Can Lucas-Kanade be used to estimate motion parallax in 3D cluttered scenes?

V. Couture and M. S. Langer

McGill University (School of Computer Science), Montreal, Quebec, H3A 2A7 Canada
email: vincent.couture@mail.mcgill.ca

Abstract

When an observer moves in a 3D static scene, the motion field depends on the depth of the visible objects and on the observer's instantaneous translation and rotation. By computing the difference between nearby motion field vectors, the observer can estimate the direction of local motion parallax and in turn the direction of heading. It has recently been argued that, in 3D cluttered scenes such as a forest, computing local image motion using classical optical flow methods is problematic since these classical methods have problems at depth discontinuities. Hence, estimating local motion parallax from optical flow should be problematic as well. In this paper we evaluate this claim. We use the classical Lucas-Kanade method to estimate optical flow and the Rieger-Lawton method to estimate the direction of motion parallax from the estimated flow. We compare the motion parallax estimates to those of the frequency based method of Mann-Langer. We find that if the Lucas-Kanade estimates are sufficiently pruned, using both an eigenvalue condition and a mean absolute error condition, then the Lucas-Kanade/Rieger-Lawton method can perform as well as or better than the frequency-based method.

1 Introduction

3D cluttered scenes such forests are characterized by the presence of a large number of discontinuities and depth layers. These scenes are challenging for classical motion estimation methods, which assume that the velocity field is locally smooth [7, 9, 3] and that a small number of depth layers is present [4]. Many methods have been invented that address discontinuities and layers, including robust approaches [5], non-linear optimization, segmentation, etc... (e.g. see references in [15, 1]). However, it is unclear whether these methods can handle 3D cluttered scenes such as in Fig. 1 since these scenes contain a large number of discontinuities and a large number of depth layers.

This paper addresses the specific problem of how a mov-

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

ing observer can estimate its own motion through a 3D cluttered 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, also known as motion parallax. Motion parallax is useful for the observer to estimate its instantaneous heading direction (translation) since the difference of two nearby motion field vectors points away from the heading direction[8].

It is not necessary to have a dense motion field to estimate either motion parallax or heading. Each local difference of velocity vectors in theory requires just two estimates, and a only a small number of motion parallax vectors over the image domain are required. (See the methods of [13] and [6], for example.) Thus, even though computing a dense motion field would be a very daunting computational problem for an observer moving in a 3D cluttered scene, it is plausible that a sparse field could be computed and that this sparse field could be enough to compute local motion

parallax and in turn the heading direction.

Mann and Langer [10, 11] argued, however, that in a 3D cluttered scene, optical flow estimates are in general unreliable since so many discontinuities and layers are present. They further conjectured that optical flow estimates cannot be used to reliably estimate motion parallax or heading. Although they proposed and studied an alternative frequency based method for solving the problem, they did not carry out experiments to test their conjecture.

Our goal in the present paper, therefore, is to examine whether the classical optical flow method of Lucas-Kanade [9] can provide reliable motion field estimates in 3D cluttered scenes and, in particular, whether these estimates can be used to estimate the local motion parallax direction. For estimating motion parallax direction, we use the simple method of Rieger and Lawton [13]. We also compare the estimates of the motion parallax direction that we obtain using the combined Lucas-Kanade/Rieger-Lawton method to those we obtained using the frequency based method of Mann-Langer [10, 11].

An overview of the paper is as follows. In Sec. 2.1 and 2.2, we review the concepts of motion field and motion parallax. In Sec. 3.1 we briefly review the algorithm of Lucas-Kanade [9] to estimate motion fields, along with two methods for pruning these estimates that are particularly important in 3D cluttered scenes. In Sec. 3.3 we review the method of Rieger-Lawton [13] for estimating the direction of local motion parallax from a given set of motion field estimates. In Sec. 4, we present results for estimating local parallax direction using Lucas-Kanade/Rieger-Lawton and compare these results to those obtained using the method of Mann-Langer. Several types of 3D cluttered scenes and camera motions are used and results are given for both synthetic and real sequences.

2 Background

2.1 The motion field

When an observer moves in a 3D static scene, a motion field is produced in the image. This field is the 2D projection of the 3D velocities of surface points (in camera coordinates) onto the imaging surface. Fig. 2 shows examples of motion fields. Fig. 2(a) shows a motion field due to a lateral camera motion. The vector magnitudes depend inversely on the depth at the corresponding image point. Fig. 2(b) shows a motion field for a camera rotation about the y-axis. This field depends on image position but not on scene depth. Fig. 2(c) shows a motion field that is due to a forward camera motion only. An exact equation for how the motion field depends on the extrinsic camera parameters, the image position, and the visible scene depths can be found in [8, 14].

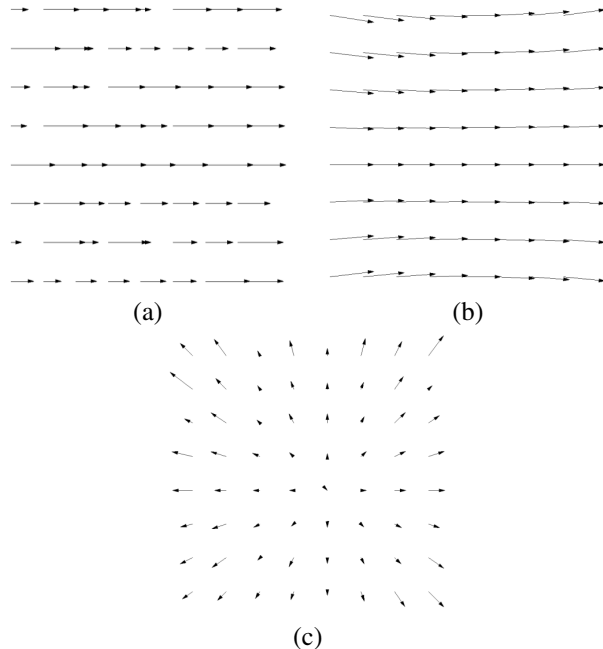Extending the arguments of [8, 13], Mann and Langer

Figure 2: Motion field due to (a) a lateral camera translation. (b) a rotating camera about the y-axis. (c) a camera translating forward.

[11] showed that, except for regions near the observer's translation direction (also known as the axis of translation (AOT)) and a few other exceptions, the motion field in a local image region can be well-approximated by vectors that fall on a *motion parallax line*:

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

Here, (τ_x, τ_y) and (ω_x, ω_y) are 2D vectors that depend on the image region and on the observer's instantaneous 3D translation and rotation, respectively. These vectors are approximately constant within the region. The vector (τ_x, τ_y) is known as the *motion parallax direction*. The scalar α depends on position x, y in the image and is inversely proportional to the depth seen at x, y. Fig. 3 illustrates the model of Eq. 1. The dotted line, parallel to vector (τ_x, τ_y), represents the set of velocities parametrized by $\alpha(x, y)$.

In Fig. 2(a), the vectors are all in the same direction and the vector lengths α are inversely proportional to depth. In Fig. 2(b), the vectors are approximately in the same direction. In Fig. 2(c), the motion field diverges away from the center of the image. At the image center, there is a singularity in the direction of the motion and the model of Eq. 1 does not apply. For further discussion of the accuracy of the model, see the Appendix in [10].
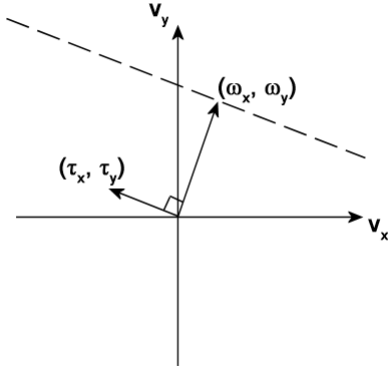
Figure 3: The set of velocities (dashed) spanned by Eq. 1.

2.2 Local motion parallax direction

Looking at Eq. 1 and taking the difference between the velocity vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ at two nearby image points, we have:

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

Using this equation, the motion parallax direction (τ_x, τ_y) can be defined as long as $\alpha_1 \neq \alpha_2$, i.e. the depths of the two corresponding neighboring regions are not equal. Eq. 2 also assumes that parameters (τ_x, τ_y) and (ω_x, ω_y) are the same for the two vectors, i.e. that Eq. 1 holds. (This is not the case near the AOT, for example.)

In the next two sections, we review two classic algorithms for estimating the local motion vectors $\mathbf{v}$ and the parallax between a small number of motion vectors that approximately satisfy Eq. 1, namely Lucas-Kanade [9] and Rieger-Lawton [13], respectively.

3 Methods used

3.1 Estimating motion field using Lucas-Kanade

Several versions of the Lucas-Kanade method have been developed since the method was invented nearly twenty years ago [2]. We will use a simple version of the method, similar to the implementation used in the Barron-Fleet-Beauchemin comparison [3]. Readers that are familiar with the method should skip on to Sec. 3.2.

Lucas-Kanade is a differential optical flow technique that estimates velocity from spatiotemporal derivatives of image intensities. We implemented a simple version that assumes local translation, i.e. for a pixel $p_i = (x, y)$ in frame t,

$$I(p_i, t) = I(p_i - \mathbf{v}t, 0)$$

where $\mathbf{v}$ is a 2D velocity $(v_x, v_y)^T$, and that intensity is conserved, $\frac{dI(p_i,t)}{dt} = 0$. This leads to the gradient constraint equation:

$$(I_x(p_i, t), I_y(p_i, t))^T \cdot \mathbf{v} + I_t(p_i, t) = 0 \qquad (3)$$

where $I_x(p_i, t)$, $I_y(p_i, t)$ and $I_t(p_i, t)$ are respectively the partial derivatives in the x,y and t directions. Derivatives are estimated as follows. First, the image sequence is low pass filtered with a spatiotemporal Gaussian filter with standard deviations of $\sigma_s = \sigma_t = 1.5$. The purpose of this filter is to remove aliasing and the effects of noise. Then, discrete derivatives are computed using the mask $(1, 0, -1)$.

Since vector $\mathbf{v}$ has two unknown components and is constrained by only one linear equation, the Lucas-Kanade method must impose another constraint, namely that the motion field is well approximated by a constant vector field within a small region. (As we will discuss later, this constraint is natural for smooth surfaces, but it is known to be problematic at depth discontinuities. For 3D cluttered scenes which contain a near-dense set of depth discontinuities, this constraint is potentially quite problematic.)

Lucas-Kanade estimates $\mathbf{v}$ using a weighted least-squares minimization of the following expression

$$argmin_v \sum_{p_i \in \Theta} W^2(p_i)[\nabla I(p_i, t)\mathbf{v} + I_t(p_i, t)]^2 \qquad (4)$$

where Θ is a small spatial region, and $W(p_i)$ is a window function that gives more weight to the center of the region. We use a 5×5 region with weights $W(p_i) = (0.0625, 0.25, 0.375, 0.25, 0.0625)$ as in [3]. In matrix form, the solution to Eq. 4 is given by:

$$\mathbf{A}^T\mathbf{W}^2\mathbf{A}\mathbf{v} = \mathbf{A}^T\mathbf{W}^2 b \qquad (5)$$

where, for n pixels $p_i \in \Theta$ at a single time t,

$$\begin{aligned} \mathbf{A} &= [\nabla I(p_1), ..., \nabla I(p_n)]^T \\ \mathbf{W} &= diag[W(p_1), ..., W(p_n)] \\ b &= -(I_t(p_1), ..., I_t(p_n))^T \end{aligned}$$

A solution to Eq. 5 can be given by

$$\mathbf{v} = (\mathbf{A}^T\mathbf{W}^2\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}^2 b.$$

3.2 Pruning unreliable velocity estimates

Although Lucas-Kanade in principle estimates a velocity vector at each image position, this estimate is known to be unreliable if the matrix $\mathbf{A}^T\mathbf{W}^2\mathbf{A}$ is near singular. For this reason, one often examines the eigenvalues of this matrix to decide whether to keep the velocity estimate or not, and discards the velocity estimate based on some threshold τ. The
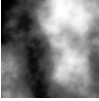
Cases without depth discontinuities	
Flat texture	**Small λ_1** Velocity cannot be estimated
2D texture	**Big λ_2** Reliable estimate of velocity
Cases involving depth discontinuities	
two flat textures	**Small λ_2** Unreliable 2D velocity (but reliable normal velocity i.e. aperture problem)
two 2D textures	**Big λ_2** Reliable 2D velocity (However, it is only *mean* velocity of the objects. See text.)

Table 1: Reliability of velocity estimates based on eigenvalue condition of Lucas-Kanade.

way this works is roughly as follows. The eigenvalues depend on the magnitudes of the spatial gradients and on their range of orientations within the local region [3]. If the local image region has near constant intensity, then the intensity gradients are small and so are both eigenvalues. In this case, one discards the estimate entirely if the larger eigenvalue λ_1 is too small ($\lambda_1 < \tau$). If only the second eigenvalue is too small ($\lambda_1 \geq \tau$ but $\lambda_2 < \tau$), then the image gradients are all in one direction and the velocity estimate suffers from the aperture problem. In this case one the normal velocity is reliably estimated. Finally, if both eigenvalues are sufficiently large ($\lambda_1 \geq \lambda_2 \geq \tau$) then the 2D velocity estimate is considered reliable and is accepted.

Table 1 summarizes the expected λ values for various cases. Of particular interest is last row of Table 1. Such a region would yield an estimate that is the roughly average of the two velocities present[1]. Note that the average of two or more velocity vectors on a motion parallax line also lies on the line.

To emphase this point, consider the more extreme yet still quite common example shown in Fig. 4. We have a local image region consisting of a T-junction formed by

---

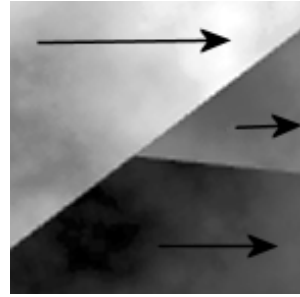[1] subject to noise caused by pixels appearing or disappearing at the occlusion boundary[12]

Figure 4: A local image region that consists of a T-junction formed by three surfaces that lie at different depths.

three surfaces that lie at different depths. Suppose this region is small e.g. 7×7 pixels. The three surfaces will have three different image velocities since velocity depends inversely on depth. Because the T-junction defines two different edge orientations, the matrix $\mathbf{A}^T\mathbf{W}^2\mathbf{A}$ would have two large eigenvalues and so the estimate would not be pruned based on the above eigenvalue conditions. *This would be true even if there were no texture on of the surfaces.* Again, an alternative method would be needed to prune away the velocity estimate for this region.

With the above examples in mind, we apply a second pruning condition to remove unreliable velocity estimates. We consider the mean absolute error (MAE) of moving a region Θ by its estimated velocity. We discard an estimate if the following expression is bigger than a threshold τ_2:

$$\sum_{p_i \in \Theta} W^2(p_i) \, |I(p_i + \mathbf{v}, t + 1) - I(p_i, t)| \qquad (6)$$

A similar pruning condition was used in [1].

We will see in Sec. 4 that the above two pruning conditions are complementary in that they nearly remove disjoint sets of motion field estimates. In short, the eigenvalue condition removes estimates for textureless image regions and for image regions containing a single high contrast edge, whereas the mean absolute error condition removes estimates from image regions typically containing multiple surfaces at different depths. What remains after the pruning – and what the visual system must rely on for estimating motion parallax – are velocity estimates in local image regions that contain one or more surfaces *near a single depth* and that contain image gradients at multiple orientations.

3.3 Estimating direction of parallax using Rieger-Lawton

To estimate the direction of motion parallax from the Lucas-Kanade motion field estimates, we follow an algorithm similar to what Rieger-Lawton proposed in [13]. Windows of 64×64 estimated velocities are considered. Only some of

the velocities in each window are used, namely the ones that are not pruned by the conditions above. The width 64 was chosen somewhat arbitrarily. In general, the window size should be large enough to contain sufficiently varying samples, but small enough to respect the local approximation of Eq. 1. The number of frames required to compute motion parallax depends on the number of frames required by the Lucas-Kanade algorithm [3] which in our case is 11 (i.e. 9 for the low pass filter plus 2 for the discrete derivative).

According to Eq. 1, all the estimated velocities in a local region (64×64) lie along a line (see Fig. 3). In particular, the *mean* of these velocities lies along the same line. It follows that the difference between each vector and this velocity mean is in the direction of local motion parallax, namely (τ_x, τ_y). The motion parallax direction can thus be estimated by computing a *residual* motion vector which is the difference between each (non-pruned) motion estimate and the mean of these estimates. (This is slightly different from [13] which removes the central velocity estimate instead of the average.) The motion parallax direction is then estimated to be the best least squares fit vector to this residual motion field. This fit also defines two eigenvalues and we can discard a motion parallax direction estimate if the ratio of the larger to smaller eigenvalue is too low. We would expect to discard motion parallax estimates in regions near the AOT, for example.

4 Experiments

We now present results for the combined Lucas-Kanade/Rieger-Lawton method on several real and synthetic sequences. We compare the results with the Mann-Langer frequency based method [10, 11], which is also based on Eq. (1).

4.1 Image sequences

The data we present are based on four 12 frame image sequences that we rendered [2] 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 2.0 to 10.0 units, with random fronto-parallel orientations but fixed size. The object scale is such that 90 % of the Lucas-Kanade regions involve at least one invalid gradient estimate ∇I, namely a gradient that samples from two different objects.

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.

---

[2]Recall that Lucas-Kanade blurs in space and time, and uses a finite difference. Hence multiple frames are needed.

$768 \times 758 \times 36$), then blurred with a Gaussian filter of standard $\sigma_s = \sigma_t = 1.0$, and finally subsampled by a factor of three. To model accurately the real sequences (see below) which were shot with a Canon XL2 video camera, a field of view is 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 about 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 by entirely by the intensity edges at the square boundaries. Second, 2D textures ($\frac{1}{f}$ noise patterns) were mapped having a 256 intensity range and all having the same mean of 128. These textures were designed to evaluate Lucas-Kanade, without corruption from aperture problems at surface edges. Figure 5(a,b) shows a frame for both cases.

For the real sequences, we used a set of indoor plants which were shot under two different camera motions, namely purely lateral motion with/and without camera tilting. The scale of the leaves corresponds roughly to the size of our synthetic squares. Again, each sequence has 12 frames, with each frame having 256×256 pixels. Motion estimates were made for the central frame only i.e. frame 6. The true motion parallax direction for both sequences is roughly horizontal. Figure 1 shows a single frame.

Examples of the sequences can be found at

`www.cim.mcgill.ca/~vchapd/CRV2007.zip`

4.2 Results

In Table 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 $\mathbf{d_e}$:

$$Error_{mp}(\mathbf{d_e}) = \arccos(\frac{\mathbf{d_t} \cdot \mathbf{d_e}}{|\mathbf{d_t}||\mathbf{d_e}|}). \qquad (7)$$

In computing this mean error, we only used estimates of motion parallax for which the the Rieger-Lawton method found a sufficient orientation in the fitting the residual motion vectors (namely, the ratio of the two eigenvalues was at least 2 - see Sec. 3.3).

Fig. 6 shows the effect of removing estimates according to the two conditions described in Sec. 3.2 for the synthetic sequences with flat texture (0D). Each subplot (a-f)
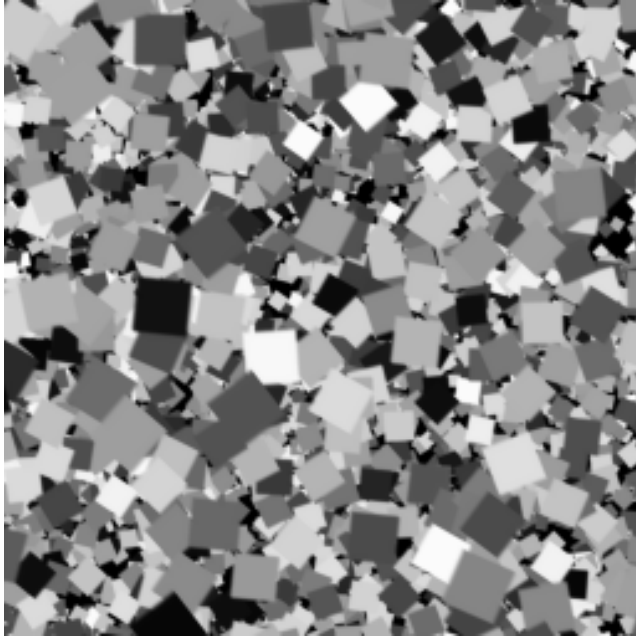
(a)
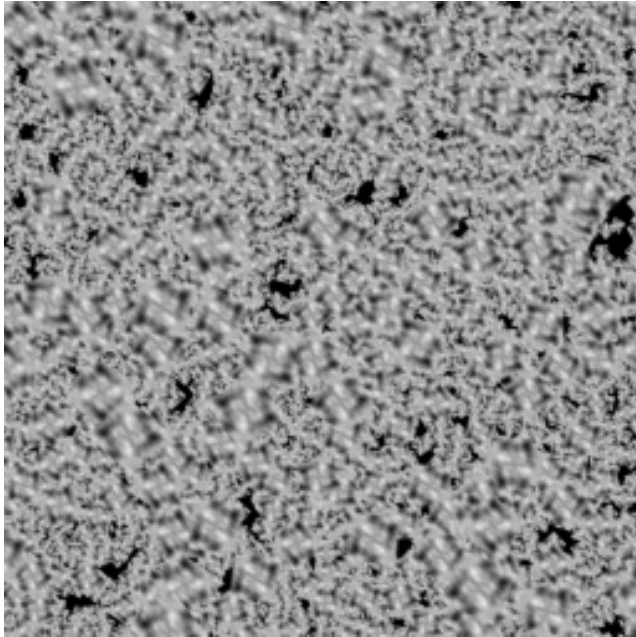
(b)

Figure 5: A frame example from the synthetic sequences with (a) flat (0D) textures. (b) 2D textures.

shows a 6×6 grid of tiles, each tile corresponding to a set of 64×64 regions over which image velocities (v_x, v_y) and then motion parallax direction (τ_x, τ_y) are estimated. Fig. 6(a,b) show the ground truth of the image velocity vectors (v_x, v_y). Each tile spans speeds of ± 2.5 pixels per frame in each v_x, v_y dimension. The data clearly lie along motion parallax lines (Eq. 1). Fig. 6(c,d) show the corresponding velocity vectors that are estimated by Lucas-Kanade. Only estimates within the speed range are plotted. Fig. 6(e,f) show the estimates that remain after the worst 40% are pruned for both pruning conditions (see table below). Pruning appears to remove outliers and reduces the spread of the velocity distribution so it better conforms to the ground truth, as intended.

Table 2 shows the mean absolute errors for estimating the motion parallax directions. For each type of camera motion and scene texture, a 3×3 table is shown which represents the error for various combinations of the two pruning conditions for the velocity estimates. (Recall Sec. 3.2.) For both pruning conditions, we removed either 0,20, or 40 percent of the estimates. (Removing 20% for both conditions removes $34 - 39\%$ in total, whereas removing 40% for both conditions removes $63 - 74\%$ in total.) We refer to these in Table 2 as the λ_2 (row) and MAE (column) pruning conditions.

One general observation is that the errors decrease when the MAE condition is used (see within rows), which was expected. However, in several cases the errors do not decrease when the λ_2 condition is used (see within columns). We believe the reason is as follows. The λ_2 condition is meant to deal with the aperture problem. But since our cluttered scenes have so many T-junctions (recall Fig. 4) and corners and relative few isolated edges, the aperture problem turns out to be less of a problem that one might anticipate.

Another general observation is that the motion parallax errors are typically lowest when the 40% threshold is applied for both pruning conditions, which is represented by the lower right corner of each subtable in Table 2. The real power seems to come from the MAE condition. Without using this condition (i.e. MAE 0%), errors are higher for the Lucas-Kanade/Rieger-Lawton method than for Mann-Langer (See right column in the table). With the MAE 40% condition, the errors are typically slightly lower than the errors found for the Mann-Langer method. (One qualifier to this comparison is that the Mann-Langer method was previously run using 32 frames [10, 11] whereas here we use only 12 frames in order to compare to a typical Lucas-Kanade implementation.)
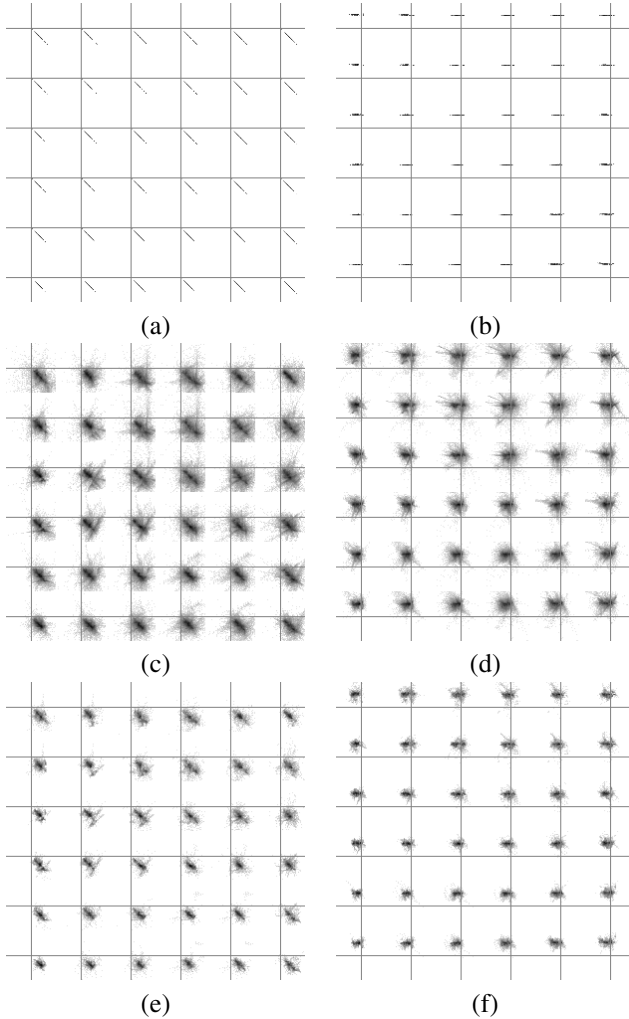
Figure 6: True velocities for the (a) synthetic Diagonal translation sequence. (b) synthetic Translation and rotation sequence. (c) Estimated velocities corresponding to sequence in (a) with 0D textures. (d) Estimated velocities corresponding to sequence in (b) with 0D textures. (e) Velocities in (c), but with $\lambda_2\% = 40$ and $MAE\% = 40$. (f) Velocities in (d), but with $\lambda_2\% = 40$ and $MAE\% = 40$.

5   Discussion

To estimate the motion parallax direction in a local image region, the surfaces in the region must have a wide range of depths. This condition is certainly met in 3D cluttered scenes. The classical way to estimate parallax is to first estimate local image velocity and then look at differences of these velocities. It is unclear whether this method could work for 3D cluttered scenes, however, since most techniques for estimating local image velocities such as Lucas-Kanade require that the underlying velocity field is smooth. This constraint clearly does not hold in 3D cluttered scenes because of the many depth discontinuities [11].

This paper addresses whether local image velocities obtained from Lucas-Kanade can be used to accurately estimate the parallax direction for 3D cluttered scenes. We identified unreliable velocity estimates in a standard way, namely those that were either susceptible to the aperture problem or that did not satify the local image translation model (intensity conservation). The reliable estimates that remained were used to estimate the motion parallax direction. These correspond to smooth velocity field regions and can be classified in two types.

The first type occurs for surfaces near the camera. Such surfaces are large in the image because of perspective and hence provide a relatively large discontinuity–free region in which velocity can be reliably measured for 2D textures (i.e. when the aperture problem is not present). Note that one such region alone is insufficient for estimating motion parallax direction since multiple depths are required. The second type of reliable estimates is more subtle and occurs when a cluster of surfaces having similar depth was present in a region. These surfaces tend to produce near–continuous velocity fields provided that their range of depths is sufficiently small and, as such, they tend to approximately satisfy the intensity conservation condition. In addition, the aperture problem does not occur at discontinuities because of the multiple texture or edge orientations. In effect, such a cluster of surfaces acts as a single larger textured object.

We found that these two types of reliable image velocity estimates were sufficient for estimating the direction of motion parallax. These parallax estimates were often as accurate or more accurate than estimates obtained using a frequency based technique [11] which does not rely on local velocity estimates. Further experiments are required to compare these two types of techniques and to examine how performance depends on scene parameters such as the amount of clutter, the surface texture, the number of frames, transparency, etc. For now, our main conclusion is that if the Lucas-Kanade estimates are sufficiently pruned then they can be used as a basis for estimating motion parallax. This conclusion is somewhat surprising, given the large number of discontinuites present in the scenes we considered.

Acknowledgements

References

[1] J. Weickert A. Bruhn and C. Schnörr. Lucas/kanade meets horn/schunck: combining local and global op-

tic 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] 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. (PAMI)*, 14(9):886–896, September 1992.

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

[7] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] 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.

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

[10] R. Mann and M. S. Langer. Estimating camera motion through a 3d cluttered scene. In *Canadian Conference on Computer and Robot Vision*, pages 472–479, London, Canada, May 2004.

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

[12] K.M. Mutch and W.B. Thompson. Analysis of accretion and deletion at boundaries in dynamic scenes. *IEEE Trans. Patt. Anal. Mach. Int. (PAMI)*, 7(2):133–138, March 1985.

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

[14] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.

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

SYNTHETIC SEQUENCES

Diagonal translation
$T = (0.2, 0.2, 0), \Omega = (0, 0, 0)$

2D textures

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	2.4	1.5	1.6	
20%	2.5	1.6	1.7	3.2
40%	2.6	1.7	1.8	

0D (flat) textures

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	39.3	7.5	28.7	
20%	6.1	5.4	5.7	4.5
40%	6.4	5.5	5.2	

Translation and rotation
$T = (0.2, 0, 0), \Omega = (-4.2, -4.2, 0)$

2D textures

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	14.3	3.8	2.7	
20%	7.8	3.1	2.3	6.8
40%	7.9	3.0	2.4	

0D (flat) textures

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	13.1	10.3	11.5	
20%	14.6	4.3	4.9	7.9
40%	14.4	4.7	5.3	

REAL SEQUENCES

Lateral translation only

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	10.8	8.9	7.3	
20%	10.8	8.2	6.2	8.5
40%	11.6	8.5	7.0	

Lateral translation and rotation

Rieger-Lawton/Lucas-Kanade				Mann-Langer
$\%\lambda_2 \backslash \%$MAE	0%	20%	40%	
0%	18.9	11.3	9.3	
20%	18.2	10.7	8.6	9.3
40%	19.0	10.2	7.9	

Table 2: Average absolute errors in motion parallax direction based on pruned velocity estimates. The removed percentages are indicated in the leftmost column ($\lambda_2\%$) and the first row of each subtable ($MAE\%$). The rightmost column shows results of the Mann-Langer method.