

# Estimating camera motion through a 3D cluttered scene

Richard Mann  
School of Computer Science  
University of Waterloo  
Waterloo, Ontario, Canada

Michael S. Langer  
School of Computer Science  
McGill University  
Montreal, Quebec, Canada

## Abstract

*Previous methods for estimating the motion of an observer through a static scene require that image velocities can be measured. For the case of motion through a cluttered 3D scene, however, measuring optical flow is problematic because of the high density of depth discontinuities. This paper introduces a method for estimating motion through a cluttered 3D scene that does not measure velocities at individual points. Instead the method measures a distribution of velocities over local image regions. We show that motion through a cluttered scene produces a bowtie pattern in the power spectra of local image regions. We show how to estimate the parameters of the bowtie for different image regions and how to use these parameters to estimate observer motion. We demonstrate our method on synthetic and real data sequences.*

## 1. Introduction

Many computer vision methods have been developed for analyzing image motion. These methods have addressed a diverse set of natural motion categories including smooth optical flow, discontinuous optical flow across an occlusion boundary, and motion transparency. Recently we introduced a new natural motion category that is related to optical flow, occlusion and transparency but that had not been identified previously. We called the motion *optical snow*. Optical snow arises when an observer moves relative to a densely cluttered 3-D scene.

Optical snow produces dense motion parallax. A canonical example of optical snow is falling snow seen by a static observer. Although snowflakes fall vertically, the image speed of each snowflake depends inversely on its distance from the camera. Since any image region is likely to contain snowflakes at a range of depths, a range of speeds will be present. A similar example is the motion seen by an observer moving past a cluttered 3D object such as a bush. Any image region will contain leaves and branches at mul-

iple depths. But because of parallax, multiple speeds will be present in the region.

Earlier work on optical snow[8] considered only lateral motion of the observer. In the present paper, we consider general observer motion, including both a forward translation component as well as camera roll component.

There are three contributions in the paper. The first is a model of motion parallax in a cluttered 3D scene. In Sec. 2, we show that the set of image velocities in a region lie on a line in velocity space, called the *motion parallax line* for that region.

The second contribution of this paper is a method for measuring the motion parallax line for an image region. The method is based on the power spectrum of the image in the region. This method generalizes that of [8] which applied to the case of lateral observer motion only. The new method is presented in Sec. 3 and experimental results are given in Sec. 3.3.

The third contribution is to use the motion parallax lines in different regions to estimate camera motion and certain properties of scene structure. The new method is similar to that of Rieger and Lawton [13] and others, but differs in that it does not rely on pre-computed image velocities. Instead, the method relies only on the estimated motion parallax lines in regions. The method is presented in Sec. 4 along with experimental results.

## 2. Models of Motion Parallax

Longuet-Higgins and Prazdny [9] described the instantaneous retinal image velocities of an arbitrary scene point, as seen by an observer moving through a static scene. Let the observer's translation velocity be  $\mathbf{T} = (T_x, T_y, T_z)$  and angular velocity be  $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_z)$ . The scene is projected to the origin of the camera coordinate frame.  $Z(x, y)$  is the depth of the surface point visible at image position  $(x, y)$ . The image plane is at depth  $Z = f$ . The image ve-

locity at  $(x, y)$  can be written:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{T_z}{Z(x,y)} \begin{bmatrix} x - x_T \\ y - y_T \end{bmatrix} + \begin{bmatrix} xy/f & -f - x^2/f & y \\ f + y^2/f & -xy/f & -x \end{bmatrix} \begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} \quad (1)$$

The first term is the translation field. It depends on inverse depth. The second term is the rotation field. It does not depend on depth. The special image positions

$$(x_T, y_T) = \frac{f}{T_z}(T_x, T_y) \quad (x_\Omega, y_\Omega) = \frac{1}{\Omega_z}(\Omega_x, \Omega_y)$$

are called the *axis of translation* (AOT) and *axis of rotation* (AOR), respectively.

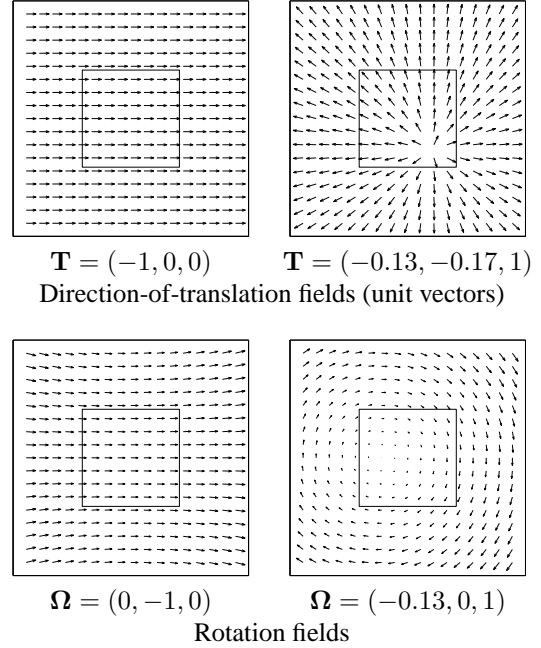
There are three standard observations about the translation and rotation fields. First, because the translation field depends on the ratio of  $T_z$  to  $Z(x, y)$ , there is a scale ambiguity. The moving observer cannot distinguish slow translation through a small scene from fast translation through a large scene. Second, each vector in the translation field points away from the AOT. Third, the speed of a translation vector depends on inverse depth. If there is a discontinuity in depth then there is a discontinuity in the magnitude of the translation field.

## 2.1. The motion parallax line

Our motion parallax model is based on further observations which concern the smoothness properties of Eq. (1). First, the *direction* of the translation field is smooth. Hence it is locally constant. The only exception is near the AOT where the direction-of-translation has a singularity. Second, the rotation field is smooth. Hence it is also locally constant. Here there are no exceptions. Although the direction of the rotation field is discontinuous at the AOR, the magnitude of the rotation field vanishes smoothly at the AOR. Hence the rotation field is smooth even at the AOR. These smoothness properties were used implicitly in [13].

Fig. 1 illustrates the two smoothness properties. Two sampled direction-of-translation fields and two sampled rotation fields are shown. Each covers a  $60^\circ$  field of view (FOV), centered at the optical axis. The central square shows a  $30^\circ$  degree FOV, which is typical for a video camera. Each plot shows a sampling of the respective field. Neighboring samples differ only slightly, with the exception of the direction-of-translation fields near the AOT.

From here on, the term *image region* refers to a  $N \times N$  subimage over which the direction-of-translation and rotation fields can be treated as locally constant. Fig. 1 suggests that for a typical video camera (say  $320 \times 240$  pixels covering a 30 degree FOV), one can safely take  $N = 64$ . These



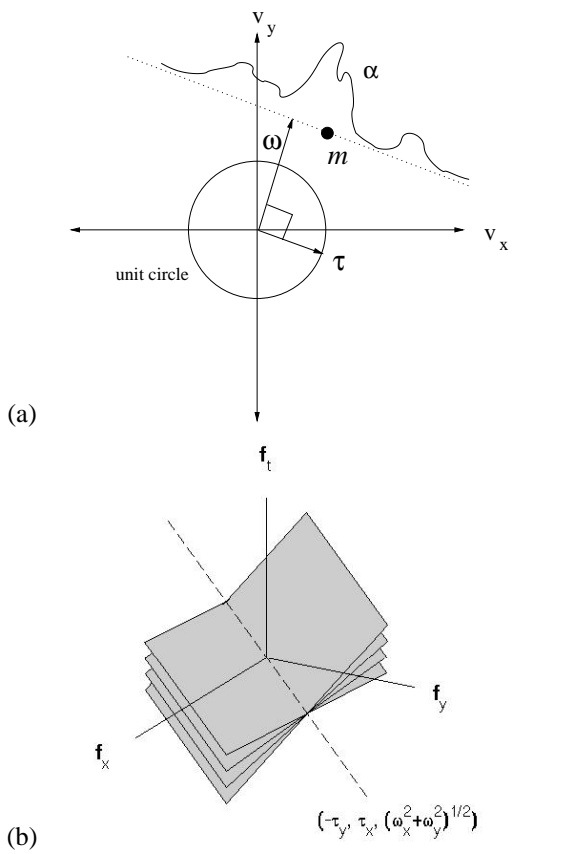
**Figure 1. Examples of sampled direction-of-translation (normalized) fields and rotation fields. The field of view (FOV) is  $60^\circ$ . Distance between samples is  $4^\circ$  at center of image and  $3^\circ$  at boundary. The interior square marks a  $30^\circ$  FOV, corresponding the FOV of a typical video camera. By inspection, the fields may be thus treated as locally constant over a region represented by a few samples, except at AOT.**

are the dimensions used in the experiments later in the paper

Because the direction-of-translation field and the rotation field are constant in an image region, it follows that *the set of image velocities in a region lies on a line in velocity space*. We refer to this line as the *motion parallax line* for that region. The line is parameterized as follows. (See Fig.2 (a).)

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \alpha \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} + \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \alpha \vec{\tau} + \vec{\omega} \quad (2)$$

$\vec{\tau}$  is the direction-of-translation vector. It is of unit length and its direction is defined up to a sign change.  $\vec{\omega}$  is the component of the rotation vector that is perpendicular to  $\vec{\tau}$ . Its length is the perpendicular distance from the origin to the motion parallax line. If the motion parallax line happens to pass through the origin, then  $\vec{\omega}$  is the zero vector. This was the case addressed in [8]. Finally,  $\alpha$  is a free parameter which indicates the  $\vec{\tau}$  component of the various im-



**Figure 2. (a) Motion parallax line (dotted).**  $\vec{\tau}$  is a unit vector in the direction-of-translation.  $\vec{\omega}$  is the component of the rotation vector that is perpendicular to  $\vec{\tau}$ .  $\alpha$  is a set of component speeds in the direction-of-translation.  $\vec{m}$  is mean velocity vector, which is used in Sec. 3. (b) Bowtie. Motion parallax in frequency domain.

age velocity vectors present in the region.

In Sec. 3, we present a method for estimating the  $\vec{\tau}$  and  $\vec{\omega}$  vectors for an image region, along with an estimate of the  $\alpha$  values in the region.

## 2.2. Motion parallax in the frequency domain

Our method for measuring the motion parallax line of an image region is based on the power spectrum. Power spectrum models have a long history in motion analysis. The simplest of these models is an image translating with velocity  $(v_x, v_y)$ . In this case, power is confined to a 2D plane in the 3D frequency domain [16]:

$$v_x f_x + v_y f_y + f_t = 0. \quad (3)$$

where  $(f_x, f_y)$  are the spatial frequencies and  $f_t$  is the temporal frequency. The  $(v_x, v_y)$  parameters of this motion plane can be estimated from the power spectrum. Computational methods have been proposed for the case of a single translation [5] as well as for a small number of superimposed translations [14, 12]. These methods find the motion plane(s) that best fit the 3D power spectrum.

In the case that velocities in a region lie on a line in velocity space as in Fig. 2a, Eq. (2) can be substituted into Eq. (3) yielding a family of planes in the 3D frequency domain:

$$(\omega_x + \alpha \tau_x) f_x + (\omega_y + \alpha \tau_y) f_y + f_t = 0. \quad (4)$$

As shown in [8], this family of motion planes intersects at a common line that passes through the origin in the frequency domain. (See Fig. 2b.) The family of planes has a *bowtie* pattern and the line of intersection of the planes is called the *axis of the bowtie*. The axis of the bowtie is in direction  $(-\tau_y, \tau_x, \sqrt{\omega_x^2 + \omega_y^2})$ .

In the next section, we show how to estimate the parameters of the bowtie for an image region, namely the axis of the bowtie and a distribution of power over the motion planes which define the bowtie. These parameters are used later in the paper to estimate the camera motion.

## 3. Estimating motion parallax

To estimate the motion parallax in an image region, we compute the 3D power spectrum and estimate the parameters of the bowtie. There are three steps:

1. Shift the motion parallax line in  $(v_x, v_y)$  space so this line passes through the origin.
2. Estimate  $\vec{\tau}$  and  $\vec{\omega}$ .
3. Estimate a histogram of speeds  $\alpha$ .

Most of our discussion concerns Step 1, which is new. Step 1 subtracts off the mean velocity in the region, bringing the motion parallax line so that it passes through the origin. This simplifies the problem of estimating  $\tau$  in Step 2 since now we can apply the method of [8].

### 3.1. Step 1: Motion compensation

To shift the motion parallax line, we perform an operation on the power spectrum that is equivalent to *motion compensation* [10]. We compute a mean velocity vector  $(m_x, m_y)$  that minimizes the sum of squared intensity differences,

$$\sum_{(x,y,t) \in N \times N \times M} \{ I(x - m_x t, y - m_y t, t+1) - I(x, y, t) \}^2$$

where the sum is over the region which is of size  $N \times N$  pixels  $\times M$  frames. Using Parseval's theorem, this minimization can be restated in the frequency domain [15]: find the vector  $(m_x, m_y)$  that minimizes

$$\sum_{(f_x, f_y, f_t) \in N \times N \times M} f_t^2 |\hat{I}(f_x, f_y, (f_t - m_x f_x - m_y f_y) \bmod M)|^2. \quad (5)$$

This is a least squares fit of a plane in  $(f_x, f_y, f_t)$ , where the error is measured in the  $f_t$  direction.

Motion compensation effectively shifts the velocities on the motion parallax line of Eq. 2 so that the mean is moved to the origin of  $(v_x, v_y)$  space. In the frequency domain, the power spectrum is sheared in the  $f_t$  direction until the bowtie axis lies in the  $(f_x, f_y)$  plane and the power is distributed roughly evenly above and below the  $(f_x, f_y)$  plane.

Three signal processing issues arise and we deal with them as follows:

**3.1.1. Windowing** We apply a triangular window (in space and in time) to the  $N \times N \times M$  region prior to taking the Fourier transform. For our examples, we take  $N = 64$  pixels and  $M = 64$  frames.

**3.1.2. Aliasing** Because of spatial and temporal sampling, the Fourier transform of the image region is a periodic function,  $\hat{I}(f_x \bmod N, f_y \bmod N, f_t \bmod M)$ . Any spatial or temporal power in the pre-sampled image irradiance that is above the Nyquist frequency ( $\frac{N}{2}$  in space,  $\frac{M}{2}$  in time) is aliased.

For real cameras, spatial aliasing effects are typically small because of optical blur and thus can be ignored. Temporal aliasing may still occur, however, especially when large image velocities are present. When a high image velocity is present, the motion plane of that velocity wraps around at the Nyquist frequency,  $f_t = \pm \frac{M}{2}$ . High positive speeds are aliased to high negative speeds. This corrupts the estimate of  $(m_x, m_y)$ . Thus, temporal aliasing cannot be ignored.

To address temporal aliasing, we *iteratively* estimate the  $(m_x, m_y)$  vector. In each step of the iteration, the power spectrum is sheared with wraparound at the  $f_t = \frac{M}{2}$ . The range of speeds after the shearing is related to the range of  $\alpha$ . An example of the iterative shearing is shown in Fig. 3.

**3.1.3. Occlusions** Motion parallax does not give rise to a perfect bowtie in the power spectrum, but rather to a smeared bowtie. This smearing is due to the image window as well as to occlusions [4, 2, 8], i.e. multiplication in space-time produces a convolution in the frequency domain.

In minimizing Eq. 5, we try to avoid frequencies where the smearing is most problematic. The smearing kernel is small in the frequency domain. However, near the origin, even a small kernel smears power between motion planes

of very different image velocities. This is problematic because natural images have power spectra that fall off as  $1/(f_x^2 + f_y^2)$  [3] and hence have large power near the origin. For this reason, we do not include low spatial frequencies in the summation of Eq. (5).

### 3.2. Steps 2 & 3: Estimate $\vec{\tau}$ , $\vec{\omega}$ , $\alpha$

Once the bowtie has been sheared, the bowtie axis lies in the  $(f_x, f_y)$  plane. This is precisely the situation that was addressed in [8]. Hence, Step 1 reduces the more general model of motion parallax of Eq. (2) to the specific model of parallel motion parallax in which  $\vec{\omega} = 0$ . The basic idea is to project the 3D power spectrum in various directions  $\vec{\tau}$  on the unit circle in the  $(f_x, f_y)$  plane, and find the direction in which the projected spectrum most resembles a bowtie.

To estimate  $\vec{\omega}$  for a region, we take the estimated  $\mathbf{m} = (m_x, m_y)$  and subtract the component that is in the direction of  $\vec{\tau}$ .

$$\vec{\omega} := \mathbf{m} - (\mathbf{m} \cdot \vec{\tau})\vec{\tau}.$$

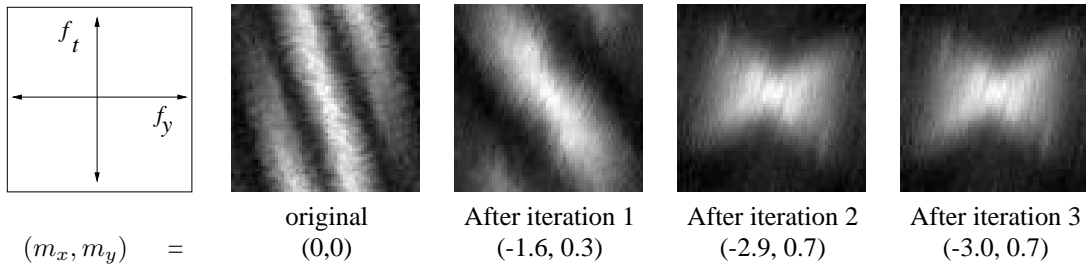
Finally we estimate the range of speeds  $\alpha$  by computing a histogram of power vs. slope in the 2D projected power spectrum, i.e. projected in direction  $\vec{\tau}$ .

### 3.3. Experiments

**3.3.1. Synthetic Image sequences** Synthetic sequences were rendered using OpenGL. The scenes consisted of squares of constant size and random orientation, positioned randomly with uniform density in a 3D view volume. Each square was texture-mapped with a checkerboard pattern. The squares had width of 0.5 units and were at depths ranging from  $Z = 5$  to 40. Each scene was viewed in perspective with a field of view of 30 degrees. Ambient lighting was used.

Two sets of image sequences were created. In the first set (Synthetic 1), the camera moved with  $\mathbf{T} = (-0.026, 0, 0.2)$  units/frame and  $\mathbf{\Omega} = (0,0,0)$ . This yielded a linear trajectory: forward and to the left by  $7.5^\circ$  off the optical axis. In the second set (Synthetic 2), the camera moved with  $\mathbf{T} = (0.026, -0.034, 0.2)$  units/frame which is  $7.5^\circ$  to the right and  $10^\circ$  down from optical axis, and  $\mathbf{\Omega} = (-0.13, 0, 0.99)$  which is a roll of 1 degree per frame. This defined a helical camera trajectory. The  $\mathbf{T}$  and  $\mathbf{\Omega}$  of Synthetic 2 correspond to the 2nd and 4th sampled fields in Fig. 1. The only other difference between Synthetic 1 and 2 sequences is that the density of squares in the latter was much greater than that of the former.

**3.3.2. Real video sequence** We also carried out experiments on several real motion sequences. Data for one typical example is reported. A video camera (Canon Optura Pi)



**Figure 3. Example of motion compensation by iterative shearing. Power spectrum is projected to  $(f_y, f_t)$  plane. Log-brightness is plotted. For this example,  $\vec{\tau}$  is near the  $x$  direction and so bowtie is visible.**

was mounted on a wheeled robot (AmigoBot). The robot drove on a circular path, moving forward to the left. The camera was pointing just over  $20^\circ$  to right of  $\mathbf{T}$ , so the AOT was just over  $20^\circ$  to the left of the image center. The camera field of view was approximately 30 degrees; hence AOT is outside FOV. The standard 4:3 image was cropped to a  $240 \times 240$  square. The image was tiled with 49 ( $7 \times 7$ ) regions, each  $N \times N = 64 \times 64$  pixels. The camera captured 30 frames per second, non-interlaced. Approximately two seconds of video were used ( $M = 64$ ).

**3.3.3. Results** Data are shown in Fig. 4. Row 4 shows the absolute errors in degrees in the  $\vec{\tau}$  estimates. Error is defined by the difference in angle between the estimated  $\vec{\tau}$  and the correct  $\vec{\tau}$  as defined by Eq. (1) where  $(x, y)$  is the central pixel in the region. For the synthetic sequences, the errors are larger near the AOT. This is expected since the model of Eq. (2) breaks down at the AOT. Errors will also be larger in any image region in which there is an insufficient image contrast or an insufficient range of depths, such as the regions in the lower left regions of the real video.

Row 5 shows the range of  $\alpha$ . Regardless of the estimate of  $\vec{\tau}$ , the range of  $\alpha$  is expected to be small near AOT. The reason is that the translation field vanishes at AOT and hence all image velocity near AOT is due to the rotation field (which is locally constant). In all examples, the range of  $\alpha$  was indeed small at AOT. In the real sequence, the range of  $\alpha$  was also small in the lower left regions, for the same reasons mentioned above.

## 4. Estimating Camera Motion

We next show how to apply the estimates of  $\vec{\tau}$  and  $\vec{\omega}$  to the problem of estimating camera motion. Our method generalizes the differential motion idea, first described in [9] and summarized as follows. At a depth boundary, two velocity vectors coexist – one for the near surface and one for the far surface. The two rotation components are identical at this common point, as are the direction-of-translation vec-

tors. However, the magnitudes of the two translation vectors differ because the magnitude depends on depth. If the observer could compute the two velocity vectors and take their difference, the difference would be in the direction of translation which points toward the AOT. The AOT may be found by intersecting lines defined by difference vectors. The basic differential motion idea was extended in [13] who used multiple velocity vectors within a region containing local depth variation, and further extended by [6] who canceled the rotation component of the velocity vectors, prior to computing AOT.

The above computational methods as well as other methods based on [9] all assume that image velocities have been pre-computed. These are very strong assumptions in the case of densely cluttered 3D scenes.

In this section, we describe a method for estimating the camera motion that relies only on the estimated  $\vec{\tau}$  and  $\vec{\omega}$  vectors in image regions. This method thus bypasses entirely the computation of point-wise image velocities.

### 4.1. Axis of translation (AOT)

Our method follows Heeger and Jepsen [6]. For any image region, let  $\mathbf{p} = [x, y, f]$  be the center of the region and let  $\mathbf{t} = [\tau_x, \tau_y, 0]$  be the (3D) vector pointing in the estimated direction  $\vec{\tau}$ . Since  $\vec{\tau}$  points toward the AOT, it follows that AOT must lie in the plane spanned by  $\mathbf{p}$  and  $\mathbf{t}$ . Thus, each region supplies a single constraint on the camera translation  $\mathbf{T}$ :

$$(\mathbf{t} \times \mathbf{p}) \cdot \mathbf{T} = 0$$

Because the estimates of  $\vec{\tau}$  for each region may be noisy, we find the AOT by computing the (unit) vector  $\mathbf{T}$  that minimizes the expression:

$$\operatorname{argmin}_{\mathbf{T}} \sum_i \|\operatorname{unit}(\mathbf{t}_i \times \mathbf{p}_i) \cdot \mathbf{T}\|^2$$

where  $i$  is the sum over regions, and  $\operatorname{unit}()$  takes the unit vector.

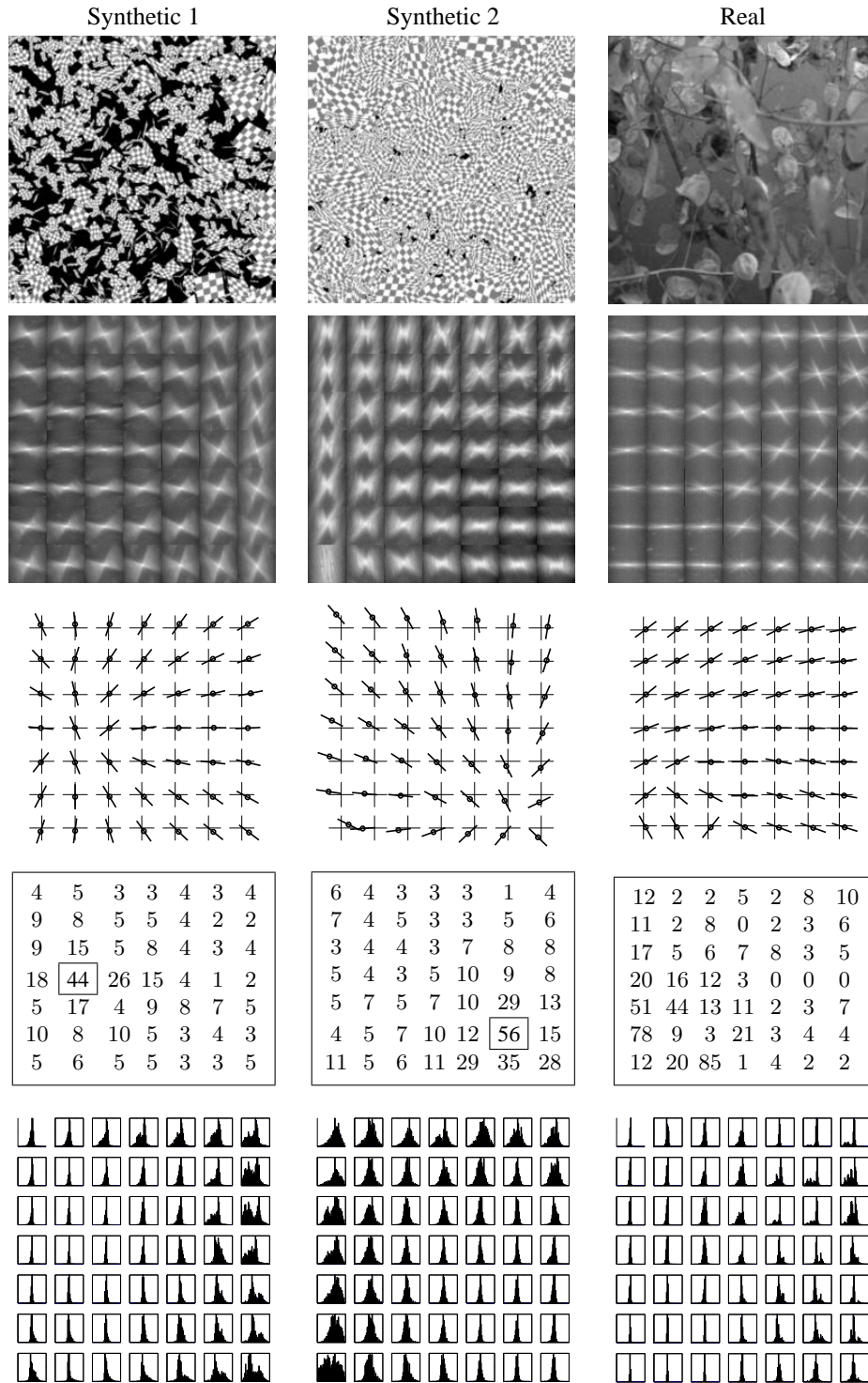


Figure 4. (row 1) Single frame from video. (row 2) Bowties found for each image region for one example of each type of sequence. (row 3) Line elements indicate estimated motion parallax line. (True  $\vec{\tau}$  points toward AOT.) Small circles are estimated  $(m_x, m_y)$ . (row 4) Numbers indicate absolute angular error in direction  $\vec{\tau}$ . For Synthetic 1 and 2, numbers are means over ten runs each. The region containing the AOT is indicated by a square. For the real video, AOT is to the left of image (hence not shown). (bottom row) Histograms of  $\alpha$ . The speed range is -4 to 4 pixels/frame.

Scene		$(T_x/T_z, T_y/T_z)$	$(\Omega_x, \Omega_y, \Omega_z)$
Synthetic 1	ground truth	(-0.13, 0)	(0, 0, 0)
	mean	(-0.123, 0.002)	(-0.001, 0.004, -0.002)
	std.dev.	(0.004, 0.005)	(0.003, 0.003, 0.005)
Synthetic 2	ground truth	(0.13, -0.17)	(0.13, 0, 0.99)
	mean	(0.11, -0.16)	(0.12, -0.013, 0.98)
	std.dev.	(0.01, 0.06)	(0.005, 0.007, 0.007)
Real	“ground truth”	(-0.41, 0)	(0, 1, 0)
	estimated	(-0.35, 0.017)	(0.017, 0.063, -0.0001)

**Table 1. Estimation of camera motion for the sequences shown in Fig. 4. For the synthetic sequences averages were computed over ten runs.**

## 4.2. Angular velocity $\Omega$

From the estimated  $\vec{\tau}$  and  $\vec{\omega}$  in each region, we can estimate  $\Omega$ . Let  $(x_i, y_i)$  denote the center point of the  $i^{th}$  region. Let  $\mathbf{B}_i$  be the rotation matrix from Eq. (1) for the  $i^{th}$  sample point,

$$\mathbf{B}_i = \begin{bmatrix} x_i y_i / f & -f - x_i^2 / f & y_i \\ f + y_i^2 / f & -x_i y_i / f & -x_i \end{bmatrix}.$$

We estimate  $\Omega$  to be a vector such that  $\mathbf{B}_i \Omega$  lies close to the estimated motion parallax line. We compute:

$$\operatorname{argmin}_{\Omega} \sum_i (\|\vec{\omega}_i\| - \operatorname{unit}(\vec{\omega}_i) \cdot \mathbf{B}_i \Omega)^2$$

where  $\operatorname{unit}(\vec{\omega}_i)$  is the unit vector perpendicular the estimate of  $\vec{\tau}_i$ .

## 4.3. Results

Table 1 presents the estimated AOT and  $\Omega$  values for ten runs of Synthetic 1 and 2, and for the real sequence. For the synthetic data the results for the AOT are near exact with mean angular errors of less than one half degree for Synthetic 1 and approximately one degree for Synthetic 2. The estimates for rotation,  $\Omega$ , are also very close. Synthetic 1 has a magnitude near 0 (no rotation) which is correct. Synthetic 2 has the correct magnitude (near 1 degree/frame) and an angular error of less than one degree.

For the real sequence, the “ground truth” of AOT and AOR was not determined using precise instrumentation, and is accurate to no better than a few degrees. Given this margin of error, the AOT estimation is acceptable.

## 5. Discussion

There are several final points worth mentioning. First, although we have developed an algorithm that is motivated by

dense motion parallax, there is nothing inherent in the algorithm which restricts its use to cluttered 3D scenes. The method only assumes an observer moving through a rigid scene. If the scene happens to contain a ground plane, transparent surfaces, or smooth surfaces, in addition to or instead of 3D clutter does not affect the validity of the motion parallax model. In future work, we will present results on scenes with other types of motion parallax such as just mentioned.

Second, with regard to motion measurement, we remind the reader that the parameters  $(\vec{\tau}, \vec{\omega}, \alpha)$  are not a complete description of the motion in an image region in that they do not specify point-wise velocities. The parameters could, however, be used to constrain methods that do estimate point-wise velocities. Current methods for estimating motion for transparency, planar deformations, occlusion, could benefit by having pre-computed estimates of  $(\vec{\tau}, \vec{\omega}, \alpha)$ .

Finally, we note that our method – like any method based on [9] – cannot resolve the many inherent ambiguities in the structure from motion problem that have been articulated over the past few years, such as the bas relief ambiguity [1, 7, 11]. In particular, understanding how these ambiguities are manifested in the estimates of  $(\vec{\tau}, \vec{\omega}, \alpha)$  is a topic of future work.

## References

- [1] G. Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:477–489, 1989.
- [2] S. Beauchemin and J. Barron. The frequency structure of 1d occluding image signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):200–206, February 2000.
- [3] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A*, 4:2379, 1987.
- [4] D. J. Fleet and K. Langley. Computational analysis of non-fourier motion. *Vision Research*, 34(22):3057–3079, 1994.

- [5] D. Heeger. Optical flow from spatiotemporal filters. In *First International Conf. on Computer Vision*, pages 181–190, 1987.
- [6] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion. *International Journal of Computer Vision*, 7(2):95–117, 1992.
- [7] J. J. Koenderink and A. J. V. Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.
- [8] M. S. Langer and R. Mann. Optical snow. *International Journal of Computer Vision*, 55(1):55–71, 2003.
- [9] H. 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.
- [10] B. Lucas and T. Kanade. Optical navigation by the method of differences. In *International Joint Conf. on Artificial Intelligence*, pages 981–984, 1985.
- [11] S. Maybank. *A Theoretical Study of Optical Flow*. PhD thesis, University of London, 1987.
- [12] P. Milanfar. Projection-based, frequency-domain estimation of superimposed translational motions. *J. Opt. Soc. Am. A*, 13(11):2151–2162, November 1996.
- [13] J. H. Rieger and D. T. Lawton. Processing differential image motion. *J. Opt. Soc. Am. A*, 2:254–260, 1985.
- [14] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 289–295, 1991.
- [15] E. P. Simoncelli. *Distributed Analysis and Representation of Visual Motion*. PhD thesis, Massachusetts Institute of Technology, 1993.
- [16] A. Watson and A. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am. A*, 2(2):322–342, 1985.