Rendering falling snow using an inverse Fourier transform

Linqiao Zhang
School of Computer Science
McGill University, Montreal

August 2003

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of M.Sc.in Computer Science

©2003 - Linqiao Zhang

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-98769-8
Our file Notre référence
ISBN: 0-612-98769-8

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Thesis advisor

Michael S. Langer

Author

Linqiao Zhang

Rendering falling snow using an inverse Fourier transform

Abstract

This thesis presents an image based falling snow rendering method which is based on spectral synthesis technique. By incorporating the natural falling snow motion property, that is, the image speed and size of the snowflakes are related to the depth, we develop a tent-like surface in frequency domain. We synthesize the power spectrum along the tent-like surface and use IFFT to bring the data function back to space-time domain, thus attain a motion parallax image sequence. Treating the motion parallax as an opacity function, we can composite it with an existing video sequence and turn it into a snowing scene. Treating the motion parallax as a stimulus for the psychophysical study, it could serve as a complex yet natural scene-like stimulus, and therefore being expected to give a new perspective to the psychophysical study.

Directeur de thèse

Michael S. Langer

Auteur

Linqiao Zhang

Rendu de chute de neige à l'aide d'une transformée de Fourier inverse

Résumé

Ce mémoire présente une méthode de rendu de chute de neige à partir d'images basée sur une technique de synthèse spectrale. En incorporant la propriété du mouvement de chute de neige naturelle – la vitesse des images et la taille des flocons dépendent de la profondeur – nous développons une surface en forme de tente dans le domaine de fréquence. Nous synthétisons le spectre de puissance le long de la surface et nous utilisons une transformée de Fourier inverse pour transférer la fonction de données au domaine espace-temps, avec laquelle nous atteignons une séquence d'images en mouvement parallaxe. En traitant le mouvement parallaxe comme une fonction d'opacité, nous pouvons le composer avec une séquence vidéo existante et obtenir une scène de chute de neige. Le mouvement parallaxe, traité comme un stimulus d'une étude psychophysiologique, peut être utilisé comme un stimulus complexe mais naturel d'une scène. Nous nous attendons alors à ce que ceci donne une nouvelle perspective à l'étude psychophysiologique.

Contents

List of Figures

Acknowledgments

Writing a thesis is a wonderful experience, as well as carrying out the research experiments. From the beginning, the big question marks dropping from heaven, to having a working result and completing a thesis, pulling out hairs, searching among piles of papers and working in the lab until midnight are the general practices throughout the duration.

Doing research certainly means leading a hard life. Often times I wonder if I could have finished my research topic if I had not loved it. To this point, I am very grateful to my supervisor, Michael Langer, who motivated me to start this topic. He is also the person who is always available for discussions and gives constructive suggestions. Often times I also wonder where my research would lead to if without all the support and guidance from my supervisor. From Michael, I learned not only knowledge, but also serious scientific spirit, which influences me to pursue a scientific career.

Editing a thesis is probably the most boring job to do. The person who is having piles of teaching, supervising and researching work to do and still selflessly give me hand to help me out is again my supervisor. From the thesis organization to the spelling error, even the punctuation, he paid meticulous attention and unlimited patience on. Checking back forth these minor details has no fun at all. However, driven by his spirit, a highly dedicated educator's spirit, I am not only able to clean up and present my thesis, but also to learn how to approach a beautiful writing.

Throughout these two years study, I was funded by the IT fellowship from McGill Computer Science School, as well as the support from NSERC and FCAR through the generosity of my supervisor. Many thanks to these funding supports helping me getting through with my study.

I thank my peer geek fellow David Bélanger, who is always able to fix my tough bugs within a finger snap, to which I have many appreciations. He also shows me many system tools, which smooth my research work and save me quite a bit of time.

I also thank my dear friend Nora Boghossian. During the first half of my research period, the most frustrated period, she gave me very supportive friendship. Together, we looked at hundreds of white noise images. Many teasings made those nonsense noise images somewhat interesting, and also, raised up my spirit. I also appreciate her help of editing part of my writing.

The forever dear thanks give to my dear family, where is the richest resource of love, understanding, encouragement, support...

Chapter 1

Introduction

This is an information technology age, where computer science techniques give us the power to imagine as wild as we like and to create as active as we can. Using computer science to take over the world is certainly the dream of a computer scientist. Here, our action is a drop in the ocean, we tried to simulate the natural phenomenon – falling snow.

As one of the most common natural phenomena, falling snow simulation is intensively used in flight simulation, film making, video game animation and meteorology study. This demands us to render snowfall computationally efficiently and visually appealingly. Research to this problem has been carrying on since 1980's, and the major approach is to use particle systems [26, 31, 17].

particle systems rendering considers each snowflake as a particle with attributes such as initial position, color, velocity, size and transparency, and also the particle's shape and lifetime. During rendering, by applying forces – gravity, friction, resilience etc. – to the particle, its attributes will be updated accordingly. The update depends

on the predefined functions, for example, Newton's first law, and a short time interval. As we can see, this rendering method has an adequate control to the motion of each particle. It therefore has wide use. Not only can it render falling snow, but also fireworks, waterfalls, explosions, falling rain etc. However, since each particle has to be computed, when the number of particles becomes big, the demanded computation increases accordingly, thus the system rendering conceivably becomes slow, and therefore prohibits its further utilization.

If the rendering purpose is to create a *visual effect* of the rendering object, then we question whether it is necessary to simulate every single particle. In this thesis, we have developed an alternative falling snow rendering method. The inspiration comes from the motion plane model [38], that is, a pure translational image motion in space-time domain produces a plane of energy in frequency domain. We intuitively incorporate the natural motion properties of the falling snow – the snowflakes in greater depth appear smaller and move slower – with the motion plane model, this results in a tent like surface in frequency domain. We then synthesisd the power specturm along the tent like surface. The final rendering result is transformed to space-time domain by using a global inverse Fourier transform, this will give us the entire falling snow image sequence.

Since our method renders the entire falling snow motion once for all, instead of approaching by rendering each single snowflake, it is therefore a computationally elegant and simple rendering method. Also, this global rendering method produces an atmospheric snowfall effect, which is the part missed out by particle systems. Moreover, our rendering method is an image-based one, the rendered image sequence can

be directly composited with 2D image instead of performing 3D to 2D transformation which is the procedure has to be taken by the scene-based rendering methods.

As we mentioned before, our rendering method integrates the depth *vs.* speed *vs.* size relationship of the falling snow, it thus provides the rendering results strong 3D depth effects. Indeed, it simulates exactly the motion effects of motion parallax. During our rendering experiments, we frequently ask questions as: "how and why do we perceive these continuous layers in depth?"; "what cues cause us perceiving the rendered motion parallax as falling snow?"; "why the white image spots are perceived closer?". Driven by our puzzles, we tried to apply perceptual transparency theories of visual psychology and the image motion theories to our rendered motion parallax. Besides the plausible explanation we could give to our puzzles, we also find that our rendered motion parallax could be an interesting stimulus to be used in the image motion study.

Two major topics – the falling snow rendering and the psychophysical study – are addressed in this thesis. We naturally organize this thesis into two parts. In Chapter 2, 3 and 4, we address the falling snow rendering problem. We present a literature background of our rendering method in chapter two. In Chapter 3, we introduce our rendering method in detail. Chapter 4 generalizes the falling snow rendering method to the case that when falling snow is seen by a panning camera. Chapter 5 and 6 are related to the topic of the psychophysical study. We give a literature review of perceptual transparency theory in Chapter 5, and present our attempted experiments and discussions in Chapter 6.

Chapter 2

Background of the Rendering Method

In this chapter, we wish to build up the literature background of our falling snow rendering method. Four topics are addressed to serve this purpose, they are:

- Spectral synthesis – an overview of the spectral synthesis method in rendering images.
- Fourier domain and its properties – the fundamental theory used to render the falling snow.
- Image motion properties – the basis of our tent like surface.
- Image compositing – the theory used to composite our snow sequence with background images.

Each topic is relatively independent, yet putting them together at the end will fund up a sufficient background to help reading through this thesis.

2.1 Spectral Synthesis

Spectral synthesis method has long been used in computer graphics to render images such as terrain [22], ocean waves [18], fluids [33], fire [6], smoke [6], cloud [6] etc. The common property of these natural scenes is that they do not have a well-defined surface or shape. This makes the usually used geometry based rendering methods hard to approach, and the existing algorithms are often suffering from the computational expensiveness. On the other hand, behind the randomness, some of the physics and natural features of these natural scenes can be easily synthesized by spectral synthesis method.

The fundamental idea of spectral synthesis is to sum up large number of sinusoidal waves and map the pixel intensity and color to the image according to the amplitude of the sinusoidal waves. Given the huge range of variation of the sinusoidal waves, the key of the spectral synthesis method is to choose a proper model to sum up the sinusoidal waves in a proper way.

The spectral synthesis method can be carried on either in space-time domain [14, 15, 23, 24, 22] or in frequency domain [33, 27, 29, 18]. To the former case, the sinusoidal wave $color(x) = sin(x)$, which is defined in space-time domain, is the primitive for synthesizing images. Further procedures such as varying its frequencies, amplitudes, biasing and weighting the obtained color function, extending its dimensionality, combining with shape function etc. would apply on this primitive to vary it. Sometimes the variation could be radical such that the primitive might be altered to be a step function or a non-periodical signal. Putting the sinusoidal waves together is usually called constructing the noise function. Methods of doing so are stochas-

tic subdivision [14, 15], fractional Brownian simulation [6, 22], interpolating random value on the discrete lattice [6] etc. Each of these methods has fairly deep and wide research space to dive in and to play around, especially when implementing these methods to the real scene simulation. Here, however, we only have a quick glance to them because our rendering method is a frequency domain based one.

The major advantage of spectral synthesis images in frequency domain is because this method is global. The focus of this method is not how to sum up all the sinusoidal waves but how to define a physics model for our desired rendering image in frequency domain. Once we set up a right model, the procedure of the inverse Fourier transform will produce the constructed noise function. We say it is global because the entire image or the entire image sequence will be synthesized once for all after the IFFT. Further, along with the globality of this method, it is also fast and efficient. However, things never can be perfect. Because of the globality, once we take the IFFT, we can not really modify each single sinusoidal wave in space-time domain. This property makes the rendering process highly rely on the physics model which we built in frequency domain. Meanwhile, this overly essential physics model is usually difficult to render, because frequency domain is not as intuitively understandable as space-time domain. But, many researchers have challenged this difficulty and rendered the scene images such as ocean waves [18], fluids [33], turbulent gaseous phenomena [27], motion under wind [29] ... and also, our falling snow quite well by using this method.

Constructing the physics model requires understanding of the power spectrum distribution of the scene we meant to render. Through analyzing the motion properties, the image structures of the scene and referring its properties in the Fourier domain,

we could approach a spatial and temporal frequency distribution model which best approximates and summarizes the physics properties of the natural scene in frequency domain. On top of this physics model, the refining steps such as 1/f noise filtering, phase shifting, color mapping, ray tracing etc. would be applied on to generate the more desirable image.

In spite of the procedural commonality of the frequency domain based spectral synthesis methods, the constructed physics models are sharply different from each other. This is expectable because one can hardly connect the physics motion of the ocean waves with the gaseous turbulent, neither with the falling snow. Therefore, studying the power spectrum distribution of each scene images becomes fundamental and primary. This requires a good knowledge of the Fourier domain. Here, the next Section will follow up to facilitate the background of the Fourier domain.

2.2 Fourier Domain and Its Properties

By applying Fourier transform [5], image data can be transformed from space-time domain to the frequency domain, and then by applying the inverse Fourier transform, the image data can be transformed back to space-time domain. The image will be unchanged after the Fourier and its inverse transforms. Therefore, Fourier transform offers another way to work on the same image data in another domain. The purpose of transforming data to frequency domain is to utilize the properties of Fourier transform, and therefore, to process image data more efficiently, or even achieve some work which is prohibitive in space-time domain. This section will give a brief introduction of the Fourier transform and a few of its properties used

in the falling snow rendering.

2.2.1 Fourier Transform

Given an one variable continuous function $f(t)$, the Fourier Transform $\mathcal{F}(\omega_t)$ is defined as

$$\mathcal{F}(\omega_t) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi t\omega_t}\mathrm{d}t \tag{2.1}$$

and the inverse transform as

$$f(t) = \int_{-\infty}^{\infty} \mathcal{F}(\omega_t)e^{j2\pi t\omega_t}\mathrm{d}\omega_t \tag{2.2}$$

where j is the imaginary unit and $e^{j\theta} = cos(\theta) + jsin(\theta)$.

In the case of discrete function, the Fourier transform is defined in the similar way, except that it sums over the discrete series instead of integrating. If we let $I(x)$ denote the discrete function with N samples, or more specifically, taking one row from an image, which is with N pixels, and let $I(x)$ represent the image intensity of this row, then the Fourier transform of $I(x)$ is defined as

$$\mathcal{I}(\omega_x) = \frac{1}{N}\sum_{x=0}^{N-1} I(x)e^{-j2\pi\omega_x x/N} \tag{2.3}$$

and the inverse Fourier transform is defined as

$$I(x) = \sum_{\omega_x=0}^{N-1} \mathcal{I}(\omega_x)e^{j2x\pi\omega_x/N} \tag{2.4}$$

$\mathcal{I}(\omega_x)$ is a complex number, i.e. $\mathcal{I}_i = \mathcal{I}_{real} + j\mathcal{I}_{image}$, it can also be expressed in terms of its magnitude $A(\omega_x)$ and phase $\alpha(\omega_x)$.

$$\begin{aligned} A(\omega_x) &= \| \mathcal{I}(\omega_x) \| = \sqrt{\mathcal{I}_{real} * \mathcal{I}_{real} + \mathcal{I}_{imag} * \mathcal{I}_{imag}} \\ \alpha(\omega_x) &= tan^{-1}(\frac{\mathcal{I}_{imag}}{\mathcal{I}_{real}}). \end{aligned}$$

$\mathcal{I}(\omega_x)$ will become:

$$\mathcal{I}(\omega_x) \;=\; A(\omega_x)\; e^{j\alpha(\omega_x)}\;.$$

The magnitude represents the energy of the signal at spatial frequency k, whereas the phase represents how many spatial frequency steps of the signal are shifted [16].

The above discussion is only about the one-variable Fourier transform. In fact, Fourier transform can be easily extended to multi-variable functions. As an example, three-variable discrete Fourier transform is discussed below. It is also the one used in falling snow rendering.

Given a discrete function $f(x, y, t)$ with samples (M, N, T), which denotes an image sequence with T frame and each frame with $M * N$ pixels. Its Fourier transform $\mathcal{F}(\omega_x, \omega_y, \omega_t)$ is defined as:

$$\mathcal{F}(\omega_x, \omega_y, \omega_t) = \frac{1}{MNT} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \sum_{t=0}^{T-1} f(x, y, t) e^{-j2\pi(\omega_x x/M + \omega_y y/N + \omega_t t/T)} \tag{2.5}$$

and its inverse transform is defined as:

$$f(x, y, t) = \sum_{\omega_x=0}^{M-1} \sum_{\omega_y=0}^{N-1} \sum_{\omega_t=0}^{T-1} \mathcal{F}(\omega_x, \omega_y, \omega_t) e^{j2\pi(x\omega_x/M + y\omega_y/N + t\omega_t/T)} \tag{2.6}$$

Here $\mathcal{F}(\omega_x, \omega_y, \omega_t)$ is a complex number in the $(\omega_x, \omega_y, \omega_t)$ space.

2.2.2 Fourier Domain

Within the $(\omega_x, \omega_y, \omega_t)$ space, i.e. the Fourier domain, the spatial frequency (ω_x, ω_y) relates to the number of cycles of the sine wave per image frame, whereas the temporal frequency ω_t relates to the number of cycles of the sine wave per image sequence. Low spatial frequency corresponds to small number of cycles per image

frame. In space-time domain, it corresponds to larger size of the image objects. Low temporal frequency corresponds to small number of cycles per image sequence. In space-time domain, it corresponds to slower image motion.

2.2.3 Periodicity Property

The Fourier sine waves are periodic signals. Using this periodicity, we can repeat the signal to obtain more samples which are beyond the original sample range. As we will see later, given an image sequence with samples (M, N, T), which is obtained from the inverse Fourier transform, one can tile each image frame side by side to obtain a larger image, or repeat the image sequence to make the motion longer.

2.2.4 Conjugacy Property

Given a real image, i.e. the $I(x, y, t)$ is a real number, its Fourier transform obeys the conjugacy property

$$\mathcal{I}(\omega_x, \omega_y, \omega_t) = \overline{\mathcal{I}(-\omega_x, -\omega_y, -\omega_t)} \tag{2.7}$$

Thus, if one synthesizes a signal in frequency domain, and wishes to get a real signal in space-time domain, then the signal should obey the conjugacy property.

After we had a quick refresh at the Fourier domain, let's look at the image motion properties in this domain.

2.3 Image Motion Properties

There has been a long and rich history about image motion analysis in Fourier domain. Moving sine wave gratings have been used as stimuli in thousands of experimental studies of visual motion perception. Such stimuli have also played a key role in computational models of visual motion processing [38, 1, 30, 8]. The model we used to render falling snow extends from two particular frequency domain models: the motion plane model introduced by Watson and Ahumada [38] and the optical snow model introduced by Langer and Mann [13, 12].

2.3.1 Motion Plane Model

Watson and Ahumada observed that an arbitrary static image that undergoes a constant translational motion in the image plane over a sequence of frames yields a plane of energy in the 3D frequency domain. That is, if one takes the 3D Fourier transform of a translating image sequence, then all the energy lies on a plane in the 3D frequency domain. Formally, suppose an image is translating with velocity (v_x, v_y) pixels per frame, that is,

$$I(x, y, t) \;=\; I(x - v_x t,\; y - v_y t,\; 0)$$

Let (ω_x, ω_y) be the spatial frequencies in the x and y directions, and let ω_t be the temporal frequency. If one takes the Fourier transform of $I(x, y, t)$, then all the energy in the 3D frequency domain lies on the plane:

$$\omega_t \;=\; -v_x\omega_x - v_y\omega_y \tag{2.8}$$

This plane passes through the origin. We refer to it as the *motion plane* for velocity (v_x, v_y).

A few details on the geometry of the motion plane helped us to develop our tent model and will help the reader's intuition as well. If one takes the intersection of the motion plane with the plane $\omega_t = 0$, one gets

$$v_x\omega_x + v_y\omega_y = 0 \tag{2.9}$$

This is a line in the spatial frequency plane (ω_x, ω_y), namely, the line along which the motion plane of Eq. (2.8) intersects the plane $\omega_t = 0$. The vector (v_x, v_y) is perpendicular to this line, and the magnitude $\sqrt{v_x^2 + v_y^2}$ is the slope of the motion plane in the direction (v_x, v_y). These properties of the motion plane are well-known [38].

To simplify the notation, we rotate the spatial frequency coordinates (ω_x, ω_y) to new coordinates (ω_v, ω_l) such that ω_v is in the direction of the motion (v_x, v_y) and ω_l is in the direction of the line of Eq. 2.9. If we let θ be the angle between ω_x and the direction of the line ω_l, then we can relate the two coordinate systems by:

$$\begin{bmatrix} \omega_v \\ \\ \omega_l \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta \\ \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \\ \omega_y \end{bmatrix} \tag{2.10}$$

In particular, Eq. (2.8) can be re-expressed as

$$v = -\frac{\omega_t}{\omega_v} \tag{2.11}$$

Eq. (2.11) will be used in Chapter 3 in our tent model from which we synthesize falling snow.

2.3.2 Optical Snow

A more recent frequency domain model of image motion is the optical snow model introduced by Langer and Mann [13]. These authors observed that falling snow produces motion in which a range of speeds is present and the direction of motion is roughly constant across the image. In the frequency domain, falling snow thus produces a family of motion planes:

$$\omega_t = -s\ v_x\omega_x - s\ v_y\omega_y \tag{2.12}$$

in which a range of speeds s is present. The family of motion planes all intersect at the line of Eq. (2.9) and each with the motion plane properties as we described in motion plane model.

In addition, Eq. (2.12) can also be expressed in a simple way by rotating the (ω_x, ω_y) coordinates to the (ω_l, ω_v) coordinates as in Eq. (2.10). The set of velocities will become sv and Eq. (2.12) can be re-expressed as

$$v = -\frac{\omega_t}{s\ \omega_v} \tag{2.13}$$

A more generalized version of the optical snow model assumes that the image velocity of the falling snow is with a range of speed and also directions [12]. For example, in the case that falling snow is seen by a moving camera, the set of image velocity is of form $(u_x + s\ v_x, u_y + s\ v_y)$, where (u_x, u_y) is a constant velocity vector. The family of motion planes will become:

$$\omega_t = -(u_x + s\ v_x)\omega_x - (u_y + s\ v_y)\omega_y \tag{2.14}$$

Eq. (2.14) defines a set of planes which all pass through the origin $(\omega_x, \omega_y, \omega_t) = (0, 0, 0)$ but do not intersect with plane $\omega_t = 0$ at a common line. In contrast with

Eq. (2.12), we name the optical snow model represented by Eq. (2.12) as parallel optical snow model, and the model represented by Eq. (2.14) as nonparallel optical snow model.

Eq. (2.14) could be rewritten as:

$$\omega_t = -s\ v_x\omega_x - s\ v_y\omega_y - u_x\omega_x - u_y\omega_y \tag{2.15}$$

Analogous to Eq. (2.10) and Eq. (2.13), we can express Eq. (2.15) using the (ω_l, ω_v) coordinates as well. If let (s_v, s_l) represents the rotated velocity vector (u_x, u_y), i.e.

$$\begin{bmatrix} s_v \\ \\ s_l \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta \\ \\ \cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} u_x \\ \\ u_y \end{bmatrix}$$

then Eq. (2.15) will become

$$\omega_t = -s\ v\ \omega_v - s_v\omega_v - s_l\omega_l$$

Given v is a constant velocity, we can simplify this equation as:

$$\omega_t = -s\ \omega_v - s_v\omega_v - s_l\omega_l \tag{2.16}$$

Eq. (2.16) will be referred in Chapter 4 to derive our sheared tent model, which we used to render the falling snow seen by a moving camera.

The optical snow motion model captures the range of velocities presented in falling snow. However, it does not explicitly account for depth effects, nor is the model used for rendering image sequences. Solving these two problems is the main original research contribution presented in this thesis. The details will be given in Chapter 3.

The above two models, the motion plane model and the optical snow model are all discussed in the frequency domain. Now, let's change our mind a bit and come back to the space-time domain. We will introduce the compositing method in the following section. This method plays an important role when we apply our rendering results to the scene images.

2.4 Image Compositing

Image compositing has been richly studied in computer graphics [25, 9, 4, 20] and widely used in visual arts and film industry. Behind those fascinating special image effects created by compositing, the basic idea of it is quite simple.

We treat multiple images as overlapping layers and combine them together to obtain the composited image. For each image which will overlay on the other image, we give an extra α channel to it. The α channel contains the value of opacity α of each pixel in the overlaying image. The α value is within range $[0, 1]$, 0 is fully transparent, 1 is fully opaque, and the intermediate values are partially transparent. Mathematically, to composite the foreground image I_2 over the background image I_1 according to the opacity α, we do

$$I(x, y) = (1 - \alpha(x, y))I_1(x, y) + \alpha(x, y)I_2(x, y) \tag{2.17}$$

to obtain the composited image I. Color images associate with three channels – RGB. If this is the case, we apply Eq. (2.17) to each channel.

Eq. (2.17) illustrates the general rule of compositing. When it is specifically applied to our application, both to the falling snow rendering and to building our

transparency study stimuli, we made slight variations to this compositing method. However, the fundamental concepts all still the same. You will see our application in Sec. 3.3 and Sec. 6.1.1.

Chapter 3

Rendering Falling Snow

We present the falling snow rendering method in this chapter. We approach this rendering problem from the frequency domain, that is, using an inverse Fourier transform. This method yields an image sequence which is designed to have similar appearance as falling snow. In this sense it produces a visual effect that is similar to falling snow. The overall idea of the method is to:

1. Define a specific surface in the 3D frequency domain, that is, in the 3D space defined by the Fourier transform of space-time (XYT). This surface is derived mathematically from a motion parallax property of falling snow, namely from a relationship between the image speed and the image size of a given snowflake and the distance of that snowflake from the viewer (see Sec. 3.1).

2. Take the inverse Fourier transform of to get a function $I(x, y, t)$.

3. Treat $I(x, y, t)$ as an opacity function and use it to composite white snow over a background image (or video).

As an example, Figure 3.1 (b) shows one frame of the falling snow image sequence, after compositing over a background image house [10], shown in Figure 3.1 (a). The entire image sequence is available from the video example *house.*

(a) (b)

Figure 3.1: (a) An image of a house on a winter day. (Painting *Jimmy's place.* Courtesy of artist Gary Johnson) (b) One frame of a rendered image sequence in which falling snow is composited over the still image in (a).

3.1 Tent Model of Falling Snow

As we reviewed in Sec. 2.3.1, the standard model for motion in the frequency domain assumes pure image translation. Falling snow differs from pure image translation in that falling snow has multiple depth layers, indeed a continuum of depth layers. There are two implications of the continuum of depth layers. First, the depth layers give rise to motion parallax. The image speed of the snowflakes in a given depth layer depends on the depth of that layer. In terms of the motion plane model, falling snow should produce energy in a range of motion planes, corresponding to

different depth layers (see Sec. 2.3.2). The second implication of the depth layers concerns the size of the snowflakes, namely that the image size of snowflakes in a given layer depends on the depth of the layer. Let us now restate these two effects using mathematical formulas, and then combine them.

Let the distance variable be d. Then the motion of falling snowflakes has two correlations associated with it:

I: *The closer the snowflake, the faster the snowflake.* For simplicity, assume all snowflakes are falling with roughly the same 3D speed. Then snowflakes closer to the camera will appear to move faster because of perspective, that is, d is inversely proportional to image speed, and so from Eq. (2.11),

$$d \propto \frac{1}{\omega_t/\omega_v} \tag{3.1}$$

II: *The closer the snowflake, the bigger the snowflake.* Bigger objects have more energy concentrated in lower spatial frequencies, i.e. smaller radius $\sqrt{\omega_v^2 + \omega_l^2}$. This means that the distance d to a snowflake is proportional to the radius of the spatial frequencies to which the snowflake contributes:

$$d \propto \sqrt{\omega_v^2 + \omega_l^2} \tag{3.2}$$

Combining Eqns. (3.1) and (3.2), we get:

$$\omega_t/\omega_v \propto \frac{1}{\sqrt{\omega_v^2 + \omega_l^2}}$$

For any fixed proportionality constant C to, we get the surface:

$$\omega_t = C \frac{\omega_v}{\sqrt{\omega_v^2 + \omega_l^2}} \tag{3.3}$$

This is our falling snow rendering surface. Figure 3.2 shows an example of a surface of Eq. (3.3). These surfaces have a tent-like form, and so we refer to Eq. (3.3) as the *tent model.*

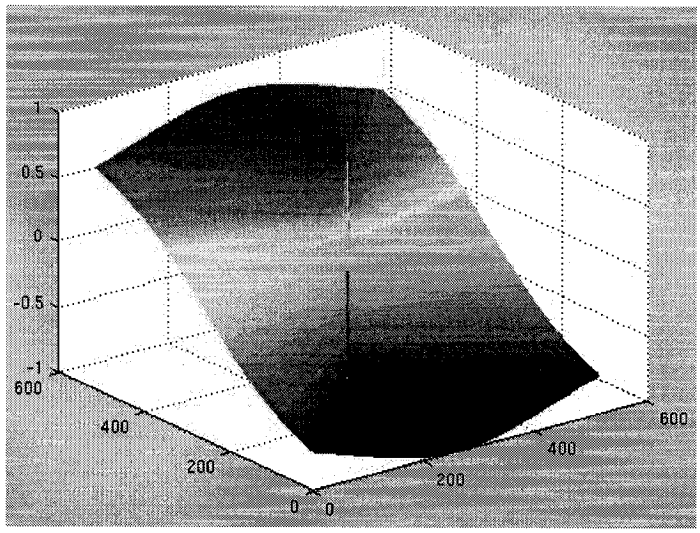


Figure 3.2: The plot of Eq. (3.3) with constant (C = 1)

If comparing our tent model with the motion plane model, i.e. $\omega_t = -v\omega_v$, and with the parallel optical snow model, i.e. $\omega_t = -sv\omega_v$, we can find some close relationship among these three. In motion plane model, ω_t relates to ω_v according to constant speed v. In parallel optical snow model, ω_t relates to ω_v according to a range of speeds sv. In particular, s is a free variable. In our tent model, ω_t relates to ω_v according to $C/\sqrt{\omega_v^2 + \omega_t^2}$, which is a set of speeds varying according to the spatial frequency $\sqrt{\omega_v^2 + \omega_t^2}$. This set of speeds captures the properties that objects at greater depth appear smaller and move slower in the image, which is the fundamental motion property of falling snowflakes. In conclusion, our tent model handles a larger range of image motion speeds than the motion plane model, and accounts more specifically

the falling snow motion property than the parallel optical snow model.

3.2 Synthesizing a Tent Surface

We synthesize falling snow sequences with T frames and $N \times N$ pixels per frame. Typical values are N=512 and $T = 64$. The basic idea is to synthesize a tent surface in the frequency domain and then take the inverse Fourier transform.

The $(\omega_x, \omega_y, \omega_t)$ space in which we synthesize the tent surface has $N \times N \times T$ cells. Each cell is assigned a complex value, with both its real and imaginary parts defined as type `double`. The cells are initialized to the value zero. We then loop through the three dimensions of the frequency domain. If any cell $(\omega_x, \omega_y, \omega_t)$ overlaps the surface defined by Eq. (3.3), we assign it a complex value with an amplitude varying between 0 to 1 and a random phase varying between 0 to 2π.

To ensure that the values after the inverse Fourier transform will be real, we enforce that the value in the cell $(\omega_x, \omega_y, \omega_t)$ is the complex conjugate of the value in the cell $(-\omega_x, -\omega_y, -\omega_t)$. This is the standard conjugacy property of Fourier transforms of real images (see Sec. 2.2.4), namely

$$\mathcal{I}(\omega_x, \omega_y, \omega_t) = \overline{\mathcal{I}(-\omega_x, -\omega_y, -\omega_t)} \tag{3.4}$$

where $I(x, y, t)$ is the image sequence and $\mathcal{I}(\omega_x, \omega_y, \omega_t)$ is its 3D Fourier transform. To achieve this conjugacy property, we only assign values to the cells within the negative ω_t halfspace. All the cells in the positive halfspace are then assigned values according to their conjugate cells.

3.2.1 Thickened tent surface

The tent in Fig. 3.2 is an infinitely thin surface. For this surface, each spatial frequency $\sqrt{\omega_x^2 + \omega_y^2}$ has a single speed associated with it. This is not quite what we want, however. Snowflakes at a given depth (and hence at a given image speed) contribute to a range of spatial frequencies. This fact suggests that we should use a thickened tent surface. We implement the thickened tent surface by using a small range of ω_t values around the tent surface. This gives slightly better renderings than if only a single layer is used.

3.2.2 Limiting the range of spatial frequencies

The sizes of the snowflakes have to be taken into account. If we allow all possible spatial frequencies to contribute to the tent surface, then the size of the snowflakes would vary from very big (the image width) to very small (the distance between pixels). This is not quite what we want to render falling snow, since snowflakes should be relatively small.

To limit the size of the snowflakes – that is, to limit the size of the moving image structure that we are synthesizing – we limit the energy in the tent surface to three octaves of relatively high spatial frequencies. This acts as cutting an annulus in the (ω_x, ω_y) plane and assigning values only to the cells whose spatial frequency components lie in the annulus. Because speed is related to size, limiting the range of spatial frequencies also limits the range of speeds.

Here are the specifics. We are synthesizing image frames of size 512×512, and so we limit the non-zero values of energy of the tent surfaces to the three octave range

from 16 to 128 cycles per image frame. Spatial frequencies lower than 16 cycles per image frame are not used in order to enforce an upper bound on size of moving image structure – that is, snowflakes are small. Spatial frequencies above 128 cycles per image frame are not used in order to stay far from the Nyquist limit, which is 256 cycles per frame.

3.2.3 1/f amplitude spectra

The final issue is to make the amplitudes of the Fourier coefficients as a function of spatial frequency. It is known that the amplitude spectra of natural images obeys a power law [7, 28]. Specifically, the amplitudes of the Fourier coefficients fall off as $1/\sqrt{\omega_x^2 + \omega_y^2}$ on average. In the image science literature, this is known as 1/f scaling. Such an amplitudes distribution will put constant amount of energy in each constant octave, thus make each band equally visible [7].

Our falling snow images are natural images as well, we therefore apply this $1/f$ spectral distribution on synthesizing our tent surface. As we mentioned at the beginning of this section that the amplitudes are set to a random value between $[0, 1]$, then now we need to divide each amplitude by the spatial frequency:

$$\mid \mathcal{I}(\omega_x, \omega_y, \omega_t) \mid \propto \frac{rand([0,1])}{\sqrt{\omega_x^2 + \omega_y^2}} \tag{3.5}$$

A little reminder is about the phases. We set them as random values between $[0, 2\pi]$ at the beginning of this section and we will keep this randomness throughout our image rendering without any modification, since phase changing relates to the image object shifting, which is out of our concern in this thesis.

3.3 Compositing Method

To obtain a falling snow sequence, we compute the inverse Fourier transform of our tent surface. This yields a real valued function which we denote $\alpha(x, y, t)$. We wish to treat $\alpha(x, y, t)$ as an opacity function, namely the density of snowflakes at a pixel (x, y) in frame t. The reason for treating $\alpha(x, y, t)$ as an opacity function is that we can then apply standard compositing (see Sec. 2.4).

To treat $\alpha(x, y, t)$ as an opacity function, we need to map it to the interval [0,1]. We also wish to take account of the fact that the human visual system is sensitive to logarithmic differences in intensity, rather than linear differences in intensity. For this reason, after mapping to [0,1], we apply a non-linear transformation

$$\alpha(x, y, t) \rightarrow \alpha(x, y, t)^{\beta}$$

where we use $\beta = 1.5$. This compresses the values of $\alpha(x, y, t)$ to the lower part of the interval [0,1] and makes the variations in opacity more visible. In particular, β is not fixed. We let $\beta \geq 1$ and vary according to the rendering purpose and the background images. Higher β value associates with lighter falling snow appearance, because the nonlinear transformation $\alpha(x, y, t)^{\beta}$ will result low opacity value, and therefore, will weaken the foreground snow and strength and background image.

To composite the snow with a background image, we set the intensity of the snow to be a constant, namely the intensity I_{amb} of the ambient light in the scene. The reason for setting the intensity of snow to a constant is that snow is white (reflectance near one). The value of I_{amb} is not known. So we take it to be a high grey level value, say 250. The reason for choosing such a value is that we would like it to be less than

the intensity of light source (which saturate the pixels at grey level 255). But we would also like the snow to appear white, so we take it to be high value.

To composite with a background image, we use the formula:

$$I(x, y, t) \;=\; I_{amb}\;\alpha(x, y, t) \;+\; (1 - \alpha(x, y, t))\;I_{bg}(x, y) \tag{3.6}$$

This is a variation of standard compositing where the foreground intensity is constant and the opacity varies with time. The background image could itself be a video, but in our example, we only use the still image.

A special feature of this compositing method is that the snow motion will become very weak or even vanish when the background image region has high intensity, or say has white color. Because in this case, we can assume that $I_{bg}(x, y) = I_{amb}$, then this results that $I(x, y, t)$ has the same value within the white image regions over all the image frames, i.e.

$$I(x, y, t) \;=\; I(x + v_x \mathrm{d}t,\; y + v_y \mathrm{d}t,\; t + \mathrm{d}t)$$

thus no falling snow motion being detected.

A fact from nature is that the snow covered surface is white. Another fact from the natural motion is that falling snowflakes would stop falling when they hit the snow covered surface. These two natural facts demand the similar visual effect from our rendered falling snow motion. And here, by fooling eyes through the intensity trick, our compositing method offers this side benefit with easy.

3.4 Tiling XYT

The opacity functions $\alpha(x, y, t)$ that we compute are periodic (see Sec.2.2.3) in all three variables XYT. The reason for the periodicity is that each of the frequency components is chosen independently. A traveling sine wave of any frequency $(\omega_x, \omega_y, \omega_t)$ is periodic in XYT, so is the sum of such waves.

The periodicity property is convenient because it means we can tile XYT with the function $\alpha(x, y, t)$ to obtain videos that are larger (in space) and longer (in time). For example, our computed $\alpha(x, y, t)$ functions are only $T = 64$ frames long. But if you loop the video, then the videos can be extended to infinite length. Notice that looping the video does not create a jumpping discontinuity.

We can also tile in space. The snow in video *sydney* (see Sec. 3.5) uses sixteen $(4 * 4)$ falling snow tiles, each one of size $N = 128$ and $T = 64$. The *sydney* image itself is $512 * 512$, which is shown in Figure (3.3). The tiles are not apparent at this size. One frame of the tiled snowing video *sydney* is shown in Figure (3.4). The tiling effect does not appear on the static image either.

When very small tiles are used (say $N = 64$), the tiles do become visible and the periodicity becomes visible as well. We present the video example *sydneysmalltiling* to show this effect. One frame of the image example is in Figure 3.5. As we can see the snow appears more like a periodic pattern than natural falling snow.

Figure 3.3: The *sydney* background image. Size is $512 * 512$.

3.5 Results

We submit three groups of MPEG videos to demonstrate the falling snow rendering results. Each group is the rendered snowing scene along with its opacity function. These videos use lossless compression (MPEG quality = 90%).

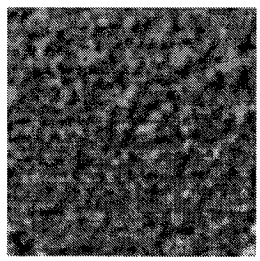
I: The video *snow* shows the opacity function $\alpha(x, y, t)$ itself. The video *house* uses a background image [10] of a scene in which there is snow on

the ground but no snow falling (see Fig. 3.1). The video shows falling snow composited over this scene.

II: The video *sydney* shows a cityscape of Sydney on a sunny day. We have composited snow over this scene to show what the city might look like on a snowy day. For *sydney*, we used the falling snow tiling method (see Sec. 3.4). The video *tile128* shows the opacity function with size $128 * 128$.

III: The video *sydneytilesmall* shows a failure example of the tiling method when the tile size is too small, as $64 * 64$. The video *tile64* shows the small tile of the opacity function with size $64 * 64$.

Notice that the snowfall direction of *house* is different from the snowfall direction of *sydney*. Indeed, the motion direction of the falling snow is a free variable (see Sec. 2.3.1), and these two examples are used to demonstrate this property.

(a) 

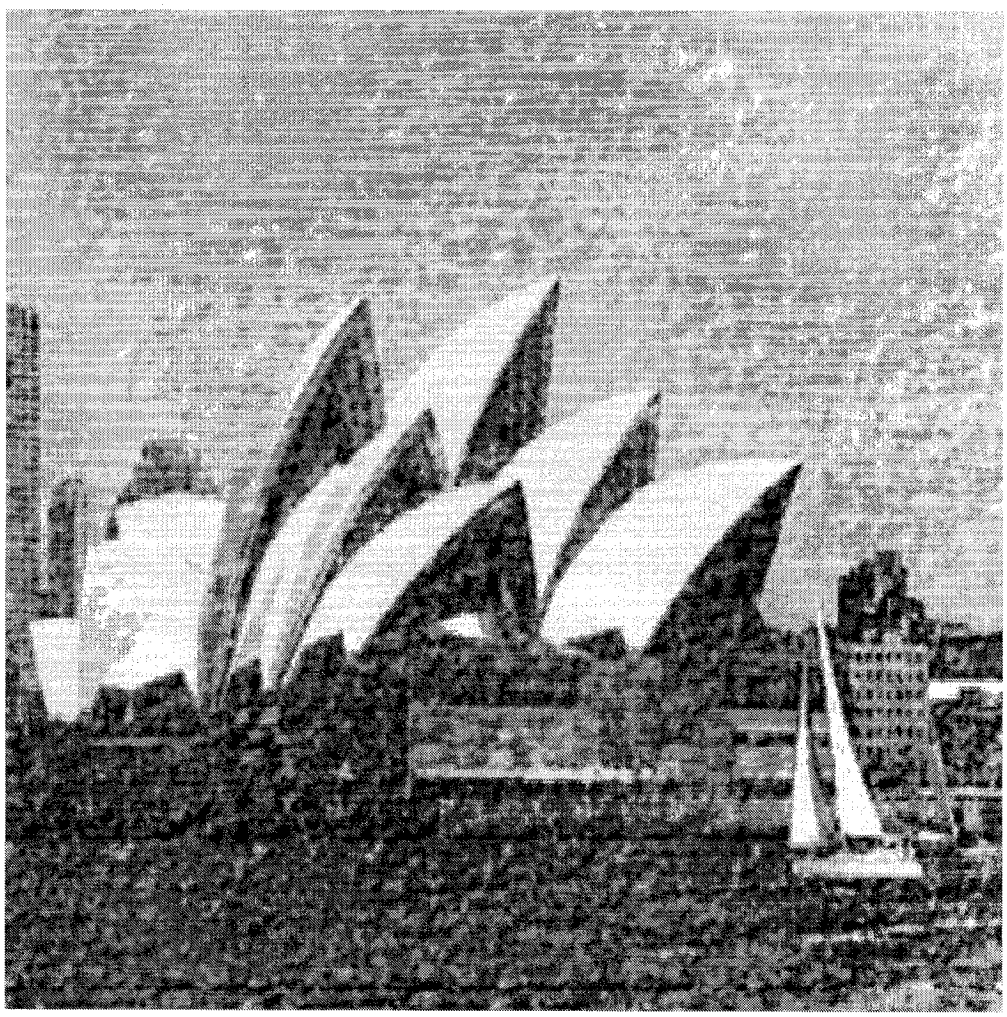
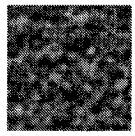
(b) 

Figure 3.4: One frame of the video example *sydney* to demonstrate the seamless tiling method. (a) One frame of the rendered falling snow, which is with size $128 * 128$. (b) The composited snowing scene *sydney*, in which the foreground snow scene uses the rendered snow in (a) and tiled sixteen $(4 * 4)$ times.

(a) 

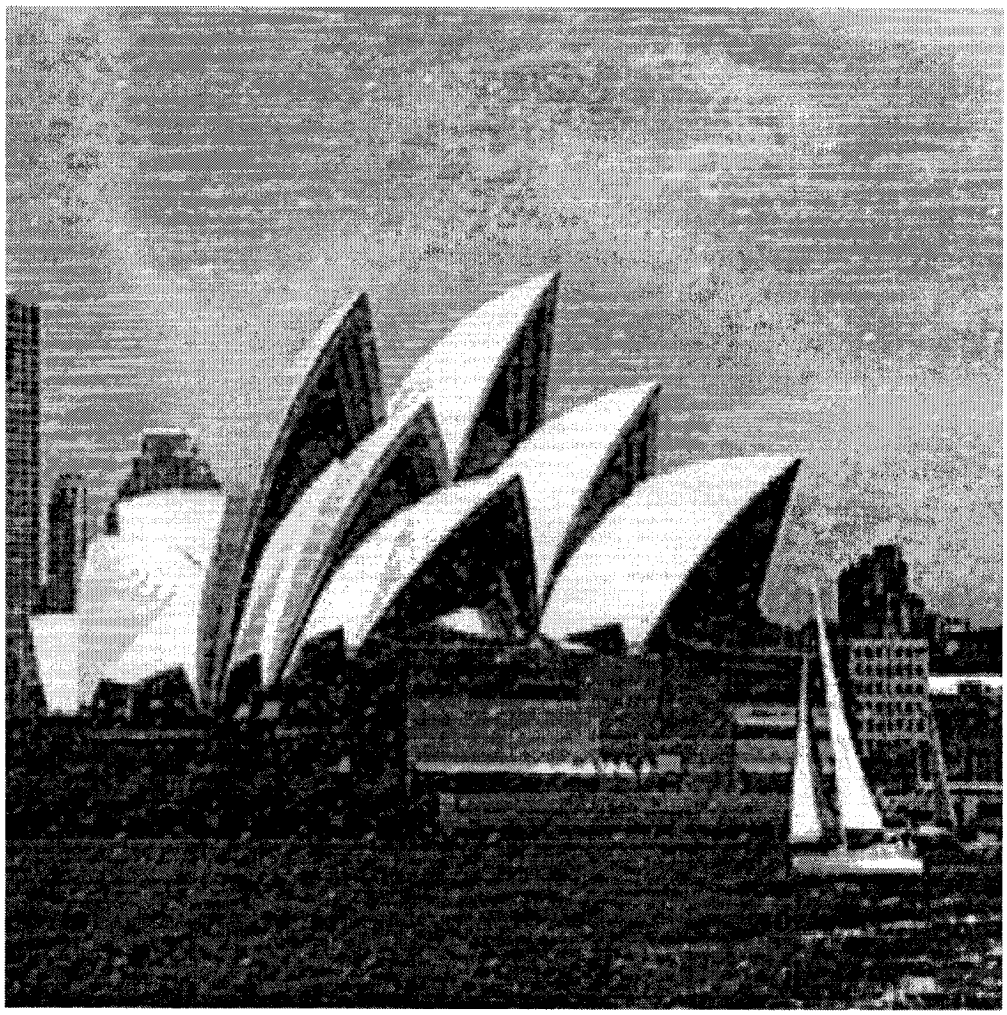
(b) 

Figure 3.5: The failure example of applying tiling method. When the size of the tile is too small to render natural scene, it appears as periodic patterns on the composited scene images. (a) One frame of a rendered snowfall tile, with size $64 * 64$. (b) The composited snowfall scene, in which the foreground snow uses $8 * 8$ tiles.

Chapter 4

Falling Snow Seen by a Panning Camera

The method we presented in Chapter 3 renders parallel falling snow. Although the motion direction of the falling snow is a free variable, once it is chosen, the image motion will be along the chosen direction only. When the falling snow is seen by a rotating camera, its motion direction will become the sum of the snowfall motion and the panning camera motion. Since we render the falling snow image velocity according to its depth, and the camera motion associates with only one image motion velocity, the sum of these two will produce a range of velocities, in which both the image motion speed and direction vary according to its depth.

4.1 The Sheared Tent

Let's recall Eq: (2.16), which expresses the image motion as the sum of a set of parallel image motion $(s, 0)$ and a panning camera rotation (s_v, s_l). This set of motion planes is re-expressed as:

$$\omega_t + s_v\omega_v + s_l\omega_l \;=\; -s\;\omega_v \tag{4.1}$$

In Chapter 3, we developed our tent model, in which Eq. (3.3) defines a set of parallel image velocities which takes into account the falling snow motion properties. More specifically, it specify the set of velocities sv of the parallel optical snow as $C/\sqrt{\omega_l^2 + \omega_v^2}$. If we substitute $C/\sqrt{\omega_l^2 + \omega_v^2}$ for s in Eq. (4.1), we obtain the equation which expresses the falling snow motion seen by a panning camera with velocity (s_l, s_v).

$$\omega_t + s_v\;\omega_v + s_l\;\omega_l \;=\; C\;\frac{\omega_v}{\sqrt{\omega_v^2 + \omega_l^2}} \tag{4.2}$$

We wish to simplify the camera panning motion and only consider the case that the camera is panning in perpendicular to the snow falling. Therefore, the camera panning velocity becomes $(s_l, 0)$, and Eq. (4.2) becomes:

$$\omega_t \;=\; C\;\frac{\omega_v}{\sqrt{\omega_v^2 + \omega_l^2}} - s_l\;\omega_l \tag{4.3}$$

Geometrically, Eq. (4.3) expresses a tent-like surface added to a plane. A plot of Eq. (4.3) with constant $C = 1$ and $s_l = 0.3$ is shown in Figure 4.1. Given that the shape of the plotted surface looks like the tent surface is sheared up, we name this rendering model as *sheared tent model.* Synthesizing the sheared tent surface and

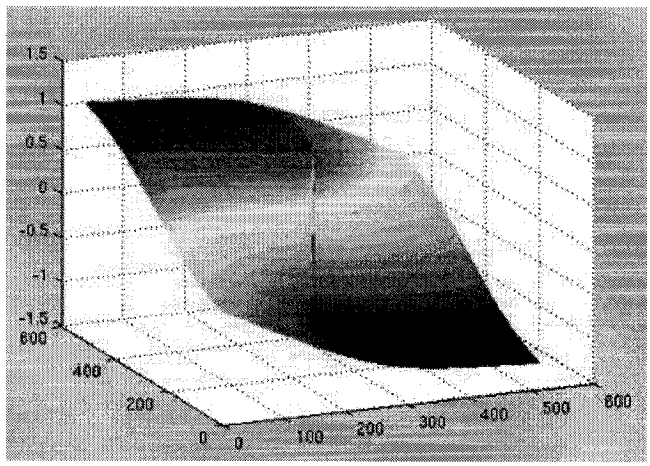


Figure 4.1: The plot of Eq. (4.3), with constant $C = 1$ and $s_{\omega_t} = 0.3$

compositing with the snowing scene follow exactly the same as in Sec. 3.2 and Sec. 3.3.

There are two coefficients in Eq. (4.3). The constant C relates to the falling snow motion speed, and s_l relates to the camera panning speed. Theoretically, the camera can pan with infinitely high speed. However, given that the constant C is limited to $[0, \pi]$ and (ω_l, ω_v) are limited to $[-\pi, \pi]$, the high camera panning speed will result in $s_l \gg C$, thus result in $s_l\ \omega_l \gg C\ \omega_v/\sqrt{(\omega_v^2 + \omega_l^2)}$. If we refer back to Eq. (4.3), we can see that this will lead to $\omega_t \approx s_l * \omega_l$. This is equivalent in saying that the snowfall motion will be diminished and the image motion will appear to be translational only. To avoid this effect, we need to keep these two coefficients compatible. Since $\omega_v/\sqrt{\omega_v^2 + \omega_l^2} \leq 1$ and $\omega_l \leq \pi$, we wish $s_l/\pi \approx C$ to ensure the compatibility of these two coefficients. One side effect of the constraint $s_l/\pi \approx C$ is that it limits the camera panning speed. This might be considered as the limitation of the shearing tent method.

4.2 Results

We present three video examples to demonstrate the rendering results of our sheared tent model. All the videos use lossless compression (MPEG quality = 90 %).

The video *panningsnow* is the falling snow opacity function which is rendered from the sheared tent model. It demonstrates the snowfall motion when it is seen by the panning camera. The video *panningsnowscene* is the *panningsnow* composited with the snowing scene *house* [10]. The panning motion of the scene *house* [10] is made by one pixel per frame. However, from the video *panningsnowscene* we can perceive that the panning motion of the background scene is faster than the *panningsnow*'s. This is because the *panningsnow* motion speed is less than one pixel per frame, for the reason of compatibility. A better result can be achieved by compositing the falling snow with a video whose shooting camera has roughly equal velocity as the *panningsnow*'s.

We also present the video *panningbgstill*, which composites the video *panningsnow* with the still background image *house* [10]. The panning motion of the falling snow is more perceivable in this video, we therefore present it as a comparative example.

To this end, we presented our falling snow rendering technique, which includes introducing the literature background, in Chapter 2, mathematically formulating and practically implementing the rendering technique, in Chapter 3, and extending the falling snow rendering as it is seen by a panning camera, in Chapter 4. Moreover, during the development of our rendering technique, we looked up and inspired by some psychophysical theories. After we developed our falling snow rendering method, we find that our rendered motion parallax could be used as a stimulus to help psychophysical study. In the following two chapters, we will present the literature background of

the psychophysical theories and show that how our rendered motion parallax could be used for psychophysical study.

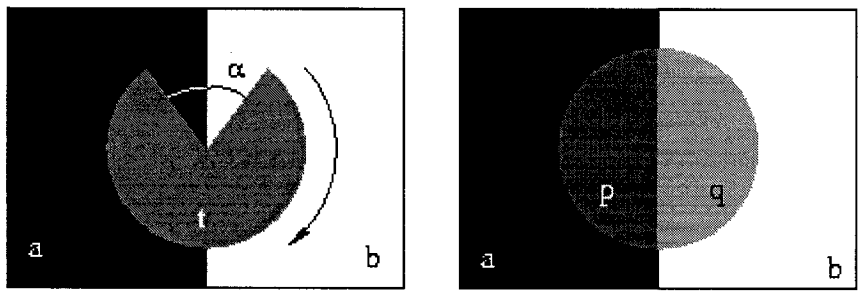
Chapter 5

Perceptual Transparency Theory Review

During studying the rendering experiments of the falling snow motion parallax, we frequently observe snowflakes passing through each other with perceivable depth. In order to make more compelling rendering effects, we wish the motion parallax with continuously many depth and wish the perceived motion as vivid as the real falling snowflakes. These rendering objectives motivate us to have a deep investigation of the transparency theory in human visual perception. Along with carrying on those interesting experiments, a nice side-effect we had is that the rendered motion parallax could be a new moving stimulus to study perceptual transparency. We will review the perceptual transparency literature in this chapter. In Chapter 6, we will introduce our preliminary transparency study experiments and discuss the results.

5.1 Metelli's Theory

Since Helmholtz(1866) spoke of seeing one color through another, Koffka(1935) related this problem to perceptual transparency, and later on many other researchers tried to answer *when* to scission [1], i.e. when the visual system initiates the percepts of transparency, and *how* to scission, i.e. how the visual system assign transmittance and reflectance values to the perceived layers, perceptual transparency theory has been richly studied. Among them, the most historical one is Metelli's "episcotister model" [19].



(a) (b)

Figure 5.1: Metelli's episcotister model. (a) a disc with reflectance t, and an open sector of relative area α, is rotated rapidly in front of a bipartite background. (b) when the disc is rotated fast enough, a transparent layer is perceived overlying the bipartite background.

Metelli built up the "episcotister model" (see Figure 5.1) and derived the *Metelli's equations* to answer how to scission. Also, he addressed the geometric and photometric conditions to approach when to scission. As showing in Figure 5.1, when a disc with reflectance t and an open sector of relative area α, is rotated fast enough in front of a bipartite background image, it will lead to perceive a transparent surface overlying

[1] Here we refer scission as visual system perceiving multiple transparency layers from single image intensity.

the bipartite background. From Talbot's law, which gives the mixing color in the region of the episcotister rotating over the background, Metelli derived the t and α as following,

$$\alpha = \frac{p-q}{a-b} \tag{5.1}$$

$$t = \frac{aq-bp}{a+q-b-p} \tag{5.2}$$

Eq. (5.1) and (5.2) are called *Metelli's equations*, since their significant influence on the perceptual transparency study.

Given the constraint that $\alpha \in [0,1]$, and from Eq. (5.1), Metelli observed the geometric condition of scission, i.e. the *magnitude constraint* and the *polarity constraint.* The former requires $|p-q| \leq |a-b|$ and the latter says that $(p-q)$ has the same sign as $(a-b)$. Besides, Metelli also observed two kinds of photometric constraints, i.e. the *contour continuity* of the underlying surface and the *boundary continuity* of the perceived transparent layer.

Albeit these insightful constraints Metelli made from Eq. (5.1), he claimed that Eq. (5.2) is too complicated to make a simple prediction. Indeed, there is a discrepancy between the prediction of the Metelli's model and the perceived transparency. Metelli's model predicts that when $(p-q)$ and $(a-b)$ meet both the *magnitude constraint* and the *polarity constraint*, a consistent transmittance value α is supposed to be derived from Eq. (5.1). However, even Metelli himself puzzled [19] that when "all other conditions being equal, the darker the transparent layer, the greater its perceived transparency". This discrepancy later on was solved by Singh and Anderson [32]. They showed that it is the variable - the Michelson contrast, which can be interpreted from Eq. (5.2), acted as a more adequate constraint to predict the

transmittance value α.

5.2 Michelson Contrast

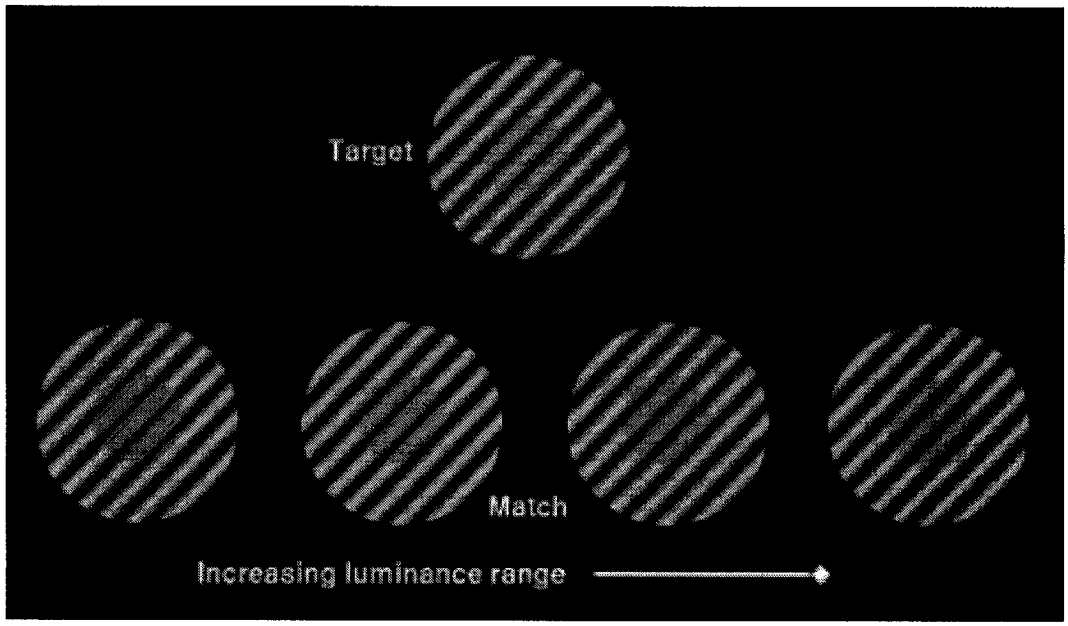


Figure 5.2: Singh and Anderson's transmittance matching experiments.

As showing in Figure 5.2, Singh and Anderson did the transmittance matching experiments to study how to scission. In their experiments, they place a smaller disk inside a large disk. The target central disk has the same mean luminance value as its sinusoidal background, and is assigned a fixed luminance range. The matching central disk is assigned a mean luminance value only, which is probably different from the target's. By adjusting the luminance range of the matching central disk, its transmittance value will change accordingly. During the experiments, the observer is instructed to find the transmittance value which matches the target's. Moreover, all

these central disks are placed on the 2D sinusoidal background, which is more general than the bipartite background as used in the Metelli's model. By examining the obtained data of matching central disk and the target disk, Singh and Anderson find that both the luminance range and the mean luminance affect the perceived transmittance value. More precisely, it is the Michelson contrast, which is the luminance range divided by twice of the mean luminance, affects the perceived transmittance.

If let M represents the Michelson contrast, let L_{range} represents the luminance range and let L_{mean} represents the mean luminance, then the Michelson contrast is defined as:

$$M = \frac{L_{range}}{2 * L_{mean}} \tag{5.3}$$

Therefore, instead of deriving the perceived transmittance value α by using *Metelli's equations* – Eq. (5.1), which only considered the luminance range, Singh and Anderson re-defined the perceived transmittance as:

$$\alpha = \frac{M_{center}}{M_{surround}} \tag{5.4}$$

Indeed, Eq. (5.4) can be derived from Eq. (5.2) ([32] : 504), which Metelli claimed as too complicated to make a simple prediction. Consequently, by applying Eq. (5.4), *Metelli's puzzle* – that a black episcotister looks more transparent than a white one – has an answer: the darker episcotister has higher Michelson contrast, so, it has higher transmittance value.

Besides addressing the Michelson contrast as the answer of how to scission, Singh and Anderson also applied it to the study of when to scission. As the *magnitude constraint* of the Metelli's model requires the central luminance range to be less than

the surround to initiate the perceptual transparency, Singh and Anderson give a counter-example as showing in Figure 5.3. In Figure 5.3, as the central luminance

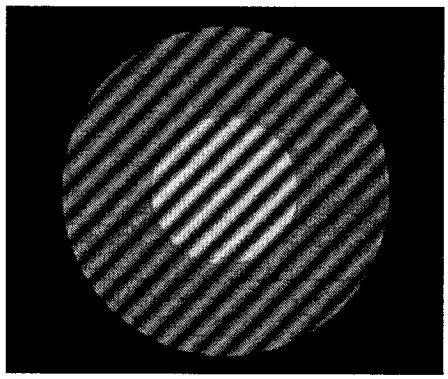


Figure 5.3: Example that the central luminance range is larger than the surround. The surround region is perceived to be transparent, and the central region is perceived as a background being seen through a hole of the transparent layer.

range is larger than the surround, the surround region is perceived to be transparent, and the central region is perceived as a background being seen through a hole of the transparent layer. Moreover, when keeping the luminance range of both the central and the surround regions unchanged but varying the mean luminance, the perceived transparency will change from perceivable to obscure. Indeed, from the experimental data results, Singh and Anderson show that it is the Michelson contrast acted as the critical variable to initiate the scission, not the luminance range. Furthermore, by placing the low contrast regions over the textured background, as show in Figure 5.4, Singh and Anderson extend the photometric condition of the Metelli's model. They find that, over the contrast varying regions, the continuity of the textures, for example, the left part of Figure 5.4, is the sufficient condition to initiate the scission. From these new geometric and photometric observations, Singh and Anderson suggest that "visual system uses changes in Michelson contrast over aligned contours and

groupable textures as a critical image property to initiate percepts of transparency" ([32]:515).

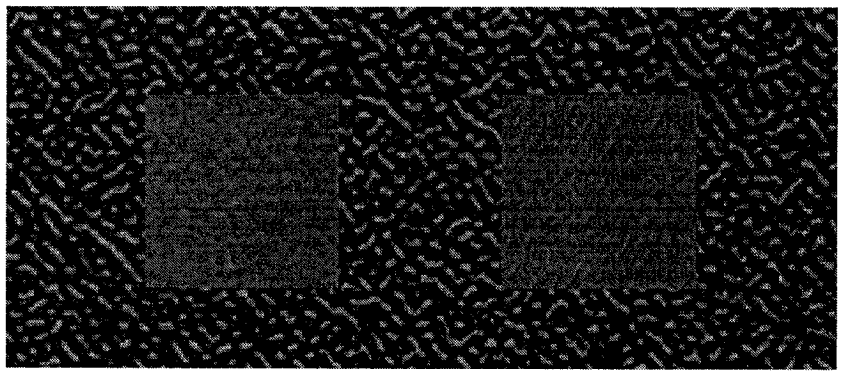
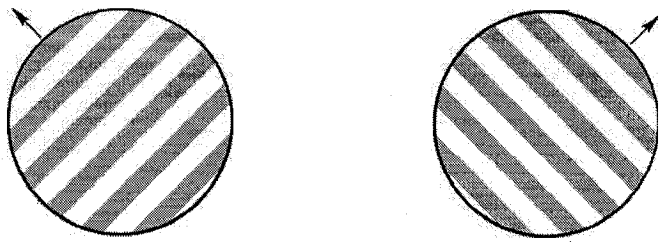


Figure 5.4: Texture background with two low contrast regions. The left low contrast region has the consistent texture with the background. The right one does not. The perceptual transparency can be perceived from the left low contrast region, but not the right one. Note that in the printed hard copy, this figure is not perceived as good as the one in the soft copy, because of the PS/PDF dithering problem. However we can at least perceive that the texture in the right low contrast region has orientation and density inconsistancy, which prevents initiating the percepts of transparency.

5.3 Coherent *vs.* Non-coherent Motion

Beyond the adequate perceptual transparency study of the static images, other research groups broaden this study to the moving images [34, 35, 36, 37, 11, 2, 21]. Stoner and Albright is one among them. They extended the stimuli of line gratings which are used by Movshon et. al [21] to the square-wave gratings and addressed the multiplicative luminance condition under which the motion of each single moving object can be perceived when there are multiple moving objects in an image. As in Figure 5.5, the model of their experiment is two square wave gratings, each with the same size, the same transmittance value, and the same background, but move up-left

and up-right accordingly. These two gratings are then superimposed one on top of another. If T_{sw} represents the transmittance value of the square wave, and let T_o



(a) (b)

Figure 5.5: Stoner and Albright's square wave gratings. (a) and (b) are with same size, same transmittance value and the same background, but move up-left and up-right accordingly.

represents the transmittance value of the overlap region. By adjusting T_o, Stoner and Albright find that when $T_{sw} * T_{sw} \leq T_o \leq T_{sw}$, the image motion is perceived as the two square waves moving in their own motion directions, i.e. non-coherent motion. Otherwise, the image motion is perceived as the two square waves moving together in a common direction, i.e. the coherent motion. Stoner and Albright therefore name the condition $T_{sw} * T_{sw} \leq T_o \leq T_{sw}$ as the multiplication condition and conclude that in order to perceive non-coherent motion, the image luminance condition has to obey the multiplication rule.

Following along with the previous three Sections, we had an overview to the perceptual transparency theory. In the next Chapter, we will see how these theories could apply on our motion parallax image sequences, and how our motion parallax image sequence could extend these theories.

Chapter 6

Perceptual Transparency Study Using Motion parallax

In Chapter 3, we rendered falling snow motion by using the tent model. The formulation was based on the motion parallax of the falling snowflakes, i.e. the size and the motion speed of the snowflakes correlate to the depth, and we refer the depth as infinitely many layers. The rendered results appear remarkably similar to the falling snow and with continuously many layers. We would like to take advantage of the rendered motion parallax and apply it to the perceptual transparency study. We carried on several informal experiments and had a preliminary study of the conditions of initiating the percepts of transparency when using the motion parallax as stimulus. These experiments are not carefully designed procedural work, but they can be considered as the initial explorations for the future psychophysics study. Also, we compared the perceived results with those perceived from the static images. We also ask how applicable are the previous transparency theories to our rendered motion

parallax.

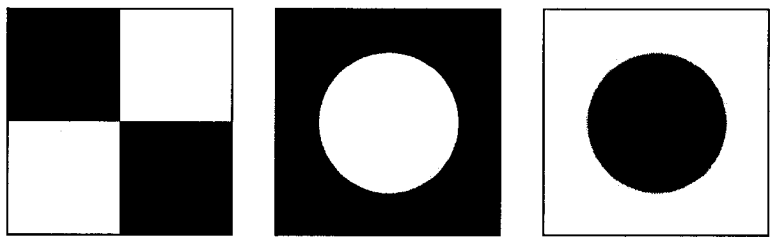
6.1 When to Scission: Conditions for Motion Parallax

In the review of the transparency study in Chapter 5, we learned of Metelli's model, as well *Metelli's puzzle.* We saw that this puzzle later on was well solved by Singh and Anderson, who discovered that the Michelson contrast is the actual variable used by the visual system to initiate scission. Besides, they also extended both of the geometric and the photometric conditions in Metelli's model, and made the conditions of when to scission more general as the following:

I: There is change in Michelson contrast over the aligned contours.

II: The texture display over the regions where Michelson contrast changes should be groupable.

However, the stimuli Singh and Anderson used are only static images. It is quite often that the visual system confront moving images and perform image decomposition on them. We therefore question when the visual system initiates perceptual transparency with the moving images. We investigate this problem by compositing the motion parallax which is rendered in Chapter 3 over the background images which are with geometric patterns.



(a) (b) (c)

Figure 6.1: The background images used to composite with the motion parallax to investigate the scission conditions of the moving images. (a) checkboard (b) moon (c) disc. The original image size is $512 * 512$.

6.1.1 Compositing Method

We render the motion parallax by using the tent model and synthesize the data in the same way as we synthesize the falling snow. After taking the inverse Fourier transform, we scale the data function within gray level intensity $[0, 255]$. The obtained image sequence is the motion parallax, which we used to study the perceptual transparency. We treat the motion parallax as the foreground image $I_{fg}(x, y, t)$ and composite it with the background image $I_{bg}(x, y)$. Since the purpose of this experiment is to study the conditions of initiating perceptual transparency when the media is the motion parallax, we design the background images as shown in Figure (6.1).

We wish the Michelson contrast of the composited image sequences changing over the geometric patterns. To achieve this purpose, we lower the Michelson contrast of the motion parallax in the white background regions and leave those in the black background regions unchanged. The compositing method is:

$$I(x, y, t) = \begin{cases} 255 * .7 + 0.3 I_{fg}(x, y, t), & I_{bg}(x, y) = 1 \\ I_{fg}(x, y, t) & I_{bg}(x, y) = 0 \end{cases} \tag{6.1}$$

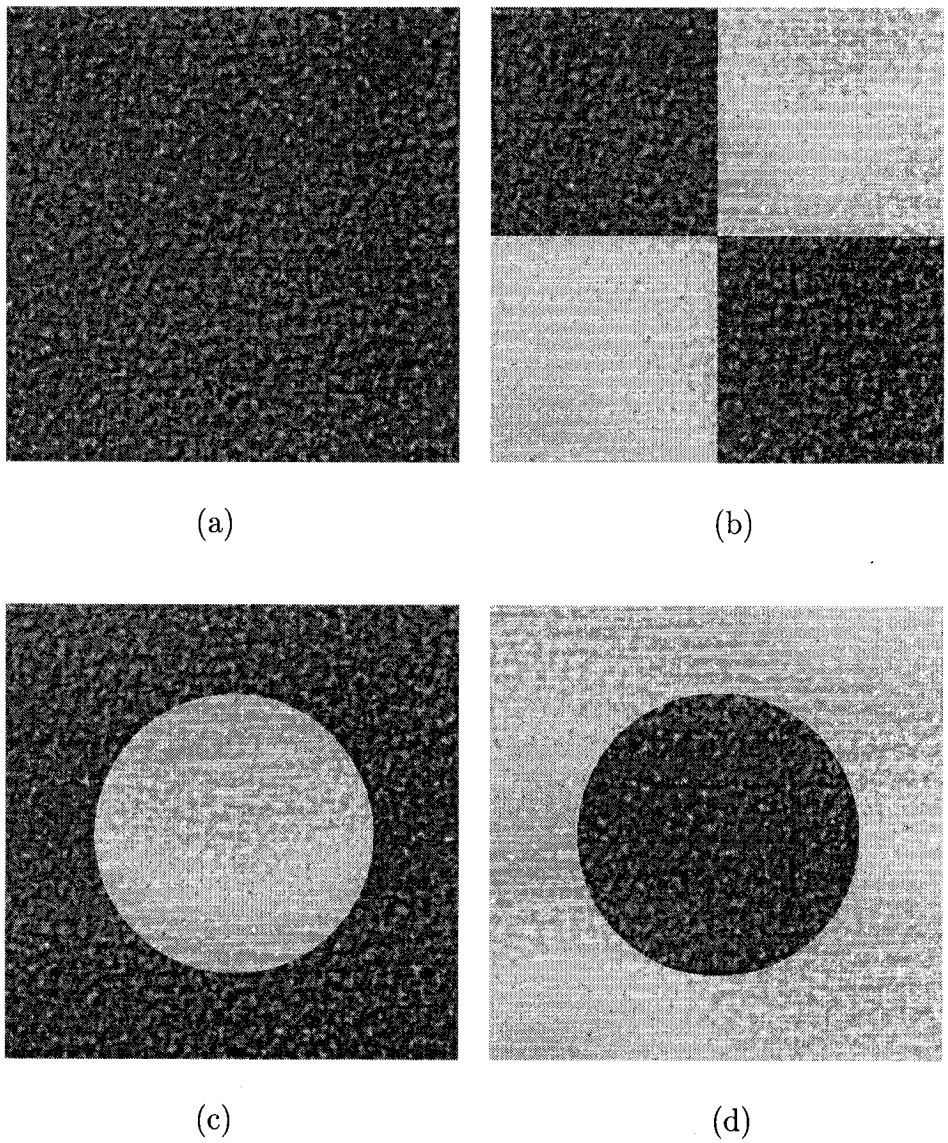
Based on this compositing method, we can compute the Michelson contrast of

both the black and white background regions. The motion parallax is obtained from the inverse Fourier transform. It is the sum of the sine waves, therefore, its mean value is 0. After we rescale the data function, the luminance range is scaled as $[0, 255]$, and the mean is scaled as 128. Let's recall Eq. (5.3), Michelson contrast is the luminance range divided by twice of the mean luminance. Since the compositing method leaves motion parallax in the black background regions unchanged, the Michelson contrast of these regions will be the Michelson contrast of the motion parallax itself, it is 1. From Eq. (6.1), and given that the image luminance difference is $255 * 0.3$ and the image mean luminance is $(255 + 255 * 0.7)/2$, the Michelson contrast of the white background regions can be easily computed as 0.176.

6.1.2 Result and Discussion

After compositing, the motion parallax changes the Michelson contrast along the geometric figural contour. The compositing results are shown in videos *checkboard*, *moon* and *disc*. The motion parallax itself is shown in the video *parallax*. One frame of each video is shown in Figure 6.2.

The static images in Figure 6.2 (b), (c) and (d) all meet the scission conditions concluded by Singh and Anderson. Therefore, it is not surprising that a transparent layer can be perceived from all of them. Indeed, the perception we got is that the geometric background images are split into two layers along the figural contours. The white regions are perceived overlaying the motion parallax patterns as a light transparent layer, and the black regions are perceived laying behind the motion parallax patterns as a black background layer. The motion parallax pattern itself is perceived



(a) (b)

(c) (d)

Figure 6.2: (a) One frame of the rendered motion parallax. (b), (c) and (d) One frame of the motion parallax composited with the background image checkboard, moon and disc respectively. In (b), (c) and (d), the low contrast regions are perceived as transparent layers overlaying on the motion parallax patterns.

with continuous 3D depth (see Figure 6.2 (a)). This fact could increase the perceived distance between the perceived transparent layer and the background layer and might cause the effect of scission stronger.

The last statement can be perceived more clearly by observing the moving images. In the moving images, not only the structure, but also the moving speed of the motion parallax gives depth cue. Since the continuous depth is evidently perceivable, when perceiving the transparent layers overlaying on the motion parallax, these transparent layers will inherit the depth cues of the motion parallax, thus are perceived as much closer than the black background layer.

Stone and Thompson investigated the effects of image contrast on image speed perception [34]. Using grating stimuli, they observed that the low-contrast gratings are perceived moving slower than the high-contrast ones. This perception effect is also shown on the motion parallax stimulus. From the video *checkboard*, *moon* and *disc*, we could perceive that the image motion speed in the low contrast regions is slower than those in the high contrast regions. However, this image speed difference does not show significant impact on perceiving transparency for our examples. There could be two issues to this fact. One is that the difference is too small to affect the overall image motion consistency. Another one is that the motion speed consistency is not the dominant effect to affect the perceptual transparency.

To clarify our doubts, we did a motion-inconsistency experiment. This experiment is quite similar to the experiment *moon*, except that the motion direction of the motion parallax in the central disc part has 90 deg angular difference from the motion direction in the surround part. The experimental result is shown in the video

motion_inconsistent, and one frame of the video is shown in Figure 6.3.

In Figure 6.3, a transparent layer can be perceived overlaying central low contrast region. This is plausible because the pattern structures of these two motion parallaxes are very similar, and the motion direction difference does not show on the still image of course. However, from the video *motion_inconsistent*, the central transparent layer could be perceived as well, even though the motion parallaxes in the central and surround regions move in two very different directions. This effect might suggest that the motion velocity consistency over the Michelson contrast changing regions is not a critical condition of perceiving transparency.

This is quite different from the photometric condition of the static images. In static images, the groupable texture over the Michelson contrast changing regions is a critical scission condition, recall Figure 5.4. However, in moving images, we are able to perform image scission from the ungroupable motion.

To this end, we hypothesize that the scission conditions – the change of the Michelson contrast is over aligned contours and over groupable texture – are applicable to the motion parallax stimulus as well. However, the groupable velocities of the motion parallaxes over the aligned contours where the Michelson contrast changes might not be necessary. In addition, the scission layers could inherit the depth cues of the motion parallax, thus might promote the scission perception.

Nevertheless, our hypotheses are only based on the limited experiments. Further detailed and extensive studies are certainly necessary in order to have a general and comprehensive suggestion to this topic.

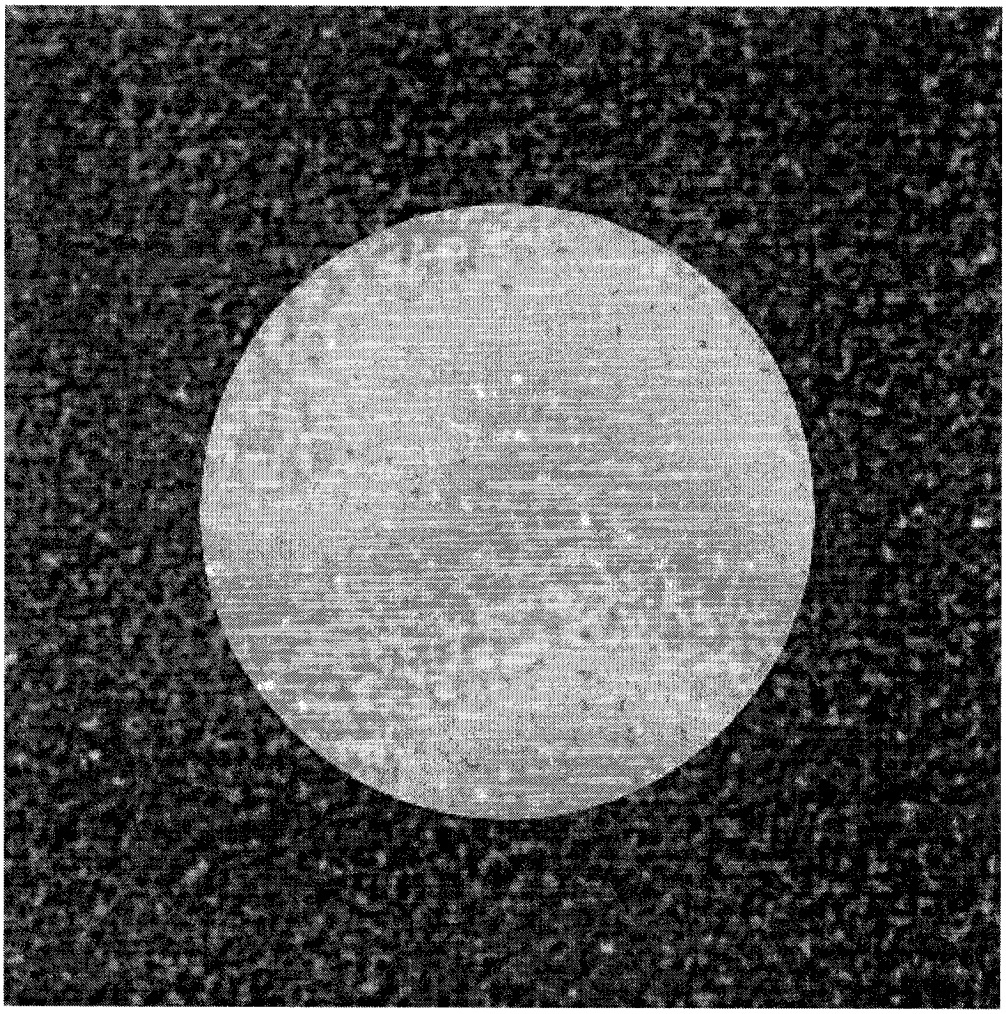


Figure 6.3: One frame of the video *motion_inconsistent.* The motion parallax in the central low contrast region moves down-left, whereas the one in the surround high contrast region moves down-right. Since the pattern structures of these two parallaxes are very similar, the motion direction difference does not show on this static image. Therefore, we are still able to perceive a transparent layer overlaying central low contrast region.

6.2 Image Motion Analysis: Using Motion Parallax as Stimulus

As we reviewed in Sec. 5.3, Stoner and Albright used gratings as stimuli and observed that the visual system computes the multiplicative transparency to decide the coherency of the motion. However the stimuli they used are quite strict, which are two superimposed gratings with the same transmittance value and moving along the fixed directions. These stimuli were generalized later on by other researchers in terms of varying the motion direction and the contrast of the gratings. Along with the generalization, the research results also became more broad and accurate [37, 35, 36, 11, 2, 21]. For example, Kim and Wilson [11] varied the relative motion directions of the two gratings and suggested that it is the motion directions of the gratings that mainly affects the coherency of the observed motion. Stoner and Albright [36] themselves later on varied the relative contrast of the gratings and observed that the luminance contrast difference of the two superimposed gratings could affect the motion coherency by acting as a depth-from-occlusion cue. More specifically, when two superimposed gratings are put on the same black background, the higher contrast grating tends to be perceived in front of the low contrast grating and acts as an occluding grating, whereas the luminance value of the overlap regions of these two gratings is not critical anymore.

From the previous examples, we could notice that the stimuli used in those research are mainly limited on gratings. When the real natural scenes are much more complicated than two superimposed gratings and there exists more than one cue,

we question whether the grating stimuli are good enough for image motion study. Intuitively, we would like the stimuli to be more similar to nature.

In this sense, the motion parallax we rendered could suggest a new perspective for the image motion study. Since we render the motion parallax in the Fourier domain, it is the sum of sine waves, it therefore appears much more complicated than the gratings. Moreover, the primary goal of rendering this stimulus is to simulate the falling snow and the formulation of this stimulus is based on falling snow motion, we therefore naturally acquire a stimulus which is very similar to the nature.

Staring at the motion parallax, we could observe that the image objects are passing through each other and the motion incoherency is obviously existing. What cues are acting here? We could suggest the luminance contrast difference as one cue. From the motion parallax, we could observe that the image patterns with higher luminance contrast are in front of those with lower luminance contrast. We could interpret our observation as higher contrast image patterns acting as the occluding objects and therefore being observed as in front, whereas the lower contrast image patterns are acting as the occluded objects and therefore being observed as behind. We might also try to apply the luminance multiplicative condition [37]. However, a difficulty here is that how do we exactly locate the overlapping regions among the objects of the motion parallax and address the luminance value?

This discussion leads us to a number of open questions. When the previous research results are based on exactly two gratings and are with exactly two layers, we ask if we still can use those theories to give an adequate image motion analysis to our rendered motion parallax? For example, when multiple layers exist in the motion par-

allax, can we apply the multiplicative rule to the luminance values of the overlapping regions of the multiple layers? When the luminance contrast differences exist among more than two objects of the motion parallax, do we interpret the depth order of the objects according to the luminance contrast monotonically? When the transparency and the depth-from-occluding cues are both existing in the motion parallax and even somewhat conflicting with each other, how does our visual system interpret them?

From these questions and uncertainties, we can see the motivation and necessity of using motion parallax as stimulus to study image motion. Although carefully designed experiments have not been carried out in this thesis, because the enormous work load and challenging questionnaire could be another research topic. However, we could predict that motion parallax could be a promising stimulus for future research.

Chapter 7

Summary

This thesis presents an image based falling snow rendering method. It is based on the spectral synthesis technique, and is carried on in frequency domain. Synthesizing the power spectrum is based on the tent model, which is developed in this thesis, and is also the major contribution of this thesis. The inspiration of developing the tent model comes from the motion plane model. It is saying that a pure translational image motion in the space-time domain produces a plane of energy in frequency domain. This motion plane comprises two important properties, they are, the common line of the motion plane (see Eq. (2.9)) relating to the motion direction and the slope of the motion plane relating to the motion speed. Associating these two properties with the natural falling snow motion property, i.e. the size and the speed of falling snowflakes relate to the depth, we develop a tent like surface, and name it as tent model. The power spectrum we synthesized in frequency domain is exactly along this tent model. We also apply $1/f$ scaling law and limiting the range of spatial frequency to the synthesized power spectrum to make the snowflakes equally visible at all the

depth and give them reasonable size. Afterwords, we bring the data function of the power spectrum to space-time domain by computing IFFT. We then treat the data function as the opacity function and composite a white unitary snow image, which is with intensity 250, with a background scene image or video to create the falling snow scene.

We also extend our falling snow rendering to the case when the motion is seen by a panning camera. Using the property of the sheared common line in the optical snow model, we shear up the tent surface to incorporate the panning camera motion. A special feature of this method is that it allows us to render the camera motion with non-integer pixels per frame without aliasing, since the rendering work is carried on in frequency domain. However, given the limitation of the shape of the tent surface, the incorporated camera panning speed is indeed always limited as less than one pixel per frame. This causes an inconsistency between the camera motion and the background image motion, because in space-time domain, the image motion is always integer pixels per frame. As a result, this special feature turns our sheared tent model to be not very useful.

The rendered falling snow motion sequence comprises the depth *vs.* the speed *vs.* the size relationships of the falling snowflakes. If treating the rendered motion parallax itself as a motion stimulus, it would include more motion cues than the conventional grating stimulus. Also, since we considered the natural image amplitude spectra distribution property and the sizes of the image objects, the rendered motion parallax per se is more similar to the natural scene than the moving gratings. We therefore suggest that our rendered falling snow motion parallax could be a novel

stimulus for the perceptual transparency study and image motion analysis.

Bibliography

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A*, 2:284 – 299, 1985.

[2] E. H. Adelson and J. A. Movshon. Phenomenal coherence of moving visual patterns. *Nature*, 300:523 – 525, 1982.

[3] B. L. Anderson. Stereoscopic surface perception. *Neuron*, 24:919 – 928, 1999.

[4] J. F. Blinn. Compositing, part I: Theory. *IEEE Computer Graphics and Applications*, 14(5):83 – 87, September 1994.

[5] R. N. Bracewell. *The Fourier Transform and Its Applications.* McGraw-Hill, NY, 1965.

[6] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley. *Texturing and Modeling – A Procedural Approach.* Morgan Kaufmann, San Francisco, CA 94104-3205, 2003.

[7] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4:2370 – 2393, 1987.

[8] D. J. Fleet. *Measurement of Image Velocity.* Kluwer Academic Press, Norwell, MA, 1992.

[9] J. Foley and W. Kim. Icl - the image composition language. *IEEE Computer Graphics and Applications*, 7(11):26 – 35, November 1987.

[10] G. Johnson. *Jimmy's place.* `http://members.aol.com/garyjohnson01/page3.html`.

[11] J. Kim and H. R. Wilson. Dependence of plaid motion coherence on component grating directions. *Vision Research*, 33:2479 – 2489, 1993.

[12] M. S. Langer and R. Mann. Optical snow. *International Journal of Computer Vision.* Accepted for publication: 10/31/02.

[13] M. S. Langer and R. Mann. Dimensional analysis of image motion. In *Eighth IEEE International Conf. on Computer Vision*, pages 155 – 162, 2001.

[14] J. P. Lewis. Generalized stochastic subdivision. *ACM Transactions on Graphics*, 6:167 – 190, 1987.

[15] J. P. Lewis. Algorithms for solid noise synthesis. *Computer Graphics*, 23(3):263 – 270, July 1989.

[16] A. W. Lohmann, D. Mendlovic, and G. Shabtay. Significance of phase and amplitude in the fourier domain. *Journal of the Optical Society of America A*, 14:2901 – 2904, 1997.

[17] D. K. MaAllister. The design of an API for particle system. In *UNC Computer Science Tech Report*, January 2000.

[18] G. A. Mastin, P. A. Watterberg, and J. F. Mareda. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications*, 7(3):16 – 23, March 1987.

[19] F. Metelli. The perception of transparency. *Scientific American*, 230:90 – 98, 1974.

[20] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceeding of SIGGRAPH '95*, August 1995.

[21] J. A. Movshon, E. A. Adelson, M. Gizzi, and W. T. Newsome. The analysis of moving visual patterns. In *Study Group on Pattern Recognition Mechanisms*, pages 117 – 151, 1985.

[22] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. *Computer Graphics*, 23(3):41 – 50, July 1989.

[23] K. Perlin. A unified texture/reflectance model. In *Proceeding of SIGGRAPH '84*, July 1984.

[24] K. Perlin. An image synthesizer. In *Proceeding of SIGGRAPH '85*, July 1985.

[25] T. Porter and T. Duff. Compositing digital images. In *Proceeding of SIGGRAPH '84*, pages 253 – 259, 1984.

[26] W. T. Reeves. Particle system — a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2:91 – 108, 1983.

[27] G. Sakas. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer*, 9:200 – 212, 1993.

[28] A. V. D. Schaaf and J. H. Hateren. Modeling the power spectra of natural images: Statistics and information. *Vision Research*, 36:2759 – 2770, 1995.

[29] M. Shinya and A. Fournier. Stochastic motion - motion under the influence of wind. In *Eurographics '92 Proceedings*, pages 119 – 128, September 1992.

[30] E. P. Simoncelli and D. J. Heeger. A model of neural responses in visual area MT. *Vision Research*, 38:743 – 761, 1998.

[31] K. Sims. Particle animation and rendering using data parallel computation. In *Proceeding of SIGGRAPH '90*, August 1990. Dallas, TX.

[32] M. Singh and B. L. Anderson. Toward a perceptual theory of transparency. *Psychological Review*, 109:492 – 519, 2002.

[33] J. Stam. A simple fluid solver based on the FFT. *Journal of graphics tools*, 6(2):43 – 52, 2001.

[34] L. S. Stone and P. Thompson. Human speed perception is contrast dependent. *Vision Research*, 32:1535 – 1549, 1992.

[35] G. R. Stoner and T. D. Albright. The interpretation of visual motion: Evidence for surface segmentation mechanisms. *Vision Research*, 36:1291 – 1310, 1996.

[36] G. R. Stoner and T. D. Albright. Luminance contrast affects motion coherency in plaid patterns by acting as a depth-from-occlusion cue. *Vision Research*, 38:387 – 401, 1998.

[37] G. R. Stoner, T. D. Albright, and V. S. Ranachandran. Transparency and coherence in human motion perception. *Nature*, 344:153 – 155, 1990.

[38] A. B. Watson and A. J. Ahumada. Model of human visual motion sensing. *Journal of the Optical Society of America A*, 2:322 – 342, 1985.