Motion parallax without motion compensation in 3D cluttered scenes

M. S. Langer[1], V. Chapdelaine-Couture[1], R. Mann[2] and S. Roy[3]

[1] McGill University (School of Computer Science), Montreal, Quebec, H3A 2A7 Canada

[2] University of Waterloo (School of Computer Science), Waterloo, Ontario, N2L 3G1 Canada

[3] University of Montreal (DIRO), Montreal, Quebec, H3T 1J4 Canada

Abstract

When an observer moves through a rigid 3D scene, points that are near to the observer move with a different image velocity than points that are far away. The difference between image velocity vectors is the direction of motion parallax. This direction vector points towards the observer's translation direction. Hence estimates of the direction of motion parallax are useful for estimating the observer's translation direction. Standard ways to compute the direction of motion parallax either rely on precomputed optical flow, or rely on motion compensation to remove the local image shift caused by observer rotation. Here we present a simple Fourier-based method for estimating the direction of motion parallax directly, that does not require optical flow and motion compensation. The method is real-time and performs accurately for image regions in which multiple motions are present.

1 Introduction

This paper addresses the problem of estimating directly the direction of motion parallax in local image regions. We are specifically interested in situations in which two or more layers are present in local regions. Such situations are abundant in nature, for example, in 3D cluttered scenes such as when trees or bushes are present. The main challenge with 3D cluttered scenes is that the pointwise image velocity field is difficult to estimate, since there are a large number of depth discontinuities and a large number of depth layers present. These discontinuities and layers violate the assumptions of standard optical flow methods namely that the velocity field is smooth [1], and also violate the key assumption of layered motion methods, namely that only a small number of layers are present [3, 6, 4, 21].

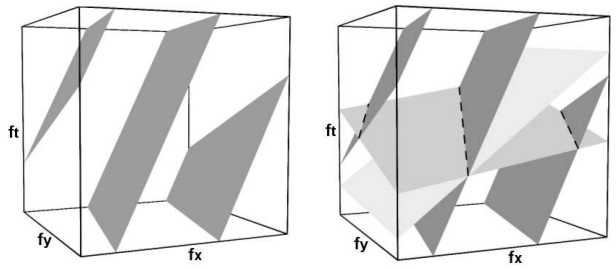
One way to address the challenge of 3D cluttered scenes is to note that, even though the standard assumptions of optical flow and layered motion are violated, there still exist strong constraints on the image velocity field. Extending the arguments of [13, 8, 17], one can show [15] that when a local image region contains visible surfaces at a wide range of depths and when this local region is not near the observer's translation direction, also known as the axis of translation (AOT), the image velocity vectors in that local region are constrained to lie near a *motion parallax line*,

$$(v_x, v_y) = (\omega_x + \alpha\ \tau_x,\ \omega_y + \alpha\ \tau_y)\ . \qquad (1)$$

Here, $\omega_x, \omega_y, \tau_x, \tau_y$ are constants that depend on the image region and on the observer's instantaneous 3D translation and rotation. The vector (τ_x, τ_y) is the *direction of motion parallax*. It points from the AOT to the center of the image region in question. The variable α depends on position x, y in the region and on the depth at (x, y).

The goal of the present paper is to introduce a simple method for estimating the direction of motion parallax (τ_x, τ_y) in local image regions. This is well-known to be an important problem. If the observer can estimate the direction of motion parallax in several local regions, it can use these to estimate the direction of heading [13, 8, 17, 15]. The emphasis in this paper is to introduce a method that estimates the direction of motion parallax, (τ_x, τ_y), without precomputed optical flow as in [15, 5]. The reason for avoiding optical flow as a prior computation is that optical flow is difficult to estimate in 3D cluttered scenes. The particular technical contribution of this paper is to show how to estimate the direction of motion parallax using a PCA method similar to [5], but without having to first perform motion compensation.

In Sec. 2, we describe the properties of motion parallax in the frequency domain. The key theoretical insight is developed in Sec. 3, where we show how to collapse the PCA method from 3D to 2D. In Sec. 4, we clarify further why PCA provides an estimate for the bowtie axis. In Sec. 5, we present experimental results.



(a) (b)

Figure 1: (a) An image translating with uniform image velocity produces a plane of energy in the 3D spatio-temporal frequency domain. Here we can see temporal aliasing caused by high velocities. (b) Motion parallax locally yields a distribution of power in the 3D frequency domain that is in the shape of a bowtie. Here, all planes intersect at a main axis, but other intersections are also present due to temporal aliasing.

2 Frequency properties of motion parallax

The method we present in this paper is based on frequency domain properties of motion parallax. The *motion plane* property [20, 9] states that an image translating with uniform image velocity (v_x, v_y) pixels per frame produces a plane of energy in the 3D spatio-temporal frequency domain:

$$v_x f_x + v_y f_y + f_t = 0\,. \tag{2}$$

Substituting Eq. (1) into Eq. (2) yields a family of planes in the frequency domain,

$$(\omega_x + \alpha\,\tau_x) f_x \;+\; (\omega_y + \alpha\,\tau_y\,) f_y + f_t \;=\; 0\,.$$

This yields a distribution of power in the 3D frequency domain that is in the shape of a *bowtie* (see Fig. 1(b)) [12]. The planes of the bowtie intersect at a common line, called the *bowtie axis*. The axis passes through the origin and is in direction $(-\tau_y,\ \tau_x,\ \sqrt{\omega_x^2 + \omega_y^2}\,)$.

When high velocities are present, temporal aliasing occurs (see Fig. 1). The usual way to deal with temporal aliasing is to use motion compensation [14, 2, 11].

Motion compensation has also been used to estimate the direction of motion parallax. In [15], a two stage frequency-based method was introduced for finding this bowtie axis for any local image region containing a range of depths. The first stage performs motion compensation by finding a best fitting motion plane in the frequency domain, and shearing the 3D power spectrum such this best fitting plane and the bowtie axis lie in the (f_x, f_y) plane. The second stage is a brute force search for the direction of the bowtie axis within the (f_x, f_y) plane [12].

An alternative frequency domain method for finding the bowtie axis is based on PCA [5]. This is the method that we improve on in the present paper. The idea of [5] is that 3D frequency components that are near the bowtie axis have more power than off-axis components, since motion planes superimpose at the bowtie axis. A PCA is performed which finds best fitting line to the power spectrum, effectively penalizing power that is off the bowtie axis. This PCA method still requires motion compensation, however. For example, if the vector (ω_x, ω_y) is large, then frequencies on or near the bowtie axis can undergo temporal aliasing and this aliased power badly corrupts the estimate of the bowtie axis parameters (τ_x, τ_y).

3 Avoiding motion compensation

To understand how motion compensation can be avoided, we first need to review how motion compensation is defined in the frequency domain, and then review how motion compensation can be expressed as a PCA problem in the frequency domain.

The usual way to perform motion compensation in the frequency domain is to find the best fitting motion plane, and to shear the frequency domain such that this best fitting plane is mapping to the (f_x, f_y) plane. If we define "best fit" by the sum of weighted distances in the f_t direction, then we wish to minimize:

$$\epsilon_1(v_x, v_y) \;\equiv \sum_{f_x, f_y, f_t} |\, W_{xyt}\,|^2\;\; (f_t + v_x f_x + v_y f_y)^2 \tag{3}$$

where the range of frequencies is $f_x, f_y \in \{-\frac{N}{2}, \ldots, \frac{N}{2} - 1\,\}$, $f_t \in \{-\frac{T}{2}, \ldots, \frac{T}{2} - 1\,\}$ (these are the frequencies below the Nyquist rate) and

$$W_{xyt} \equiv W(f_x, f_y, f_t)$$

is a set of weights which depend on the 3D Fourier transform of the image $I(x, y, t)$. Typically, W_{xyt} is the 3D amplitude spectrum [19]. Later, in Sec. 4, we will let W_{xyt} be the power spectrum.

The sum of weighted squared distance $\epsilon_1(v_x, v_y)$ in Eq. (3) can be re-expressed in terms of matrices. Let $\mathbf{F}$ be the $N^2T \times 3$ matrix that lists the N^2T frequency triplets (f_x, f_y, f_t). Let $\mathbf{W}$ be an $N^2T \times N^2T$ diagonal matrix of weights $W(f_x, f_y, f_t)$. Define the 3×3 matrix $\mathbf{F}^T\mathbf{W}^2\mathbf{F}$ by

$$\begin{pmatrix} \sum f_x^2\, W_{xyt}^2 & \sum f_x\, f_y\, W_{xyt}^2 & \sum f_x\, f_t\, W_{xyt}^2 \\ \sum f_x f_y\, W_{xyt}^2 & \sum f_y^2\, W_{xyt}^2 & \sum f_y\, f_t\, W_{xyt}^2 \\ \sum f_x f_t\, W_{xyt}^2 & \sum f_t f_y\, W_{xyt}^2 & \sum f_t^2\, W_{xyt}^2 \end{pmatrix} \tag{4}$$

where each $\sum$ is a triple summation over f_x, f_y, f_t below the Nyquist limits.

For any (v_x, v_y) let $\mathbf{v} \equiv (v_x, v_y, 1)^T$ be a vector normal to the motion plane of Eq. (2). Thus we can write Eq. (3) as:

$$\epsilon_1(v_x, v_y) \equiv \mathbf{v}^T \mathbf{F}^T \mathbf{W}^2 \mathbf{F} \mathbf{v} . \quad (5)$$

The motion compensation problem then is to find the vector (v_x, v_y) that minimizes $\epsilon_1(v_x, v_y)$.

This formulation needs to be modified slightly to account for temporal aliasing, which occurs when there are high image speeds present, i.e. $|v_x^2 + v_y^2|$ is large. Temporal aliasing causes motion planes to wrap around at the Nyquist frequency, which is not accounted for by the distance function of Eq. 5 (see Fig. 1). The traditional way to perform motion compensation is to use a multiscale method [14, 2]. For example, when minimizing Eq. (3), an initial estimate of (v_x^0, v_y^0) is obtained using low spatial frequencies only. The weighting function $W(f_x, f_y, f_t)$ is then sheared iteratively in the f_t direction, with wraparound where it exceeds Nyquist limit. At the n^{th} iteration, we write this shear transformation with wrapround as:

$$W^n(f_x, f_y, f_t) \leftarrow W(f_x, f_y, (f_t + v_x^n f_x + v_y^n f_y) \bmod T) . \quad (6)$$

The minimization is then applied to the sheared spectrum $W^n(f_x, f_y, f_t)$ to obtain an incremental (v_x, v_y), and this incremental velocity is added to the currently velocity estimate,

$$(v_x^{n+1}, v_y^{n+1}) \leftarrow (v_x^n, v_y^n) + (v_x, v_y).$$

Higher spatial frequencies are included in each iteration, since temporal aliasing is less likely to occur when the motion has been partly compensated. The computation terminates when the motion is fully compensated, that is, when the incremental velocity (v_x, v_y) is sufficiently close to $(0, 0)$.

To pose the problem in terms of PCA, we now change the formulation slightly. We minimize the perpendicular distance to a motion plane, rather than the distance in the f_t direction [7, 18, 10]. This method is known as *orthogonal distance regression* [16]. For any frequency (f_x, f_y, f_t) and velocity (v_x, v_y), the perpendicular distance from (f_x, f_y, f_t) to the motion plane defined by (v_x, v_y) is:

$$| (f_x, f_y, f_t) \cdot (v_x, v_y, 1) | / \sqrt{v_x^2 + v_y^2 + 1}$$

and the sum of squared weighted perpendicular distances is

$$\epsilon_2(\mathbf{v}) \equiv \frac{1}{\mathbf{v}^T \mathbf{v}} \sum_{f_x, f_y, f_t} W_{xyt}^2 \, |f_t + v_x \, f_x + v_y \, f_y|^2 \quad (7)$$

or equivalently,

$$\epsilon_2(\mathbf{v}) = (\mathbf{v}^T \mathbf{F}^T \mathbf{W}^2 \mathbf{F} \mathbf{v}) / (\mathbf{v}^T \mathbf{v}). \quad (8)$$

If the weights $W(f_x, f_y, f_t)$ have been sheared by motion compensation such that $\epsilon_1(\mathbf{v})$ has a minimum at $(0, 0, 1)$, then one can show that $\epsilon_2(\mathbf{v})$ also has a minimum at $(0, 0, 1)$.

Since the matrix $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$ is real and symmetric, it has three orthogonal eigenvectors. Moreover, since Eq. (8) is a *Rayleigh quotient*, the eigenvectors correspond to the maxima and/or minima of $\epsilon_2(\mathbf{v})$. Since $\epsilon_2(\mathbf{v})$ has a minimum at $(v_x, v_y, 1) = (0, 0, 1)$, this vector must be an eigenvector of $\mathbf{F}^T \mathbf{W}^2 \mathbf{F}$. The remaining two eigenvectors thus lie in the (f_x, f_y) plane. It follows Eq. (4) must have the form:

$$\begin{pmatrix} \sum f_x^2 \, W_{xyt}^2 & \sum f_x \, f_y \, W_{xyt}^2 & 0 \\ \sum f_x f_y \, W_{xyt}^2 & \sum f_y^2 \, W_{xyt}^2 & 0 \\ 0 & 0 & \sum f_t^2 \, W_{xyt}^2 \end{pmatrix} \quad (9)$$

The symbol $\sum$ denotes a triple sum over f_x, f_y, f_t, below the Nyquist limit.

It follows that the (f_x, f_y) components of the remaining two eigenvectors are identical to the eigenvectors of the following 2×2 matrix, which is just the upper left part of matrix (9):

$$\begin{pmatrix} \sum f_x^2 \, W_{xyt}^2 & \sum f_x \, f_y \, W_{xyt}^2 \\ \sum f_x f_y \, W_{xyt}^2 & \sum f_y^2 \, W_{xyt}^2 \end{pmatrix} \quad (10)$$

This leads to the *main technical insight* of the paper. Although we reached the above conclusions by assuming the weighting function W_{xyt} is first motion compensated – that is, sheared in the f_t direction so that the best fitting motion plane is the (f_x, f_y) plane – the four elements of (10) are unaffected by this shearing, since the shearing only amounts to translatation within each (f_x, f_y) column in the 3D frequency domain. The sum over f_t in Eq. (10) can be treated as an inner loop, namely a sum over f_t of W_{xyt}^2. This collapses the 3D eigenvector problem into a 2D eigenvector problem. In particular, we can compute the eigenvectors of the matrix in (10) *without first performing motion compensation.* This is useful since, as we see in the next section, the eigenvector with the larger eigenvalue provides an excellent estimate of the (f_x, f_y) component of the bowtie axis. Hence this eigenvector is an excellent estimate of the direction of motion parallax.

4 Principal eigenvector, bowtie axis, and SSNP

For any Raleigh quotient, the eigenvector with largest eigenvalue is called the *principal eigenvector*. The principal eigenvector defines the "best fit line" that passes through

the origin, in that the sum of weighted orthogonal distances to this line is minimized [16]. For the 2D matrix (10), the principal eigenvector is the best fit 2D line in the 2D plane (f_x, f_y). There is one sample point for each (f_x, f_y). We wish to choose W_{xyt} such that $\sum_{f_t} W_{xyt}^2$ is relatively large for spatial frequencies (f_x, f_y) near the bowtie axis line $(-\tau_y, \tau_x)$. Such a W_{xyt} should yield a principal eigenvector of (10) that is roughly aligned with $(-\tau_y, \tau_x)$.

To choose W_{xyt}, we consider the distribution of power within constant (f_x, f_y) columns of the 3D frequency domain, with f_t varying within a column. Assuming a bowtie is present, columns that intersect the bowtie axis will have large power $|\hat{I}(f_x, f_y, f_t)|^2$ for f_t at or near the bowtie axis, but small power for f_t away from the bowtie axis. For columns that do not intersect the bowtie axis, the power will be more uniform over f_t.

With this distribution property in mind, we define W_{xyt} to be the *normalized power spectrum*

$$W_{xyt} \equiv \frac{|\hat{I}(f_x, f_y, f_t)|^2}{\sum_{f_t} |\hat{I}(f_x, f_y, f_t)|^2}. \quad (11)$$

Clearly $0 \leq W_{xyt} \leq 1$ and, for fixed (f_x, f_y),

$$\sum_{f_t} W_{xyt} = 1 .$$

We can think of W_{xyt} as the whitened power spectrum [9], so that each spatial frequency (f_x, f_y) has the same total normalized power.

With the above definition of W_{xyt} as the normalized power spectrum, we now define the *sum of squared normalized power*

$$\text{SSNP} \equiv \sum_{f_t} W_{xyt}^2 .$$

which is a function of spatial frequency (f_x, f_y). In Sec. 5, we will show several examples of SSNP for different motion parallax sequences.

We expect to find relative large values of SSNP for spatial frequencies (f_x, f_y) near the bowtie axis line $(-\tau_y, \tau_x)$. Two extreme examples illustrate why. One extreme is an (f_x, f_y) such that $|\hat{I}(f_x, f_y, f_t)|$ is non-zero for only one value of f_t, which is the ideal case that the (f_x, f_y) column intersects the bowtie axis. In this case, W_{xyt} is 1 for that f_t and is zero for all other f_t, and so SSNP $= 1$. The other extreme is an (f_x, f_y) such that $|\hat{I}(f_x, f_y, f_t)|$ has the same value for all f_t. Since there are T values of f_t, we would have $W_{xyt} = 1/T$ for all f_t and so SSNP $= \frac{T}{T^2} = \frac{1}{T}$. It is easy to see that the general case lies between these two extremes and so

$$\text{SSNP} \in [\frac{1}{T}, 1]. \quad (12)$$

As argued above, the left and right limits tend to correspond, respectively, to frequencies (f_x, f_y) far from and to near to the bowtie axis line. Thus, this choice of W_{xyt} should yield a principal eigenvector of (10) that is roughly aligned with the bowtie axis line $(-\tau_y, \tau_x)$.

5 Experiments

The experiments below show there are larger values of SSNP near the bowtie axis, and that the principal eigenvector of Eq. (10) yields an excellent estimate of the direction of motion parallax (τ_x, τ_y) (see Eq. (1)). We also show that the method performs well even in the presence of a non-zero "camera rotation" vector (ω_x, ω_y).

5.1 Synthetic motion layers

The first set of experiments uses a scene that obeys the model of Eq. (1) exactly. See Fig. 2. Each video is defined by shifting a set of tiles in a set of depth layers from frame to frame according to the model of Eq. (1) and ensuring that near tiles occlude far tiles. Each video is defined by a subset of up to five layers, $\alpha \in \{1, 2, \ldots, 5\}$. Each layer is a set of opaque 2D square tiles which are dropped in sequence at randomly chosen positions in a 2D image plane.

In addition to capturing occlusion relations between layers, the size and density of tiles in each layer is chosen according to the laws of linear perspective. (This is not necessary for the model, but it is a natural property and so we enforce it.) Thus the width of each tile in layer α is chosen to be proportional to α (and so the area of each tile is proportional to α^2), and the number of tiles in layer α is chosen to be inversely proportional to α^2.

Figure 2 shows an example of a single 64×64 pixel frame. For the figure, each tile is given a single random intensity. For the videos that we used in the experiments, each tile was given a 1/f noise pattern. We do not use 1/f noise in Fig. 2 because then tiles would be much less visible for the reader.

Videos were created for various numbers of frames T, by shifting the layers as described above. Before taking the 3D Fourier transform of a video sequence, we multiplied by a raised cosine window in all three dimensions in order to reduce boundary artifacts in the Fourier transform. Once the 3D Fourier transform of the windowed image sequence is computed, we examine the SSNP and whether there are larger values near the true bowtie axis, i.e. in direction $(-\tau_y, \tau_x)$.

Fig. 3 shows the results for one set of parameters, namely $(\tau_x, \tau_y) = (1, 1)$ and $(\omega_x, \omega_y) = (0, -3)$. Each plot shows the average SSNP as a function of (f_x, f_y). The average is taken over 100 videos of size $N \times N$ pixels and T frames, where $N = 64$ and $T \in \{2, 4, 8, 16, 32\}$. Each plot

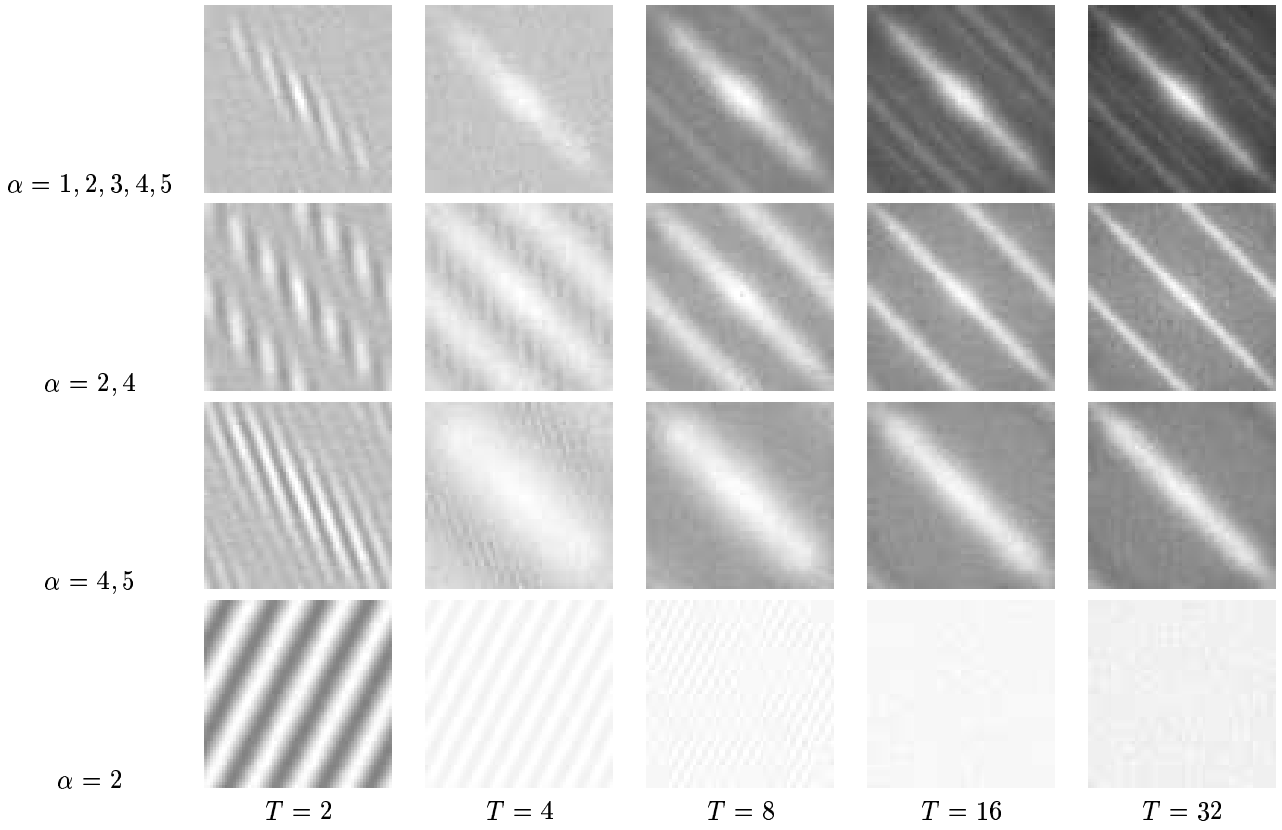


Figure 3: Example plots of SSNP as a function of (f_x, f_y), averaged over 100 videos with varying number of frames T and layers n. Left column shows the α values of the layers for each row, so that $n = 5$ for the top row, $n = 2$ for the middle two rows, and $n = 1$ for the bottom row. Each column corresponds to a fixed number of frames T. The intensities in each plot have been normalized to have the same maximum value for illustration purposes.

in Fig. 3 has been normalized to have the same maximum grey level intensity.

We observe three effects in these data. The first is the hypothesized one. It arises for larger number of frames T (say $T \geq 8$) and for more than one motion layer (Fig. 3 upper three rows). It features larger values of SSNP in the direction of the bowtie axis, $(-\tau_y, \tau_x) = (-1, 1)$, namely the line, $f_y = -f_x$, from the top left corner to the bottom right corner of each plot. (The origin is at the center). These larger values increases with T as illustrated by the increase of the contrast of the plots.

Second, we observe larger values of SSNP along lines that are parallel to the bowtie axis (these appear in the upper two rows only). These lines are the result of temporal aliasing. When a motion plane reaches the temporal Nyquist frequency, aliasing occurs. The motion plane wraps around and can then intersect other motion planes below the Nyquist frequency (see Fig. 1(b)). This intersection of an aliased plane and another motion plane yields larger values of SSNP. For a large number of motion layers (top row, $n = 5$), these secondary lines are smaller than the primary one on the bowtie axis. The reason is that the secondary lines arise from two two intersecting motion planes only (e.g. $\alpha = 2, 4$ or $\alpha = 1, 5$) whereas the primary one on the bowtie axis is due to the intersection of all $n = 5$ motion planes. Note that in the second row, where $n = 2$, the secondary lines are as bright as the primary lines. Also note that, in the third row, there are two motion planes and both are aliased at high spatial frequencies. However, they do not intersect each other and hence no secondary lines result.

A third effect occurs for small values of T, in particular, $T = 2$. Here, the temporal frequency values f_t are coarsely quantized ($f_t = 0, 1$), and a banding structure occurs in the SSNP. To see why, note that for some spatial frequency (f_x, f_y), the power must be distributed in some proportion p and $1 - p$ between $f_t = 0$ and 1, respectively. The squared normalized of each is then p^2 and $(1 - p)^2$, respectively, and so the SSNP is is $p^2 + (1 - p)^2$, which depends on p. This proportion p varies with (f_x, f_y), hence the banding in the $T = 2$ case. This third effect is unrelated to the bowtie axis and occurs even in the case of a single motion plane. (See Fig. 3 bottom row and leftmost column.) This banding

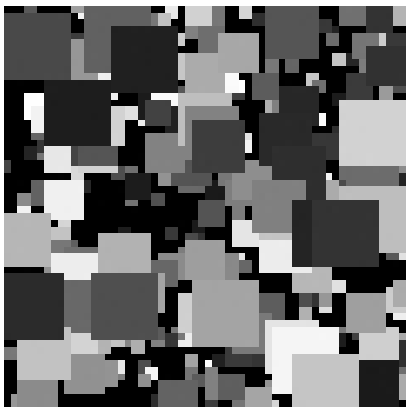


Figure 2: One 64×64 pixel frame of a synthetic video. There are five layers, $\alpha = 1, \ldots, 5$ ($n = 5$). For the actual motion sequences, each tile is given a 1/f noise.

disappears as T increases, since the more finely quantized f_t is better able to represent the single motion plane.

We have shown that the SSNP has larger values along the bowtie axis and that there are larger values at other spatial frequencies as well. The question thus arises: are the larger values near the bowtie axis sufficient for estimating the bowtie axis? To answer this question, we computed the angular error between the direction of the 2D principal eigenvector and the bowtie axis, for each of 100 videos in each condition of Fig. 3, in particular, the first three rows. We use spatial frequencies up to half the spatial Nyquist limit $|f| < \frac{N}{4} = 16$, to reduce the effects of aliasing above the Nyquist frequency. Table 1 shows the median error for each condition. The errors are large for $T = 2$, but are less than six degrees for $T \geq 4$.

	T				
	2	4	8	16	32
five layers($\alpha = 1, \ldots, 5$)	19.7	6.0	2.6	2.5	2.5
two layers ($\alpha = 2, 4$)	14.8	3.9	3.2	4.6	5.9
two layers ($\alpha = 4, 5$)	17.8	4.6	2.4	2.5	2.9

Table 1: Median of absolute angular differences (in degrees) between the bowtie axis direction and the computed principal eigenvector, for layered motion scenes illustrated in Fig. 2

5.2 Real sequence

The second experiment shows how the method performs on real image sequences, each frame having 256×256 pixels. A scene composed of plants was shot under two different camera motions. Sequence A was shot by a camera undergoing a lateral translation only. Sequence B was shot by a camera undergoing lateral translation while being gradually tilted, which produces a rotation component perpendicular to the direction of motion parallax. The true motion parallax direction for both sequences is roughly horizontal. Both sequences contain 128 frames. We use lateral motion mainly for clarity in presenting the data, since all patches have the same τ and ω vectors. However, the method does not require lateral motion [15].

Figure 4(a) shows a single frame from sequence A. Figure 4(b) and (c) show the SSNP for a column of regions (see dotted lines in Fig. 4(a)) from sequences A and B respectively with different temporal windows. As the number of frames is increased, the bowtie axis becomes much more clear. For both sequences, the bowtie axis converges to a vertical line since motion parallax is horizontal.

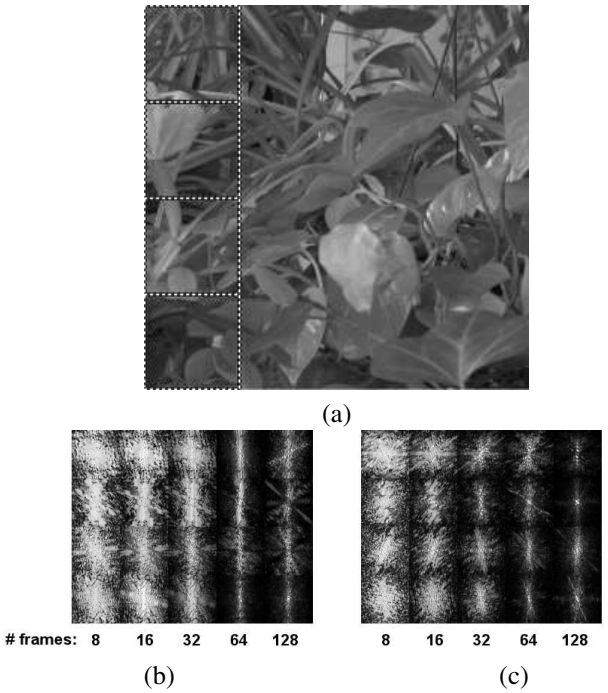
Figure 5 shows that increasing $N \times N$, the size of regions, lowers the errors. This is expected only when the observer's translation is lateral [15]. Also note that the errors are bigger for the real sequences than for the synthetic ones since the surfaces in the real sequences (the leaves) have much less texture on them. (Recall the synthetic squares had 1/f texture.) In addition, the errors are slightly larger for sequence B in which there is camera rotation. One possible reason for this is the higher image velocities which leads to more temporal aliasing.

5.3 Performance Comparison

Computation time for any region is roughly proportional to the number of frames T. For a region of width $N = 64$ pixels and for $T = 32$, the following times are typical. The 3D FFT took about 0.3 seconds. The PCA computation including computation of SSNP took an additional 0.08 seconds. This is much faster than the method of [15] which used motion compensation followed by a brute force search for the bowtie axis. That method typically used about 6 seconds total per local region (0.65 sec for motion compensation and over 5 sec for the brute force bowtie search. The new method is also must faster than the previous 3D PCA method [5]. The reason is that the 3D PCA method requires motion compensation as a first stage (0.65 seconds, as quoted above). Quoted values are on an AMD Athlon 1.6 GHz cpu, running Matlab version 6.5.

The accuracy of the new 2D PCA method is near identical to the 3D PCA method [5]. It was shown previously [5] that the 3D PCA method has similar accuracy to the method of [15]. Thus there is no tradeoff of accuracy for the decrease in computation time for the new method.

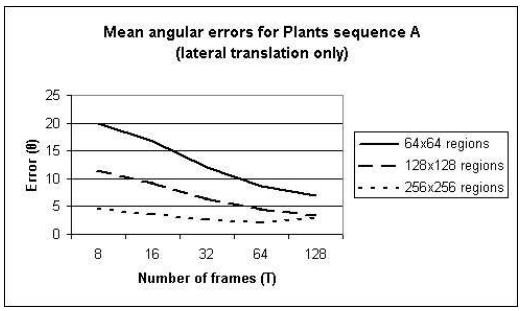
Finally, a natural question to ask is how well standard methods for computing motion parallax which are based on motion compensation perform on these sequences. We ran the Lucas-Kanade optical flow method [14], and fed the velocity vectors into the motion parallax method of [17]. The



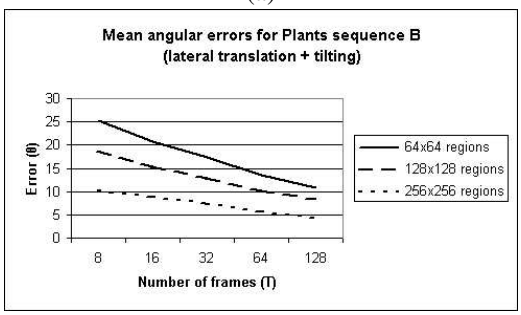
(a)

(b) (c)

Figure 4: (a) A single frame from Plants sequence A. (b) SSNP of a column of regions from Plants sequence A (lateral translation only) and using a varying number of frames $T = 8, \ldots, 128$. (c) SSNP for a column of regions from Plants sequence B (lateral translation + tilting) and using a varying number of frames.



(a)



(b)

Figure 5: (a) Mean absolute angular errors for Plants sequence A (lateral translation only). (b) Mean absolute angular errors for Plants sequence B (lateral translation + tilting).

optical flow was computed over $M \times M$ pixel regions, and motion parallax was computed using a 5×5 grid of vectors, subsampled from a $5M \times 5M$ pixel region. We expected this method to perform well when the image velocities are roughly constant over a distance M (scale of objects) but vary over a distance $5M$ (multiple objects). This is indeed what we found. (Data not shown.)

6 Conclusions

We have addressed the problem of estimating the direction of motion parallax in local image regions, for the case of an observer moving through a rigid 3D cluttered scene. As in [15], we have taken a frequency domain approach, exploiting a bowtie distribution of power that arises from the intersection of a set of motion planes. As in [5], we have used a PCA method. The key contribution of the present paper is to show how to collapse the 3D PCA problem to a 2D PCA problem, and thereby avoid motion compensation.

The experiments showed that the method performs better when there are several different velocities present in the local image region. The reason is that power is spread out over more motion planes. Since the motion planes intersect at the bowtie axis, relatively large values of squared-power are found near the bowtie axis, whereas relatively small values are found off-axis. This concentration of squared-power near the bowtie axis drives the principal eigenvector of the PCA toward the bowtie axis.

We also found that the estimates of the bowtie axis were more accurate when the number of frames was sufficiently large. For $T = 2$ or 4 frames only, we found artifactual concentrations of squared-power which arose from limited sampling of the temporal frequency f_t. For $T \geq 8$ frames, these artifacts are avoided.

The key feature of the new method is that it avoids motion compensation. This significantly speeds up the estimate of the motion parallax direction, which allows the vision system to use these estimates more quickly. Once the motion parallax directions have been estimated, the visual system can use them to immediately estimate the observer's direction of translation. This heading estimate can be done with traditional methods [17], which require as input only the directions of local motion parallax. The key contribution of the present paper is to show how to obtain these estimates of motion parallax very quickly from a sequence of images, without relying on optical flow and without having to perform motion compensation.

Acknowledgements

This research was supported by an FCAR Team Grant to M.S. Langer and S. Roy.

References

[1] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *Int. J. Comp. Vis.*, 12(1):43–77, February 1994.

[2] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *European Conference on Computer Vision*, pages 237–252, Santa Margherita Ligure, Italy, 1992.

[3] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. Patt. Anal. Mach. Int.*, 14(9):886–896, September 1992.

[4] M.J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow-fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[5] V. Chapdelaine-Couture, S. Roy, M. S. Langer, and R. Mann. Principal components analysis of optical snow. In *British Machine Vision Conference*, pages 799–808, London U.K., 2004.

[6] T. Darrell and A.P. Pentland. Cooperative robust estimation using layers of support. *IEEE Trans. Patt. Anal. Mach. Int.*, 17(5):474–487, May 1995.

[7] N.M. Grzywacz and A.L. Yuille. A model for the estimate of local image velocity by cells in the visual cortex. *Proceedings of the Royal Society of London. B*, 239:129–161, 1990.

[8] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion. *Int. J. Comp. Vis.*, 7(2):95–117, 1992.

[9] D.J. Heeger. Optical flow from spatiotemporal filters. In *First International Conference on Computer Vision*, pages 181–190, 1987.

[10] C.L. Huang and Y.T. Chen. Motion estimation method using a 3d steerable filter. *Image and Vision Computing*, 13:21–32, 1995.

[11] M. Irani, B. Rousso, and S. Peleg. Recovery of egomotion using region alignment. *IEEE Trans. Patt. Anal. Mach. Int.*, 19(3):268–272, 1997.

[12] M. S. Langer and R. Mann. Optical snow. *Int. J. Comp. Vis.*, 55(1):55–71, 2003.

[13] H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, B-208:385–397, 1980.

[14] B.D. Lucas and T. Kanade. Optical navigation by the method of differences. In *International Joint Conference on Artificial Intelligence*, pages 981–984, 1985.

[15] R. Mann and M. S. Langer. Spectral estimation of motion parallax and application to egomotion. *J. Opt. Soc. Am. A*, 22:1717–1731, 2005.

[16] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[17] J. H. Rieger and D. T. Lawton. Processing differential image motion. *J. Opt. Soc. Am. A*, 2:254–260, 1985.

[18] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conf. Comp. Vis. Patt. Rec. (CVPR)*, pages 289–295, 1991.

[19] E. P. Simoncelli. *Distributed Analysis and Representation of Visual Motion*. PhD thesis, Massachusetts Institute of Technology, 1993. Department of Electrical Engineering and Computer Science.

[20] A. Watson and A. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am*, 2(2):322–342, 1985.

[21] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Conf. Comp. Vis. Patt. Rec. (CVPR)*, pages 520–526, 1997.