Layered motion field visualization: perceptual issues

M. S. Langer*, D. Rekhi, J. Pereira, A. Bhatia
School of Computer Science, Center for Intelligent Machines (CIM)
McGill University, Montreal, Canada

Abstract

Layered motion fields arise in natural vision in many situations, including self-motion in a cluttered scene, motion of a fluid, and transparency. Layered motion fields have the property that there are multiple velocities present near each 2D spatial location. As such, standard 2D motion visualization methods do not apply, since they allow for only a single velocity vector at each image position. This paper examines perceptual issues that arise in visualizing layered motion fields. A key issue is that the human visual system is severely limited in how well it can process such fields. We give a thorough review of the relevant psychophysical literature, and focus on experiments that test how well the human visual system can detect spatial discontinuities and discrete layers in motion fields. We then present a specific layered motion visualization method. We demonstrate the limitations of the human visual system in perceiving the layered motions produced by this method.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1 Introduction

Although the human visual system (HVS) is capable of processing an enormous amount of information, it does have its limits. Of all possible image sequences with P pixels and T frames that can be defined on a pixel grid, the vast majority are interpreted by the HVS as meaningless patterns of flickering light (white noise). Rather than being able to understand any arbitrary time-varying image, the HVS is specialized for a very restricted subset, those that correspond to motions of objects.

There are many types of motions. These include optical flow that is generated when an observer moves through a 3D scene, motion texture such as fluid flow and smoke, motion of animals walking or running, layered motions that are produced by transparency and specularities off smooth surfaces.

In certain cases, motion can be described as pure 2D deformations, namely, 2D intensity patterns that are morphed over time along a smooth 2D vector field. In other cases, however, this 2D model is insufficient to capture the richness of the motion. The motion field can possess many discontinuities in space-time, and it can also have multiple layers. In this paper, we discuss the problem of how to visualize such multi-layered motion fields.

It is clear from everyday experience that the HVS can process layered motions *to some extent*. The question that motivates this research is, to what extent? First, the HVS can process only a limited range of speeds, sizes, and spatial contrasts of in a moving image. Second, it can process only a limited set of spatial and temporal *gradients* in the 2D motion field. For example, although the HVS can represent two moving layers quite well, and three layers to some extent, distinguishing more than three layers is near impossible. In this paper, we review some of what is known about these basic constraints. We then apply these constraints to an earlier method we proposed for visualizing layered motions.

*email: {langer, drekhi, javeen, aditya}@cim.mcgill.ca

2 Background

2.1 Psychophysics

We begin by briefly reviewing some of the findings obtained from human psychophysics in the processing of motion layers.

2.1.1 Sensitivity to spatial gradients

Various experiments have been conducted to estimate the sensitivity of the HVS to spatial gradients in the motion field. Image sequences were composed either of moving dots or moving 2D sinusoids. The sequences were synthesized such that a single frame of the sequence contains no information about where the motion boundary is. This was done to study how the motion information is processed, independently of pure spatial intensity cues such as edges.

In [van Doorn and Koenderink 1982c] the view field was divided into two halves, each half having a random dot pattern moving at a different velocity. Experiments were conducted with different parameters (magnitude and direction) for the average velocity and the difference velocity between the two half-fields. A Weber type law was found such that the ability to perceive a gradient depends on the ratio of the the difference velocity to the average velocity. The experimental results showed that this ratio (also known as the Weber fraction) had to be nearly 100% for the observer to notice the gradient. This indicates that humans have a limited ability in perceiving such 2D motion gradients.

A similar experiment was conducted in [van Doorn and Koenderink 1982b]. The view field was divided into horizontal stripes containing white noise with alternating stripes having velocities in opposite directions. For broader stripes, the observer was able to perceive the alternating stripes moving in opposite directions. As the stripes became narrower, the observer wasn't able to distinguish the different stripes and the entire image field appeared incoherent. Interestingly for very narrow stripes, the image field gave the appearance of two transparent layers moving in opposite directions.

Returning to the case of spatial gradients, [Watson and Eckert 1994] estimated the sensitivities using bandpass filtered noise rather than random dots. The motion directions alternated between stripes, where the stripes were now blended together to avoid within-frame spatial cues. The task was to identify when the stripes were present. Detection performance was studied as a function of the ratio of spatial frequency of stripes (envelope) to the peak spatial frequency of the translating intensities (carrier). It was found that perception of the stripes was poor if this ratio was either too high or low, indicating the HVS is limited in detecting spatial frequencies in the 2D motion field that are either too low or too high.

In [Mestre et al. 2001], a random dot stimulus was used and the display field was again divided into horizontal stripes. However this time only the speeds (and not the direction) between the alternating stripes were varied. In order to segment the alternating stripes, subjects required a higher speed difference between the stripes when the stripes were narrower. This again shows that the HVS sensitivity to motion gradients falls when the bars become narrower, i.e. high spatial frequencies in the motion field.

2.1.2 Sensitivity to motion layers

The above studies showed indirectly that the HVS is able to perceive two moving layers. Numerous studies have addressed this question in a more direct manner. [Adelson and Movshon 1982] investigated two layers defined by moving parallel lines and the rules by which the HVS perceives two vs. one moving layer. [Stoner et al. 1990] extended this method to layered transparency phenomena, using the rules of compositing.

While the HVS can easily perceive up to three superimposed global motion layers, more than three layers is much more difficult to perceive. [Andersen 1989] used a random dot pattern and assigned different speeds to different groups of dots so as to simulate different moving layers. Experiments were carried out with one, two, three, four and five layers and subjects were asked to determine the number of layers in each of these different cases. The subjects were able to determine accurately the number of layers when the number of layers was less than or equal to three, but were not able to distinguish between the three, four and five layer case.

Despite this evidence that the HVS can perceive up to three distinct moving layers, most studies have investigated the question of two vs. one layer. For example, suppose we had just two overlapping layers moving at different speeds. What should the difference in speeds be, so that the HVS can segregate the two moving layers? [Masson et al. 1999] presented subjects with moving random dots and investigate the minimum speed difference that was required between the two moving layers so that subjects could perceive two and only two layers. The speed discrimination threshold, defined as the ratio of speed difference (at which the subject can accurately segregate two layers) to the average speed, was determined for different average speeds and stimulus durations. The threshold increased as the average speed was increased and decreased as the stimulus duration was increased. This showed a longer stimulus durations helped the subject in segregating two moving layers more easily. For example, for duration of 500 ms, a difference of 20% in speed was needed.

[Braddick 1997] asked the subjects to judge whether the angle between the directions of two superimposed groups of moving dots appeared less than or greater than 90 degrees. The precision with which the subjects judged the relative angle was compared with the case where the subjects were presented with just a single group of moving dots and were asked to judge the angle between a fixed straight line and the direction of dot motion. The standard deviation in the case of two layers of dots were found to be nearly three times higher than for the motion of one layer compared to a static line.

A final example illustrates some of the subtleties of the HVS in processing motion layers. [Qian et al. 1994a] manipulated two fields of dots moving with opposite velocities. When the dots in each field were chosen independently, two superimposed layers were perceived as expected. However, when the dots in the two fields were chosen in pairs (one dot per pair from each field) and each pair of dots consisted of two nearby dots moving in opposite directions, then the HVS no longer could perceive the two layers. These findings can be explained in terms of the underlying physiology of HVS [Qian et al. 1994b], and further emphasize that there are concrete spatial limits in how well the HVS can process motion fields.

2.2 Graphics methods for 2D motion visualization

In Sec. 3, we will discuss a particular method for visualizing 2D layered motion fields. Before we do so, we briefly review several non-layered methods.

Perhaps the most straightforward method is to use particle systems. Moving particles are excellent for representing very local aspects of the flow, since the human visual system can track single particles with great accuracy using smooth pursuit eye movements.

Many other interesting methods have been invented which are not based on particles, however. One example is spectral synthesis methods [Mandelbrot 1977; Voss 1988]. These create a motion by adding up 2D sinusoids and varying the phase of the sinusoids from frame to frame. The phase change is determined by a dispersion relation. Examples of rendered motions include ocean waves [Mastin et al. 1987] and dynamic clouds [Sakas 1993]. (See [Ebert et al. 2003] for a review.)

The above spectral methods create global motion patterns. Others have created local motion patterns by varying local phase. [Freeman et al. 1991] created an illusion of motion, using a quadrature pair of oriented filters that are applied to a single image frame. The percept of motion is produced by phase-shifting these filters over a sequence of frames.

Other methods have relied on spatial orientation cues to motion. The LIC (Line Integral Convolution) algorithm [Cabral and Leedom 1993] convolves a white noise texture along streamlines of a 2D motion field, using a kernel filter. It produces orientated spatial structure parallel to the direction of motion. In the basic version, a static image frame is produced. The orientation of the flow is given by the local orientation of the (blurred) image. The speed and direction of the flow can be represented by using a time varying phase-shifted periodic convolution kernel to generate a sequence of animation frames. Various improvements have been made to this basic LIC technique, like extending it to 3D [Interrante and Grosch 1998], and to non-rectilinear grids [Forssell and Cohen 1995].

A related method is spot noise [Wijk 1991]. The basic primitive is a spot, e.g. an ellipse. These spots have random intensities and are randomly distributed to generate a texture. The orientation and size of each spot are related to the flow vector where the spot is placed. Each spot is like a streak in the direction of this vector, with longer streaks indicating faster flow. Like LIC, the direction of flow is ambiguous up to 180 degrees. Only by turning the spots into a moving particle system can this ambiguity be resolved.

These flow visualization methods yield interesting results, but they were not intended for visualizing *layered* motions. In this paper, we discuss a spectral based method that is used for layered motions. Although the method only motion cues, it can be extended to spatial cues as well e.g. by adding particles [Langer et al. 2004] or motion streaks.

3 Method

The examples that we present later in the paper use two types of layered motion fields. The first type is the motion seen by an observer moving through a static scene. This is commonly called "egomotion". The second is the motion generated by a fluid model. We begin this section by briefly reviewing these motion field models. We then present the details of our method. In the following section, we demonstrate the method with a several examples.

3.1 Two types of layered motion fields

3.1.1 Egomotion

[Longuet-Higgins and Prazdny 1980] developed a model of motion fields seen by an observer moving through a static 3D scene. The

image velocity of a point at a given instance of time is as follows. Let (T_x, T_y, T_z) be the instantaneous translation component of the camera motion in the x, y and z direction and $(\Omega_x, \Omega_y, \Omega_z)$ be the rotation component of the camera motion in x, y and z direction. Assume that the image projection plane is located at a depth f from the camera (centered at the origin) and $z(x,y)$ represents the projection of a 3D point to co-ordinates (x,y) in the image plane. For layered motions, multiple $z(x,y)$ values are needed for each x, y, namely one for each layer.

According to this model, the image velocity at (x,y) is then :

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \frac{T_z}{z(x,y)} \begin{bmatrix} x - x_T \\ y - y_T \end{bmatrix} + \begin{bmatrix} xy/f & -f - x^2/f & y \\ f + y^2/f & -xy/f & -x \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}$$

The image location $(x_T, y_T) = \frac{f}{T_z}(T_x, T_y)$ is called the direction of heading, also called the *focus of expansion* (FOE).

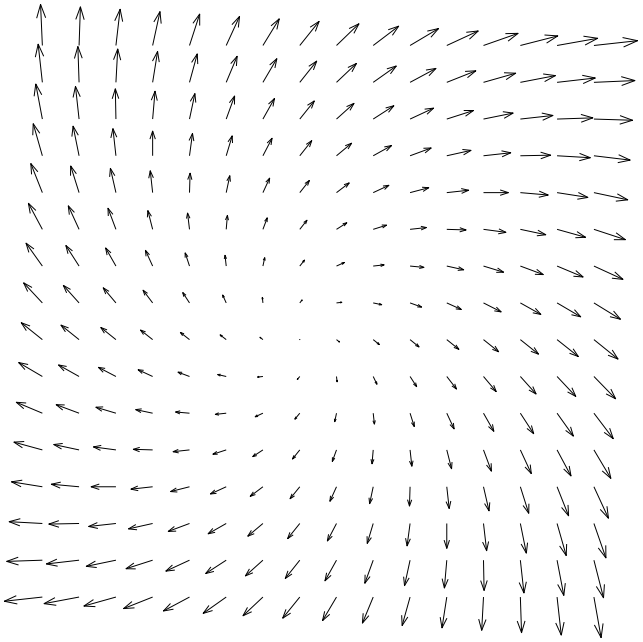

Figure 1: An example of a vector field for a single motion layer (constant depth). Here the camera is translating forward and rotating about the optical (z) axis. The axis labels indicate viewing angle, where (0,0) is the direction of the optical axis.

For any single depth layer, the 2D image velocity field thus is the vector sum of translational component and rotational component. The translation components of the vector field always points away from the direction of heading. The length of the vector for that depth layer is inversely proportional to $z(x,y)$. Figure 1 is an example of a vector field generated using above equations, where the translation direction is straight ahead and the rotation is about the optical axis. The velocity field for just one depth layer, $z = constant$, is shown.

3.1.2 Fluid Flow

The second type of motion is that of a 2D fluid. We used the simple fluid solver of [Stam 2001] to generate a "turbulent" time-dependent 2D velocity field. The method generates velocity fields for incompressible fluids and dynamically reacts to the forces supplied by the user. To get a turbulent fluid flow we designed a time-dependent force field that is a sum of two components. The first component is a smooth vector field. The second component is a high frequency vector field with a zero mean. Figure 2 shows an example of fluid flow field, at one particular instant of time, generated using this method.

For the demonstrations we show later, we chose the motion in the different layers to differ in their speed only. In the context of a fluid, one can think of the speed as a decreasing function of height, that is, decreasing toward a boundary layer.

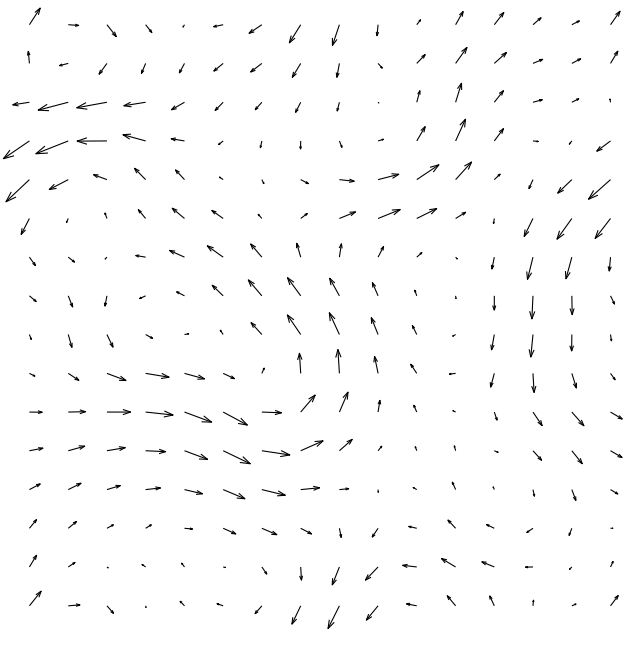

Figure 2: An example of the vector field generated by the fluid flow technique.

3.2 Dispersion relation for layered motion

Given a vector field generated by either of the above methods, our method generates a layered motion field using a phased-based spectral synthesis method [Langer et al. 2004]. For the case of egomotion flows, different depth layers move at different speeds because of linear perspective, i.e. because of the $1/z(x,y)$ term in the egomotion equation above. For the case of fluids, we can think of different layers as having different speeds because of constraints from a boundary layer.

To visualize the various layers, we related the size of moving intensities patterns in each layer to the speed of these patterns. For example, in the case of egomotion and in the absence of a camera rotation ($\Omega \equiv 0$), high 2D spatial frequency components (small wavelengths) correspond to far depth layers and move with lower image speed and low spatial frequency components (large wavelengths) correspond to near depth layers and move with higher speed. As derived in [Langer et al. 2004], this perspective effect is captured by the following *dispersion relation* between the spatial frequencies (f_x, f_y) and the temporal frequency f_t.

$$f_t = C \frac{\cos\theta \; f_x + \sin\theta \; f_y}{\sqrt{f_x^2 + f_y^2}} \tag{1}$$

If camera rotation were present, then a constant rotation component would need to be added. For the sake of simplicity, we ignore the rotation component for now.

The angle θ is the image direction of the motion. The spatial frequencies f_x, f_y are measured in cycles per M pixels and the temporal frequencies are measured in cycles per T frames. For reasons we will see later, we use $M = 64$ and $T = 128$.

What is the constant C ? Consider the case of a velocity vector with the magnitude (i.e. speed) of V pixels/frame. If this speed V is associated with the lowest spatial frequency f_{min} that we render, then we define:

$$C \equiv -\frac{T}{N} V f_{min} . \quad (2)$$

3.3 Tiles and discrete layers

Because the motions fields are not constant over the image domain, the variables C and θ in the dispersion relation must be made to depend on (x,y). To account for these spatially varying velocities, we synthesized the motion in small tiles of size $M \times M$ where $M = 64$. We considered two different tile overlaps (16 and 32 pixels). C and θ are constant within each tile, but vary between tiles.

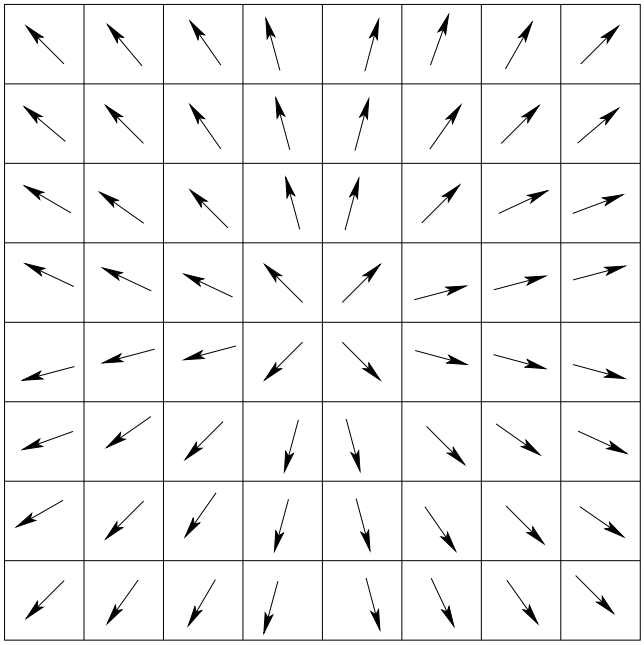

Figure 3: A tiled field for the case of pure forward motion, and no overlap between tiles. The arrows represent the *direction* of motion, which is constant within each tile. At the focus of expansion (FOE), the motion direction has a singularity. The magnitudes of the motion field (not shown) go to zero at the FOE, regardless of the depth of the various layers.

As explained in Sec. 2.1, the HVS can perceive up to three overlapping moving layers. This implies that if we render each tile separately, then it is sufficient to render three layers in each tile. Having more layers than this will be of little use, since the HVS will not perceive them. Therefore, as in [Langer et al. 2004], we add up a set of 2D sinusoids over the three octave range $\sqrt{f_x^2 + f_y^2} \in [2,16)$, but rather than allowing for a continuum of layers in the range, we partition this three octave range into n layers of constant octave width, where $n = 1,2,3$. For example, with $n = 3$ layers, each is one octave wide, namely $\sqrt{f_x^2 + f_y^2} \in [2,4),[4,8),[8,16)$.

Each tile is given a constant velocity in n layers, with a maximum of $n = 3$.. The image velocity for the lowest frequency of each layer is assigned by the dispersion relation, with the speed for the lowest frequency layer determined by Eq. (2). For the rest of the frequencies in each layer, the denominator of Eq. (1) has a constant f_i instead of $\sqrt{f_x^2 + f_y^2}$. This constant f_i is the lowest spatial frequency of the layer. Thus within layer i all the frequencies have the same image velocity,

$$f_t = C\frac{\cos\theta\ f_x + \sin\theta\ f_y}{f_i} \quad (3)$$

where $i = 1,2,3$ in the case of $n = 3$. In the higher frequency layers, the image velocity decreases as per the dispersion relation (Eq. 1).

The motion layers in each tile are rendered using the method of [Langer et al. 2004], but now with only n layers. A random amplitude spectrum $\hat{I}(f_x, f_y)$ is generated once for each tile, such that the amplitudes have a $1/f$ distribution and the phases are random. Then, at each time step, each of the frequency components is phase-shifted. This is done in the frequency domain, by modifying the phase spectrum at each time step t as follows,

$$\phi(f_x, f_y, t+1) := \phi(f_x, f_y, t) + C(t)\ \frac{\cos\theta(t)\ f_x + \sin\theta(t)\ f_y}{f_i} . \quad (4)$$

After each phase change, a 2D Inverse Fourier Transform is computed, which yields the next frame of the sequence for that tile.

When the speed within a tile is too great, motion blur occurs. Technically speaking, this blurring occurs when the temporal frequency f_t exceeds the temporal Nyquist frequency $\frac{T}{2}$, *i.e.*

$$\left| C\frac{\cos\theta\ f_x + \sin\theta\ f_y}{f_i} \right| > \frac{T}{2} .$$

This causes high speeds in direction θ to appear as high speeds in direction $\theta + 180°$. To avoid temporal aliasing, we zero the amplitudes of any spatial frequency components that satisfy the above relation. This blurs the spatial structure of the image in the direction of the motion. The result is an effect similar to animated LIC. Locally in the image, oriented spatial structure occurs in the direction of motion, acting as a spatial cue [Geisler 1999], and there are high speed (but low spatial frequency) phase changes in the direction of the oriented structure. This motion blurring is the only case in which we use a spatial cue in our motion visualization method. Here we have no choice, since otherwise the motion would suffer from temporal aliasing.

3.4 Blending across tiles and layers

In order to reduce the visibility of the tile boundaries, neighboring tiles are overlapped and linear blending is performed in the overlapped region. If I_1 and I_2 are the intensities in the neighboring tiles and I is the corresponds to the intensity of their overlap region, then $I = w_1 I_1 + w_2 I_2$ where w_1 and w_2 are weights for the neighboring tiles. These weights depend on (x,y) and are chosen as shown in Fig. 4.

It turns out that we cannot choose weights in the naive way, namely $w_1 + w_2 = 1$, since the variance (or contrast) would be incorrect. The reason is as follows. Since two tiles I_1 and I_2 are random 1/f noise images, the means are zero, the variances $var(I_1)$ and $var(I_2)$ are the same, and I_1 and I_2 are uncorrelated. The variance of the blended image is:

$$\begin{aligned} var(I) &= var(w_1 I_1 + w_2 I_2) \\ &= var(w_1 I_1) + var(w_2 I_2) \end{aligned}$$

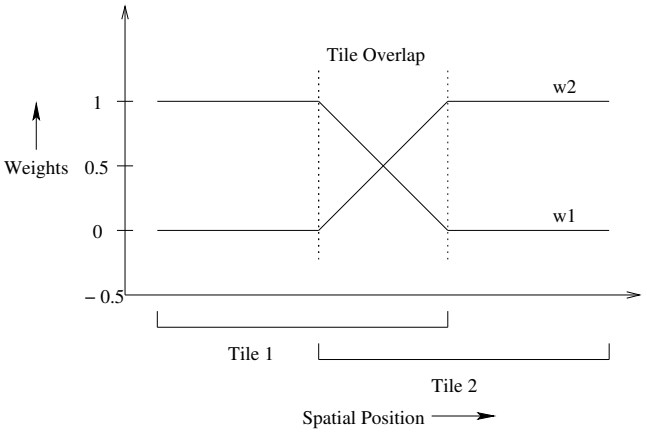

Figure 4: The weights $w1$ and $w2$ used for blending are chosen as shown above. The spatial dimension signifies either the X or the Y image axis. In fact, a non-linear weighting scheme is used (see text).

$$= \quad w_1^2\ var(I_1) + w_2^2\ var(I_2)$$
$$= \quad (w_1^2 + w_2^2)\ var(I_1)$$

If we choose w_1 and w_2 to add to 1, i.e. $w_1 + w_2 = 1$, then $var(I) = (w_1^2 + (1-w_1)^2)\ var(I_1)$ which would vary with w_1. This is not what we want. To keep the variance constant, we instead weight the image tiles I_1 and I_2 by $\frac{w_1}{\sqrt{w_1^2+w_2^2}}$ and $\frac{w_2}{\sqrt{w_1^2+w_2^2}}$, respectively, where $w_1 + w_2 = 1$. In doing so, the blended image now has the same variance as the images I_1 and I_2 themselves. A similar method was used by [Watson and Eckert 1994].

We next discuss how layers are blended. According to the above description, the different layers are linearly summed. We will compare the effect of using this method with classical compositing, which is implemented as follows. For the case of $n = 3$ layers, a frame is generated for each layer by $n = 3$ separate inverse Fourier transforms. The layers are composited to form the image frame $I(x,y,t)$ using the relation:

$$R(x,y,t) = F(x,y,t) + (1 - \frac{F(x,y,t)}{255}) * B(x,y,t) \qquad (5)$$

where R represents the result image after compositing, F represents the foreground image and B the background image. Both have intensities in the range 0 to 255. For $n = 3$, the frame generated from the lowest spatial frequencies is taken as the foreground and the frame generated from the next highest spatial frequencies is the background. The result image is then taken as the new foreground image and the third layer is taken as the background to generate the final image I.

One of the prime advantages of the linear summing was that, for each tile, we could perform the inverse fast Fourier transform (ifft) operation just once per frame, irrespective of the number of layers. However in the compositing case, we would need to perform as many ifft operations as the number of layers and in addition perform the compositing operations. The compositing method is thus a factor of n slower than the linear summing method.

4 Demonstrations

The videos discussed below are found in `APGV05-Langer.tar` which is available from the first author's website.

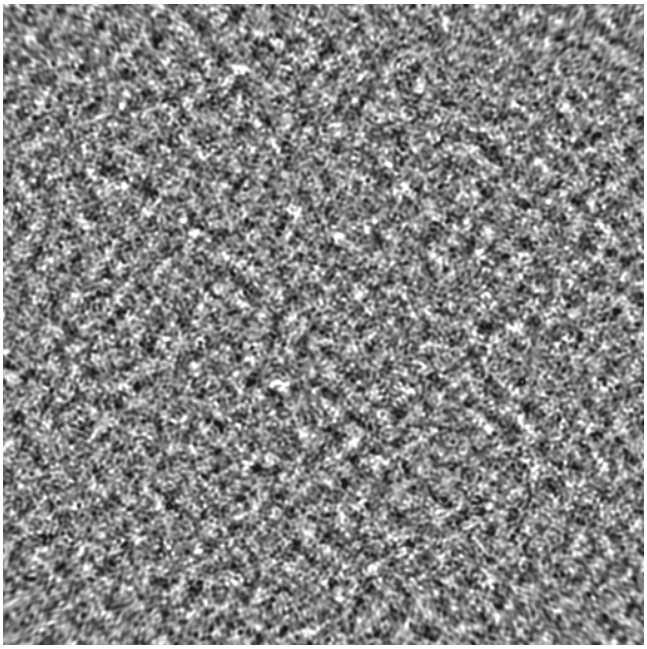

Figure 5: An example of a single image frame from one of our sequences. When the tiles are blended appropriately at their boundaries, the boundaries are invisible insofar as no purely spatial cues to the motion are present.

Each of the videos was $N \times N = 512 \times 512$ pixels, and 24 frames per second. The fluid flow videos were about 5 seconds each (128 frames) and the egomotion videos were about 2.5 seconds each (64 frames). For our videos, the tile sizes were always $M = 64$ pixels wide, and the tile overlap was either 16 or 32 pixels.

4.1 Fluid Flow

Our first example is the video `1layer-16`. It shows a single layer per tile, and a steady flow (constant 2D velocity over time). The tile overlap of 16 pixels. This is insufficient as tile boundaries are clearly visible to the HVS.

Our second example is the video `1layer-32`. Here we have increased the tile overlap to 32 pixels. Using a greater tile overlap better hides the tile boundaries by reducing the spatial gradient, i.e. the velocity transition between tiles occurs over a distance of 32 rather than 16 pixels.

In terms of performance, however, larger tile overlap implies both a greater number of tiles and more float ops for the overlapped regions, hence, leading to poor speed performance.

The video `2layer-32-Lin` uses two layers and a high tile overlap (32 rather than 16). If we compare this video with the one generated for the single layer and 32 pixel overlap, we notice that some of the tiles in the single layer video exhibit motion blur while the two layer video doesn't exhibit this. The reason is that, in the two layer video, the higher spatial frequencies have lower image velocity than they do in the single layer video. According to Eq. (3), for a given spatial frequency, higher image velocities give rise to higher temporal frequencies, which leads to aliasing. In our method, these high temporal frequencies are cut-off which yields motion blur. (See Sec. 3.3).

The greater the number of layers, the more difficult it is to perceive the individual layers. This reduced layer visibility means less visible tile boundaries. For the example `3layer-16-Lin`, the tile boundaries are just barely visible even though a tile overlap of of

only 16 pixels is used. When an overlap of 32 pixels is used as in `3layer-32-Lin`, the tile boundaries disappear completely. However, the wider overlap tends to be more expensive.

Next we consider videos with layers that are blended using classical compositing. The compositing method for three layers is about three times slower than the linear method. An example video is shown in `3layer-16-Comp`, for which 3 layers are used and the tile overlap is 16 pixels. The visual effect is slightly different, owing to the non-linearities of compositing. With `3layer-32-Comp` the overlap is 32 pixels. We again have good motion layer effect, but the wider tile overlap tends to be more expensive.

We next show an example with unsteady flow, i.e. the 2D vector field is time-dependent. The results `3layer-32-Lin-U` are similar to the case of steady flow with same parameters, except that the velocity field changes with time according to the fluid flow properties. In this example, the velocity dampens over time due to viscosity of the fluid. The force field is supplied only in the first frame.

4.2 Egomotion

The egomotion equation of Sec. 3.1.1 is used to generate the case of a translating and rotating camera. In `lhp-translation`, the viewer is translating along the direction of z-axis. The Focus of Expansion is at the center of the image. The video `lhp-rotation` shows the field produced by a camera rotating about x-axis. The field has a parabolic non-linearity, which is due to perspective. For correct perspective, the viewer should observe this field up close to the monitor. Finally, `lhp-transrot` shows the camera translating in the direction of the z-axis, while rotating about the z-axis as well. See figure 1. For each of the videos, the tile size is 64 pixels with 32 pixel overlap and uses 3 layers.

4.3 Performance Statistics

With tile width $M = 64$ and no tile overlap we are able to render 34.5 frames per second (fps). With a 16 pixel overlap, we get 20.3 fps and with 32 pixel overlap we get 11.6 fps. The performance statistics quoted here were performed on a Pentium 4 3.0 GHz with 2 GB RAM and ATI Radeon 5900 Pro 256 MB with image size of 512×512. Our program occupies only 10 MB of memory. Rather than writing every frame to disk we are directly displaying it using OpenGL.

5 Summary

In this paper, we have related what is known about the perception of layered motion and motion gradients to the applied problem of motion layer visualization. Our goal was not to map parametric constraints from the HVS to a specific parameters for visualization, but rather to present a set of relevant constraints that have appeared in the psychophysics literature and shown how these play a role in the motion visualization problem.

Two specific types of layered motion fields were addressed: a motion field generated by the observer motion in a scene with multiple depths (e.g. a densely cluttered scene) and motion field corresponding to the layered motion of the fluid.

The specific layered method we presented is an adaptation of the model presented in [Langer et al. 2004]. In that earlier method, a continuum of layers was used. We found that because the HVS is limited in how many layers it can detect, the method could be simplified by reducing the number of discrete layers. Examples demonstrate effective layered motion using purely motion cues, and do so with good performance statistics.

Acknowledgements

M. Langer was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

ADELSON, E., AND MOVSHON, J. 1982. Phenomenal coherence of moving visual patterns. *Nature 300*, 523 – 525.

ANDERSEN, G. 1989. Perception of three-dimensional structure from optic flow without locally smooth velocity. *Journal of Experimental Psychology: Human Perception and Performance 15*, 2, 363–371.

BRADDICK, O. 1997. Local and global representations of velocity: transparency, opponency and global direction perception. *Perception 26*, 995–1010.

CABRAL, B., AND LEEDOM, L. 1993. Imaging vector fields using line integral convolution. In *Proceedings of ACM SIGGRAPH 1993*, 263–272.

EBERT, D., MUSGRAVE, F., PEACHEY, D., PERLIN, K., AND WORLEY, S. 2003. *Texturing and Modeling – A Procedural Approach*. Morgan Kaufmann, San Francisco, CA 94104-3205.

FORSSELL, L., AND COHEN, S. 1995. Using line integral convolution for flow visualization: Curvilinear grids, variable-speed animation, and unsteady flows. In *IEEE Transactions on Visualization and Computer Graphics*, 133–141.

FREEMAN, W., ADELSON, E., AND HEEGER, D. 1991. Motion without movement. *ACM Computer Graphics 25*, 4, 27 – 30.

GEISLER, W. S. 1999. Motion streaks provide a spatial code for motion direction. *Nature 400*, 65–69.

INTERRANTE, V., AND GROSCH, C. 1998. Visualizing 3d flow. *IEEE Computer Graphics and Applications 18*, 4, 49–53.

LANGER, M., ZHANG, L., KLEIN, A., BHATIA, A., PEREIRA, J., AND REKHI, D. 2004. A spectral-particle hybrid method for rendering falling snow. In *Rendering Techniques 2004 Eurographics Symposium on Rendering*, 217–226.

LONGUET-HIGGINS, H., AND PRAZDNY, K. 1980. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B B-208*, 385–397.

MANDELBROT, B. 1977. *Fractals: Form, Chance, and Dimension*. Freeman, San Francisco.

MASSON, G., MESTRE, D., AND STONE, L. 1999. Speed tuning of motion s egmentation and discrimination. *Vision Research 39*, 26, 4297–4308.

MASTIN, G., WATTERBERG, P., AND MAREDA, J. 1987. Fourier synthesis of ocean scenes. *IEEE Computer Graphics and Applications 7*, 3 (March), 16 – 23.

MESTRE, D., MASSON, G., AND STONE, L. 2001. Spatial scale of motion segmentation from speed cues. *Vision Research 41*, 21, 2697–2713.

QIAN, N., ANDERSON, R. C., AND ADELSON, E. H. 1994. Transparent motion perception as detection of unbalanced motion signals i: Psychophysics. *Journal of Neuroscience 14*, 7357–7366.

QIAN, N., ANDERSON, R. C., AND ADELSON, E. H. 1994. Transparent motion perception as detection of unbalanced motion signals iii: Modelling. *Journal of Neuroscience 14*, 7381–7392.

SAKAS, G. 1993. Modeling and animating turbulent gaseous phenomena using spectral synthesis. *The Visual Computer 9*, 200 – 212.

STAM, J. 2001. A simple fluid solver based on the fft. *Journal of Graphics Tools 6*, 43–52.

STONER, G., ALBRIGHT, T., AND RANACHANDRAN, V. 1990. Transparency and coherence in human motion perception. *Nature 344*, 153 – 155.

VAN DOORN, A., AND KOENDERINK, J. 1982b. Spatial properties of the visual detectability of moving spatial white noise. *Experimental Brain Research 45*, 188–195.

VAN DOORN, A., AND KOENDERINK, J. 1982c. Visibility of movement gradients. *Biological Cybernetics 44*, 167–175.

VOSS, R. 1988. *The Science of Fractal Images*. Springer-Verlag, New York Berlin Heidelberg, ch. Fractals in nature: From characterization to simulation.

WATSON, A., AND ECKERT, M. 1994. Motion-contrast sensitivity: visibility of motion gradients of various spatial frequencies. *Journal of the Optical Society of America A-Optics Image Science and Vision 11*, 2.

WIJK, J. 1991. Spot noise-texture synthesis for data visualization. In *Computer Graphics (Proceedings of ACM SIGGRAPH 91)*, 309–318.