MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1334 April 1992

Towards Autonomous Motion Vision

M. Ali Taalebinezhaad

Abstract

Earlier, we introduced a direct method called *fixation* for the recovery of shape and motion in the general case. The method uses neither *feature correspondence* nor *optical flow*. Instead, it directly employs the spatio-temporal gradients of image brightnesses.

This work reports the experimental results of applying some of our fixation algorithms to a sequence of real images where the motion is a combination of translation and rotation. These results show that parameters such as the *fixation patch size* have crucial effects on the estimation of some motion parameters.

Some of the critical issues involved in the implementation of our autonomous motion vision system are also discussed here. Among those are the criteria for automatic choice of an optimum size for the fixation patch, and an appropriate location for the *fixation point* which result in good estimates for important motion parameters.

Finally, a calibration method is described for identifying the real location of the rotation axis in imaging systems.

Keywords: Motion, shape, fixation, least-squares, direct motion vision, normalized error, fixation point, camera calibration.

Acknowledgments: This paper describes the research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the artificial intelligence research at this laboratory is provided in part by DARPA under the ONR contract N00014-85-K-0124.

1 Introduction

Recovery of relative motion between an observer and an environment as well as the structure of the environment, from time varying images, is the goal in motion vision. Much of the earlier work on recovering motion has been based either on establishing correspondences between the prominent features in the images of a sequence (*correspondence*) or establishing the velocity of points over the whole image, commonly referred to as the *optical flow*.

In general, identifying features here means determining gray-level corners. For images of smooth objects, it is difficult to find good features or corners. Furthermore, the *correspondence* problem has to be solved, that is, feature points from consecutive frames have to be matched. On the other hand, the computation of the local flow field exploits a constraint equation between the local brightness changes and the two components of the *optical flow*. This only gives the components of flow in the direction of the brightness gradient. To compute the full flow field, one needs additional constraints such as the heuristic assumption that the flow field is locally smooth [5, 4]. This leads to an estimated optical flow field which may not be the same as the true motion field.

The use of *optical flow* or *correspondence* techniques for solving motion vision problems has proven to be rather unreliable and computationally very expensive [16, 15, 7]. This has motivated the investigation of *direct methods* which use the image brightness information directly to recover the motion and shape.

Previous work in direct motion vision has used the *Brightness-Change Constraint Equation* (BCCE) for rigid body motion [8]

$$E_t + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{\mathbf{s} \cdot \mathbf{t}}{Z} = 0 \tag{1}$$

to solve special cases such as *known depth* [5], *pure translation* or *known rotation* [6], *pure rotation* [6], and *planar world* [8]. All these direct methods are restricted in the types of the motion or shape that they can handle.

Recently, we introduced a direct method called *fixation*[1] for solving the motion vision problem in the general case without placing restrictions on the motion or the shape [12, 13, 11]. The fixation method is based on the theoretical proof that for a sequence of fixated images (a sequence of images with one stationary image point in them), the 3D rotational velocity $\boldsymbol{\omega}$ can always be explicitly expressed in terms of a linear function of the 3D translational velocity $\mathbf{t}$. Namely,

$$\boldsymbol{\omega} = \omega_{\mathbf{R}_o} \hat{\mathbf{R}}_o + \frac{1}{\|\mathbf{R}_o\|}(\mathbf{t} \times \hat{\mathbf{R}}_o) \tag{2}$$

where $\hat{\mathbf{R}}_o$ is the unit vector along the position vector of the *fixation point* (a point in the image plane which stays stationary) and $\omega_{\mathbf{R}_o}$ is the component of rotational velocity about the fixation axis $\mathbf{R}_o$.

It should be emphasized that we do not need to know the real fixation point, if there is any, to take advantage of this *fixation constraint equation* (FCE), eqn. (2). In fact, our algorithm

[1]The terms and notations used in this paper have been defined in our previous work such as [10] or [12]. For a review of the necessary background, the reader is encouraged to consult one of those references.

allows us to choose virtually any point as the fixation point by a simple manipulation of one of the images and obtain a sequence of fixated images [12, 13].

The combination of the *Fixation Constraint Equation* (FCE), eqn. (2), and the BCCE, eqn. (1) offers a solution to the motion vision problem of arbitrary motion relative to an arbitrary rigid environment. That is, it allows recovery of the depth map Z, total 3D rotational velocity, and 3D translational velocity t without putting severe restrictions on the motion or the shape [12, 13].

Fixation does not necessarily mean tracking! Our technique for obtaining *fixated images* is not only simpler than the previous tracking methods, but also is more general. For example, Aloimonos & Tsakiris [2] propose a method for tracking a target of known shape; Bandopadhay et al. [3] use optical flow and feature correspondence for tracking the principal point in order to find the motion in a special case (They assume that there is no rotation along the optical axis.) without considering noise; and Sandini & Tistarelli [9] use an optical flow based tracking method for finding the depth in a special case (no rotation along the optical axis). Also, Thompson [14] introduces an optical flow method for recovering the motion in special case where the rotational velocity along the optical axis is zero. His method requires a sequence of tracked images at the principal point but he acknowledges that the actual implementation of such tracking requirement in engineering systems is not possible yet.

On the other hand, our *fixation method* does not require tracked images as its input. Instead, it introduces a *pixel shifting process* which constructs a sequence of fixated images at any arbitrary point, chosen as *fixation point*, and for any input sequence of images [12, 13]. This is done entirely in software without physically moving the camera for *tracking*. Besides being reliable, our *pixel shifting process* is much simpler than those tracking methods.

This work reports the experimental results of applying some of the fixation algorithms to real image sequences where the motion is a combination of translation and rotation. Finding the *fixation velocity* (velocity at the fixation point) and the component of rotational velocity about the fixation axis, $\omega_{\mathbf{R}_o}$, are important steps in our fixation method [12, 13]. The results here show that the fixation velocity and $\omega_{\mathbf{R}_o}$ can be estimated satisfactorily if proper parameters values are used.

Some of the crucial implementation issues of our fixation technique are also discussed here. Among those are the autonomous selection of an optimum size for the fixation patch (the image patch around the fixation point) based on an error norm (*normalized error*), and the choice of an appropriate location for the fixation point.

And finally, a calibration method is described for identifying the real location of the rotation axis in imaging systems.

2 The Effect of Fixation Patch Size

Finding the fixation velocity (velocity at the fixation point), and the component of rotational velocity about the fixation axis, $\omega_{\mathbf{R}_o}$, is an important step in our fixation method for recovering the shape and motion from an arbitrary sequence of input images. This is because in our method a *pixel shifting process* uses the fixation velocity to construct a sequence of fixated images from an arbitrary sequence of input images. We also need $\omega_{\mathbf{R}_o}$ for computing the total rotational velocity [12].

The algorithms used for recovering the fixation velocity and $\omega_{\mathbf{R}_o}$ obtain their input information from the fixation patch (an image patch around the fixation point) [12, 13]. In order to study the effect of the fixation patch size on the estimation of the desired motion parameters, we have used a sequence of real images acquired at the *Imaging Laboratory of Carnegie Mellon University.* Figures 1 and 2 show two of these 16 bits grey levels, 576×384 pixel images. The camera has a nominal focal length of 24 *mm*, and a pixel size of 0.02×0.02 *mm*. The calibrated *principal point* has been used as a fixation point. In the raster format system (origin at the top left corner of the image), the principal point is located near the center of image, pixel $(275, 205)$. The frontal depth of this point is about 1450 *mm*.

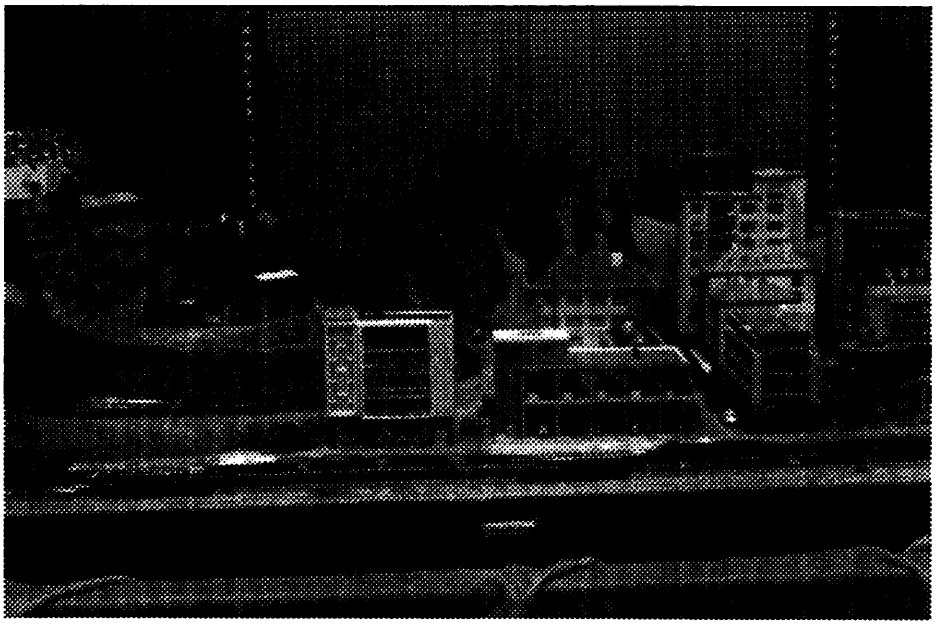

Figure 1: The First image in the landscape image sequence.

The real motion between these two images has both translational and rotational components. The real rotation is -0.3 *degree* about the optical axis Z. The real translation is -2 *mm* along the horizontal axis X. Testing our algorithms using such real images is valuable because the observed motion is relatively large (more than subpixel motion in the image plane). For very large motions it is enough to use higher frame grabbing rates. These days, there are commercially available frame grabbers which are capable of capturing up to 7,500 frame per second at 12-bit gray scale resolution on personal computers [1].

2.1 Estimation of rotational velocity component, $\omega_{\mathbf{R}_o}$

The motion field velocity due to the component of the rotational velocity of an observer relative to an environment along $\mathbf{R}_o$ is given by $-(\boldsymbol{\omega}_{\mathbf{R}_o} \times \mathbf{r}) = -\omega_{\mathbf{R}_o}(\hat{\mathbf{R}}_o \times \mathbf{r}) = -\frac{\omega_{\mathbf{R}_o}}{\|\mathbf{r}_o\|}(\mathbf{r}_o \times \mathbf{r})$, where $\hat{\mathbf{R}}_o = \hat{\mathbf{r}}_o$ is the unit vector along $\mathbf{r}_o$, position vector of the fixation point in a viewer

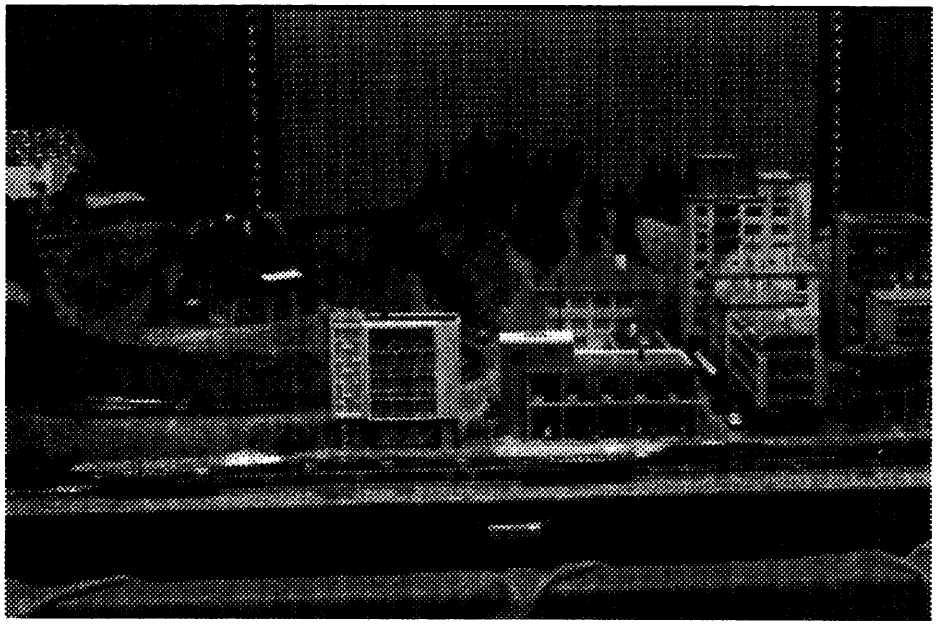

Figure 2: The second image in the landscape image sequence after undergoing a real motion of -0.3 *degree* rotation about the nominal optical axis Z, and -2 *mm* translation along the horizontal axis X.

centered coordinate system. Assuming that depth is approximately the same on the *fixation patch* (a small patch around the fixation point) and substituting for $\mathbf{r}_\circ = (x_\circ \; y_\circ \; 1)^T$ and $\mathbf{r} = (x \; y \; 1)^T$, we can write the components of the total motion field velocity due to fixation velocity and $\omega_{\mathbf{R}_\circ}$ as

$$\left\{ \begin{array}{rcl} x_t & = & u_\circ - \frac{\omega_{\mathbf{R}_\circ}}{\|\mathbf{r}_\circ\|}\hat{\mathbf{x}} \cdot (\mathbf{r}_\circ \times \mathbf{r}) \;=\; u_\circ + \bar{\omega}_{\mathbf{R}_\circ}(y - y_\circ) \\ y_t & = & v_\circ - \frac{\omega_{\mathbf{R}_\circ}}{\|\mathbf{r}_\circ\|}\hat{\mathbf{y}} \cdot (\mathbf{r}_\circ \times \mathbf{r}) \;=\; v_\circ - \bar{\omega}_{\mathbf{R}_\circ}(x - x_\circ) \end{array} \right. \tag{3}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are the unit vectors along the x and y axes and $\bar{\omega}_{\mathbf{R}_\circ}$ is a notation for $\frac{\omega_{\mathbf{R}_\circ}}{\|\mathbf{r}_\circ\|}$.

Ideally, the BCCE must be satisfied at any point on the fixation patch as [5]

$$x_t E_x + y_t E_y + E_t = 0. \tag{4}$$

Substituting for x_t and y_t from eqn. (3) into the BCCE, eqn. (4), gives

$$[u_\circ + \bar{\omega}_{\mathbf{R}_\circ}(y - y_\circ)]E_x + [v_\circ - \bar{\omega}_{\mathbf{R}_\circ}(x - x_\circ)]E_y + E_t = 0. \tag{5}$$

Due to noise, eqn. (5) does not necessarily hold for any pixel (x, y) so we can find $u_\circ, v_\circ$ and $\bar{\omega}_{\mathbf{R}_\circ}$ by minimizing the sum of squares of errors over the fixation patch. In other words we want to minimize

$$\iint [(u_\circ + \bar{\omega}_{\mathbf{R}_\circ}(y - y_\circ))E_x + (v_\circ - \bar{\omega}_{\mathbf{R}_\circ}(x - x_\circ))E_y + E_t]^2 dx\, dy \tag{6}$$

with respect to u_o, v_o and $\bar{\omega}_{\mathbf{R}_o}$. This results in a system of three linear equations that can be solved for the three unknowns

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} u_o \\ v_o \\ \bar{\omega}_{\mathbf{R}_o} \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}. \tag{7}$$

Matrix **A** is symmetric and its elements are given by

$$\begin{cases} a_{12} = \iint E_x E_y dx\, dy \\ a_{13} = \iint E_x[E_x(y-y_o) - E_y(x-x_o)]dx\, dy \\ a_{23} = \iint E_y[E_x(y-y_o) - E_y(x-x_o)]dx\, dy \\ a_{11} = \iint E_x^2 dx\, dy \\ a_{22} = \iint E_y^2 dx\, dy \\ a_{33} = \iint [E_x(y-y_o) - E_y(x-x_o)]^2 dx\, dy \end{cases} \tag{8}$$

and the of components of vector **C** are as follows:

$$\begin{cases} c_1 = -\iint E_t E_x dx\, dy \\ c_2 = -\iint E_t E_y dx\, dy \\ c_3 = -\iint E_t[E_x(y-y_o) - E_y(x-x_o)]dx\, dy. \end{cases} \tag{9}$$

Considering that the fixation point coordinates x_o and y_o are known, then the sets of equations in (8) and (9) show that the elements of matrix **A** and the components of vector **C** are fully computable. After finding $\bar{\omega}_{\mathbf{R}_o}$, we can easily calculate $\omega_{\mathbf{R}_o}$ as

$$\omega_{\mathbf{R}_o} = \bar{\omega}_{\mathbf{R}_o} \sqrt{x_o^2 + y_o^2 + 1}. \tag{10}$$

In the special case where the fixation point is at the principal point, $x_o = y_o = 0$, elements of matrix **A** and the components of the vector C are simplified further and $\omega_{\mathbf{R}_o}$ becomes equal to $\bar{\omega}_{\mathbf{R}_o}$.

Using these algorithms, we can find $\omega_{\mathbf{R}_o}$ for any given fixation patch size. Figure 3 shows that for small patch sizes (less than 30×30 pixel in this case) the estimated value for $\omega_{\mathbf{R}_o}$ is oscillating wildly and results in unacceptable values. As the patch size increases, the estimated $\omega_{\mathbf{R}_o}$ converges towards the real value of rotation. For large patch sizes (around 100×100 pixel in this case) the estimated rotation, -0.309 *degree*, becomes roughly the same as the real rotation, -0.3 *degree*.

It can be seen that the size of fixation patch has a critical effect on the estimated values of the component of rotational velocity about the fixation axis, $\omega_{\mathbf{R}_o}$. A small patch size results in a value for $\omega_{\mathbf{R}_o}$ which is usually far distant from the real value. This is possibly because in a small patch, small translations can be interpreted as large rotations. Figure 4 shows a hypothetical situation where (a) and (b) are a sequence of a small 3×3 pixel patch. The real motion in this case is most likely a pixel heigh vertical translation. But if we try to interpret it as a rotation about the patch center we will end up with a 45 *degree* of rotation which is not acceptable, considering the assumed small motion between images. As a conclusion, we should use relatively large patch sizes in order to obtain good estimates for the rotational velocity component about the fixation axis, $\omega_{\mathbf{R}_o}$.

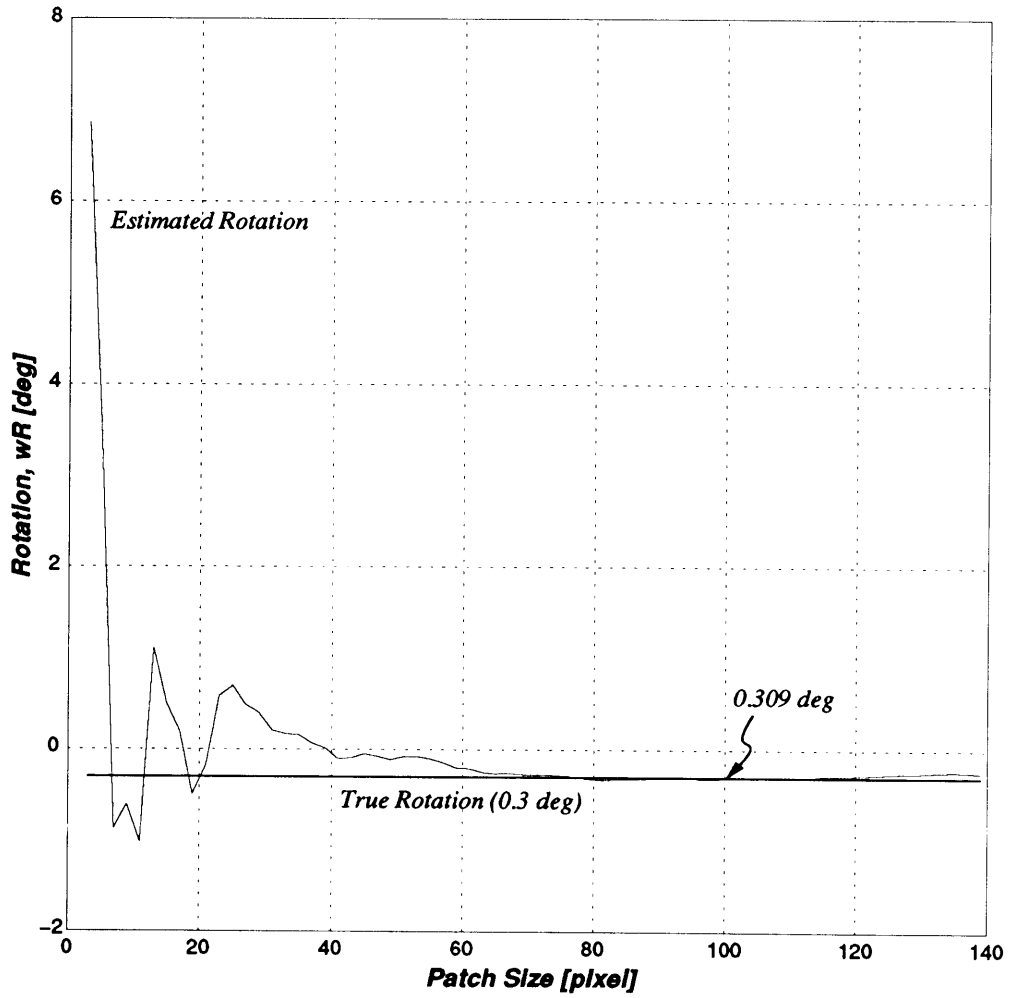

Figure 3: Estimated value of the component of rotation velocity about the fixation axis, $\omega_{\mathbf{R}_o}$, versus the fixation patch size for the landscape image sequence. For large patch sizes, the estimated value of $\omega_{\mathbf{R}_o}$ converges towards the real value of $\omega_{\mathbf{R}_o}$, -0.3 *degree*.

3 Autonomous Choice of Optimum Fixation Patch Size

The experimental results and explanations in the previous section suggest that relatively large patch sizes should be used in order to get a good estimate for the component of the rotation along the fixation axis, $\omega_{\mathbf{R}_o}$. On the other hand, we know that in general a large patch size will result in a wrong estimate for the fixation velocity because depth variations generally increase as the patch size increases. In this section, we will describe a technique for choosing an optimum fixation patch size which results in a good estimate for the fixation velocity.

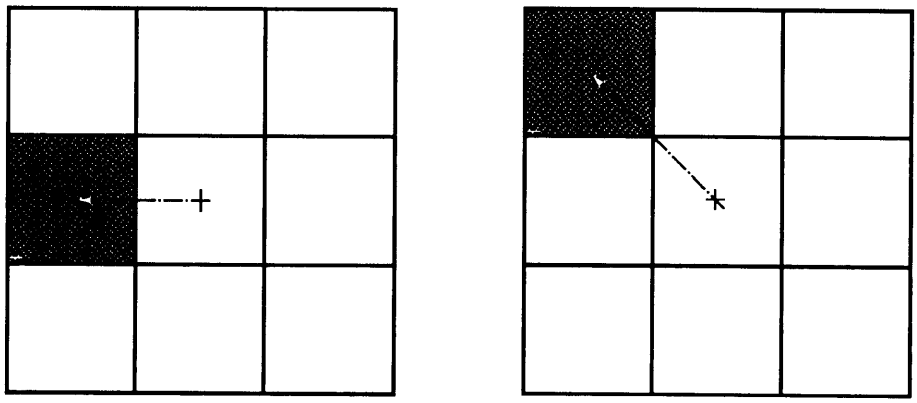

Figure 4: Using small fixation patch can result in wrong interpretation of large rotation. In a patch of 3×3 pixel, a pixel heigh vertical translation can be seen as 45 *degree* rotation which is not an acceptable answer at all, considering the finite motion between images.

3.1 Computing the fixation velocity

We can find a good estimate for $\omega_{\mathbf{R}_o}$ using a relatively large patch but the corresponding fixation velocity estimate from such large patches is not usually reliable. Using only the acquired estimate for $\omega_{\mathbf{R}_o}$ from a large patch, we can write the total motion field at any point (x,y) on a small patch around the fixation point (*fixation patch*). As we showed in subsection 2.1

$$\begin{cases} x_t &= u_o + \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(y-y_o) \\ y_t &= v_o - \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(x-x_o) \end{cases} \tag{11}$$

where (x_o, y_o) is the position of fixation point (located in the image plane), and (u_o, v_o) is the fixation velocity that we are about to estimate. After substituting for x_t and y_t into the BCCE, eqn. (4), we will have

$$\left(u_o + \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(y-y_o)\right) E_x + \left(v_o - \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(x-x_o)\right) E_y + E_t = 0. \tag{12}$$

However, due to noise, the above equation does not necessarily hold for any pixel. As a result, we can find u_o and v_o by minimizing the sum of the errors over the whole fixation patch. Namely, by minimizing

$$\iint_p \left[\left(u_o + \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(y-y_o)\right) E_x + \left(v_o - \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}}(x-x_o)\right) E_y + E_t\right]^2 dx\, dy \tag{13}$$

with respect to u_o and v_o. This will result in the following system of linear equations,

$$\begin{bmatrix} \iint_p E_x^2 dx\, dy & \iint_p E_x E_y dx\, dy \\ \iint_p E_x E_y dx\, dy & \iint_p E_y^2 dx\, dy \end{bmatrix} \begin{pmatrix} u_o \\ v_o \end{pmatrix} = \begin{pmatrix} \iint_p \left(\frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}} ((x-x_o)E_y - (y-y_o)E_x) - E_t \right) E_x dx\, dy \\ \iint_p \left(\frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2+y_o^2+1}} ((x-x_o)E_y - (y-y_o)E_x) - E_t \right) E_y dx\, dy \end{pmatrix} \quad (14)$$

that can be solved for the two unknowns u_o, and v_o. Note that $\omega_{\mathbf{R}_o}$ has been already computed and is a known value in this equation.

Figure 5 shows the estimated values of the horizontal translation $U = \frac{f u_o}{Z_o}$ for the landscape image sequence for different sizes of the fixation patch where f is the focal length and Z_o is depth at the fixation point. It can be seen that U nicely converges towards the real horizontal translation, $-2\ mm$. The dependency of U on the patch size is quite clear in this figure.

In practice, we do not know the real fixation velocity, and therefore we cannot select an appropriate fixation patch size by checking the computed values of fixation velocity. In order to solve this problem, we should find an autonomous way of choosing an optimum size for the fixation patch.

3.2 Normalized error

We showed that for any given size of the fixation patch, we can find the fixation velocity components, u_o and v_o. Also the component of the rotational velocity about the fixation axis, $\omega_{\mathbf{R}_o}$, can be estimated using a relatively large patch. Knowing these values, the motion field velocity (x_t, y_t) at any point (x, y) in the image plane is given by eqn. (11). Ideally, for any given image point (x, y) the BCCE, eqn.(4), must be satisfied. However, in practice we are dealing with real images which are noisy and as a result, the term $x_t E_x + y_t E_y + E_t$ does not usually become zero. This term can be considered as an error term for the corresponding pixel. In a patch of size $p \times p$ pixel, we can add these error terms to define the *normalized error* as

$$e = \frac{\sum [x_t E_x + y_t E_y + E_t]^2}{p^2}. \quad (15)$$

This definition allows us to compare the performance of different patch sizes by studying the behavior of the normalized error e with respect to the changes in the patch size p. This consideration makes it possible for us to find an optimum patch size.

3.3 Case I: Small changes in relative depth as p increases

Figure 6 shows the normalized error versus the fixation patch size for the landscape image sequence. Although this plot corresponds to a specific image and motion, it shows one of the two typical representations of the normalized error behavior as the patch size increases. As shown in this figure, the normalized error first increases with the patch size and reaches a peak and then dips down.

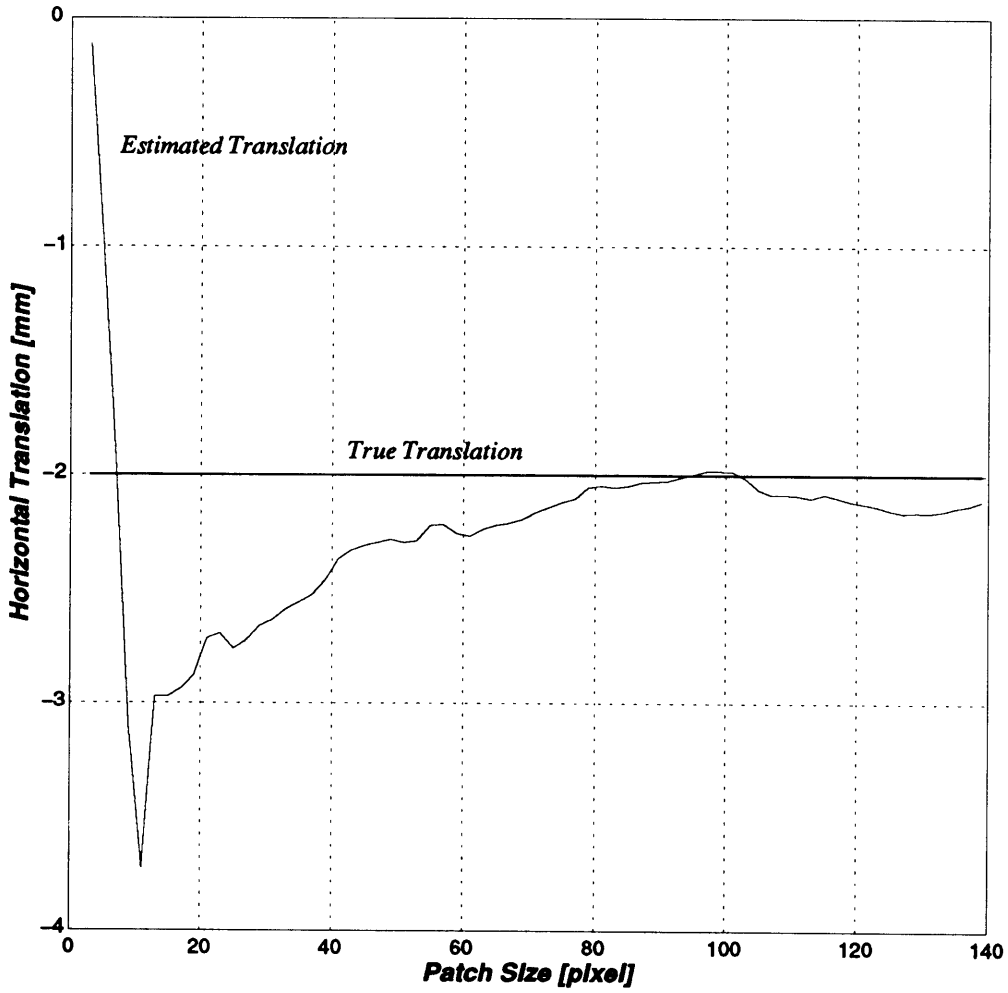

Figure 5: Estimated value of the horizontal component of translational velocity, along the X-axis, versus the fixation patch size for the landscape image sequence.

This is because initially for the smallest patch size (3×3 pixel) the algorithm finds the motion estimates that makes the BCCE error term ($x_t E_x + y_t E_y + E_t$) as small as possible. In a 3×3 pixel patch, there is only one BCCE error term which corresponds to the central pixel of the patch. The algorithm does a good job in minimizing this error term but the motion estimates are usually very bad at this level because basically there is not enough data available to the algorithm.

In the next level, we have a patch of 5×5 pixel size which includes 9 different permutations of the basic block of 3×3 pixel patch. There is still not enough data for the algorithm to come up with good motion estimates but it finds parameters which minimize the the sum of the BCCE error terms. Usually, the algorithm is not as successful as it was for the 3×3 pixel patch size because it should deal with 9 error terms instead of one and this will result in higher normalized error.

As we increase the patch size, the struggle between providing more data to the algorithm

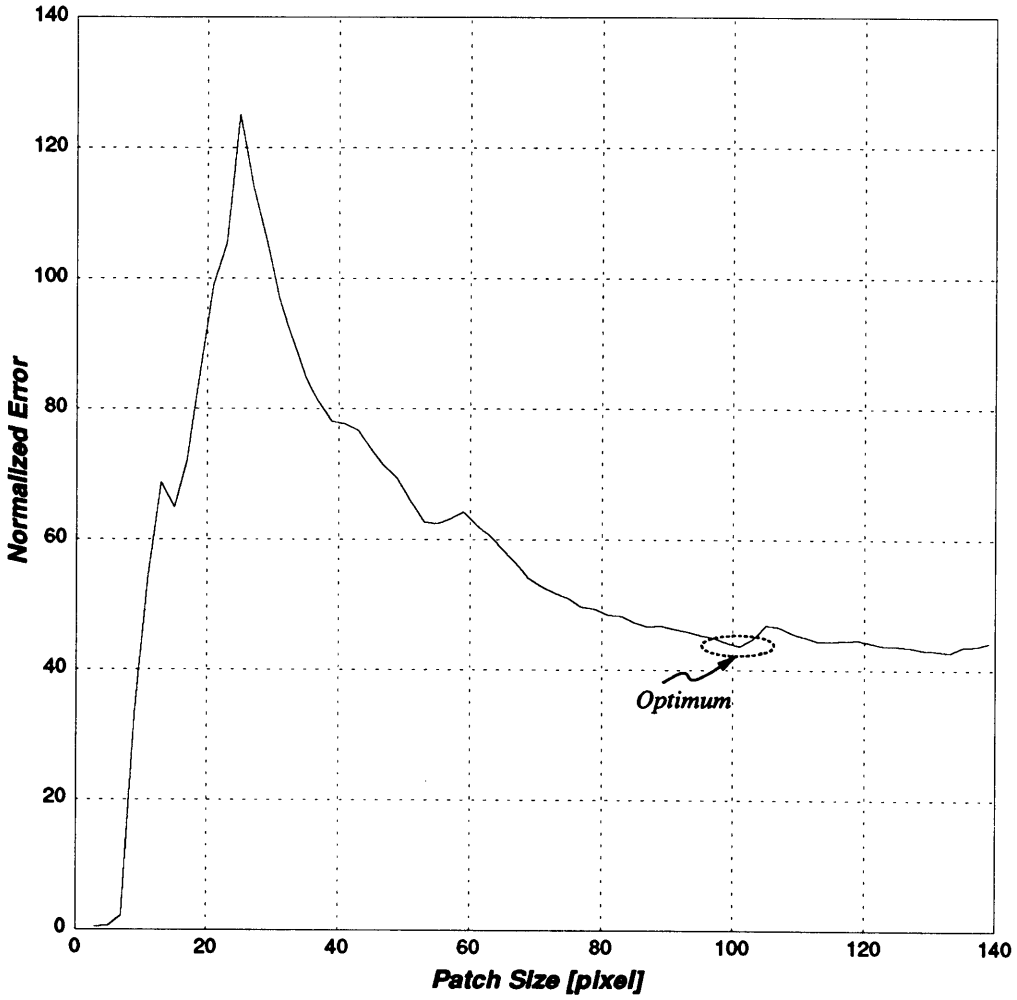

Figure 6: Estimated value of the normalized error e versus the fixation patch size for the landscape image sequence.

and satisfying more error terms continues and for relatively small patch sizes results in higher normalized error. The normalized error increases until it reaches a peak point where the rule of more input data becomes more important than satisfying more error terms. Then by increasing the patch size, we are providing more useful data to the algorithm and this will give a better motion estimate and results in a smaller normalized error.

After dipping down, the normalized error stays roughly the same in this case because the relative depth variation does not change much with the patch size, (Fig. 6). The optimum patch size in this example occurs around 100×100 pixel which corresponds to the start of small normalized error slope (roughly flat) after the first peak. In this example, relative depth changes are small (1250 mm to 1625 mm, about 30% difference) and stay roughly the same as the patch size increases.

3.4 Case II: Significant changes in relative depth as p increases

In this section we will study another image sequence where there is considerable relative depth changes as we increase the patch size. The purpose is to see how the normalized error behaves in this case. Figures 7 and 8 show a sequence of two 227×280 pixel, 32-bit images (cup images). The real motion of the camera is a horizontal translation of 2.5 mm to the

Figure 7: The first image in the cup image sequence.

right. The camera has a nominal focal length of 18.66 mm, pixel-width of 0.032 mm, and pixel height of 0.029 mm. We have used the nominal principal point (image center) as our fixation point.

Figure 9 shows the estimates for the horizontal translation, vertical translation, and the rotational velocity component $\omega_{\mathbf{R}_\circ}$ which are obtained using the same algorithms used for the landscape image sequence. It is obvious that the estimated values depend strongly on the size of the fixation patch. However, we can find good estimates for these motion parameters if we choose the right fixation patch size.

The normalized error for this sequence of cup images is shown in Fig. 10. As before, the normalized error first increases and after reaching a peak it dips down and then grows with the patch size again. This is because in the beginning, insufficient information results in extremely wrong estimates specially for the rotational component and this causes the normalized error to increase with the patch size. As we are providing more and more data to the algorithm, we obtain better estimates for the motion components and this decreases the normalized error. If we increase the patch size beyond an optimum patch size, which occurs at about 50 pixel in this example, the normalized error starts increasing again. In this 50×50 pixel patch, we have a considerable amount of relative depth change (from 584 mm to 914 mm, about 60 % increase). Such significant relative depth variation leads to wrong fixation velocity estimates which in turn results in a larger normalized error.

Figure 8: The second image in the cup image sequence after 2.5 *mm* horizontal motion of the camera to the right.

As one might expect, the optimum fixation patch size depends on the patch topology and texture which may vary not only from image to image but also from patch to patch in a single image. However, the general pattern of the normalized error allows us to autonomously find an optimum fixation patch size which gives good estimates for the fixation velocity components. This optimum fixation patch size corresponds to either the minimum normalized error after the first peak (in case where there is considerable change in the relative depth, as in the cup image sequence), or where there the normalized error starts changing slowly with the patch size (in the case where the relative depth does not change significantly with the patch size, as in the landscape image sequence).

4 Autonomous Choice of an Appropriate Fixation Point

In general, our fixation algorithms do not put any restrictions on the choice of the fixation point location and virtually any point can be chosen as the fixation point. Among all points, the choice of principal point (image center) makes the formulations simpler. However, in practice, one should take some measures in choosing an appropriate fixation point. Most significantly, the motion of the chosen fixation point should be detectable using the information from its corresponding patch. To clarify this, we can consider a patch which has a uniform brightness. Choosing the center of a such patch as the fixation point will not be useful. Because the motion of such point is irrecoverable using only the information from that patch.

Similar to 3.1 (with the exception that $\omega_{\mathbf{R}_o} = 0$ here), the least square method can be applied to the BCCE terms to obtain the following system of linear equations for the uniform

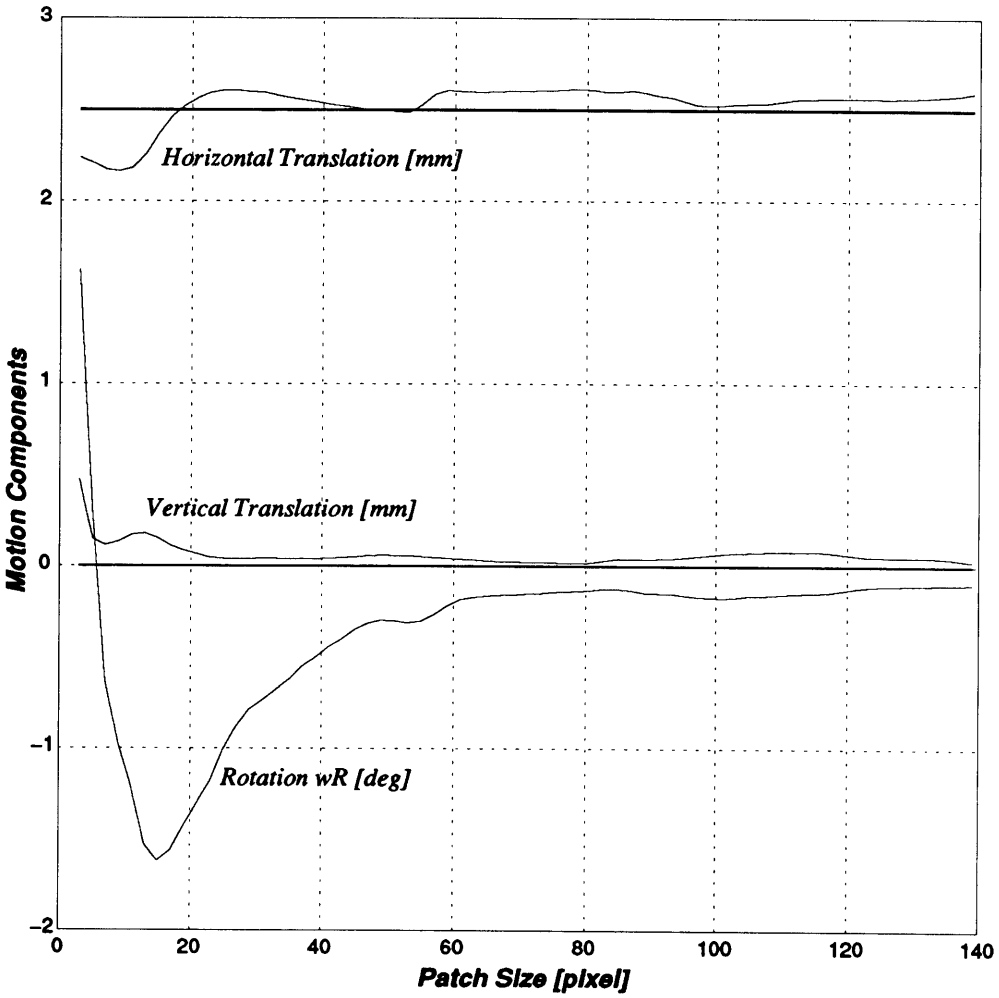

Figure 9: The estimated values for the horizontal translation, vertical translation, and the rotational velocity component, $\omega_{\mathbf{R}_o}$, versus fixation patch size in the cup image sequence.

motion field (u, v) on the patch as

$$\begin{bmatrix} \iint_p E_x^2 dx\, dy & \iint_p E_x E_y dx\, dy \\ \iint_p E_x E_y dx\, dy & \iint_p E_y^2 dx\, dy \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\iint_p E_t E_x dx\, dy \\ -\iint_p E_t E_y dx\, dy \end{pmatrix}. \tag{16}$$

It is obvious that the solution for (u, v) exists if the determinant of the above matrix

$$D = (\iint_p E_x^2 dx\, dy)(\iint_p E_y^2 dx\, dy) - (\iint_p E_x E_y dx\, dy)^2 \tag{17}$$

is not zero. But this is still not enough because it does not guarantee that the patch is an appropriate one.

If we denote the smaller eigenvalue of the coefficient matrix in eqn. (16) with λ_s,

$$\lambda_s = \frac{1}{2}\left[\iint_p (E_x^2 + E_y^2) dx\, dy - \sqrt{\iint_p (E_x^2 - E_y^2)^2 dx\, dy + 4(\iint_p E_x E_y dx\, dy)^2}\right] \tag{18}$$

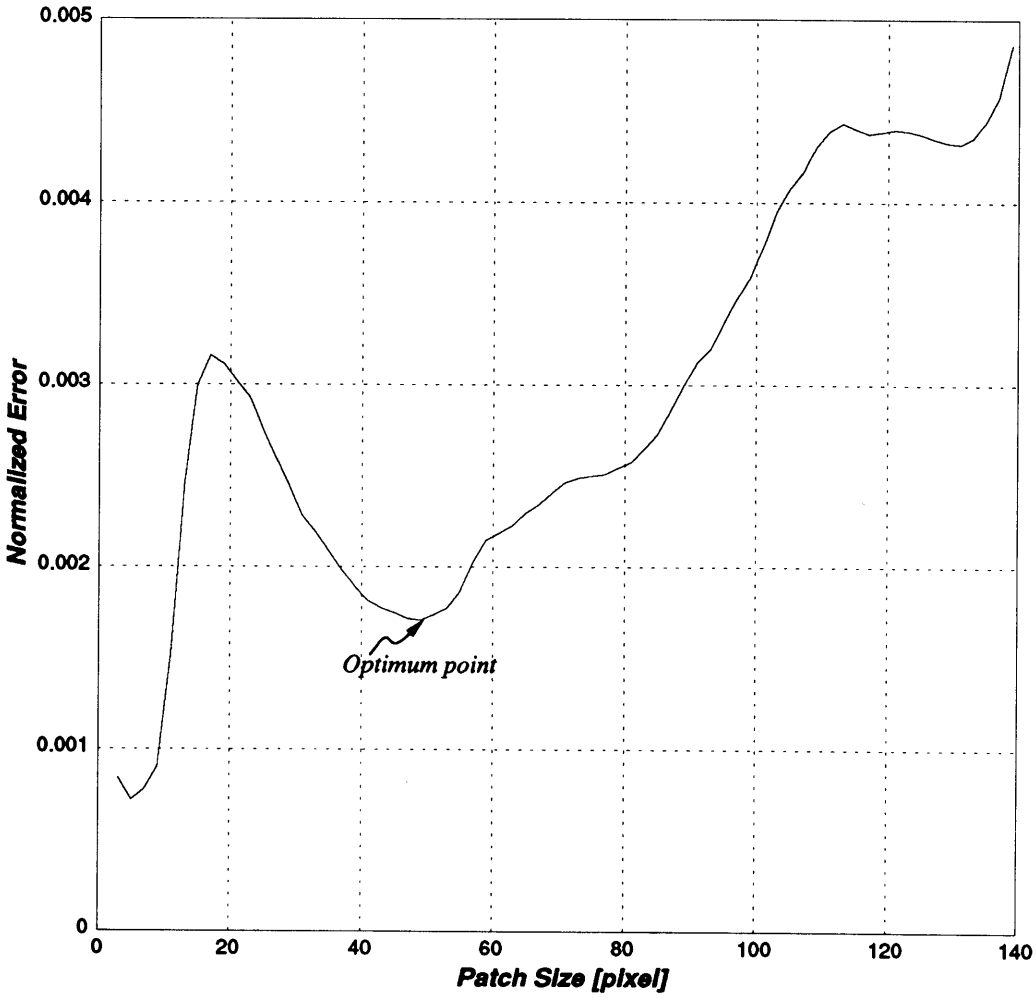

Figure 10: The normalized error versus fixation patch size for the cup image sequence.

then we can define a good fixation point as a point whose corresponding patch has the largest λ_s. Using such patch not only guarantees a solution ($D \neq 0$) but also ensures that our solution (u, v) is not sensitive to noise errors in the coefficient matrix of eqn. (16). It is simple to implement this criteria for autonomous choice of a good fixation point even in real noisy images.

We have addressed the question of finding an appropriate fixation point (the center of a fixation patch) among a number of given patches. But which patches should we check in the first place? We can search the whole image for a globally optimum location of a fixation point as follows:

1- Divide the whole image into 4 quadrants and find the corresponding λ_s for each quadrant.

2- Use the quadrant with the largest λ_s as a new base image.

3- Repeat steps 1 & 2 until reaching a quadrant with an acceptable size.

Doing such comprehensive search may not always be necessary. Instead, we can check a

limited number of neighboring patches (near the principal point, for convenience) and choose the center of the one with the largest λ_s as the fixation point.

5 Calibration of the Rotation Axis

In our experiment on the landscape images, we have not explicitly applied any vertical translation (along Y axis). However, the experimental results in Fig. 11 show a vertical translation of about -0.9 mm. This is mainly because the real rotation axis does not pass through the center of projection[2]. To clarify this, we should mention that in motion vision, it is assumed that the rotation axis passes through the origin of the viewer centered coordinate system, i.e the center of projection. But at the *CMU Imaging Laboratory*, the rotation mechanism is not set up so as to make the Z axis of rotation coincide with the optical axis. To obtain this experimental result, we have used fixation algorithms which assume that the rotation axis passes through the center of projection which is not true here.

According to the basic kinematics, the compensating translation which results from shifting the rotation axis is given by

$$\mathbf{V}_o = -\boldsymbol{\omega} \times \mathbf{B}. \tag{19}$$

Where B is a vector extending from a point on the real rotation axis to a point on the shifted rotation axis. In our special case, $\mathbf{V}_o = -(\omega\hat{z}) \times (b\hat{x})$. In this experiment, $\mathbf{V}_o = -0.9\hat{y}\ mm$, and $\omega = -0.3$ *degree*. As a result we conclude that the real rotation axis is located at about $b = -(-0.9)/((-0.3 \times \pi)/180) = -172\ mm$ perpendicular distance from the optical axis in the horizontal plane.

A similar method can be used for the calibration of the rotation axis which is parallel to the optical axis in any camera system arrangement. In order to find the real location of the rotation axis, the following steps should be taken:

1- Apply a pure rotation about the axis which is supposed to be the optical axis.

2- If $\omega_{\mathbf{R}_o}$ is not accurately known, compute it by applying the algorithms given in section 5 of [12] to a relatively large patch around the principal point.

3- Find the translational motion (u_o, v_o) at the principal point using eqn. (14).

4- Find the real location of the rotation axis using,

$$\begin{cases} b_x &= -\frac{v_o f}{Z_o \omega_{\mathbf{R}_o}} \\ b_y &= +\frac{u_o f}{Z_o \omega_{\mathbf{R}_o}} \end{cases} \tag{20}$$

where Z_o is depth at the principal point, and f is the focal length of the camera. As a result, the real rotation axis is parallel to the optical axis and intersects the image plane at point (b_x, b_y).

[2]If the CCD edges are not accurately aligned with the horizontal and vertical axes of the camera frame, i.e. the CCD is mounted at an angle with respect to the camera coordinate system, such kind of errors happen in both vertical and horizontal directions. But it is not the case here because the inaccuracy of motion has happened only in the vertical direction.

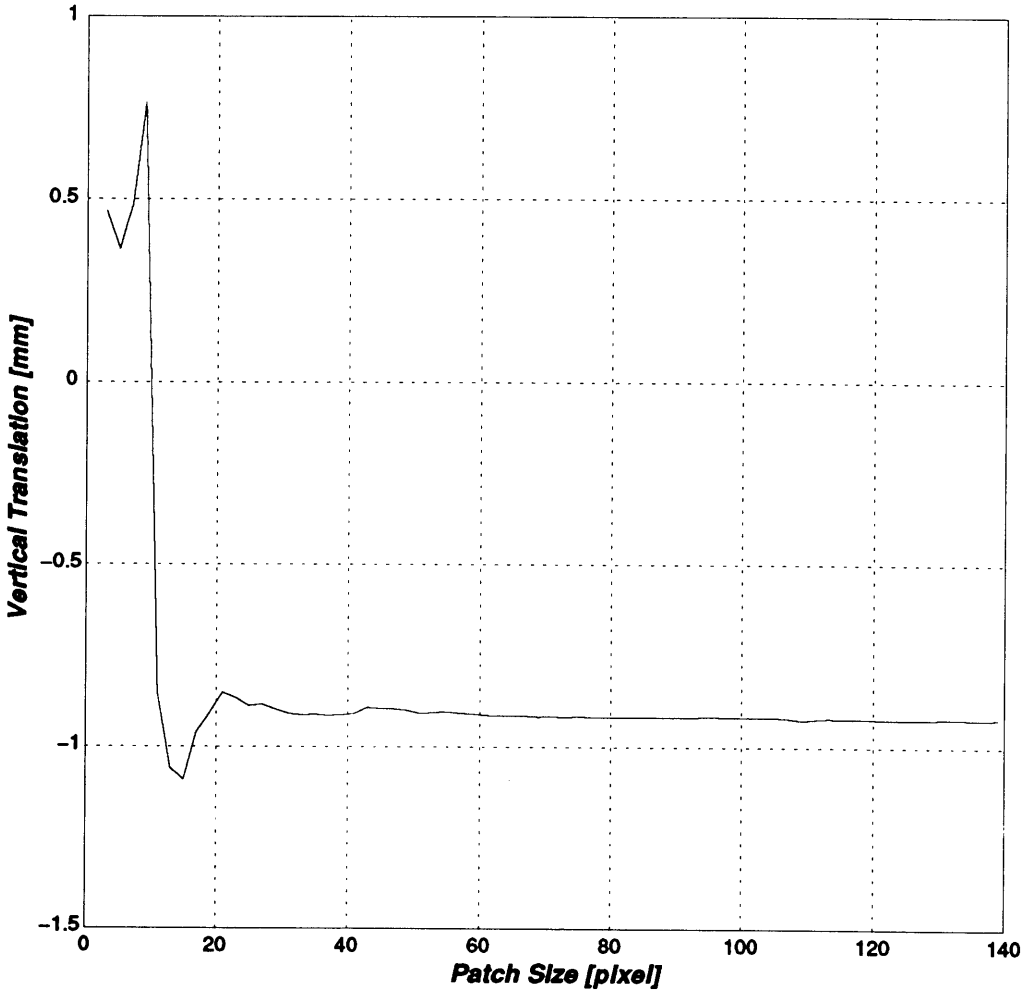

Figure 11: Estimated value of the vertical component of translational velocity, along the Y axis, versus the size of fixation patch for the landscape images.

6 Conclusions

Recovery of fixation velocity and the component of the rotational velocity along the fixation axis, $\omega_{\mathbf{R}_o}$, are important important steps in our fixation method. The experimental results presented here show that the fixation velocity and $\omega_{\mathbf{R}_o}$ can be computed satisfactorily using only the information from a small patch around the fixation point. The corresponding optimum patch sizes in these experiments is equivalent to a field of view of about 2×2.4 *degree*. Obtaining such good motion estimates while using only a small field of view ensures the feasibility of our fixation method. This is especially important if we consider that the nominal (not calibrated) focal length and pixel size are used in the computations.

The presented techniques for the autonomous choice of an appropriate fixation point, and an optimum fixation patch size allows us to find good estimates for the motion parameters. Also, the method described for the calibration of the real rotation axis offers a simple solution

to an important practical problem. This problem can result in considerable error in the motion estimates if it is not detected and compensated for.

Our goal has been to design a general motion vision system which takes any sequence of images as its input and recovers the motion and shape without any need to check, choose, and adjust parameters. Our fixation technique offers such a general system and this paper answers the critical issues involved in the full implementation of such an autonomous motion vision system.

7 Acknowledgments

Professor Berthold K. P. Horn has had a crucial role in shaping this work. The author would like to thank him for his insightful comments and suggestions. Also thanks to Professors W. Eric L. Grimson, and Tomaso Poggio for providing me with their valuable comments.

References

[1] Data sheets for mc4256 and mc6464 fast framing camera. Technical report, United Technologies, Adaptive Optics Associates, Cambridge, MA, 1992.

[2] J. Aloimonos and D. P. Tsakiris. On the mathematics of visual tracking. Technical Report CAR-TR-390, Computer Vision Laboratory, University of Maryland, MD, Sep. 1988.

[3] A. Bandopadhay, B. Chandra, and D. H. Ballard. Active navigation: Tracking an environmental point considered beneficial. In *Proc. of IEEE Workshop on Motion: Representation and Analysis*, pages 23–29, Kiawash Island, May 7-9 1986.

[4] E. C. Hildreth. *The Measurement of Visual Motion.* MIT Press, Cambridge, MA, 1984.

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

[6] B. K. P. Horn and E. J. Weldon Jr. Direct methods for recovering motion. *Int'l Journal of Computer Vision*, 2:51–76, 1988.

[7] J. J. Little and A. Verri. Analysis of differential and matching methods for optical flow. Technical Report AI memo 1066, MIT AI Laboratory, Cambridge, MA, Aug. 1988.

[8] S. Negahdaripour and B. K.P. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):168–176, Jan. 1987.

[9] G. Sandini and M. Tistarelli. Active tracking strategy for monocular depth inference over multiple frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):13–27, Jan. 1990.

[10] M. A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. Technical Report AI Memo 1187, MIT AI Laboratory, Cambridge, MA, Mar. 1990.

[11] M. A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. In *Proc. of IEEE International Conf. on Computer Vision*, Osaka, Japan, Dec. 1990.

[12] M. A. Taalebinezhaad. Partial implementation of fixation method on real images. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Maui, Hawaii, June 1991.

[13] M. A. Taalebinezhaad. Direct recovery of motion and shape in the general case by fixation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, in press, mid 1992.

[14] W. B. Thompson. Structure-from-motion by tracking occlusion boundaries. *Biological Cybernetics*, 62:113–116, 1989.

[15] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. Technical Report AI Memo 917, MIT AI Laboratory, Cambridge, MA, Dec. 1986.

[16] A. Verri and T. Poggio. Against quantitative optical flow. In *Proc. of the First IEEE Intl. Conf. on Computer Vision*, pages 171–180, Washington, DC, 1987.