and the estimated parameters are given by $\mathcal{P}_{j^*}$. This requires approximately 20 seconds for a dataset of 5000 images. More sophisticated forms of non-parametric regression could also be used, i.e., applying kernel regression to $\mathcal{P} = f(\mathcal{F})$. However, these did not improve the resulting estimates (see discussion in §6.7), likely because the distance from the photo to the manifold of rendered images is generally larger than the distances between neighboring samples on the manifold. As a result, a large kernel spans too many neighboring samples while a small kernel effectively results in nearest-neighbor selection.

6.5 Perceptual Evaluation

We performed an experiment that provides a perceptual evaluation of the EMD-based metric used by the estimation procedure. Our goal is to verify that if the metric predicts that renderings (and thus the appearance parameters) are similar to photographs, human observers also find them similar; while if the metric predicts a large difference, humans agree with this prediction.

Subjects are asked to make relative assessments as to which of two renderings they find to be most similar in appearance to a given photo. The two renderings use the same hairstyle, which avoids confusing hairstyles with hair appearance. The experimental setup thus exactly mimics decisions of the type that need to be made during the parameter estimation (§6.4.4), and it fits naturally into a two-alternative forced choice (2AFC) protocol. The experiment does not provide a measure of *absolute similarity* of a rendering to a photograph: This is a very hard problem, and there is no established way to do this. In addition, such a hypothetical test would also involve the evaluation of the quality of the rendering and lighting model, which are beyond the scope of this work.

Given a photo, P, and two renderings, A and B, we expect that our metric will be weakly predictive of the choice of closest image for cases where $\delta^P_{AB} = |\delta(\mathcal{F}_P, \mathcal{F}_A) - \delta(\mathcal{F}_P, \mathcal{F}_B)|$ is small, i.e., the metric finds that images A and B are roughly equidistant to the photo, P. Similarly, we expect the metric to be strongly predictive as δ^P_{AB} becomes large. We thus define three categories producing almost equidistant, quite different and very different renderings: $\Delta_1 : 0.2 \leq \delta^P_{AB} \leq 0.5$ (almost equidistant), $\Delta_2 : 2 \leq \delta^P_{AB} \leq 3$, and $\Delta_3 : 4 \leq \delta^P_{AB} \leq 10$.

We choose ten photographs from our results data set that approximately span the space of hair appearances. For any given photo, we first build a base bin, B, of three images, where each image i is chosen to be close to the photograph, P by satisfying $\delta_{NN} \leq \delta_{Pi} \leq \delta_{NN} + b_w$, where δ_{NN} is the distance of the nearest neighbor (best match) in the dataset to the photo, and b_w is the base-bin width. We also always include the nearest neighbor as one of the three images in this bin. For all image pairs shown in the test, one of the images will come from this base bin. This helps ensure that distance differences are only compared for a similar reference distance. For each photograph, we then select 9 pairs of renderings such that there are 3 pairs for each of the categories, Δ_1, Δ_2, and Δ_3. This experimental setup results in 90 comparisons per test session, where the subject must choose the "rendering with hair appearance most similar to the photograph". The test has a duration of approximately 15 minutes. An example screenshot of the web survey can be seen in Fig.6.6.

We used an Internet-based survey, distributed to three university or research institutions. Participants were instructed not to complete the survey if they were color-blind and were recommended to use a bright, good quality monitor with sufficient resolution so that the photo and the image-pair could all appear on-screen simultaneously. A total of 47 participants completed the survey, giving a total of 1410 evaluations for each distance category (47 participants $\times$ 10 photos $\times$ 3 tests per category). We consider separate hypotheses for each Δ_1, Δ_2, and Δ_3. The null hypothesis for each case is that users will be at chance in having their selection of the closest image match that given by the metric. The alternate

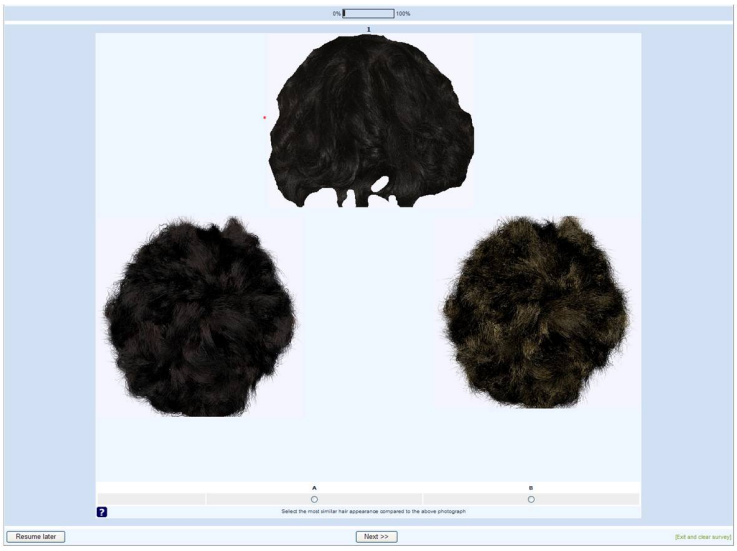


Figure 6.6: *Screenshot of the perceptual experiment.*

hypothesis is that the metric helps predict the participant's choice of most similar appearance.

For category Δ_1, the subject's choice agreed with the metric in 71.4% of all tests, with a standard deviation across subjects of σ_s=8%. The null hypothesis can be rejected: $\chi^2(1, n = 1410) = 276, p < 10^{-5}$. For category Δ_2, the agreement rises to 75.3% (σ_s=7), and the null hypothesis can be rejected: $\chi^2(1, n = 1410) = 360, p < 10^{-5}$. Lastly, for the large predicted differences of category Δ_3, the agreement is 93.6% (σ_s=5%) and the null hypothesis can be rejected: $\chi^2(1, n = 1410) = 1074, p << 10^{-5}$.

Our experiment thus shows that human observers agree with our feature-based distance for judgments of similarity between photos and rendered images.

6.6 Results

We test the appearance estimation on a set of 64 photographs. Figure 6.7 shows a subset of the results and the associated model parameters are given in Table 6.1. The new lighting conditions shown in the rightmost column are chosen manually to best match the corresponding photographs. The complete set is given in the additional material of the paper [Bonneel *et al.* 2009a] [1]. Part of these results are shown in Figure 6.8. Over the full set of photos, the best-match EMD distance spans the range $1.35 < \delta < 6.13$, with a mean $\mu = 2.92$, and a standard deviation $\sigma = 1.01$.

Animation: We can easily animate our estimated hair appearance, which remains a challenge for other methods such as [Paris *et al.* 2008]. Animations of two hair appearances are shown in the video (see footnote 1) and Figure 6.9.

[1] Additional materials: http://www-sop.inria.fr/reves/Basilic/2009/BPVDD09/hairAdditionalMaterial.zip
See also the paper's video available at http://www-sop.inria.fr/reves/Basilic/2009/BPVDD09/hairVideo.avi

$\hat{m}_1$	$\hat{m}_2$	α_R	α_{TT}	α_{TRT}	N_g	δ
0.001	0.123	11.9	18.3	9.9	0.927	2.75
0.355	0.572	11.4	20.0	23.8	0.788	3.15
0.052	0.267	14.0	5.3	16.4	0.561	1.63
0.081	0.955	16.0	12.7	16.0	0.969	1.99
0.362	0.916	17.3	19.0	23.5	0.544	3.81
0.910	0.086	11.3	18.9	14.9	0.760	3.70
0.033	0.436	8.1	7.3	12.9	0.400	3.42

Table 6.1: *Estimated model parameters for Figure 6.7, as well as their distances, δ, to the photo.*

Application scenario: Our method is suited to end user-driven content creation such as for the design of avatars for games, in contrast to the methods used in high-end visual effects for film where expert artists are available to help set parameters. We have used a subset of our input photographs to create a prototype interface for character design, as shown in Figure 6.10 and the video, see footnote 1. In this interface, the user can create a library of hairstyles for game characters by taking flash photographs and picking a template geometry. These hairstyles are then available in the game as shown in our prototype (see video).

6.7 Discussion

We first discuss the various estimation methods we tested before adopting the approach presented here, then discuss a number of robustness tests we performed and conclude with a discussion of limitations.

Other estimation methods: We tried several machine learning approaches based on features inspired by domain-specific knowledge on hair appearance, but none of them worked as well as the approach that we finally propose. We tested regression methods that seek a function f such that the parameters $\mathcal{P}$ can be expressed as $\mathcal{P} = f(\mathcal{F})$, where $\mathcal{F}$ is a set of image features. The critical part is the choice of $\mathcal{F}$. Low-dimensional feature vectors require only a small training set to cover the feature space. But such feature vectors are exceedingly difficult to design since they are almost of the same size as the parameters, that is, finding good feature vectors is almost equivalent to the initial problem. Larger feature vectors do not suffer from this problem but require a larger training set to obtain f, which quickly becomes the limiting factor. We tested feature vectors based on means and medians of the color components, as well as responses to banks of oriented spatial filters. We implemented Gaussian process or Nadaraya-Watson kernel regression to predict the hair absorption parameters, but none produced reliable estimates, as tested using cross-validation with rendered images. We also tested several segmentation methods to correlate highlights with corresponding lobe parameters; none reliably identified the desired highlights, because the R and TRT regions overlap too significantly.

Our final use of a color histogram solves this dilemma by using a high-dimensional feature, the weighted color distribution, a perceptually-validated metric on this feature vec-

tor, and the use of a sampling and a nearest-neighbor scheme to resolve the final parameter estimates. A somewhat surprising aspect of the solution is that it can be effective without relying on spatial features. The advantage of this lack of spatial sensitivity makes our approach robust to spatial variations.

Dataset sampling: We compute a simple statistic to confirm our intuition that the photo and rendered-image manifolds are widely spaced compared to the sample spacing used by the dataset (see Figure 6.5). Specifically, we compute the ratio $r = \delta_{12}/\delta_{1P}$ over the full set of photos, where δ_{12} is the distance between the first and second nearest neighbors, and δ_{1P} is distance between the photo and the nearest neighbor. The resulting small values of r, $0.22 < r < 1.08$, $\mu = 0.62, \sigma = 0.17$, support the described intuition regarding this geometry. We also note that for particular regions of the parameter space, the rendered image is locally invariant to some parameter values, which allows us to use a smaller number of samples. For example, the appearance of black hair is largely invariant to TT or TRT lobe widths. The use of only 5000 samples to cover a seven-dimensional parameter space is partly enabled by the dimensionality reduction and perceptually-uniform sampling of the hair color space and a priori knowledge of the limited expected range for the remaining parameters.

Match Sensitivity: We examine how the distance to the photo changes as a function of the model parameters in the region surrounding the nearest neighbor. Figure 6.11 shows normalized plots of how the distance metric changes as individual parameters are varied around their final estimated value while the remainder are held fixed. The absorption parameters and the R-lobe parameter have well-defined local minima. The TT and TRT lobes should not take on values smaller than their nominal value, but have a minimal effect on the overall appearance for larger values. The noise parameter exhibits a shallow local minimum for this example, although in general it can have a stronger variation – the two images shown in Figure 6.4, which vary only in their noise parameter, have a relatively large distance of $\delta = 3.2$.

Robustness with respect to lighting variation: To test for the effect of changing the lighting, we compare the result of using flash-lighting placed 15 degrees above the lens with the result of flash lighting placed 15 degrees below the lens. For the case of a person with black hair, the original and modified-lighting parameter estimates are $\mathcal{P}_0 = \{0.0518, 0.1133, 13°, 3.5°, 12°, 0.82\}$ and $\mathcal{P}_1 = \{0.0073, 0.6543, 11°, 15°, 17°, 0.82\}$, respectively. While the estimated absorptions are different in the melanin space, they are very similar when seen in the rgb space. α_R also remains very similar. The TT and TRT lobe widths receive different estimates, although this is not unexpected given the negligible role of transmissive scattering in black hair. The distance to the best match changes only marginally ($\delta_0 = 1.93$, $\delta_1 = 2.02$). We also apply the same test for a person with lighter-colored hair, and in this case the same nearest-neighbor is returned ($\delta = 2.36$), thus yielding an unchanged parameter estimate.

Impact of hairstyle geometry: A fundamental question to ask is the extent to which the choice of hairstyle geometry affects the parameter estimation. A first way to measure this is to match a photo using different hairstyles and to observe the resulting parameters and their images. Figure 6.12 shows an input photo and the nearest neighbors for three different hairstyles. The estimated parameters for the manually chosen

target hairstyle are: $\mathcal{P}_0 = \{0.0845, 0.9232, 16°, 13°, 16°, 0.97\}$. With the shown alternate hairstyles, this changes to: $\mathcal{P}_1 = \{0.0137, 0.8786, 21°, 3.9°, 20°, 0.76\}$ and $\mathcal{P}_2 = \{0.0495, 0.9093, 23°, 14°, 23°, 0.68\}$. The differences in estimated parameter values can be attributed in part to the choice of hairstyle and in part to the fact that parameters such as α_{TT} have little effect for dark hair and so they may not be estimated in a consistent fashion.

The impact of hairstyle geometry can also be measured by using each of the seven different hairstyles to do parameter estimation and then using the resulting parameter estimates to render images using the user-selected best-matching hairstyle. We can then compute their respective similarities to the input photo using our metric. The results of this computation show that parameter estimates that come from the user-selected hairstyle are always among the best results, and that largely different hairstyles produce inferior results. This confirms the importance of using similar hairstyles for matching and rendering.

Our library is currently based on seven hairstyle models because these approximately span the range of hairstyles observed in our test set of 64 input photographs, and because of the difficulty of obtaining or creating additional hairstyles. Enlarging our library of hairstyles may help improve the final rendered likeness to the input photos, and possibly result in small improvements to the parameter estimation.

Limitations: Hair may be dyed, have sun-bleached or dyed highlights, or have a partial distribution of gray hairs, violating our assumption of constant hair absorption values. Visible scalp will also affect the estimations. Our synthetic appearance model does not model wet or greasy hair. Figure 6.13 illustrates our two worst results as judged by the metric. The gap between the rendered images and the photos for our dataset of tens of heads highlights some of the remaining challenges in hair modeling/capture and rendering. Our appearance estimation technique inherits the limitations of current rendering techniques, but also stands to directly benefit from future advances in hair rendering. In particular, our rendering implementation was unable to obtain realistic images for front lit blond hair, possibly due to the disciplined hair approximation or the straight light paths for large scale scattering [Zinke *et al.* 2008]. As such we do not present results yet for very ("Scandinavian") blond hair. Some of our current lightest hair can be seen in the second row of Figure 6.7, in Figure 6.9, and the white hair in Figure 6.13. We expect better matches by improving the rendering method or by using an existing slower but more accurate approach ([Moon *et al.* 2008]), and possibly by increasing the number of hairstyles in our database.

6.8 Conclusions

We have presented a novel method for estimating hair appearance parameters from a single flash-lit photo. We develop a hair absorption model based on melanin-based pigmentation, and introduce geometry noise as an appearance parameter. A suitable image feature and distance are defined for measuring hair appearance similarity, and we conduct a perceptual evaluation of this metric, which gives a strong indication of the validity of our choices. The technique has been used to estimate hair appearance parameters for 64 photographs, for which we provide side-by-side comparisons of the input photos and renderings. To our knowledge, this is a significantly larger set of comparative results than those presented to

date in prior art on hair modeling and rendering. We analyze the robustness and sensitivity of the appearance estimates in several ways.

Hair appearance can be captured with a wide range of techniques. Our proposed approach lies at one end of this spectrum, requiring a single flash photograph as input, and producing an estimate of seven appearance parameters in minutes. The technique of Paris et al. [Paris *et al.* 2008] lies at the other end of the spectrum, requiring a light-stage setup consisting of 16 cameras, 3 DLP projectors, 150 programmable LED lights, 40 000 images, 20 minutes of capture time, hours of compute time, and producing a detailed model of geometry and reflectance that requires 4.3Gb of data during rendering.

There are a number of exciting avenues for further exploration. With the help of lighting and white-point estimation techniques, it may well be possible to do parameter estimation from one or more photos taken in unstructured lighting conditions. We wish to add further geometric expressivity to the model by estimating meso-level geometry such as clumps and wisps of hair that may be identifiable from the photo. Recent progress has been made on this [Wang *et al.* 2009]. The distance we use could be used to define a hair appearance manifold from a large collection of input photos, independent of hair rendering techniques. We believe that the general style of parameter estimation approach may be applicable to other types of phenomena in graphics.

Figure 6.7: *Seven parameter estimation results, showing, from left to right: input photo, nearest-neighbor match in the rendered database, photo of new lighting condition, rendered new lighting condition using manually matched lighting. Quality of matches is discussed in section 6.7.*

Figure 6.8: *Seven additional estimation results.*

Figure 6.9: *Animating hair using estimated hair appearance parameters, with the input photo shown on the left.*

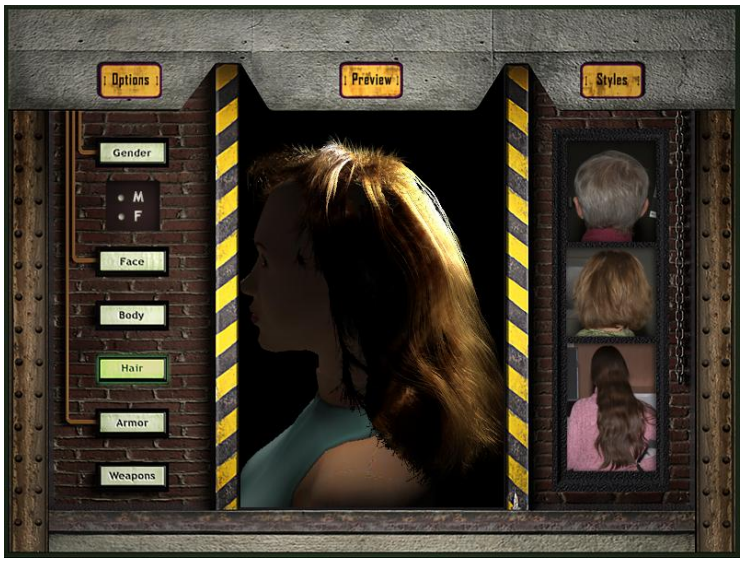


Figure 6.10: *The use of hair appearance in a prototype character customization interface for a game.*

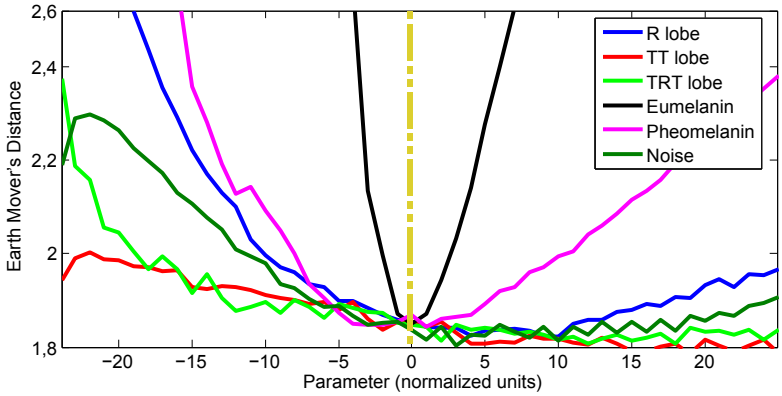


Figure 6.11: *Example variation of the distance as a function the underlying model parameters around the nearest neighbor. The x-axis spans the following spreads for each model parameter:* $\alpha_R : 28°, \alpha_{TT} : 21°, \alpha_{TRT} : 28°, \hat{m}_1 : 0.65, \hat{m}_2 : 1.9, N_g : 1.92$.

Figure 6.12: *The effect of hairstyle choice on parameter estimation: input photo, manually chosen target hairstyle, alternate hairstyle 1, alternate hairstyle 2.*

Figure 6.13: *The photos and nearest-neighbors for our worst two matches, as measured by the metric. Left pair:* $\delta = 6.13$*. Right pair:* $\delta = 5.08$*.*

CHAPTER 7

A Texture-Synthesis Approach for Casual Modeling

Contents

The contributions in this chapter have been submitted for publication and are under review at the time of writing.

7.1 Introduction

In the previous chapter, we have seen how a single photograph could be used to infer parameters to a physical model, in particular in the case of hair. This allowed us to obtain a similar appearance as the input photograph. However, it can be interesting to obtain the same appearance as an input photograph for the whole rendering. This is achieved by

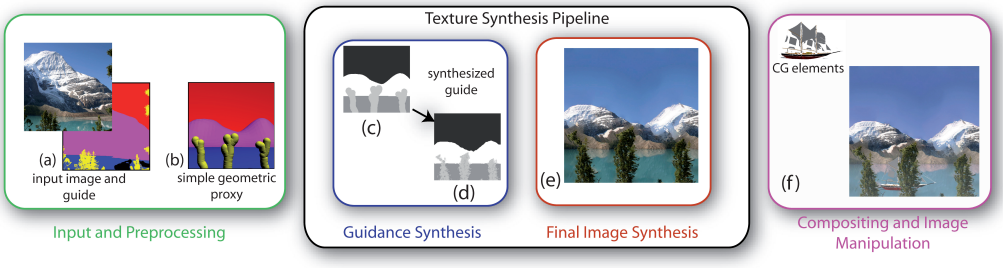


Figure 7.1: *Rendering simple proxy geometry using texture-synthesis shading. Given a source photo and proxy geometry, the goal is to produce visually-rich depictions of the 3D proxy. Our approach also allows for consistent integration of standard CG elements with texture-synthesis elements. Each colored box represents a separate stage of our approach.*

infering the photograph style to the rendering and using portions of the input photograph directly as rendering primitives rather than estimating parameters. We will see how this can be done, by using a texture synthesis shader, in this chapter.

Creating convincing computer graphics images is a challenging task: detailed models are hard to create, they can be very difficult to texture and the final rendering can require significant tweaking. In many cases the resources to do this are simply not available. The widespread development of 3D-capable applications, such as simple 3D games on phones or portable devices, or simple 3D previsualization, motivates alternative approaches for rapid user-driven generation of convincing 3D content.

We propose a new approach for the casual modeling of natural scene elements using texture synthesis. As illustrated in Figure 7.1(a), the user starts by selecting a source photo, from a pre-annotated set, which corresponds to the desired style of visual detail. The user then creates a crude 3D model of the desired geometry, which we call the *proxy* (Figure 7.1(b)). We can then generate new images (Figure 7.1(e)) based on the proxy and using the rich visual detail available in the source photo. Multiple viewpoints can be generated for a given scene. We can incorporate 3D computer graphics elements into these scenes (Figure 7.1(f)), add atmospheric effects such as fog, and interactively modify the synthesized elements, such as applying a color change to the sky, land, ice, or water (Figure 7.7).

The simplified proxy geometry does not provide any detail along the silhouettes of the generated objects (Figure 7.1(c)). We add this detail by introducing a texture synthesis stage to create a final guidance map. The smooth texture-element borders coming from the proxy geometry are replaced with the richer border detail between texture elements that can be found in the source photo (Figure 7.1(d)). A final image synthesis stage then produces the final image.

Both of the texture synthesis stages manipulate or compare guide images that contain discrete texture labels rather than the continuously-valued colors that form the staple of traditional texture synthesis. To this end, we introduce the use of Chamfer distances in the texture synthesis process in order to compare regions of texture labels in a principled fashion.

Our contributions can be summarized as follows:

- The introduction of label-based guidance synthesis to achieve detailed silhouettes and the use of the Chamfer distance to achieve high-quality results.
- A three-step final texture synthesis approach to produce high-quality visual results. These steps consist of initialization with a set of patches created using the guidance, pixel-based guided synthesis, and a final gradient-based Poisson correction.
- A prototype system which allows the integration of 3D synthetic elements into the final casually-modeled images and the addition of several image-based effects (color changes, fog etc.), based on image layers.

Our approach allows the rapid creation of images that are rich in visual detail, with the ability to change the viewpoint and manipulate the resulting scene in interesting ways.

7.2 Previous work

Our work is related to two main domains, that of casual modeling and texture synthesis. The surveys presented in [Wither 2008] and [Wei *et al.* 2009] provide good overviews of the extensive previous work in these two areas. We discuss only the most closely related work in each domain.

7.2.1 Casual Modeling

Fast creation of visually-rich depictions of natural scenes is a difficult task, particularly for non-expert users. Sketch-based modeling techniques are a promising approach developed for a variety of classes of geometry, including landscapes. In [Cohen *et al.* 2000], a user draws contours of mountains from which an approximate geometry is then estimated. Non photorealistic rendering (NPR) is then performed to render the sketched scene. In [Watanabe & Igarashi 2004], a related approach is developed; they also report on artifacts of the method used in [Cohen *et al.* 2000] when the geometry is seen from a different angle. The use of NPR rendering is a key aspect of these techniques as the smooth and approximate geometry does not lend itself well to realistic rendering. A variety of NPR rendering techniques have been developed specifically for landscapes and digital elevation models [Cohen *et al.* 2000, Watanabe & Igarashi 2004, Pierre-Loup Lesage 2002, Whelan & Visvalingam 2003, Mat & Visvalingam 2002]. However, if a richly-textured image is desired, the creation of the appropriate textures remains a challenging task.

The approach of [Zhou *et al.* 2007] allows users to easily create large-scale landscape geometry, i.e., mountains, hills, or large canyons. The user sketches a desired feature map in a top view and a novel terrain-specific texture synthesis approach is then used to generate the desired model given example digital elevation model data. The terrain geometry is rendered using the Terragen terrain system with procedural textures as determined by the terrain height and slope. A method of generating complex natural terrain, including arches

is recently developed in [Peytavie *et al.* 2009]. Impressive terrains are generated by leveraging a hybrid representation for terrain geometry, optimized tools for user editing of this terrain, and procedural elements that are specific to specific geological formations such as rock piling.

Our approach is largely complementary to these techniques. In our pipeline, a simple geometric proxy is first rapidly modeled and then we achieve an implicit addition of geometric detail in a first rendering stage (the guide synthesis) followed by the addition of visual detail in a second rendering stage.

7.2.2 Texture synthesis

The problem of synthesizing textures from an example image has attracted much interest over the past two decades. For a comprehensive review please refer to the recent survey by [Wei *et al.* 2009].

Our work is most closely related to guided texture synthesis [Ashikhmin 2001, Hertzmann *et al.* 2001]. In particular, Hertzmann et al. introduced the idea of texture–by–numbers. The input to the system is a color image and a map segmenting its content through shades of basic RGB colors (red, purple, yellow, etc.), called *labels* in the following. Given a guide — a new map using similar labels but a different layout — the algorithm synthesizes a new color image with a corresponding layout. Synthesis is performed by matching square neighborhoods in a multi-resolution, coarse to fine process. The similarity metric is an L^2 norm comparing both colors and labels. The algorithm produces impressive results on a variety of images. In [Ramanarayanan & Bala 2007], while achieving better quality and similar applications using patched based synthesis and energy minimization, the agreement metric remains the same L^2 norm on RGB labels.

While this work is of great inspiration to us, our approach has key differences. First, the use of RGB labels limits the number of classes which could be properly defined. Indeed, the different labels are averaged together during the multi-resolution synthesis process, leading to ambiguities. Most importantly, using the L^2 metric on neighborhoods made of discrete labels does not provide a faithful metric: Labels close in value are not necessarily visually similar. We address this problem by modifying all synthesis steps to truly consider discrete labels.

Second, in the texture-by-number scheme the user is responsible for painting the *target* map. Instead, we do not want to require the user to precisely draw shape outlines: Our approach automatically enriches with details the simplistic guidance map obtained by rendering the 3D proxy. This approach is hinted at in the work of Zhang et al. [Zhang *et al.* 2003], in which a binary map describing the shape of texture elements is synthesized prior to colors. However, in our case the map is multi-labeled and must correspond to the rendering of the proxy geometry.

Finally, for fast high-quality texture synthesis we rely on a parallel algorithm [Lefebvre & Hoppe 2005]. This algorithm exploits coherence during synthesis – in particular it uses only local information around each synthesized pixel during the neighborhood matching process (a k-coherent candidate mechanism [Tong *et al.* 2002]). However, adapting such an algorithm for guided synthesis has not been attempted before, and doing

so is non-trivial since the guide prevents the local search approach.

7.3 Input and Preprocessing

As input, we require simple proxy geometry for the scene elements that will use the texture-synthesis shading, a source image with the desired texture categories, and any desired standard 3D CG elements.

Proxy geometry: We provide a set of basic tools to quickly model an approximate scene. Since we focus our efforts on the rendering process, we restrict ourselves to the use of a simple *terrain tool* and *sphere tool* to create the proxy geometry for our examples. This limited toolset proved to be sufficiently flexible, and could be augmented or replaced by many other alternatives. The terrain tool simply pushes and pulls vertices of a height-field with a Gaussian region of influence with the distance to the cursor. The sphere tool instances spheres along a given path, thereby enabling the easy creation of topologies that are not possible with the heightfield model, such as the arches in Figure 7.6(middle).

Source photo: Given the geometry, the user needs to select the photo which will provide the rich detail for the texture-synthesis shading. The photo should have the suitable texture categories and have a roughly similar point of view as that desired for the proxy geometry. The texture categories in the source photo need to be labeled, which we accomplish through a segmentation process. In our case this involves about half an hour of manual work per image, which is a one-time preprocessing overhead for using a particular source image. In large scale application of the technique, we envisage users selecting from among a library of pre-segmented images. Although the accuracy of the segmentation is not critical, the details in the boundaries will be transfered to a synthesized smooth guide, so some details should be present in the input segmentation.

Proxy guide: An image of texture category labels is obtained by rendering the 3D proxy geometry into a *proxy guide*, using flat shading with the appropriate texture label associated with each component of the proxy. The labels should match those assigned to the segmented source image. In the illustrated images of the proxy guide, e.g., Figure 7.1, we visualize the discrete texture category labels using distinctive colors. However, all processes in the shading pipeline will treat labels as having strictly discrete semantics, rather than continuously-valued colors. A proxy depth map is created at the same time as the proxy guide, and will be leveraged later to allow for depth-consistent compositing into the rendered scheme.

7.4 Guidance synthesis

The first stage of the rendering pipeline produces a synthesized guide and an associated depth map. These contain important silhouette details which are not present in the guide produced by rendering the simple proxy geometry. As illustrated in Figure 7.2, this is accomplished using standard texture synthesis, initialized with a down-sampled version of the rendered proxy. The goal is to replace the smooth, unnatural borders and silhouettes of the proxy geometry with the richer border structure that is available from the source photo

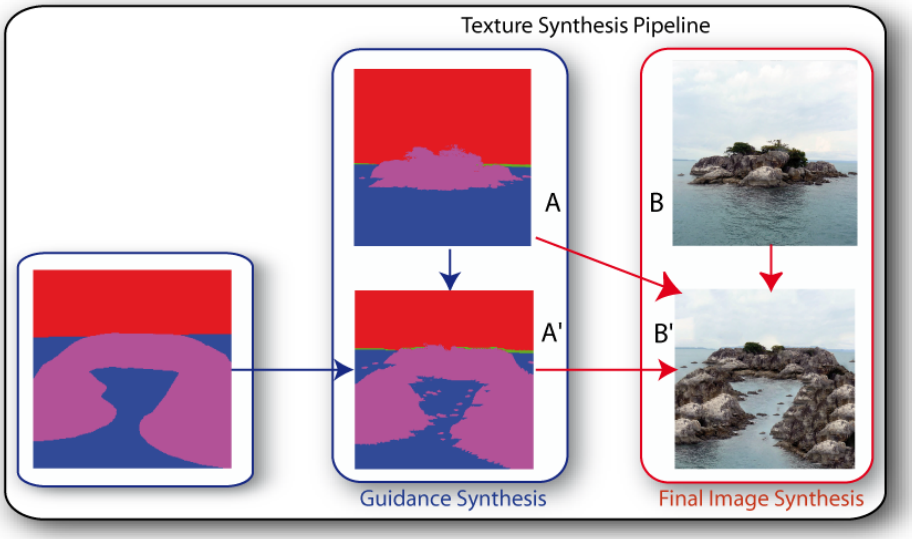


Figure 7.2: *The two-stage shading pipeline.*

via the source guide. However, the standard L^2-norms commonly used in texture synthesis are ill-adapted to this problem because the image pixels represent discrete texture category labels, and not continuously-valued colors. Addressing this issue requires a number of modifications to state-of-the-art texture synthesis algorithms. The guidance synthesis phase also needs to produce a depth map corresponding to the synthesized guide if we wish to be able to integrate other CG elements into the scene. Both stages employ texture synthesis methods which use guide images consisting of texture labels, not colors. As we describe next, we propose texture synthesis methods based on Chamfer distances to address this issue.

7.4.1 Chamfer distance

Chamfer distances compute a generalized distance between edges, or more generally, sets of points, and it is applied in many variations in shape matching applications [Barrow *et al.* 1977, Borgefors 1988]. It is given by the mean of the minimal distances of each point in one set to the closest point in another set. For our purposes, we define it as the sum over each pixel in neighborhood A of the L_∞-distance between the pixel in A and the closest pixel in neighborhood B sharing the same label. The symmetric Chamfer distance that we use is the sum of the Chamfer distance between neighborhoods A and B, and the distance between neighborhoods B and A. This allows us to account for differences in the geometry of the labels and for the fact that classes are intrinsically discrete. In cases where there is no corresponding label, we use an assigned distance corresponding to the size of the image neighborhood.

Figure 7.3 illustrates an example where the Chamfer distance computes a more mean-

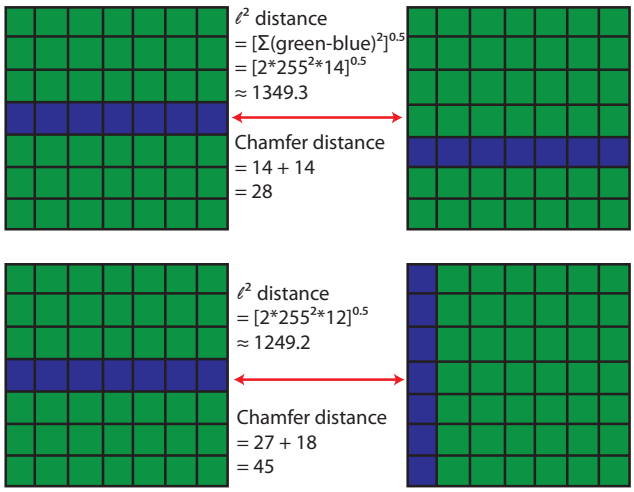


Figure 7.3: *A comparison of L^2 and Chamfer distances for two cases. According to the Chamfer distance, the top image pair is more similar, while an L^2 distance metric finds the bottom pair to be more similar.*

ingful result than an L^2 metric on pixel colors. The use of a binary same-category / different-category distance metric between pixels that effectively measures the spatial overlap of regions would also fair poorly in such cases. For non-overlapping pixels, any such a metric does not distinguish between a pixel being near to other pixels of the same label and being far away from such pixels. Any such metric which compares only directly corresponding pixels will be problematic for small texture regions, where overlap is a poor proxy for similarity. We also note that while the blending of color-based texture labels (as happens in the texture synthesis pyramids) can be seen as a type of mixed labeling, five of our six examples have more than the three texture labels that could be represented independently in an RGB color space.

7.4.2 Synthesis process

Our guidance map texture synthesis uses an approach similar to [Lefebvre & Hoppe 2005]. However, the use of discrete identifiers precludes building a multiresolution Gaussian stack or a Gaussian pyramid [Hertzmann *et al.* 2001]. We thus build a *voting stack* from which we can extract a *voting pyramid*. This step is performed by keeping the identifier which is the most present in a $N \times N$ kernel around each pixel. The kernel size is scaled by a factor of two at each level of the stack. The pyramid is extracted from the stack by sampling the stack every 2^l pixels, where l is the stack level.

We initialize the synthesis at a level which is not too coarse in order to preserve the large scale structure of the proxy guide. To synthesize a 256×256 guide, we start the synthesis at the 64×64 level. We do not use the jittering step of [Lefebvre & Hoppe 2005] and use 7×7 neighborhoods. We use a coherence $\kappa = 0.35$ and $k = 5$ coherent matches. These k coherent matches are found by choosing the k pixels that are furthest apart (in image

space) which are closest (in feature space) to the current pixels in a set of $4k$ candidates. This is performed using an Hochbaum-Shmoys heuristic in the pre-processing step without significant loss in performance.

7.4.3 Acceleration techniques

The use of an L^2 distance allows for the use of efficient libraries such as ANN[1] and acceleration structures such as kd-trees. However, dealing with non-Minkowski metrics such as the Chamfer distance makes their use problematic. In particular, kd-trees cannot be used since we cannot define planes in this space and decide where a point lies with respect to the plane. Also, the use of principle component analysis (PCA) in prior work for accelerating computations cannot be done in our discrete space. We implement an acceleration structure based on ball partitioning of the space of neighborhood [Samet 2005]. Note that this step only depends on the source image and can therefore be precomputed once and then reused for all synthesis done using this image. The use of more involved data structures such as vantage point-trees [Samet 2005] or GPU nearest neighbor searches [Garcia *et al.* 2008] would improve performance. For example, [Garcia *et al.* 2008] report a $120\times$ GPU-based speedup.

7.5 Final Image Synthesis

We now have synthesized a detailed guide containing silhouettes visually similar to the example, while following the layout of the proxy rendering. We next compute the final color image by a guided texture synthesis step, similar in spirit to [Hertzmann *et al.* 2001].

To provide reasonably fast feedback and state-of-the-art synthesis quality we choose to rely on a parallel texture synthesis algorithm [Lefebvre & Hoppe 2005]. This algorithm obtains best results by exploiting coherence during synthesis. That is, it tends to form patches during synthesis, and prunes the search space of neighborhood matching by only exploiting local information within the synthesized image.

However, it is not straight-forward to use the algorithm for guided synthesis. A key difficulty is that in an area with poor label matching the local search will only find neighborhoods with incorrect labels. To overcome this, our key insight is to initialize synthesis with an approximate result already enforcing labels. This will ensure that, locally, neighborhoods with appropriate labels are found, while synthesis will essentially improve colors.

Our algorithm proceeds in three steps: First, we use the example guide A (Fig. 7.4(a)) and the synthesized guide B (Fig. 7.4(b)) to grow color patches and approximate a first synthesis result (Fig. 7.4(c) and (d) for the color coded patches). While this approximation is reasonable in terms of matching labels, it will have many artifacts in the color channel. Second, we downsample the approximation and use it as an initialization for a parallel neighborhood matching synthesis scheme, using Chamfer distance to compare label neighborhoods (see Fig. 7.4(e)). For performance reasons we do not synthesize until the maximum resolution but stop at an intermediate level. Third, we supersample the synthesis

[1] `www.cs.umd.edu/~mount/ANN/`

result to recover the initial resolution, removing any seams through gradient transfer and Poisson stitching (see Fig. 7.4(g)).

Similarly to [Lefebvre & Hoppe 2006] we enrich the neighborhoods with the distance of each pixel to the closest contour in the label map, computed using [Danielsson 1980]. This helps synthesis better capture the image appearance around boundaries in the label map. The distance map is used in neighborhood comparisons.

Patch growth for initialization The purpose of this step is to grow patches on top of the synthesized guide (B in Fig. 7.4). At the beginning no pixels are covered by a patch. We randomly pick an uncovered pixel and find its closest match in the original guide (A), using a weighted combination of the Chamfer distance between labels and the distance map. We use the following distance:

$$d(p_A, p_B) = 2.4\ ||\mathcal{N}_{D_A}(p_A) - \mathcal{N}_{D_B}(p_B)||_2 + 10\ C(p_A, p_B) \tag{7.1}$$

where $D_{A,B}$ are the distance maps, $\mathcal{N}_{\mathcal{D}_\mathcal{A}}(p_A)$ the 7×7 neighborhood around p_A in D_A, and $C(p_A, p_B)$ the Chamfer distance between neighborhoods around p_A and p_B in the label maps I_A and I_B. We use the best match as a seed to perform a flood fill in both A and B which stops either at already covered pixels or when d is larger than a given threshold (typically, 25% more than the closest distance + 10). This gives us a patch around the uncovered pixel in B. Each patch defines a mapping between pixels in B and pixels in the example image (color and labels). This process iterates until all pixels in B are covered.

The result is a set of patches as shown color-coded in Fig. 7.4(d). These patches define an image correct in terms of labels, but with many artifacts in the color channels (Fig. 7.4(c)). This first step is performed at the target synthesis resolution, typically 256x256.

Pixel-Based Guided Synthesis The second step performs guided synthesis, similar in spirit to [Hertzmann *et al.* 2001] but using a parallel algorithm. We synthesize B' (Fig. 7.4(e)) using k-coherent synthesis, following [Lefebvre & Hoppe 2005]. As described previously, we use a voting stack for the Chamfer distances to perform multi-scale synthesis.

The distance metric is the same as in Eq. 7.1, augmented with the luminance channel (we found it unnecessary to compare RGB colors when selecting neighborhoods):

$$\begin{aligned} d(p_S, p_E) = \quad & 2.4\ ||\mathcal{N}_{L_S}(p_S) - \mathcal{N}_{L_E}(p_E)||_2 \\ & +1.4\ ||\mathcal{N}_{D_S}(p_S) - \mathcal{N}_{D_E}(p_E)||_2 \\ & +10\ C(p_S, p_E) \end{aligned} \tag{7.2}$$

where p_S is the pixel being synthesized and p_E the candidate in the example image, $L_{S,E}$ the luminance, $\mathcal{N}$ are 7×7 neighborhoods. The weights are determined experimentally. Synthesis starts at a coarse resolution of 32x32 and we typically use a coherence value $\kappa = 0.35$.

To reduce repetitions sometimes introduced by texture synthesis, we reject candidates which are already present in a 9x9 neighborhood around the current pixel in the image

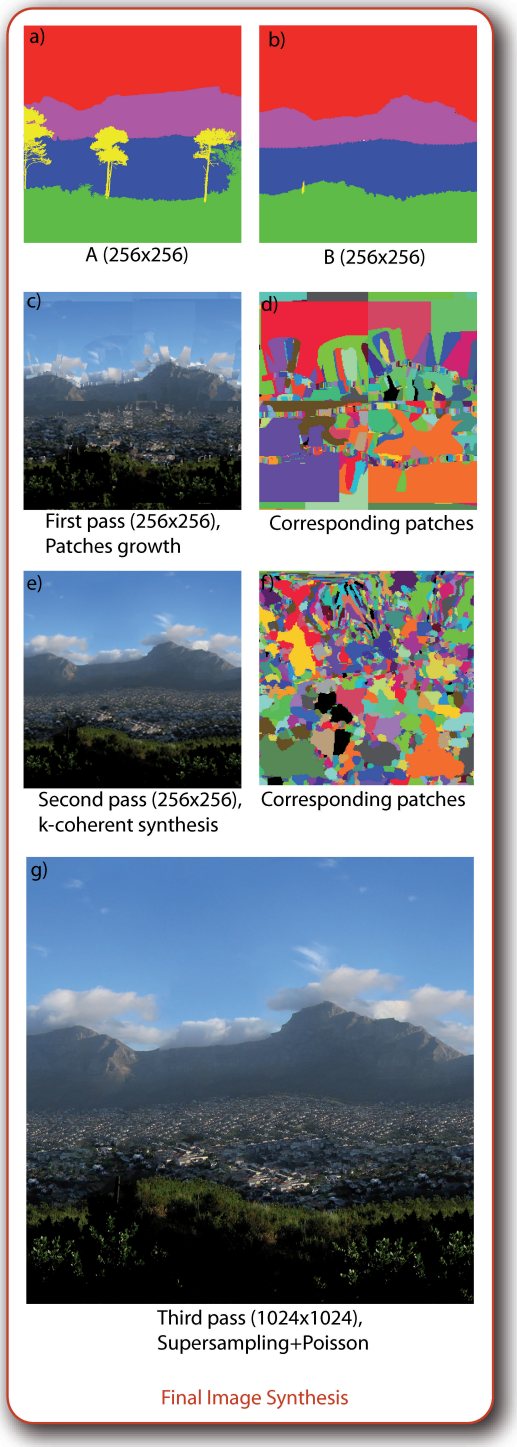


Figure 7.4: *A three step synthesis*

being synthesized. We also reject the candidate if any of its neighbor is present in a radius of 2^{l-1}, where l is the current synthesis level. We use 4 correction subpasses at each level of the pyramid.

Gradient transfer In the first two steps, we synthesize images at a resolution of 256x256 for efficiency. In the final pass we supersample the image to the resolution of the input image (typically 1024x1024) and perform a final Poisson synthesis step to attenuate any remaining artifacts.

The texture synthesis step can lead to visible seams when it is impossible to find good matching neighborhoods in the example. This often happens if a global gradient is present in the example image, for instance in the sky. In an approach inspired by [Agarwala *et al.* 2004] we transfer gradients instead of colors in areas where error is too high. We do this by building an error map consisting of the distance of each synthesized pixel to its best match in the photograph. We linearly supersample this error map to the photograph resolution and perform a Poisson synthesis step for the 60% of the pixels with the highest errors. The remaining 40% of the pixels remain unmodified since we are confident in their value. We force the gradient to zero at the boundary between patches, where it is not well defined. We also supersample the synthesized guide (Sec. 7.4.2) and ensure that boundaries between regions of different labels are left unchanged: We fix their values by using them as Dirichlet boundary conditions.

7.6 Compositing and Image Manipulation

Our approach supports several ways of enriching and manipulating the casually modeled scene. In particular, CG elements can be integrated with the synthesized elements and cast shadows into the scene. In addition, the layered representation of the scene allows image manipulation (e.g., changing the sky color) or other compositing effects (such as fog). Consistent depth values are needed for all pixels to enable these operations. We use the available depth map for the proxy guide to develop a depth map corresponding to the synthesized guide. First, the pixels in the synthesized guide that have the same labels as the corresponding pixels in the proxy guide are assigned the proxy depth. For each pixel in the remaining regions, we assign the depth of the closest pixel having the same label both in the proxy and the synthesized guide. This results in an extended depth map.

The depth allows us to directly composite 3D CG elements into the scene, using the camera parameters used to generate the proxy image. We can also perform simple shadow mapping using the depth. In the examples shown, we cast shadows from the CG elements using the extended depth map and attenuate the pixel values.

The label segmentation results in well defined layers. This allows us to interactively manipulate the colors of each layer, for example the sky. Fog can also be interactively composited into the scene. The boat example in Figure 7.7 shows these effects.

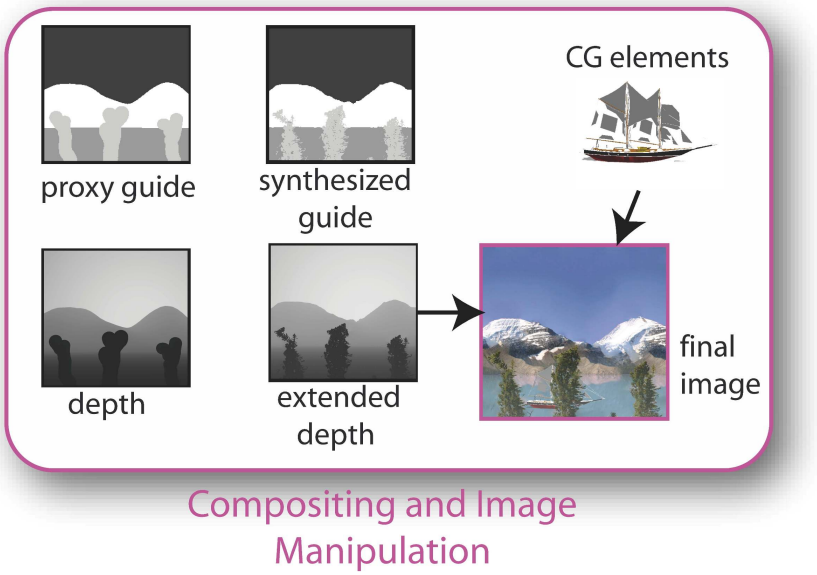


Figure 7.5: *Main elements of the manipulation phase.*

7.7 Results and Implementation

We model a number of example scenes to demonstrate the method. Figure 7.6 shows examples of icebergs, stone arches, and mountains. Two synthesized views are shown for each scene, and these views have approximate 3D consistency because of their shared use of the proxy geometry. The resulting scenes appear to have rich geometry, such as the complex silhouette and faceted appearance of the icebergs, despite the simple underlying proxy geometry.

Figure 7.7 further shows the successful integration of regular CG elements into the synthesized scenes. Figure 7.8 illustrates the ability to change the camera point of view for a synthesized iceberg scene.

7.7.1 Implementation

To solve the Poisson equation, we make use of the TAUCS library and its multifrontal sparse factorization method for symmetric positive definite matrices using 3 threads on the 1024×1024 final image (one thread of each color channel). The matrix is of size $p \times p$, where there are typically $p = 631$k pixels to be synthesized, with 5 non zero elements per row, which is solved in 38 seconds.

7.7.2 Performance

Given our implementation, we are able to synthesize 1024×1024 images in approximately $122s$. We summarize our current implementation performance in table 7.1. Although we parallelize part of the code on the CPU and our code benefits from inte-

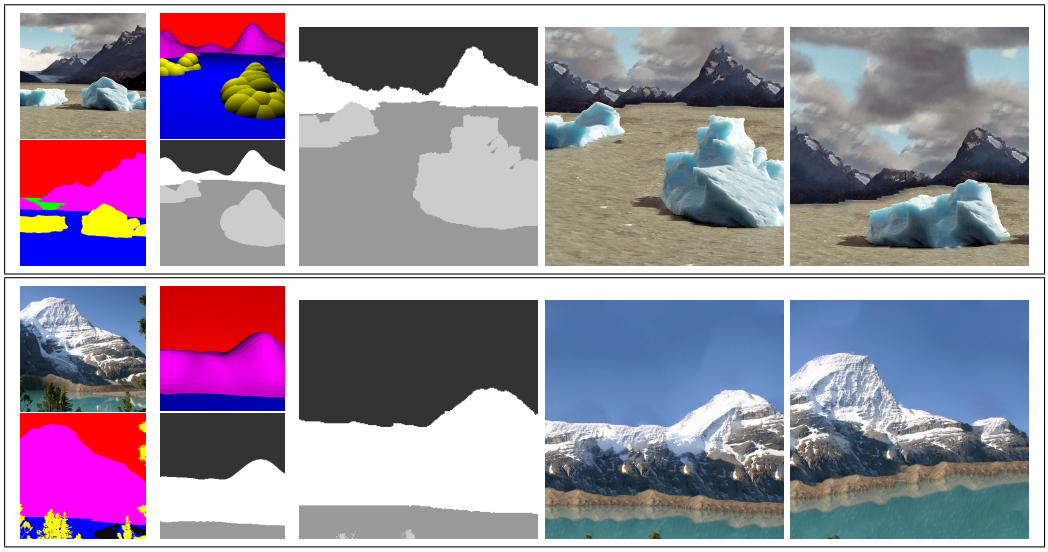


Figure 7.6: *Results, part one. The small images for each scene include (in clockwise order) the source photo, the 3D proxy, the proxy guide, and the source labels. The large images (from left to right) show the synthesized guide, the final image, and a synthesized image from another viewpoint.*

Op.	kNN	Stacks	Guide init	Synth	Img init	Img synth	Poisson	Depth
Time	1h	13s	12s	6s	13s	40s	38s	<1s
threads	1-4	1	4	4	4	4	3	1

Table 7.1: *Performance of each step: kNN refers to the precomputation which is partly single threaded. Stacks refer to the computation of various voting stack. These numbers are indicative and our code could directly benefit from massively multiprocessor architectures.*

ger computations, our code remains unoptimized since our focus is not on synthesis speed, but rather on a particular usage of texture synthesis and the final image quality. Many possibilities for acceleration exist, for example GPU methods such as multigrid Poisson resolution [McCann & Pollard 2008], or brute force nearest neighbor solutions [Garcia *et al.* 2008]; VP-trees [Samet 2005] or the use of recent multi-core architectures [Seiler *et al.* 2008].

7.7.3 Impact of Guide Synthesis

Figure 7.9 illustrates the result of image synthesis with and without the guide. The smooth silhouette of the proxy guide is largely preserved in the synthesis without the guide, which gives the image an artificial flavor. The guide synthesis adds the necessary detail to the guide, resulting in the addition of trees and the addition of small rocks surrounding the main island.

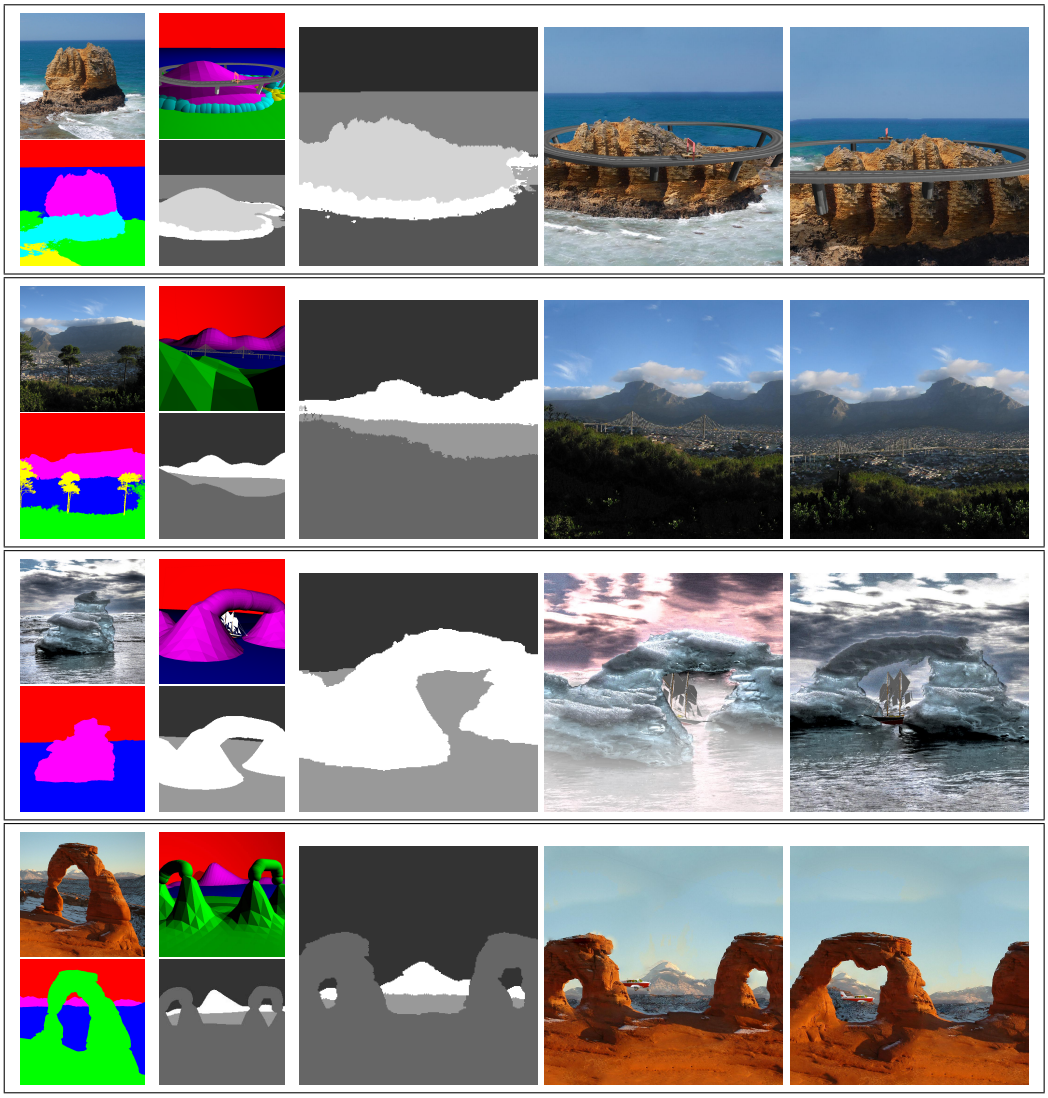


Figure 7.7: *Results, part two, illustrating depth-consistent integration of static or animated CG elements.*

Figure 7.8: *Camera rotating around an iceberg. Notice the consistent shadows and reflections on the water.*

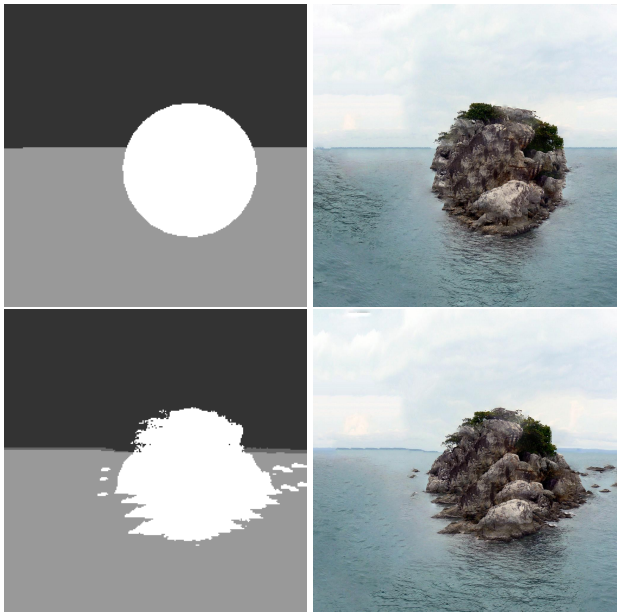


Figure 7.9: *Impact of guide synthesis for two examples. The first row shows the guide and synthesis result without guide synthesis. The second shows the same with guide synthesis.*

7.7.4 Comparison to Image Analogies

The third stage of our method, the guided texture synthesis step, is directly inspired by the texture-by-numbers approach suggested in the seminal image-analogies work of [Hertzmann *et al.* 2001]. Nevertheless, the third stage of our synthesis method differs from this previous work in a number of significant respects. This includes the use of Chamfer distance to respect the discrete nature of texture labels, the introduction of a pyramid based on a voting scheme, and the three-step synthesis.

To see the qualitative impact of these differences, we compare the result of substituting the original image analogies method of [Hertzmann *et al.* 2001] for our two step synthesis process, i.e., with the proxy guide as input. We use the original code made available by the authors and first verify its correct usage by reproducing results from the original paper. Figure 7.10 shows the results for a subset of our examples. While for some images the original image analogies method obtains reasonable results, other cases exhibit strong artifacts. We did significant experimentation with the κ parameter for the image analogies method in order to ensure that this was not the source of the visible artifacts. For our method we keep all parameters fixed to the previously described values Our own method is also subject to many limitations (also shared by Image Analogies) as we discuss next.

Although [Ramanarayanan & Bala 2007] shown improved quality results compared to Image Analogies, their use of patch based synthesis makes applications more difficult (see Section 7.8.2).

A comparison between the use of an L^2 distance over RGB colored labels and a Chamfer distance over discrete IDs in our pipeline is provided in Figure 7.11. This comparison

shows that, as expected, labels gets blended in the multiresolution process when using RGB colors as identifiers thus creating new undesirable regions. Our method, based on discrete IDs does not suffer from this problem.

Figure 7.10: *Output of Image Analogies. The stone arches did not use a blurred guide, while the other two images do. The results use the input given in Figures 7.7, 7.9, and 7.7, respectively.*

7.8 Limitations and Future work

7.8.1 Limitations

Two failure cases are illustrated in Figure 7.12. Our current segmentation does not distinguish between texture borders that arise from 3D adjacency and those that arise from occlusions. This is illustrated by the first failure example using the cliffs of Dover. The

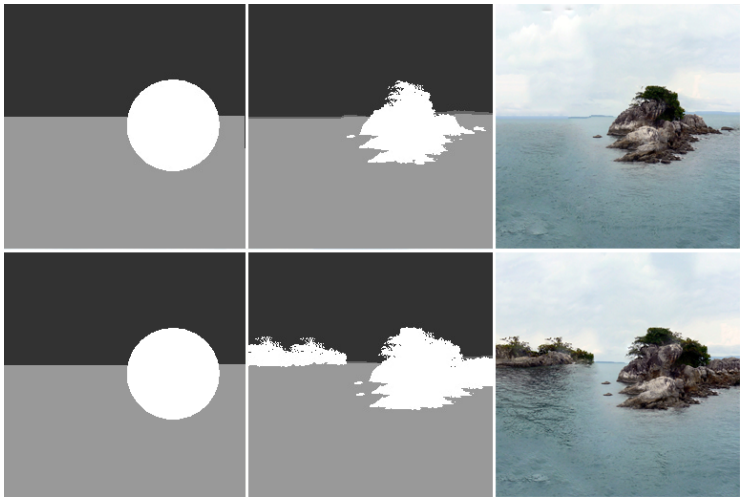


Figure 7.11: *From left to right: Initial guide, synthesized silhouettes, synthesized color result. Top row: Our pipeline using the Chamfer distance. Bottom row: Our pipeline using an L^2 distance. The multiresolution process blends labels together thus creating new undesirable regions between the sky and the sea on the left.*

final synthesized image matches the proxy guide by creating what is visually a tall cliff rather than the alternative shape modeled by the 3D proxy, consisting of a long coastline with sloping cliffs instead of plunging cliffs. The second failure example illustrates what happens when the relevant combinations of borders between texture classes are not shared between the source photo and the proxy guide. This particular view of the proxy demands a cyan-colored shoreline category leading down into a blue-colored water category. Because this combination does not exist in the source image, the synthesized image is of poor quality.

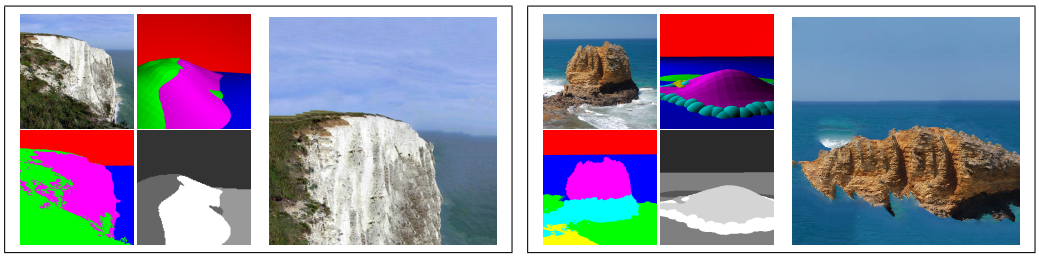


Figure 7.12: *Failure cases. Left: Cliffs of Dover. Right: Rocky Isle. The cause of failure is discussed in §7.7.*

Our method inherits many of the limitations of previous texture synthesis methods. In particular, due the lack of semantic information, the guide synthesis can lead to unnatural results such as trees with two trunks, given that the method does not model the kind of abstract constraints needed to prevent this. Artifacts are common along structured border regions such as the horizon line if this is placed at a different angle in the proxy guide than in the original image.

Our synthesized skies still occasionally yield artifacts which can be fixed by allowing the Poisson synthesis to be performed on larger areas or even on all pixels.

We are currently limited to working with a single source image to provide the visual detail. A promising direction of future work is to investigate the use of texture categories from multiple source images. Particular texture labels could be declared as being semantically equivalent, for example. This would potentially allow for much wider classes of texture-synthesis shading, as well as indexing based on other attributes, such as normals and lighting conditions.

7.8.2 Temporal coherence and lighting variations

Despite the fact that we can create multiple views, we do not solve the hard problem of temporal coherence. Solving this would allow for our tool to be usable in production rendering of animations and remains an exciting challenge.

Our latest implementation shows that we can handle the issue of temporal coherence by considering only local coherence. Our approximate extended depth allows reprojection of the synthesized pixels using a 3D connected mesh cut at label boundaries. Thus, only unoccluded pixels need to be resynthesized, and spatial coherence in newly synthesized pixels allow for a locally temporally coherent synthesis: although views appear coherent

when moving the camera, there is no guarantee that moving far away in the scene and coming back to the original position will yield exactly the same synthesized view. We further accelerated the synthesis process, partly by performing computations on the GPU, to allow for a synthesis in the order of a second.

However, we cannot currently provide complete freedom in viewing the 3D proxy. The viewpoint must be chosen so as to approximately preserve the same overall scale, occlusion ordering, and inclination angle as in the source photo.

We also tried using the same synthesized patches on different frames of a time lapse photograph, in order to provide lighting variations in a straightforward way. Although this approach could lead to discontinuities in areas where shadows have been cut by a patch at a different time of the day, this approach worked well in practice which is mainly due to the final Poisson synthesis.

We thus hope to provide a full image based interactive rendering pipeline with controled lighting, usable for production and games.

7.9 Conclusions

We have proposed a method for using texture synthesis as a shading technique which supplies rich visual detail to scenes modeled from simple geometry. Regular CG elements and texture-synthesis elements are integrated in a consistent fashion, and the layering of the naturally-segmented textured elements allows for easy further interactive image manipulation. In support of these goals, we introduce the notion of guidance synthesis, the use of a Chamfer distance metric as a principled means to achieve guided synthesis, and a three-step final synthesis method.

Conclusion

The two main goals of this thesis were the use of crossmodal perception to improve audio and visual algorithms, and the use of photographs to enrich visual CG renderings. As we have seen throughout this thesis, we believe that these goals were reached to a large extent.

In the first part, we used spatio-temporal tolerance windows for clustering and scheduling of sounds. We also studied the perception of audio-visual materials, and derived an LOD mechanism to perceptually choose both the visual and audio quality at the same time. We integrated these results into a game engine, demonstrating the practical interest of crossmodal algorithms.

In the second part, we used a single photograph to extract parameters and match the hair appearance for further CG renderings. We also used a photograph to infer its style for "casual modeling", and rapidly sketch and render 3D scenes.

The first part lasted two years, while the second part took the remaining third year of the PhD. We will thus conclude with some in depth insights on our crossmodal research, and more briefly conclude on the use of photographs for CG rendering.

Part I: Crossmodal Perceptual Rendering

As we have seen throughout this thesis, using perception can improve algorithm performance and can be used to validate algorithms. Our main results in crossmodal perception show that we can use spatio-temporal windows to provide flexibility in the audio rendering process, both spatially and temporally in order to speedup computations. We have also shown that the visual rendering of a material could be degraded if accompanied by a high quality sound which is cheaper to compute.

Insights

In general terms benefits of using crossmodal algorithms alone are somewhat lower than our original expectations. For example, it is unclear how the simple experiment we ran for crossmodal clustering (Chapter 2) directly extends to highly complex scenes.

A large body of crossmodal results have been obtained in the neuroscience community [Spence & Driver 2004] for extremely simple stimuli. Also, preliminary crossmodal effects have been shown in computer graphics [Tsingos *et al.* 2004] in more complex contexts. In consequence, the small crossmodal effect on complex scenes would have been difficult to predict on the basis of knowledge about the state of the art at the outset of this thesis.

During the first two years of the thesis, we also designed and ran a few other pilot crossmodal-related studies which remained unpublished due to the lack of significant results. For example, following the previous work which reported improvements in eye sensitivity in the presence of sound [Lippert *et al.* 2007, Stein *et al.* 1996, Bolognini *et al.* 2005, Vroomen *et al.* 2000], we developed a pilot test in which the subject had to detect a frog in a forest scene (Fig.7.13, left). The contrast of the frog was varied as well as its sound. Although we were able to reproduce results similar to [Stein *et al.* 1996] on a simple experiment [Baker 1949] including sounds, we did not obtain significant results using a more

complex environment such as the forest scene. Even if significant detection improvements with the sound had been found, it would have been difficult to directly use this effect for algorithmic gain. The use of a Threshold vs. Intensity (TVI) curve taking into account sounds when building a threshold map would have been a possible scenario. Studies have also shown the influence of sound quality on the perceived quality of degraded images [Storms & Zyda 2000]. We thus studied the impact of the presence of sound on the perceived quality of shadows in a baseball game (Fig.7.13, right) but did not obtain significant results.

Figure 7.13: *Left: A forest containing frogs used in an experiment. Right: A baseball bat projecting soft shadows for an experiment.*

Looking back, we believe that some of the main difficulties we encountered in perceptual and crossmodal research for computer graphics can be summarized as follows:

- Thresholds are small: Although perception can be used to improve algorithms by neglecting imperceptible details, the visual thresholds are small. In a paper not presented in this thesis, we used visual perception to control visual levels of detail [Drettakis *et al.* 2007]. However, this approach resulted in computational gain only in the case of highly complex scenes with high masking (due to shadows or tiny complex geometries such as in trees, see Fig.7.14, left). Audio thresholds tend to be higher; consequently, we found that it was more beneficial to control audio algorithms using perception [Moeck *et al.* 2007, Bonneel *et al.* 2008, Grelaud *et al.* 2009].

- Psychophysics use simple stimuli: Stimuli used in psychophysics are very simple, generally involving LEDs and beeping loudspeakers (Fig. 7.15, left) and results obtained *cannot be directly extended to more complex scenes*. The applied perception community uses somewhat more complex stimuli (Fig. 7.15, right), although still not reaching the complexity of virtual environments. However, cognitive science deals with complex stimuli, and up to now, mostly studied the attention shifts due to crossmodal events [Driver & Spence 1998]. We believe that an interesting application of studies on the crossmodal attention space is the design of interfaces, and it would be interesting to further investigate this domain. We proposed an initial contribution,

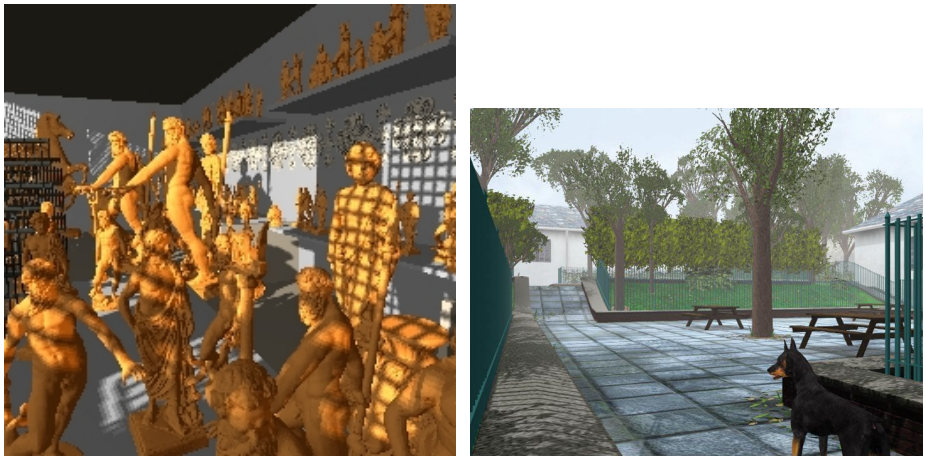


Figure 7.14: *Left: High masking due to shadows and geometry in complex scenes can be exploited for algorithmic gain [Drettakis et al. 2007]. Right: A virtual environment to cure dog phobia [Viaud-Delmon et al. 2008]*

not described in this thesis, by developing a virtual environment aimed at curing dog phobia [Viaud-Delmon *et al.* 2008], including barking dogs (see Fig.7.14, right). Ultimately, audiovisual conflicts could be used for this purpose.

- Results in psychophysics show small crossmodal effects: Although psychophysics show statistically significant and reproducible results when comparing unimodal and crossmodal conditions, these effects remain small (Fig.7.16). Enven though they help us understand brain mechanisms, this reduces the hope for high gain in computation time.

- Crossmodal effects are not easy to convert to algorithms: For example, in a paper not presented in this thesis, we demonstrated that reaction times are reduced when both audio and visuals are present when recognizing an object [Suied *et al.* 2009]. However, we did not find direct algorithmic applications, even in the context of task oriented performance measurements (e.g., a VR version of "finding Waldo").

- Experiments are hard to design: In [Bonneel *et al.* 2010], we found interesting audio-visual perceptual results by carefully designing the experiment. We went through several versions of the experimental protocol before defining the appropriate setup and corresponding question allowing us to identify the crossmodal effects. The use of two senses at the same time makes the experiments much harder to design: although side by side comparisons are commonly used in the visual perception community, this cannot be done when sounds are present. On the other hand, A/B comparisons used in audio research may not be appropriate (e.g., popping). Also, sounds are more meaningful in animated sequences which also makes comparisons harder. In addition, the task of the participant is not always well defined when both sounds and visuals are present. A concept merging both cues has to be found to

avoid having the participant focus on a single modality. For example, the concept of *materials* in [Bonneel *et al.* 2010] allowed us to merge impact sounds and visual representation of materials in a single more abstract representation, and ask a meaningful question to participants.

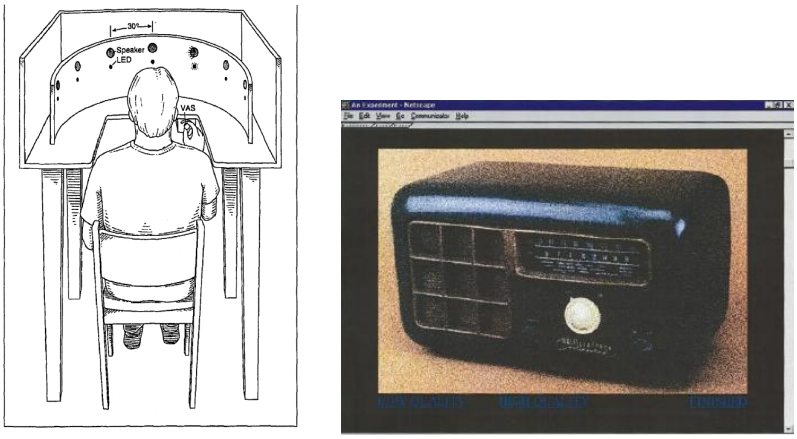


Figure 7.15: *Left: Experimental setup used in [Stein et al. 1996]: the participant is sat in front of equally spaced pairs of LED and loudspeakers, adjusting the intensity of a LED to match another one. Right: Experimental setup used in [Storms & Zyda 2000]. The image of the radio is presented at different qualities (noise added, or compression artifacts) and a corresponding sound is simultaneously emitted at different qualities (noise added or sampling rate varied).*

Much work remains to be done in the field of crossmodal audiovisual rendering. While the contributions we present on crossmodal interactions yield significant, albeit not spectacular, improvements to algorithms, we believe our results have put research on crossmodal effects for Virtual Environments on more solid ground. We hope that our current results will open a new way to crossmodal perceptual algorithms and provide an initial interesting scientific basis on the topic.

Future work

We believe that crossmodal effects could have a significant impact on user interfaces. Indeed, user's attention can be shifted toward particular locations with sounding and visual events. Also, studies reported smaller reaction times in presence of both sounds and visuals [Suied *et al.* 2009, Kinchla 1974] in detection tasks. This could be used to improve human's performance in virtual environments when performing particular tasks.

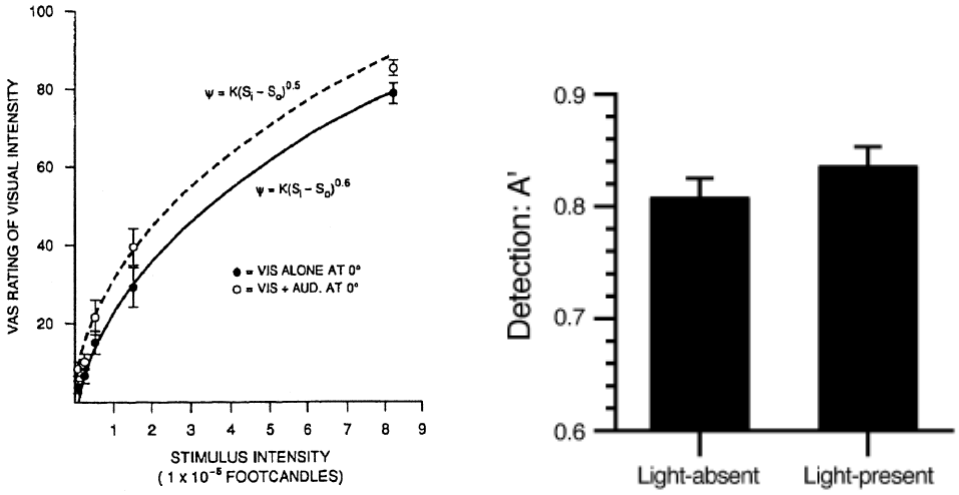


Figure 7.16: *Left: Results of [Stein et al. 1996] showing a statistically significant increase of perceived light intensity when accompanied by a sound. Right: Results of [Lovelace et al. 2003] showing a statistically significant decrease of sound detection threshold when accompanied by an irrelevant light.*

Part II: Visual rendering using a single photograph

Insight

In the second part of this thesis, we demonstrated how a single photograph could be used to enrich visual renderings. Specifically, we first tried to reproduce hair appearance, and then tried to reproduce a rendering of a sketched scene similar to a photograph. We believe that user guided synthesis, with the help of an input photograph, can drastically simplify the rendering process. Indeed, in modern rendering techniques and hardware setups for computer graphics, at one end of the technological spectrum we have highly complex and expensive setups. These setups provide very accurate renderings through image based techniques [Paris *et al.* 2008] or through best fits on physical measurements [Matusik *et al.* 2003] with computationally expensive rendering models (e.g., PBRT [Pharr & Humphreys 2004]), and are typically preferred by the film industry. At the other end of the spectrum, no input is required and simple rendering models are applied (e.g., ambient occlusion [Zhukov *et al.* 1998]), which is typically preferred by the game industry. We proposed a way to close the loop between real world photographs and simple rendering models using an intermediate solution, where the capture setup is extremely simple (a single photograph) and cheap rendering models are used. This provides an adequate solution for typical usage of low-end devices (PDA, mobile phones) or non expert users (avatar customization, sketching).

As we have seen in our work, a single photograph can be used either to find parameters of a physical simulation (the hair reflectance parameters [Bonneel *et al.* 2009a]), or directly to synthesize based on image pixels for rendering (Chapter 7). Although a single

image appears to contain too little information to fit a full 4D reflectance model which depends on the camera and light positions, a perceptual metric allowed us to succeed in this technical challenge. This drastically reduces the complexity of previous hair capture setups, needing a full dome of lights and cameras capturing more than 40,000 images [Paris *et al.* 2008]. We also used a single photograph to provide a tool for the "casual modeling" of 3D sketched scenes. The pixels from the photos are directly extracted by patches, and are placed and refined during the rendering. This provides a realistic rendering using the style of the photo.

Future work

One of our most promising results relates to the casual modeling of 3D scenes using an input photograph. Using this approach, a user can very quickly design a rough 3D scene in a few seconds, and have it realistically rendered in 2 minutes based on the style of a photograph. Synthetic 3D objects can then be integrated in this rendering, while properly handling occlusions.

We believe that this approach can be extended to realtime rendering, particularly using recent GPUs. Interactivity makes it interesting to have temporal coherence as well in order to continuously navigate in the scene. However, since a human is likely to forget small salient features location over a long period of time, a simpler local temporal coherence can be justified and easily achieved. This would provide continuous motions without requiring a full temporal coherence of small scale features. This would provide a fresh approach to computer graphics rendering, with potential applications in realtime 3D games.

More generally, the computer vision community has solved most problems related to 3D reconstruction from photograph. However, their solution typically deal with reconstructing the real world. We believe that it is important to have more tools to only infer the geometric style of a photograph without reconstructing the scene faithfully, thus allowing more freedom in the creation process. In particular, it is important to provide the tools to create a variety of *new objects*, albeit based on the input photographs. Other features could then be extracted from the photograph such as textures, materials and lighting conditions, in order to transfer them to synthesized new scenes.

APPENDIX A

Appendix

A.1 Some Elements of Distribution Theory

Applying a distribution T defined by a locally integrable function to a smooth test function f with local support, implies the following operation:

$$< T, f >= \int_{-\infty}^{\infty} Tf(x)dx. \tag{A.1}$$

A commonly used distribution is the Dirac distribution (note that this is not the Kronecker delta) which has value 0 everywhere, except at 0. $< \delta_k, f >= f(k)$ is commonly used in signal processing (Dirac combs). We use the following properties of distributions:

$$\delta_0 \star f = f \quad \delta_0^{(n)} \star f = f^{(n)} \tag{A.2}$$

where $f^{(n)}$ denotes the n^{th} derivative of f.

$$\delta_a(t) \star f(t) = f(t-a) \quad \delta_a^{(n)}(t) \star f(t) = f^{(n)}(t-a) \tag{A.3}$$

$$\mathcal{F}\left(f(t)g(t)\right) = \frac{1}{2\pi}\mathcal{F}(f(t)) \star \mathcal{F}(g(t)) \tag{A.4}$$

A.2 Formulas for energy computation

We present here several expressions which we use in the computation of energy for modes.

The instant energy of a mode is given by:

$$\begin{aligned}\int_{t_1}^{t_2} \left(\sin(\omega x)e^{-\alpha x}\right)^2 dx \quad = \quad & \frac{1}{4}\frac{e^{-2\alpha t_1}\left(\alpha^2+\omega^2-\alpha^2\cos(2\omega t_1)+\alpha\omega\sin(2\omega t_1)\right)}{\alpha(\alpha^2+\omega^2)} \\ & -\frac{1}{4}\frac{e^{-2\alpha t_2}\left(\alpha^2+\omega^2-\alpha^2\cos(2\omega t_2)+\alpha\omega\sin(2\omega t_2)\right)}{\alpha(\alpha^2+\omega^2)}\end{aligned} \tag{A.5}$$

To compute the total energy of two modes as $\|s\|^2 =< \sum_i a_i f_i, \sum_j a_j f_j >= \sum_i \sum_j a_i a_j < f_i, f_j >$, the expression $< f_i, f_j >$ is given below, using Eq. A.5:

$$\begin{aligned}\int_0^{\infty} \left(\sin(\omega_1 x)e^{-\alpha_1 x}\right).\left(\sin(\omega_2 x)e^{-\alpha_2 x}\right) dx = \\ \frac{2\omega_1\omega_2(\alpha_1+\alpha_2)}{((\alpha_1+\alpha_2)^2+(\omega_1-\omega_2)^2)\left((\alpha_1+\alpha_2)^2+(\omega_1+\omega_2)^2\right)}\end{aligned} \tag{A.6}$$

Similarly, the scalar product of two modes in a given interval (t,t+dt) is given as follows (we substitute α_2 by $\alpha_2 + \alpha_1$ and ω_2 by $\omega_2 + \omega_1$):

$$\begin{aligned}\int_t^{t+dt} \left(\sin(\omega_1 x)e^{-\alpha_1 x}\right) \; & . \left(\sin(\omega_2 x)e^{-\alpha_2 x}\right) dx = \\ & (e^{-\alpha_2\,(t+dt)}\,((\omega_2^3 - 2\,\omega_1\,\omega_2^2 \\ & +\alpha_2^2\,\omega_2 - 2\,\alpha_2^2\,\omega_1)\,\sin((t+dt)\,\omega_2 + (-2\,t - 2\,dt)\,\omega_1) \\ & +(-\alpha_2\,\omega_2^2 - \alpha_2^3)\,\cos((t+dt)\,\omega_2 + (-2\,t - 2\,dt)\,\omega_1) \\ & +e^{\alpha_2\,dt}\,((-\omega_2^3 + 2\,\omega_1\,\omega_2^2 - \alpha_2^2\,\omega_2 + 2\,\alpha_2^2\,\omega_1)\,\sin(t\,\omega_2 - 2\,t\,\omega_1) \\ & +(\alpha_2\,\omega_2^2 + \alpha_2^3)\,\cos(t\,\omega_2 - 2\,t\,\omega_1) + (\omega_2^3 - 4\,\omega_1\,\omega_2^2 \\ & +(4\,\omega_1^2 + \alpha_2^2)\,\omega_2)\,\sin(t\,\omega_2) \\ & +(-\alpha_2\,\omega_2^2 + 4\,\alpha_2\,\omega_1\,\omega_2 - 4\,\alpha_2\,\omega_1^2 - \alpha_2^3)\,\cos(t\,\omega_2)) \\ & +(-\omega_2^3 + 4\,\omega_1\,\omega_2^2 + (-4\,\omega_1^2 - \alpha_2^2)\,\omega_2)\,\sin((t+dt)\,\omega_2) \\ & +(\alpha_2\,\omega_2^2 - 4\,\alpha_2\,\omega_1\,\omega_2 + 4\,\alpha_2\,\omega_1^2 + \alpha_2^3)\,\cos((t+dt)\,\omega_2))) \\ & /(2\,\omega_2^4 - 8\,\omega_1\,\omega_2^3 \\ & +(8\,\omega_1^2 + 4\,\alpha_2^2)\,\omega_2^2 - 8\,\alpha_2^2\,\omega_1\,\omega_2 + 8\,\alpha_2^2\,\omega_1^2 + 2\,\alpha_2^4)\end{aligned} \tag{A.7}$$

This expression can be computed, after appropriate factorization using 17 additions, 24 multiplications, 8 cosine/sine operations, 2 exponentials and 1 division.

We then found a simpler expression for the above formula:

$$\begin{aligned}\int_{t_1}^{t_2} \left(\sin(\omega_1 x)e^{-\alpha_1 x}\right) . \left(\sin(\omega_2 x)e^{-\alpha_2 x}\right) dx = \\ (e^{-t_1\,(\alpha_1+\alpha_2)} - e^{-t_2\,(\alpha_1+\alpha_2)}) \int_0^{\infty} \left(\sin(\omega_1 x)e^{-\alpha_1 x}\right) . \left(\sin(\omega_2 x)e^{-\alpha_2 x}\right) dx\end{aligned} \tag{A.8}$$

This result is much more computationally efficient. In particular, part of it does not depend on time, and can thus be precomputed.

Bibliography

[Agarwal *et al.* 2003] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie and Henrik Wann Jensen. *Structured importance sampling of environment maps*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, pages 605–612, 2003. 81

[Agarwala *et al.* 2004] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin and Michael Cohen. *Interactive digital photomontage*. ACM Transactions on Graphics (ACM SIGGRAPH 2004), vol. 23, no. 3, pages 294–302, 2004. 117

[Alais & Burr 2004] D. Alais and D. Burr. *The ventriloquism effect results from near-optimal bimodal integration*. Current Biology, vol. 14, pages 257–262, 2004. 20

[Alais & Carlile 2005] D. Alais and S. Carlile. *Synchronizing to real events: subjective audiovisual alignment scales with perceived auditory depth and speed of sound*. Proceedings of the National Academy of Sciences of the USA, vol. 102, no. 6, pages 2244–7, 2005. 20

[Arthur & Vassilvitskii 2007] D. Arthur and S. Vassilvitskii. *k-means++: The advantages of careful seeding*. In ACM-SIAM Symposium on Discrete Algorithms, pages 1027–1035, 2007. 93

[Ashikhmin 2001] Michael Ashikhmin. *Synthesizing natural textures*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pages 217–226, New York, NY, USA, 2001. ACM. 110

[Baker 1949] Howard Dehaven Baker. *The Course of Foveal Light Adaptation Measured by the Threshold Intensity Increment*. Journal of the Optical Society of America, vol. 39, no. 2, pages 172–179, 1949. 127

[Barrow *et al.* 1977] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles and H. C. Wolf. *Parametric correspondence and chamfer matching: Two new techniques for image matching*. In Proceedings of the 5th International Joint Conference on Artificial Intelligence, pages 659–663, 1977. 112

[Begault 1999] Durand Begault. *Auditory and non-auditory factors that potentially influence virtual acoustic imagery*. In Proceedings of the AES 16th International Conference on Spatial Sound Reproduction, pages 13–26, 1999. 20

[Ben-Artzi *et al.* 2006] Aner Ben-Artzi, Ryan Overbeck and Ravi Ramamoorthi. *Real-time BRDF editing in complex lighting*. In ACM Transactions on Graphics (ACM SIGGRAPH 2006), pages 945–954, 2006. 55

[Bertails 2006] Florence Bertails. *Simulation de Chevelures Virtuelles*. PhD thesis, Institut National Polytechnique de Grenoble, 2006. 89

[Blauert 1997] J. Blauert. Spatial hearing : The psychophysics of human sound localization. M.I.T. Press, Cambridge, MA, 1997. 27

[Bolin & Meyer 1995] Mark R. Bolin and Gary W. Meyer. *A frequency based ray tracer*. In SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages 409–418, New York, NY, USA, 1995. ACM. 19, 20

[Bolognini *et al.* 2005] Nadia Bolognini, Francesca Frassinetti, Andrea Serino and Elisabetta Làdavas. *"Acoustical vision" of below threshold stimuli: interaction among spatially converging audiovisual inputs*. Experimental Brain Research, vol. 160, no. 3, pages 273–282, January 2005. 5, 127

[Bonneel *et al.* 2008] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon and Doug James. *Fast Modal Sounds with Scalable Frequency-Domain Synthesis*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), vol. 27, no. 3, pages 1–9, August 2008. 4, 6, 33, 128

[Bonneel *et al.* 2009a] Nicolas Bonneel, Sylvain Paris, Michiel van de Panne, Frédo Durand and George Drettakis. *Single Photo Estimation of Hair Appearance*. Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering), June 2009. 4, 7, 88, 97, 131

[Bonneel *et al.* 2009b] Nicolas Bonneel, Michiel van de Panne, Sylvain Lefebvre and George Drettakis. *A Texture-Synthesis Approach for Casual Modeling*. Submitted, 2009. 4, 7

[Bonneel *et al.* 2010] Nicolas Bonneel, Clara Suied, Isabelle Viaud-Delmon and George Drettakis. *Bimodal perception of audio-visual material properties for virtual environments*. ACM Transactions on Applied Perception (in press), 2010. 4, 6, 53, 129, 130

[Borgefors 1988] G. Borgefors. *Hierarchical chamfer matching: A parametric edge matching algorithm*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pages 849–865, 1988. 112

[Brainard *et al.* 2008] David Brainard, Larry Maloney and Anya Hurlbert, editeurs. Perception of material properties in 3d scenes (workshop), Pennsylvania, 2008. 21

[Chiu 2008] Dey-Fuch Chiu. Penta G - a game engine for real-time rendering research. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, June 2008. 72, 78, 80

[Cohen *et al.* 2000] Jonathan M. Cohen, John F. Hughes and Robert C. Zeleznik. *Harold: a world made of drawings*. In NPAR '00: Proceedings of the 1st International Symposium on Non-Photorealistic Animation and Rendering, pages 83–90, New York, NY, USA, 2000. ACM. 109

[Cook & Torrance 1982] R. L. Cook and K. E. Torrance. *A Reflectance Model for Computer Graphics*. ACM Transactions on Graphics, vol. 1, no. 1, pages 7–24, 1982. 54

[Danielsson 1980] P. E. Danielsson. *Euclidean distance mapping*. Computer Graphics and Image Processing, vol. 14, pages 227–248, 1980. 115

[Debevec *et al.* 2000] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin and Mark Sagar. *Acquiring the Reflectance Field of a Human Face*. In Kurt Akeley, editeur, Proceedings of ACM SIGGRAPH 2000, pages 145–156. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000. 3

[Debevec 1998] Paul Debevec. *Rendering with natural light*. In ACM SIGGRAPH 98 Electronic art and animation catalog, page 166, New York, NY, USA, 1998. ACM Press. 56

[Dellepiane *et al.* 2008] Matteo Dellepiane, Nico Pietroni, Nicolas Tsingos, Manuel Asselot and Roberto Scopigno. *Reconstructing head models from photographs for individualized 3D-audio processing*. Computer Graphics Forum (Proceedings of Pacific Graphics), vol. 27, no. 7, pages 1719–1727, 2008. 14

[Disney 2009] Disney. `http://adisney.go.com/disneyvideos/animatedfilms/wall-e/media/downloads/WALLEProductionNotes.pdf`, *Wall-E production notes, accessed June*, 2009. 3

[Donnelly & Lauritzen 2006] William Donnelly and Andrew Lauritzen. *Variance shadow maps*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pages 161–165, New York, NY, USA, 2006. ACM. 56

[Donner *et al.* 2008] C. Donner, T. Weyrich, E. D'Eon, R. Ramamoorthi and S. Rusinkiewicz. *A layered, heterogeneous reflectance model for acquiring and rendering human skin*. ACM Transactions on Graphics (ACM SIGGRAPH ASIA 2008), 2008. 90

[Drettakis *et al.* 2007] George Drettakis, Nicolas Bonneel, Carsten Dachsbacher, Sylvain Lefebvre, Michael Schwarz and Isabelle Viaud-Delmon. *An Interactive Perceptual Rendering Pipeline using Contrast and Spatial Masking*. In Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering). Eurographics, June 2007. 4, 19, 128, 129

[Driver & Spence 1998] Jon Driver and Charles Spence. *Attention and the crossmodal construction of space*. Trends in Cognitive Sciences, vol. 2, no. 7, pages 254–262, July 1998. 128

[Ducheneaut *et al.* 2009] Nicolas Ducheneaut, Ming-Hui Wen, Nicholas Yee and Greg Wadley. *Body and mind: a study of avatar personalization in three virtual worlds*.

In CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, pages 1151–1160, New York, NY, USA, 2009. ACM. 7, 88

[Fleming *et al.* 2003] Roland W. Fleming, Ron O. Dror and Edward H. Adelson. *Real-world illumination and the perception of surface reflectance properties*. Journal of Vision, vol. 3, no. 5, pages 347–368, July 2003. 21, 22, 56, 81

[Fouad *et al.* 1997] H. Fouad, J.K. Hahn and J.A. Ballas. *Perceptually Based Scheduling Algorithms for Real-time Synthesis of complex sonic environments*. Proceedings of International Conference on Auditory Display, 1997. 21

[Fujisaki *et al.* 2004] Waka Fujisaki, Shinsuke Shimojo, Makio Kashino and Shin'ya Nishida. *Recalibration of audiovisual simultaneity*. Nature Neuroscience, vol. 7, no. 7, pages 773–778, July 2004. 20

[Funkhouser *et al.* 1999] Thomas Funkhouser, Patrick Min and Ingrid Carlbom. *Real-time acoustic modeling for distributed virtual environments*. In Proceedings of ACM SIGGRAPH 99, pages 365–374, 1999. 13

[Funkhouser *et al.* 2004] T. Funkhouser, N. Tsingos, I. Carlbom, G. Elko, M. Sondhi, J. West, G. Pingali, P. Min and A. Ngan. *A Beam Tracing Method for Interactive Architectural Acoustics*. Journal of the Acoustical Society of America, vol. 115, no. 2, pages 739–756, February 2004. 13

[Garcia *et al.* 2008] Vincent Garcia, Eric Debreuve and Michel Barlaud. *Fast k Nearest Neighbor Search using GPU*, Apr 2008. 114, 119

[Gardner *et al.* 2003] A. Gardner, C. Tchou, T. Hawkins and P. Debevec. *Linear light source reflectometry*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, no. 3, pages 749–758, 2003. 90

[Giordano & McAdams 2006] Bruno L. Giordano and Stephen McAdams. *Material identification of real impact sounds: Effects of size variation in steel, glass, wood, and plexiglass plates*. Journal of the Acoustical Society of America, vol. 119, no. 2, pages 1171–1181, 2006. 21, 57, 58

[Grabli *et al.* 2002] S. Grabli, F. Sillion, S. R. Marschner and J. E. Lengyel. *Image-based hair capture by inverse lighting*. In Proceedings of Graphics Interface, 2002. 89

[Green 2003] Robin Green. *Spherical Harmonic Lighting: The Gritty Details*. Archives of the Game Developers Conference, March 2003. 55

[Grelaud *et al.* 2009] David Grelaud, Nicolas Bonneel, Michael Wimmer, Manuel Asselot and George Drettakis. *Efficient and Practical Audio-Visual Rendering for Games using Crossmodal Perception*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), New York, NY, USA, 2009. ACM. 4, 6, 69, 71, 128

[Grewin 1993] C. Grewin. *Methods for Quality Assessment of Low Bit-Rate Audio Codecs*. In Proceedings of the AES 12nd International Conference, pages 97–107, 1993. 28

[Guest *et al.* 2002] Steve Guest, Caroline Catmur, Donna Lloyd and Charles Spence. *Audiotactile interactions in roughness perception*. Experimental Brain Research, vol. 146, pages 161–171, 2002. 22

[Guski & Troje 2003] R. Guski and N.F. Troje. *Audiovisual phenomenal causality*. Perception and Psychophysics, vol. 65, no. 5, pages 789–800(12), July 2003. 5, 20, 45, 48

[Hairston *et al.* 2003] W. D. Hairston, M. T. Wallace, J. W. Vaughan, B. E. Stein, J. L. Norris and J. A. Schirillo. *Visual Localization Ability Influences Cross-Modal Bias*. Journal of Cognitive Neuroscience, vol. 15, no. 1, pages 20–29, 2003. 4, 5, 20, 21

[Halal & Schoon 2001] John Halal and Douglas D. Schoon. Hair structure and chemistry simplified. Cengage Learning, 2001. 89

[Hartley & Zisserman 2004] R. I. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, ISBN: 0521540518, second édition, 2004. 7

[Herder 1999] Jens Herder. *Optimization of Sound Spatialization Resource Management through Clustering*. The Journal of Three Dimensional Images, 3D-Forum Society, vol. 13, no. 3, pages 59–65, September 1999. 26

[Hertzmann *et al.* 2001] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless and David H. Salesin. *Image Analogies*. In Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pages 327–340, August 2001. 110, 113, 114, 115, 121

[Hochbaum & Schmoys 1985] Dorit S. Hochbaum and David B. Schmoys. *A best possible heuristic for the ık-center problem*. Mathematics of Operations Research, vol. 10, no. 2, pages 180–184, May 1985. 13

[Hormander 1983] Lars Hormander. The analysis of linear partial differential operators i. Springer-Verlag, 1983. 38, 39

[Howell 1992] David C. Howell. Statistical methods for psychology. PWS-Kent, 1992. 49

[HPD 2008] HPD. `http://graphics.ucsd.edu/~will/download/HairPhotobooth/` *Hair photobooth data [Paris et al. 2008], accessed June*, 2008. 90

[InCrysis 2009] InCrysis. `http://www.incrysis.com/index.php?option=com_content\&task=view\&id=559`*, interview with C.Yerli, CEO at Crytek, accessed June*, 2009. 3

[International Telecom. Union 1994] International Telecom. Union. *Methods for subjective assessment of small impairments in audio systems including multichannel sound systems.* Rapport technique, International Telecom. Union, 1994. 28

[International Telecom. Union 2003] International Telecom. Union. *Method for the subjective assessment of intermediate quality level of coding systems*. Recommendation ITU-R BS.1534-1, 2001-2003. 48, 49

[IRCAM 2009] IRCAM. *http://recherche.ircam.fr/equipes/salles/listen/, The LISTEN HRTF Database, accessed June*, 2009. 13, 27, 48

[James *et al.* 2006] Doug L. James, Jernej Barbic and Dinesh K. Pai. *Precomputed acoustic transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources*. ACM Transactions on Graphics (ACM SIGGRAPH 2006), vol. 25, no. 3, pages 987–995, July 2006. 17, 34, 46, 57

[Kajiya & Kay 1989] J. T. Kajiya and T. L. Kay. *Rendering fur with three dimensional textures*. In Computer Graphics (Proceedings of SIGGRAPH 89), pages 271–280, 1989. 89

[Katz 2001] B. Katz. *Boundary element method calculation of individual head-related transfer function. par I: Rigid model calculation*. Journal of the Acoustical Society of America, vol. 110, no. 5, pages 2440–2448, 2001. 14

[Kautz *et al.* 2002] J. Kautz, P. Sloan and J. Snyder. *Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics*. In Proceedings of the 13th Eurographics workshop on Rendering, pages 291–296, 2002. 54, 55, 56, 58

[Kinchla 1974] R. A. Kinchla. *Detecting target elements in multielement arrays - A confusability model (visual letter detection tasks)*. Perception and Psychophysics, vol. 15, pages 149–158, February 1974. 5, 130

[Klatzky *et al.* 2000] R. Klatzky, D. Pai and E. Krotkov. *Perception of material from contact sounds*. Presence: Teleoperators and Virtual Environments, pages 399–410, 2000. 6, 21, 54

[Kristensen *et al.* 2005] Anders Wang Kristensen, Tomas Akenine-Möller and Henrik Wann Jensen. *Precomputed local radiance transfer for real-time lighting design*. ACM Transactions on Graphics (ACM SIGGRAPH 2005), vol. 24, no. 3, pages 1208–1215, 2005. 55

[Larcher 2001] V. Larcher. *Techniques de spatialisation des sons pour la réalité virtuelle*. PhD thesis, Université Paris 6 (Pierre et Marie Curie), 2001. 14

[Larsson *et al.* 2002] P. Larsson, D. Västfjäll and M. Kleiner. *Better presence and performance in virtual environments by improved binaural sound rendering*. Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, pages 31–38, June 2002. 34

[Lefebvre & Hoppe 2005] Sylvain Lefebvre and Hugues Hoppe. *Parallel controllable texture synthesis*. ACM Transactions on Graphics (ACM SIGGRAPH 2005), vol. 24, no. 3, pages 777–786, 2005. 110, 113, 114, 115

[Lefebvre & Hoppe 2006] Sylvain Lefebvre and Hugues Hoppe. *Appearance-space texture synthesis*. In ACM Transactions on Graphics (ACM SIGGRAPH 2006), pages 541–548, New York, NY, USA, 2006. ACM. 115

[Lewald *et al.* 2001] Jörg Lewald, Walter H. Ehrenstein and Rainer Guski. *Spatio-temporal constraints for auditory–visual integration.* Behavioural Brain Research, vol. 121, no. 1-2, pages 69–79, 2001. 21

[Lippert *et al.* 2007] M. Lippert, N. K. Logothetis and C. Kayser. *Improvement of visual contrast detection by a simultaneous sound.* Brain research, vol. 1173, pages 102–109, October 2007. 5, 127

[Lokki *et al.* 2002] Tapio Lokki, Lauri Savioja, Riitta Väänänen, Jyri Huopaniemi and Tapio Takala. *Creating Interactive Virtual Auditory Environments*. IEEE Computer Graphics and Applications, vol. 22, no. 4, pages 49–57, 2002. 13

[L'Oréal 2008] L'Oréal. *http://www.hair-science.com, Hair Science, accessed Nov*, 2008. 89, 91

[Lovelace *et al.* 2003] C. T. Lovelace, B. E. Stein and M. T. Wallace. *An irrelevant light enhances auditory detection in humans: a psychophysical analysis of multisensory integration in stimulus detection.* Cognitive brain research, vol. 17, no. 2, pages 447–453, July 2003. 5, 131

[Luebke & Hallen 2001] David Luebke and Benjamin Hallen. *Perceptually Driven Simplification for Interactive Rendering*. In Proceedings of the 12nd Eurographics workshop on Rendering, pages 223–234, June 2001. 19

[Luebke *et al.* 2002] David Luebke, Benjamin Watson, Jonathan D. Cohen, Martin Reddy and Amitabh Varshney. Level of detail for 3d graphics. Elsevier Science Inc., New York, NY, USA, 2002. 4, 19

[Marks *et al.* 1997] J. Marks, B. Mirtich, B. Andalman, H. P Ster, S. Gibson, J. Hodgins, T. Kang, P. A. Beardsley, W. Ruml and W. Freeman. *Design galleries: A general approach to setting parameters for computer graphics and animation*. In Proceedings of ACM SIGGRAPH 1997, pages 389–400, 1997. 90

[Marschner *et al.* 2003] Stephen R. Marschner, Henrik Wann Jensen, Mike Cammarano, Steve Worley and Pat Hanrahan. *Light scattering from human hair fibers*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, no. 3, pages 780–791, 2003. 88, 89, 90, 91, 92, 94

[Mastoropoulou *et al.* 2005] Georgia Mastoropoulou, Kurt Debattista, Alan Chalmers and Tom Troscianko. *The influence of sound effects on the perceived smoothness of rendered animations*. In APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization, pages 9–15, New York, NY, USA, 2005. ACM. 22

[Mat & Visvalingam 2002] Ruzinoor Che Mat and Mahesh Visvalingam. *Effectiveness of Silhouette Rendering Algorithms in Terrain Visualisation*. In Proceedings of the National Conference on Computer Graphics and Multimedia (CoGRAMM; Melaka, October 2002), 2002. 109

[Matusik *et al.* 2003] Wojciech Matusik, Hanspeter Pfister, Matt Brand and Leonard McMillan. *A Data-Driven Reflectance Model*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, no. 3, pages 759–769, July 2003. 3, 54, 58, 131

[McAdams *et al.* 2004] S. McAdams, A. Chaigne and V. Roussarie. *The psychomechanics of simulated sound sources: Material properties of impacted bars*. Journal of the Acoustical Society of America, vol. 115, pages 1306–1320, March 2004. 57

[McCann & Pollard 2008] James McCann and Nancy S. Pollard. *Real-Time Gradient-Domain Painting*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), vol. 27, no. 3, August 2008. 119

[Mihashi *et al.* 2003] T. Mihashi, C. Tempelaar-Lietz and G. Borshukov. *Generating Realistic Human Hair for "The Matrix Reloaded"*. In ACM SIGGRAPH 2003 Sketches and Applications Program, 2003. 88

[Moeck *et al.* 2007] Thomas Moeck, Nicolas Bonneel, Nicolas Tsingos, George Drettakis, Isabelle Viaud-Delmon and David Aloza. *Progressive Perceptual Audio Rendering of Complex Scenes*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pages 189–196. ACM, April 2007. 4, 5, 13, 14, 25, 26, 30, 34, 40, 44, 46, 48, 72, 128

[Møller 1992] Henrik Møller. *Fundamentals of Binaural Technology*. Applied Acoustics, vol. 36, pages 171–218, 1992. 27

[Moon & Marschner 2006] J.T. Moon and S.R. Marschner. *Simulating multiple scattering in hair using a photon mapping approach*. ACM Transactions on Graphics (ACM SIGGRAPH 2006), vol. 25, no. 3, pages 1067–1074, 2006. 88, 89, 95

[Moon *et al.* 2008] Jonathan T. Moon, Bruce Walter and Stephen R. Marschner. *Efficient Multiple Scattering in Hair Using Spherical Harmonics*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), vol. 27, no. 3, 2008. 88, 89, 100

[Myszkowski 1998] Karol Myszkowski. *The Visible Differences Predictor: applications to global illumination problems*. In Proceedings of the 9th Eurographics workshop on Rendering, pages 223–236, June 1998. 19

[Ng *et al.* 2003] Ren Ng, Ravi Ramamoorthi and Pat Hanrahan. *All-frequency shadows using non-linear wavelet lighting approximation*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, no. 3, pages 376–381, 2003. 55

[Ngan & Durand 2006] A. Ngan and F. Durand. *Statistical acquisition of texture appearance*. In Proceedings of the Eurographics Symposium on Rendering, pages 31–40, 2006. 90

[O'Brien & Hodgins 1999] James F. O'Brien and Jessica K. Hodgins. *Graphical modeling and animation of brittle fracture*. In Proceedings of ACM SIGGRAPH 99, pages 137–146, August 1999. 15

[O'Brien *et al.* 2002] James F. O'Brien, Chen Shen and Christine M. Gatchalian. *Synthesizing sounds from rigid-body simulations*. In ACM SIGGRAPH Symposium on Computer Animation, pages 175–181, July 2002. 14, 15, 16, 34, 46, 57

[Odgaard *et al.* 2003] E. C. Odgaard, Y. Arieh and L. E. Marks. *Cross-modal enhancement of perceived brightness: sensory interaction versus response bias.* Perception and Psychophysics, vol. 65, no. 1, pages 123–132, January 2003. 5

[Oppenheim *et al.* 1999] Alan V. Oppenheim, Ronald W. Schafer and John R. Buck. Discrete-time signal processing (2nd edition). Prentice-Hall, 1999. 37, 41

[O'Sullivan *et al.* 2004] Carol O'Sullivan, Sarah Howlett, Yann Morvan, Rachel McDonnell and Keith O'Conor. *Perceptually Adaptive Graphics*. In Christophe Schlick and Werner Purgathofer, editeurs, Eurographics STAR Report, numéro STAR-6 de State of the Art Reports, pages 141–164. INRIA and the Eurographics Association, 2004. 4, 19

[Pai *et al.* 2001] Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond and Som H. Yau. *Scanning Physical Interaction Behavior of 3D Objects*. In Proceedings of ACM SIGGRAPH 2001, pages 87–96, August 2001. 17

[Paris *et al.* 2004] S. Paris, H.M. Briceño and F.X. Sillion. *Capture of hair geometry from multiple images*. ACM Transactions on Graphics (ACM SIGGRAPH 2004), vol. 23, no. 3, pages 712–719, 2004. 89

[Paris *et al.* 2008] Sylvain Paris, Will Chang, Wojciech Jarosz, Oleg Kozhushnyan, Wojciech Matusik, Matthias Zwicker and Frédo Durand. *Hair Photobooth: Geometric and Photometric Acquisition of Real Hairstyles*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), vol. 27, no. 3, 2008. 3, 88, 90, 92, 97, 101, 131, 132, 139

[Pellacini *et al.* 2000] Fabio Pellacini, James A. Ferwerda and Donald P. Greenberg. *Toward a psychophysically-based light reflection model for image synthesis*. In Proceedings of ACM SIGGRAPH 2000, pages 55–64, 2000. 81

[Peytavie *et al.* 2009] Adrien Peytavie, Eric Galin, Stephane Merillou and Jerome Grosjean. *Arches: a Framework for Modeling Complex Terrains*. Computer Graphics Forum (Proceedings of Eurographics), vol. 28, pages 457–467, 2009. 110

[Pharr & Humphreys 2004] Matt Pharr and Greg Humphreys. Physically based rendering: From theory to implementation. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. 131

[Pierre-Loup Lesage 2002] Mahes Visvalingam Pierre-Loup Lesage. *Towards Sketch-Based Exploration of Terrain*. Computers and Graphics, vol. 26, no. 2, pages 309–328, 2002. 109

[Pixar 2009] Pixar. *http://www.pixar.com/howwedoit/, accessed June*, 2009. 3

[Press *et al.* 1992] W. H. Press, Saul A. Teukolsky, W. T. Vetterling and Brian P. Flannery. Numerical recipes in C: The art of scientific computing. Cambridge University Press, 1992. 37

[Raghuvanshi & Lin 2006] Nikunj Raghuvanshi and Ming C. Lin. *Interactive sound synthesis for large scale environments*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pages 101–108, 2006. 14, 17, 18, 34, 41, 46, 47, 50, 57

[Ramamoorthi & Hanrahan 2001a] R. Ramamoorthi and P. Hanrahan. *On the relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object*. The Journal of the Optical Society of America, 2001. 56, 67

[Ramamoorthi & Hanrahan 2001b] Ravi Ramamoorthi and Pat Hanrahan. *An Efficient Representation for Irradiance Environment Maps*. In Eugene Fiume, editeur, Proceedings of ACM SIGGRAPH 2001, pages 497–500, 2001. 56, 67

[Ramamoorthi & Hanrahan 2002] Ravi Ramamoorthi and Pat Hanrahan. *Frequency space environment map rendering*. ACM Transactions on Graphics, vol. 21, no. 3, pages 517–526, 2002. 55

[Ramanarayanan & Bala 2007] G Ramanarayanan and K Bala. *Constrained texture synthesis via energy minimization.* IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 1, pages 167–78, 2007. 110, 121

[Ramanarayanan *et al.* 2007] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter and Kavita Bala. *Visual Equivalence: Towards a New Standard for Image Fidelity*. ACM Transactions on Graphics (ACM SIGGRAPH 2007), page 76, aug 2007. 4, 21, 69

[Ramasubramanian *et al.* 1999] Mahesh Ramasubramanian, Sumanta N. Pattanaik and Donald P. Greenberg. *A Perceptually Based Physical Error Metric for Realistic Image Synthesis*. In Proceedings of ACM SIGGRAPH 99, pages 73–82, August 1999. 4, 19

[Reinhard *et al.* 2002] Erik Reinhard, Michael Stark, Peter Shirley and James Ferwerda. *Photographic tone reproduction for digital images*. Proceedings of ACM SIGGRAPH 2002, vol. 21, no. 3, pages 267–276, 2002. 57

[Rodet & Depalle 1992] Xavier Rodet and Philippe Depalle. *Spectral Envelopes and Inverse FFT Synthesis*. In Proceedings of the 93rd AES Convention, San Francisco, 1992. 37

[Rubner *et al.* 2000] Y. Rubner, C. Tomasi and L.J. Guibas. *The Earth Mover's Distance as a Metric for Image Retrieval*. International Journal of Computer Vision, vol. 40, no. 2, pages 99–121, 2000. 93, 94

[Rushmeier 2008] Holly Rushmeier. *The perception of simulated materials*. In ACM SIGGRAPH 2008 classes, pages 1–12, New York, NY, USA, 2008. ACM. 6, 21

[Samet 2005] Hanan Samet. Foundations of multidimensional and metric data structures (the morgan kaufmann series in computer graphics and geometric modeling). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005. 114, 119

[Sarlat *et al.* 2006] L. Sarlat, O. Warusfel and I. Viaud-Delmon. *Ventriloquism aftereffects occur in the rear hemisphere*. Neuroscience Letters, vol. 404, pages 324–329, 2006. 27

[Sarna & Swartz 1988] T. Sarna and H. M. Swartz. *The physical properties of melanins*. In The Pigmentary System. Oxford University Press, 1988. 91

[Seiler *et al.* 2008] Larry Seiler, Doug Carmean, Eric Sprangle, Tom Forsyth, Michael Abrash, Pradeep Dubey, Stephen Junkins, Adam Lake, Jeremy Sugerman, Robert Cavin, Roger Espasa, Ed Grochowski, Toni Juan and Pat Hanrahan. *Larrabee: a many-core x86 architecture for visual computing*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), pages 1–15, 2008. 119

[Sekuler *et al.* 1997] R. Sekuler, A. B. Sekuler and R. Lau. *Sound alters visual motion perception*. Nature, vol. 385, no. 6614, page 308, 1997. 20

[Si 2003] Hang Si. TETGEN*: A 3D Delaunay Tetrahedral Mesh Generator*, 2003. `http://tetgen.berlios.de`. 15

[Sloan *et al.* 2002] Peter-Pike Sloan, Jan Kautz and John Snyder. *Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments*. Proceedings of ACM SIGGRAPH 2002, pages 527–536, 2002. 55

[Sloan *et al.* 2005] Peter-Pike Sloan, Ben Luna and John Snyder. *Local, deformable precomputed radiance transfer*. ACM Transactions on Graphics (ACM SIGGRAPH 2005), vol. 24, no. 3, pages 1216–1224, July 2005. 55, 81

[Snavely *et al.* 2006] Noah Snavely, Steven M. Seitz and Richard Szeliski. *Photo Tourism: Exploring Photo Collections in 3D*. In ACM Transactions on Graphics (ACM SIGGRAPH 2006), pages 835–846. Press, 2006. 7

[Spence & Driver 2004] Charles Spence and Jon Driver. Crossmodal space and crossmodal attention. Oxford University Press, USA, June 2004. 19, 127

[Stein *et al.* 1996] B. E. Stein, N. London, L. K. Wilkinson and D. D. Price. *Enhancement of perceived visual intensity by auditory stimuli: a psychophysical analysis*. Journal of Cognitive Neuroscience, vol. 8, no. 6, pages 497–506, 1996. 5, 127, 130, 131

[Storms & Zyda 2000] Russell L. Storms and Michael Zyda. *Interactions in Perceived Quality of Auditory-Visual Displays*. Presence: Teleoperators and Virtual Environments, vol. 9, no. 6, pages 557–580, 2000. 5, 22, 128, 130

[Sugita & Suzuki 2003] Y. Sugita and Y. Suzuki. *Audiovisual perception: Implicit estimation of sound-arrival time*. Nature, vol. 421, no. 6926, page 911, 2003. 20

[Suied *et al.* 2009] Clara Suied, Nicolas Bonneel and Isabelle Viaud-Delmon. *Integration of auditory and visual information in the recognition of realistic objects*. Experimental Brain Research, 2009. 129, 130

[Tobin 2008] D. J. Tobin. *Human hair pigmentation: biological aspects*. International Journal of Cosmetic Science, vol. 30, no. 4, 2008. 91

[Tong *et al.* 2002] Xin Tong, Jingdan Zhang, Ligang Liu, Xi Wang, Baining Guo and Heung-Yeung Shum. *Synthesis of bidirectional texture functions on arbitrary surfaces*. In Proceedings of ACM SIGGRAPH 2002, pages 665–672, New York, NY, USA, 2002. ACM. 110

[Tsingos *et al.* 2004] Nicolas Tsingos, Emmanuel Gallo and George Drettakis. *Perceptual Audio Rendering of Complex Virtual Environments*. ACM Transactions on Graphics (ACM SIGGRAPH 2004), vol. 23, no. 3, pages 249–258, July 2004. 4, 5, 6, 13, 14, 19, 21, 26, 27, 30, 34, 44, 48, 72, 127

[Tsingos *et al.* 2007] Nicolas Tsingos, Carsten Dachsbacher, Sylvain Lefebvre and Matteo Dellepiane. *Instant Sound Scattering*. In Rendering Techniques (Proceedings of the Eurographics Symposium on Rendering), 2007. 14

[Tsingos 2005] Nicolas Tsingos. *Scalable Perceptual Mixing and Filtering of Audio Signals using an Augmented Spectral Representation*. In Proceedings of the International Conference on Digital Audio Effects, pages 277–282, September 2005. 14, 34, 42, 51

[van den Doel & Pai 1998] Kees van den Doel and Dinesh K. Pai. *The Sounds of Physical Shapes*. Presence: Teleoperators and Virtual Environments, vol. 7, no. 4, pages 382–395, 1998. 14, 51

[van den Doel & Pai 2003] Kees van den Doel and Dinesh K. Pai. *Modal Synthesis for Vibrating Objects*. Audio Anecdotes, 2003. 14, 17, 34, 39, 40, 41, 46, 47, 50, 54

[van den Doel *et al.* 2001] Kees van den Doel, Paul G. Kry and Dinesh K. Pai. *FoleyAutomatic: Physically-based sound effects for interactive simulation and animation*. In Proceedings of ACM SIGGRAPH 2001, pages 537–544, 2001. 50

[van den Doel *et al.* 2002] Kees van den Doel, Dinesh Pai, T. Adam, L. Kortchmar and K. Pichora-Fuller. *Measurements of Perceptual Quality of Contact Sound Models*. Proceedings of International Conference on Auditory Display, pages 345–349, 2002. 18, 57

[van den Doel *et al.* 2004] Kees van den Doel, Dave Knott and Dinesh K. Pai. *Interactive simulation of complex audiovisual scenes*. Presence: Teleoperators and Virtual Environments, vol. 13, no. 1, pages 99–111, 2004. 17, 18, 34

[Vangorp *et al.* 2007] Peter Vangorp, Jurgen Laurijssen and Philip Dutré. *The influence of shape on the perception of material reflectance*. ACM Transactions on Graphics (ACM SIGGRAPH 2007), vol. 26, no. 3, page 77, August 2007. 6, 21, 22, 53, 58

[Viaud-Delmon *et al.* 2008] Isabelle Viaud-Delmon, Feryel Znaïdi, Nicolas Bonneel, Clara Suied, Olivier Warusfel, Khoa-Van N'Guyen and George Drettakis. *Auditory-visual virtual environments to treat dog phobia*. In Proceedings 7th ICDVRAT with ArtAbilitation, September 2008. 129

[Vroomen *et al.* 2000] Jean Vroomen, Beatrice De Gelder and Jean Vroomen. *Sound enhances visual perception: Cross-Modal effects of auditory organization on vision*. Journal of Experimental Psychology. Human perception and performance., vol. 26, pages 1583–1590, 2000. 5, 127

[Wand & Straßer 2004] Michael Wand and Wolfgang Straßer. *Multi-Resolution Sound Rendering*. In Symposium on Point-Based Graphics, 2004. 26

[Wang *et al.* 2009] Lvdi Wang, Yizhou Yu, Kun Zhou and Baining Guo. *Example-Based Hair Geometry Synthesis*. ACM Transactions on Graphics (ACM SIGGRAPH 2009), 2009. 101

[Ward *et al.* 2007] K. Ward, F. Bertails, T.Y. Kim, S.R. Marschner, M.P. Cani and M.C. Lin. *A Survey on Hair Modeling: Styling, Simulation, and Rendering*. IEEE Transactions on Visualization and Computer Graphics, pages 213–234, 2007. 89

[Watanabe & Igarashi 2004] Nayuko Watanabe and Takeo Igarashi. *A sketching interface for terrain modeling*. In ACM SIGGRAPH 2004 Posters, page 73, New York, NY, USA, 2004. ACM. 109

[Wei *et al.* 2005] Y. Wei, E. Ofek, L. Quan and H.Y. Shum. *Modeling hair from multiple views*. ACM Transactions on Graphics (ACM SIGGRAPH 2005), vol. 24, no. 3, pages 816–820, 2005. 89

[Wei *et al.* 2009] Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra and Greg Turk. *State of the Art in Example-based Texture Synthesis*. In Eurographics STAR Report, 2009. 109, 110

[Whelan & Visvalingam 2003] J. C. Whelan and M. Visvalingam. *Formulated Silhouettes for Sketching Terrain*. In TPCG '03: Proceedings of the Theory and Practice of

Computer Graphics 2003, page 90, Washington, DC, USA, 2003. IEEE Computer Society. 109

[Williams *et al.* 2003] Nathaniel Williams, David Luebke, Jonathan D. Cohen, Michael Kelley and Brenden Schubert. *Perceptually Guided Simplification of Lit, Textured Meshes*. In ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D), pages 113–121, April 2003. 19

[Wither 2008] Jamie Wither. *Sketching and annotation for the procedural modelling of complex phenomena*. PhD thesis, Institut National Polytechnique de Grenoble, 2008. 109

[Yu 2001] Y. Yu. *Modeling realistic virtual hairstyles*. In Proceedings of Pacific Graphics, pages 295–304, 2001. 88, 92

[Zhang *et al.* 2003] Jingdan Zhang, Kun Zhou, Luiz Velho, Baining Guo and Heung-Yeung Shum. *Synthesis of progressively-variant textures on arbitrary surfaces*. ACM Transactions on Graphics (ACM SIGGRAPH 2003), vol. 22, no. 3, pages 295–302, 2003. 110

[Zhou *et al.* 2007] H. Zhou, J. Sun, G. Turk and J.M. Rehg. *Terrain synthesis from digital elevation models*. IEEE Transactions on Visualization and Computer Graphics, pages 834–848, 2007. 109

[Zhukov *et al.* 1998] Sergej Zhukov, Andrej Inoes and Grigorij Kronin. *An Ambient Light Illumination Model*. In George Drettakis and Nelson Max, editeurs, Rendering Techniques '98, Eurographics, pages 45–56. Springer-Verlag Wien New York, 1998. 131

[Zinke *et al.* 2008] Arno Zinke, Cem Yuksel, Andreas Weber and John Keyser. *Dual Scattering Approximation for Fast Multiple Scattering in Hair*. ACM Transactions on Graphics (ACM SIGGRAPH 2008), vol. 27, no. 3, 2008. 88, 89, 90, 92, 94, 100

[Zinke 2007] Arno Zinke. *Light Scattering from Filaments*. IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 2, pages 342–356, 2007. 88, 89

[Zölzer 2002] Udo Zölzer. Digital audio effects (DAFX), chapter 8. Wiley, 2002. 37

Audio and Visual Rendering with Perceptual Foundations

Abstract:

Realistic visual and audio rendering still remains a technical challenge. Indeed, typical computers do not cope with the increasing complexity of today's virtual environments, both for audio and visuals, and the graphic design of such scenes require talented artists.

In the first part of this thesis, we focus on audiovisual rendering algorithms for complex virtual environments which we improve using human perception of combined audio and visual cues. In particular, we developed a full perceptual audiovisual rendering engine integrating an efficient impact sounds rendering improved by using our perception of audiovisual simultaneity, a way to cluster sound sources using human's spatial tolerance between a sound and its visual representation, and a combined level of detail mechanism for both audio and visuals varying the impact sounds quality and the visually rendered material quality of the objects. All our crossmodal effects were supported by the prior work in neuroscience and demonstrated using our own experiments in virtual environments.

In a second part, we use information present in photographs in order to guide a visual rendering. We thus provide two different tools to assist "casual artists" such as gamers, or engineers. The first extracts the visual hair appearance from a photograph thus allowing the rapid customization of avatars in virtual environments. The second allows for a fast previewing of 3D scenes reproducing the appearance of an input photograph following a user's 3D sketch.

We thus propose a first step toward crossmodal audiovisual rendering algorithms and develop practical tools for non expert users to create virtual worlds using photograph's appearance.

Keywords: Audio-visual 3D rendering, perception, guided rendering