

Two-frame frequency-based estimation of local motion parallax direction in 3D cluttered scenes

V. Couture¹, M. S. Langer¹, A. Caine² and R. Mann²

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

email: {vchapd,langer}@cim.mcgill.ca

Abstract

When an observer moves in a 3D static scene, the resulting motion field depends on the depth of the visible objects and on the observer’s instantaneous translation and rotation. It is well-known that the vector difference – or motion parallax – between nearby image motion field vectors points toward the direction of heading and so computing this vector difference can help in estimating the heading direction. For 3D cluttered scenes that contain many objects at many different depths, it can be difficult to compute local image motion vectors because these scenes have many depth discontinuities which corrupt local motion estimates and thus it is unclear how to estimate local motion parallax. Recently a frequency domain method was proposed to address this problem which uses the space-time power spectrum of a sequence of images. The method requires a large number of frames, however, and assumes the observer’s motion is constant within these frames. Here we present a frequency-based method which uses two frames only and hence does not suffer from the limitations of the previously proposed method. We demonstrate the effectiveness of the new method using both synthetic and natural images.

1 Introduction

3D cluttered scenes such as foliage (see Fig. 1) are challenging for classical motion estimation methods, which assume that the velocity field is locally smooth [11, 16, 3]. Although many motion estimation methods have been proposed that address non-smooth situations containing depth discontinuities and multiple depth layers, [19, 18, 22, 23, 5, 26, 4, 1], it is unclear whether these methods can handle 3D cluttered scenes since such scenes contain a very large number of discontinuities and a very large number of depth layers.



Figure 1. Example of a cluttered scene composed of plants.

This paper is motivated by the specific problem of how a moving observer can estimate its own motion through scene. The constraints on this “heading” estimation problem are well-known. In any local region of the image, objects that lie at different depths move at different image speeds, and the difference of two neighboring motion vectors (called “motion parallax”) points toward or away from the instantaneous direction of heading. Motion parallax is thus useful for informing the observer of its instantaneous heading direction [15].

Mann and Langer [17] argued that in a 3D cluttered scene, pointwise motion field estimates are difficult to obtain since so many depth discontinuities and layers are present. Instead of estimating local parallax direction from (unreliable) pointwise motion estimates, they introduced a

method that was based on local space-time Fourier transforms of the image sequence, where the transforms are computed over image regions much larger than those used in pointwise motion estimation. Their method avoids the difficulties of estimating pointwise motions and was shown to produce very good estimates of parallax direction and heading.

There are several limitations with Mann and Langer’s method, however. First, it requires a long and dense sequence of image frames ($N \geq 16$). This introduces temporal delays in the computation which may be undesirable in a robotic vision setting. Second, it requires that the camera motion parameters (3D translation and rotation) are near constant over the sequence, which is not guaranteed in practice for long image sequences. For example, if a camera is translating along a linear trajectory and panning at the same time, then the direction of heading within the image varies continuously, which violates their model. Such a camera motion is quite common in natural vision systems e.g. during tracking.

In this paper, we address these limitations by presenting a frequency based method that requires only two frames. Unlike the previous method which used the power spectrum only, our method uses both the power spectrum and phase spectrum.

An overview of the paper is as follows. In Sec. 2 we briefly review the model of local motion parallax on which the method is based. We then describe previous frequency based methods for analysing motion, first for estimating point-wise motion vectors, second for estimating the direction of motion parallax. The novel contributions begin in Sec. 3 where we derive a frequency domain model of how the Fourier transform of an image region changes from frame-to-frame in the presence of motion parallax. We use this model to derive a method for estimating direction of motion parallax without pointwise motion estimates. Experiments and results are presented in Sec. 4.

2 Background

2.1 Motion parallax

When an observer moves in a 3D static scene, a motion field is produced which is the 2D projection of the 3D velocities of visible surfaces (relative to camera coordinates) onto the imaging surface. The motion field vector magnitudes depend inversely on the depth at the corresponding image point. The standard equation for how the motion field depends on the camera translation and rotation in 3D, on the image position, and on the depth of the visible scene depths was first derived by Longuet-Higgins and Prazdney [15] and can be found in standard computer vision textbooks [24].

It can be shown that, in a local image region, the vectors (v_x, v_y) in a motion field typically fall on a line called the *motion parallax line* [17],

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

where (τ_x, τ_y) and (ω_x, ω_y) are constant orthogonal 2D vectors that depend on the image region and on the observer’s instantaneous 3D translation and rotation [17]. In particular, the (τ_x, τ_y) vector is called the *motion parallax direction*. It depends only on the observer’s 3D translation component, namely it defines a unique line that passes through the center of the region and is the direction of heading. The scalar α depends on position (x, y) in the region and on the depth of the surface that is visible at (x, y) . The size of the image region over which this model holds depends on the observer’s 3D motion. Fig. 2 illustrates the model of Eq. 1. See Appendix of [17] for more details.

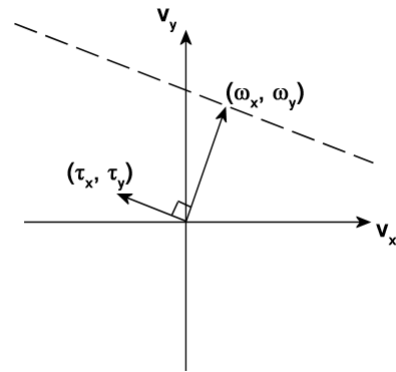


Figure 2. The set of velocities (dashed) spanned by Eq. 1.

If one could estimate the pointwise image velocities in a region, then one could use Eq. 1 to estimate the direction of motion parallax [20]. For example, let \mathbf{v}_1 and \mathbf{v}_2 be the velocities at two points in a region. Taking the vector difference and rearranging, we get

$$(\tau_x, \tau_y) = \frac{\mathbf{v}_1 - \mathbf{v}_2}{\alpha_1 - \alpha_2} \quad (2)$$

and so (τ_x, τ_y) is parallel to the vector difference.

We next briefly review a few classical methods for estimating pointwise image velocities. Since this paper concentrates on frequency based methods, we skew our review towards these methods.

2.2 Motion field estimation

One early example of frequency-based pointwise motion estimation method is the biologically inspired model

of Heeger [10]. This model uses a set of space-time Gabor filters and carries out a mean squared error fit on the outputs of these filters. Fleet and Jepson and others also used Gabor filters, but considered both the power and phase of the filter responses [21, 9, 8, 7]. Using power and phase together improves the performance significantly over a power-only method. Indeed, performance using Fleet and Jepson’s method was often better [3] than classical methods such as Lucas-Kanade [16] and Horn and Schunk [11].

Frequency-based methods are typically based on the motion plane property [25] which 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:

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

where f_x, f_y are the spatial frequency variables and f_t is temporal frequency. The intuition behind this motion plane equation is that when an image translates with this velocity, each spatial frequency component of an image translates with this velocity as well and this yields a linear relationship between the temporal frequency f_t at a point (x, y) , and the spatial frequencies. Heeger’s method for estimating motion essentially finds the plane that best fits the power spectrum of the filtered image sequence.

Heeger’s method relies on an image sequence with many frames so that there is sufficient resolution of temporal frequency. For a frequency based method to use only two frames, phase information must be used [21, 9, 8, 7]. Phase-based methods are based on the Fourier shift theorem, namely shifting an image $(\Delta x, \Delta y)$ produces a change in phase $\Delta\phi(f_x, f_y)$ that is proportional to the spatial frequency and spatial shift [13],

$$\Delta\phi(f_x, f_y) = \frac{2\pi}{N}(f_x, f_y) \cdot (\Delta x, \Delta y)$$

where N is the width of the image region over which the Fourier transform is computed. The phase-based methods cited above use Gabor-filtered images, such that the (f_x, f_y) could be considered as roughly constant, namely the center frequency of the Gabor. This allows for the estimation of $(\Delta x, \Delta y)$ from the change in phase $\Delta\phi$. (Only the component of $(\Delta x, \Delta y)$ that is parallel to the center frequency of the Gabor can be estimated using this equation. This is the “normal velocity.” A typical example is stereo vision where the filter is oriented vertically so that $f_y = 0$ and one seeks to estimate the horizontal motion or disparity Δx .)

Fleet and Jepson [7] showed that, although the phase-based estimates are not reliable everywhere, unreliable estimates could be identified and rejected. For example, when the power of Gabor filter response is very small, the phase change from frame to frame is very sensitive to noise and

the estimate can be rejected. (A similar rejection of unreliable estimates can be carried out with other motion estimation methods, e.g. Lucas-Kanade rejects motion estimates when the contrast is too low or when the local image gradients are dominated by a single direction only [16, 3].)

The key limitation in the above methods for estimating local image motion is that they assume the velocity field is locally constant. While many other methods have been developed that allow the velocity field to be more general e.g. affine or two layered motions, these methods typically do not attempt to deal with much more extreme situation of very cluttered 3D scenes (see Fig. 1), in which there are a very large number of depth discontinuities and layers. Thus it is unclear how well such methods can estimate pointwise velocities in 3D cluttered scenes, and whether the velocities that one computes are sufficiently reliable to estimate local motion parallax (recall Eq. 2). This observation motivated Mann and Langer to develop an alternative frequency-based method for estimating local motion parallax direction which does not require local velocity estimates. We next briefly describe this method.

2.3 Power spectrum method for estimating local parallax direction

Motion parallax occurs when multiple velocities are present and these lie along a motion parallax line (recall Eq. 1). In the frequency domain, multiple velocities translates into multiple motion planes, namely substituting Eq. (1) into Eq. (3) yields a family of planes [14],

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

These planes intersect at a common line that passes through the origin and is in direction $(-\tau_y, \tau_x, \sqrt{\omega_x^2 + \omega_y^2})$ in the 3D frequency domain. The (f_x, f_y) component of this line is $(-\tau_y, \tau_x)$ which is perpendicular to the direction of motion parallax (τ_x, τ_y) . Hence, estimating the direction of this line amounts to estimating the direction of motion parallax.

Mann and Langer introduced a two stage frequency-based method for finding this bowtie axis for a local image region containing a range of depths [17]. The first stage uses a standard motion compensation method to remove the estimated mean velocity in the region [16]. Since all the velocities in the region lie on the motion parallax line, the mean velocity lies along the motion parallax line as well. Hence, the motion plane corresponding to this mean velocity also contains the motion parallax line and thus, motion compensation changes the direction of the motion parallax line to $(-\tau_y, \tau_x, 0)$. The second stage of the method finds this bowtie axis using a brute force template match.

Although the method performs well on 3D cluttered scenes, there are limitations. The method requires many

image frames ($N \geq 16$) in order to achieve sufficient resolution in the temporal frequency direction. In particular, for two frames only, the method is not even well defined. Second, it requires that the camera translation and rotation parameters be constant over this long sequence of image frames. This need not be the case, in general, and indeed it is not even the case in a typical situation such as tracking.

These limitations motivated us to develop a two-frame method by using phase information in addition to power. This is exactly analogous to how frequency based methods in estimating local image velocities first used the power spectrum only [10] and then used both power and phase [21, 9, 8]. (Recall Sec. 2.2.)

3 New method

The general idea is as follows. As in [12, 17], the new method first performs *motion compensation* to get a residual flow. The motion parallax direction is then estimated from this residual flow using the new two-frame frequency method which uses phase information.

In this section, we first review the motion compensation step. We then introduce a frequency domain model which shows explicitly where the information about parallax lies in the frequency domain. The main difference between the new model and the previous frequency domain model is that the new model is expressed in the 2D spatial frequency domain (f_x, f_y) only, whereas the model of [17] is expressed in the 3D spatio-temporal frequency domain. After introducing the model, we explain how to estimate the motion parallax direction (τ_x, τ_y) in local image regions.

3.1 Motion compensation

A local region is first motion compensated to bring its mean velocity to zero. This removes all effects of camera rotation as well as the mean of the translation component. For an $N \times N$ region in frame t , we find a corresponding $N \times N$ region in frame $t + 1$ such that the average velocity from frame t to frame $t + 1$ is minimised. We then shift the region by this average velocity.

We estimate this average velocity using the Lucas-Kanade algorithm¹ with spatial derivatives computed using a $(-0.5, 0, 0.5)$ filter and temporal derivatives computed using the filter $(-1, 1)$. Prior to computing the spatial derivative, we blur spatially by convolving each frame with a radially symmetric Gaussian filter of 1.5 pixel deviation. No blurring is applied temporally because we are using two-frames only.

We will assume from now on that motion compensation has been pre-computed, and focus our attention on what is done after motion compensation.

¹Here we use only a simple version of Lucas-Kanade. For a discussion of more general versions of Lucas-Kanade, see [2].

3.2 Two-frame frequency model of motion parallax

We model the image intensities in a region of frame $t + 1$ (which has already been motion compensated) as a set of local shifts of the same region in intensities of frame t ,

$$I(x, y, t + 1) = I(x - \alpha\tau_x, y - \alpha\tau_y, t). \quad (4)$$

where α depends on (x, y) . The velocities $(\alpha\tau_x, \alpha\tau_y)$ stand for the residual velocity, after motion compensation.² We rewrite the above equation as a sum of layers having different α 's,

$$I(x, y, t + 1) = \sum_{\alpha} I_{\alpha}(x - \alpha\tau_x, y - \alpha\tau_y, t) \quad (5)$$

where I_{α} is an image ‘‘layer’’ whose pixels are moving with a velocity that corresponds to a particular α value in Eq. (5).

Taking the 2D Fourier transform over variables x, y with t fixed, we get:

$$\hat{I}(f_x, f_y, t + 1) = \sum_{\alpha} \hat{I}_{\alpha}(f_x, f_y, t) e^{i \frac{2\pi}{N} \alpha (\tau_x, \tau_y) \cdot (f_x, f_y)} \quad (6)$$

The exponent represents the phase change. Note the units. Since (τ_x, τ_y) is in pixels per frame, we need to divide by the number of pixels per image region width N to convert velocity to ‘‘image region widths’’ per frame. f_x and f_y are in cycles per image region. For example, if $\vec{v} = (1, 0)$ pixel per frame and $(f_x, f_y) = (1, 0)$, then the phase changes by $\frac{2\pi}{N}$ radians per frame i.e. from t to $t + 1$.

From frame t to the motion compensated frame $t + 1$, the Fourier transform changes as follows:

$$\Delta \hat{I}(f_x, f_y, t) = \hat{I}(f_x, f_y, t + 1) - \hat{I}(f_x, f_y, t). \quad (7)$$

Applying Eq. (6) gives:

$$\Delta \hat{I}(f_x, f_y, t) = \sum_{\alpha} \hat{I}_{\alpha}(f_x, f_y, t) (e^{i \frac{2\pi}{N} \alpha (\tau_x, \tau_y) \cdot (f_x, f_y)} - 1) \quad (8)$$

When $(\tau_x, \tau_y) \cdot (f_x, f_y)$ is small, we can take a Taylor series approximation,

$$e^{i \frac{2\pi}{N} \alpha (\tau_x, \tau_y) \cdot (f_x, f_y)} \approx 1 + i \frac{2\pi \alpha}{N} (\tau_x, \tau_y) \cdot (f_x, f_y)$$

which yields

$$\Delta \hat{I}(f_x, f_y, t) \approx (\tau_x, \tau_y) \cdot (f_x, f_y) \frac{2\pi i}{N} \sum_{\alpha} \alpha \hat{I}_{\alpha}(f_x, f_y, t)$$

The **key observation** is that $(\tau_x, \tau_y) \cdot (f_x, f_y) \approx 0$ for spatial frequencies that are orthogonal to the direction of

²This model does not explicitly account for occlusion, namely surfaces that appear in frame t or $t + 1$ but not both. Such pixels are considered as noise.

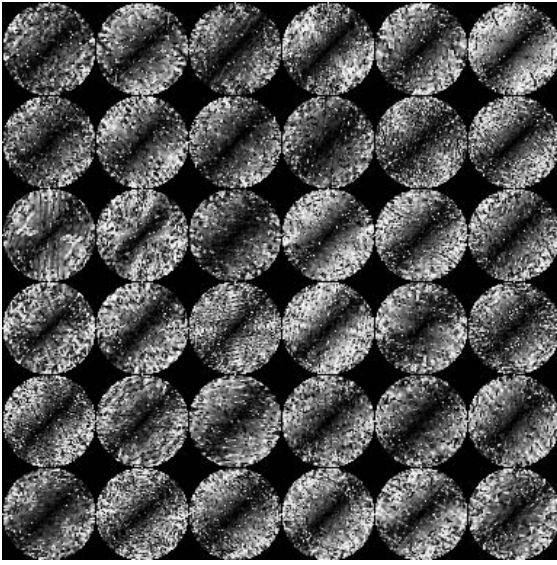


Figure 3. Phase change for a diagonal camera motion. Whiter pixels correspond to larger phase changes. The tiles are disk-shaped because frequencies above the Nyquist limit ($\frac{N}{2}$) are removed.

motion parallax. This is illustrated in Fig. 3 which shows the phase change when the motion parallax direction is diagonal, and nearly no phase change occurs near the diagonal. (See Sec. 4 for further description of the sequences.) The above observation suggests that the phase component can be used to estimate the direction of the (τ_x, τ_y) vector [6]. We next describe a new method for doing so.

3.3 Method for estimating parallax direction

Given an $N \times N$ region in frame t and the corresponding motion compensated region in frame $t+1$, namely $I(x, y, t)$ and $I(x, y, t+1)$ respectively, we apply a 2D Hanning window (raised cosine) to both regions to reduce windowing effects. The 2D Fourier transforms are then computed, giving $\hat{I}(f_x, f_y, t)$ and $\hat{I}(f_x, f_y, t+1)$.

The phase change is defined as

$$\theta(f_x, f_y, t) - \theta(f_x, f_y, t+1)$$

where

$$\theta(f_x, f_y, t) = \arctan\left(\frac{\text{Im}(\hat{I}(f_x, f_y, t))}{\text{Re}(\hat{I}(f_x, f_y, t))}\right)$$

and

$$\theta(f_x, f_y, t+1) = \arctan\left(\frac{\text{Im}(\hat{I}(f_x, f_y, t+1))}{\text{Re}(\hat{I}(f_x, f_y, t+1))}\right).$$

As we argued above, we expect the phase change to be small for spatial frequencies perpendicular to the direction of parallax and large in the direction of (τ_x, τ_y) . With this in mind, we perform a *principal components analysis* on the set of spatial frequencies (f_x, f_y) , such that we weight each spatial frequency by the phase change,

$$|\theta(f_x, f_y, t) - \theta(f_x, f_y, t+1)| \cdot (f_x, f_y). \quad (9)$$

We only use spatial frequencies $|(f_x, f_y)| < \frac{N}{2}$ where $N/2$ is the Nyquist frequency of the local region (see disks in Fig. 3). *The estimated direction of motion parallax (τ_x, τ_y) is the direction of the principal component with the larger eigenvalue.*

The phase changes $|\theta(f_x, f_y, t) - \theta(f_x, f_y, t+1)|$ in (9) can be unstable if either $\hat{I}(f_x, f_y, t+1)$ or $\hat{I}(f_x, f_y, t)$ (or both) is small. As a first attempt to discard unstable phase changes, we considered alternative weights that replace the change in phase by the magnitude of the change of the Fourier coefficients, namely

$$|\Delta \hat{I}(f_x, f_y, t)| \cdot (f_x, f_y). \quad (10)$$

Following similar reasoning used after Eq. 7, we note that $\Delta \hat{I}(f_x, f_y)$ tends to be small when $(\tau_x, \tau_y) \cdot (f_x, f_y) \approx 0$ and so the new weights tend to be small in this case as well.

In Sec. 4.2, we refer to the method using $|\theta(f_x, f_y, t) - \theta(f_x, f_y, t+1)|$ weights as the *phase method*, and to the method using $|\Delta \hat{I}(f_x, f_y, t)|$ weights as the *phase & amplitude method*.

4 Experiments

The new method was tested on synthetic sequences as well as a real sequence. These are described next.

4.1 Image sequences

The synthetic data we present are image sequences that we rendered in OpenGL. The scenes are composed of fronto-parallel squares uniformly distributed in space. With the camera initially located at the world origin, the squares were placed at depths ranging from 2 to 10 units, with random fronto-parallel orientations but fixed size.

Each frame has a 256×256 pixel resolution. To avoid spatial and temporal aliasing in the rendering, each sequence was rendered at three times its desired size (i.e. 768×768), then blurred with a Gaussian filter of standard $\sigma_s = 1.0$, and finally subsampled by a factor of three. A field of view of 35° was used and noise with standard deviation of 4.0 was added.

We tested two camera motions: first, a diagonal lateral translation motion with translation $T = (0.2, 0.2, 0.0)$ and

rotation $\Omega = (0.0, 0.0, 0.0)$; second, a horizontal lateral camera motion $T = (0.2, 0.0, 0.0)$ with a rotation around a diagonal image axis $\Omega = (-4.2, -4.2, 0.0)$. The true motion parallax directions are $(1, 1)$ and $(1, 0)$ respectively.

Two types of textures were used. First, constant intensity “textures” were used with values chosen independently from $[0, 256]$ for each object. In this case, the motion estimate is driven entirely by the intensity edges at the square boundaries. Second, 2D textures ($\frac{1}{7}$ noise patterns) were mapped having a 256 intensity range and all having the same mean of 128. Figure 4(a,b) shows a frame for both cases.

The real sequence was shot with a Canon XL2 video camera. We used a set of indoor plants which were shot under a purely lateral motion, so the motion parallax direction is $(1, 0)$ everywhere. The scale of the leaves corresponds roughly to the size of our synthetic squares. Figure 1 shows a single frame.

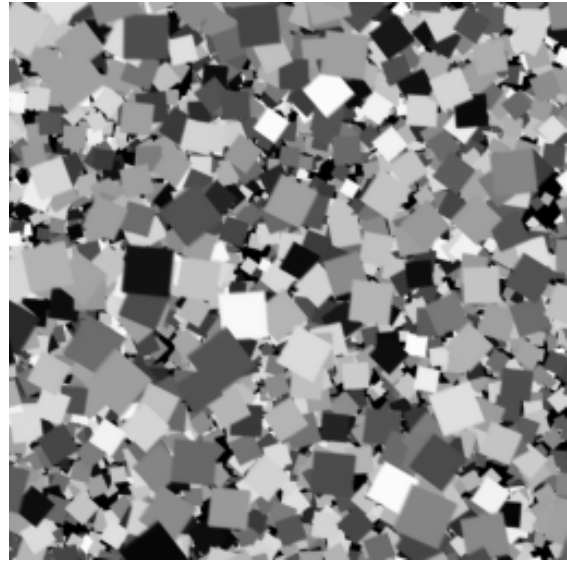
Both real and synthetic sequences are composed of 12 frames and the range of image speeds did not exceed 2 pixels per frame. Results presented in the next section have been computed on every pair of neighboring frames. The first frame of each pair is divided in a 6×6 grid of tiles, each tile corresponding to a 64×64 region over which motion parallax direction (τ_x, τ_y) is estimated. This amounts to 396 motion parallax direction estimations per sequence (36 estimations for each of the 11 pairs of frames). Here, we use $N = 64$ but in general the region size should be large enough to contain sufficiently varying scene depths, but small enough to respect the local approximation of Eq. 1.

4.2 Results

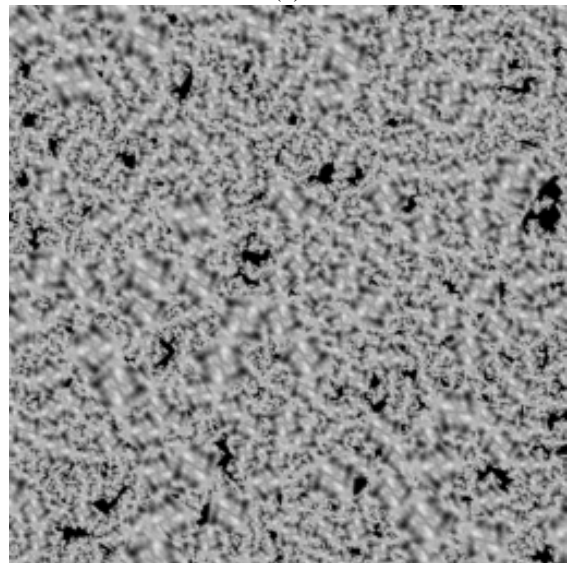
In Tables 1 and 2, we show the mean (absolute value) error of the motion parallax directions for all scenes. These errors are defined as the 2D angle between the true motion parallax direction \mathbf{d}_t and the estimated direction \mathbf{d}_e :

$$Error_{mp}(\mathbf{d}_e) = \arccos\left(\frac{\mathbf{d}_t \cdot \mathbf{d}_e}{|\mathbf{d}_t||\mathbf{d}_e|}\right).$$

In general, we observe that the *phase* method usually performs better than the *phase & amplitude* method. This was somewhat surprising since the latter was used to reduce the ill-effects of unstable phase changes. One explanation is that the amplitude spectrum depends highly on scene content (edges, gradients, contrast) and is dominated by a small set of orientations in each local region. These dominant orientations can bias the estimates.



(a)



(b)

Figure 4. A frame example from the synthetic sequences with (a) flat (0D) textures. (b) 2D textures.

SYNTHETIC SEQUENCES	
Diagonal translation $T = (0.2, 0.2, 0), \Omega = (0, 0, 0)$	
2D textures	
Phase Method	Phase & Amplitude Method
4.2	5.7
0D (flat) textures	
Phase Method	Phase & Amplitude Method
6.8	10.0
Translation and rotation $T = (0.2, 0, 0), \Omega = (-4.2, -4.2, 0)$	
2D textures	
Phase Method	Phase & Amplitude Method
9.0	14.2
0D (flat) textures	
Phase Method	Phase & Amplitude Method
15.5	11.4

Table 1. Average absolute errors (degrees) in motion parallax direction (τ_x, τ_y) for both two-frame frequency methods applied to the synthetic sequences.

REAL SEQUENCE	
Lateral translation only	
Phase Method	Phase & Amplitude Method
13.8	20.4

Table 2. Average absolute errors (degrees) in motion parallax direction (τ_x, τ_y) for both two-frame frequency methods applied to the real sequence.

5 Conclusion

A new frequency method is presented to estimate the motion parallax direction for local regions. By using both phase and power information, it extends Mann and Langer's method and can be applied to two frames only. As opposed to many classical methods for motion estimation, the new method does not assume that the velocity field is locally smooth nor that the camera motion is constant for many frames. The method performed well on the challenging 3D cluttered scenes we tested. Finally, this new method could be extended, following a scheme similar to [12] for instance, to estimate the full 3D camera motion using two frames only.

References

- [1] J. Weickert A. Bruhn and C. Schnörr. Lucas/kanade meets horn/schunck: combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.
- [2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.
- [3] 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.
- [4] M. J. Black and D. J. Fleet. Probabilistic detection and tracking of motion discontinuities. In *International Conf. on Computer Vision*, pages 551–558, Corfu, Greece, Sept. 1999.
- [5] 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.
- [6] Allan Caine. *Dense Motion Parallax Estimation from Phase*. 2005. M.Math Thesis, University of Waterloo.
- [7] D. J. Fleet. *Measurement of Image Velocity*. Kluwer Academic Press, Norwell, MA, 1992.
- [8] D.J. Fleet and A.D. Jepson. Computation of component image velocity from local phase information. *Int. J. Comp. Vis.*, 5(1):77–104, August 1990.
- [9] D.J. Fleet, A.D. Jepson, and M.R.M. Jenkin. Phase-based disparity measurement. *Computer Vision, Graphics, and Image Processing*, 53(2):198–210, March 1991.
- [10] D.J. Heeger. Optical flow from spatiotemporal filters. In *First International Conference on Computer Vision*, pages 181–190, 1987.
- [11] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
- [12] M. Irani, B. Rousso, and S. Peleg. Recovery of egomotion using region alignment. *IEEE Trans. Patt. Anal. Mach. Int. (PAMI)*, 19(3):268–272, 1997.
- [13] C.D. Kuglin and D.C. Hines. The phase correlation image alignment method. In *IEEE Conference on Cybernetics and Society*, pages 163–165, 1975.
- [14] M. S. Langer and R. Mann. Optical snow. *Int. J. Comp. Vis.*, 55(1):55–71, 2003.

- [15] 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.
- [16] B.D. Lucas and T. Kanade. Optical navigation by the method of differences. In *International Joint Conference on Artificial Intelligence*, pages 981–984, 1985.
- [17] R. Mann and M. S. Langer. Spectral estimation of motion parallax and application to egomotion. *J. Opt. Soc. Am. A*, 22:1717–1731, 2005.
- [18] D. W. Murray and B. F. Buxton. Scene segmentation from visual motion using global optimization. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-9(2)*, 9(2):220–228, March 1987.
- [19] H.H. Nagel. Constraints for the estimation of displacement vector fields from image sequences. In *In Proc. Eighth Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 945–951, Karlsruhe, Aug. 812, 1983.
- [20] J. H. Rieger and D. T. Lawton. Processing differential image motion. *J. Opt. Soc. Am. A*, 2:254–260, 1985.
- [21] T. D. Sanger. Stereo disparity computation using gabor filters. *Biological Cybernetics*, 59:404–418, 1988.
- [22] B. G. Schunck. Image flow segmentation and estimation by constraint line clustering. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 11(10):1010–1027, October 1989.
- [23] W. B. Thompson and al. Dynamic occlusion analysis in optical flow fields. *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)*, 7(4):374–383, July 1985.
- [24] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.
- [25] A. Watson and A. Ahumada. Model of human visual-motion sensing. *J. Opt. Soc. Am*, 2(2):322–342, 1985.
- [26] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Conf. Comp. Vis. Patt. Rec. (CVPR)*, pages 520–526, 1997.