ROBOT MOTION VISION
Part II: Implementation

M. A. Taalebinezhaad

MIT Artificial Intelligence Laboratory
Cambridge, MA 02139, USA

Abstract The idea of *Fixation* introduced a direct method for general recovery of shape and motion from images without using either *feature correspondence* or *optical flow* [1,2]. There are some parameters which have important effects on the performance of fixation method. However, the theory of fixation does not say anything about the autonomous and correct choice of those parameters. This paper presents the effect of those parameters' on the experimental results of implementing some of the fixation algorithms on real images where the motion is a combination of translation and rotation. The results show that important motion components can be estimated accurately if the right parameters are used. Some of the critical issues involved in the implementation of autonomous robot motion vision are also discussed. Among these are the criteria for autonomously choosing an optimum size for the fixation patch, and appropriate choice of the fixation point location. Finally, a calibration method is described for precisely determining the location of real rotation axis in imaging systems.

Key Words Active Vision, Computer Vision, Feature Correspondence, Fixation, Motion Vision, Optical Flow, Pixel Shifting, Robot Vision

چکیده نظریه «تثبیت» (Fixation) روش مستقیمی را ارائه داد که بدون استفاده از «رابطه بین نقاط ممیزه» (Feature Correspondence) یا «جریان نوری» (Optical flow)، می توان شکل و حرکت را به کمک یک سری تصاویر الکترونیکی بدست آورد [دید حرکتی ربات (قسمت اول: تئوری)]. دراین میان پارامترهائی وجود دارند که تأثیر قابل توجهی روی عملکرد روش تثبیت می گذارند. با این وجود، نظریه «تثبیت» چیزی در مورد انتخاب خودمختار (Autonomous) و صحیح آن پارامترها، بیان نمی دارد. این مقاله به توضیح اثرات آن پارامترها روی نتایج تحقیقاتی حاصل از بکارگیری بعضی از الگوریتم های تثبیت برروی تصاویر حقیقی مربوط به ترکیب حرکات دورانی و خطی می پردازد. این نتایج، نشان می دهند که درصورت انتخاب صحیح آن پارامترها، می توان مؤلفه های سرعت دورانی و خطی را بطور دقیق محاسبه نمود. بعضی از مسائل مهم مربوط به اجراء دید حرکتی خودمختار نیز دراین مقاله مورد بررسی قرار گرفته اند. دراین میان، روشهائی برای انتخاب خودمختار بهترین اندازه «ناحیهٔ تثبیت» (Fixation Patch) ارائه شده اند. در پایان نیز، روش تعیین دقیق محور دوران در سیستمهای تصویربرداری توضیح داده شده اند.

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 robot 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. Moreover, 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 [3,4]. This leads to an estimated optical flow field which may not be the same as the true motion field.

Techniques for solving the *correspondence* problem, and computing *optical flow*, have proven to be rather unstable and computationally very expensive. 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 solving special cases such as *known depth* [3], *pure translation* or *Known rotation* [5], *pure rotation* [5], and *planar world* [6]. All these direct methods are severely restricted in the types of the motion or shape that they can handle.

Recently, a general direct method called *fixation* has been introduced for solving the motion vision problem in the general case without *placing* restrictions on the motion or the shape [1,2]. The fixation method is based on a theoretical proof that for a sequence of fixated images (a sequence of images with one arbitrary stationary image point in them), the rotational velocity ω can always be explicitly expressed in terms of a linear function of translational velocity $\mathbf{t}$. Namely,

$$\omega = \omega_{\mathbf{R}_o} \widehat{\mathbf{R}}_o \frac{1}{\|\mathbf{R}_o\|} (\mathbf{t} \times \widehat{\mathbf{R}}_o). \qquad (1)$$

Where $\widehat{\mathbf{R}}_o = \widehat{\mathbf{r}}_o$ is the unit vector along the position vector of an arbitrary fixation point, an arbitrary point in the image plane chosen for fixation, and $\omega_{\mathbf{R}_o}$ is the component of rotational velocity about the fixation axis $\mathbf{R}_o$. The combination of this *Fixation Constraint Equation* (FCE) and the BCCE offers a solution to the motion vision problem of arbitrary motion relative to an arbitrary rigid environment. That is, it recovers the depth map Z, rotational velocity ω, and translational velocity $\mathbf{t}$ without putting any severe restrictions on the motion or the shape [1].

Fixation is not tracking! The *fixation method* is not only different from the previous tracking methods, but also is general. For example, Aloimons & Tsakiris [7] propose a method for tracking a target of known shape; Bandopadhay et al. [8] use optical flow and feature correspondence for tracking the principal point in order to find the motion in special case (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 [10] 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.

In contrast to these tracking methods, the *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 *fixation point* for any input sequence of images [1,2]. This is done entirely in software without any use of camera motion for *tracking*.

This work reports the experimental results of applying some of the fixation algorithms to real images where the motion is a combination of translation and rotation. Finding the fixation velocity and the component of rotational velocity about the fixation axis ($\omega_{\mathbf{R}_o}$) is one of the most important steps in the fixation method [1,2]. The results here show that the fixation velocity and $\omega_{\mathbf{R}_o}$ can be estimated accurately. Some of the crucial implementation issues of autonomous robot motion vision are also discussed here. Among those are autonomous selection of an optimum size for the fixation patch based on an error

norm called normalized error, and autonomous choice of an appropriate fixation point. Finally, an effective calibration method is described which identifies the location of real rotation axis in imaging systems.

THE EFFECT OF FIXATION PATCH SIZE

Finding the fixation velocity and the component of rotational velocity about the fixation axis, ω_{R_z}, is an important step in the fixation method for recovering the shape and motion from an arbitrary sequence of input images. Namely, in fixation method, the pixel shifting process uses the fixation velocity to construct a sequence of fixated images from an arbitrary sequence of input images, and ω_{R_z} is needed for computing the total rotational velocity [1].

The algorithms used for recovering the fixation velocity and ω_{R_z} obtain their input information from a patch around the fixation point. In order to study the effect of the fixation patch size, we have used a sequence of real images acquired at the *Imaging Laboratory of Carnegie Mellon University*. Figure 1 shows one of these 576 × 384 pixel images which have 16 bits of resolution. The camera has a nominal focal length of 24 *mm*, and pixel size of 0.02 × 0.02 *mm*. The calibrated principal point has been used as the fixation point is about 1450 *mm*.

Figure 1: An image in the sequence where the real motion is -0.3 *degrees* rotation about the optical axis Z and -2 *mm* translation along the horizontal axis X.

The real motion between these two images has both translational and rotational components. The real rotation is -0.3 *degrees* and is supposed to be about the optical axis Z. The real translation is -2 *mm* along the horizontal axis X.

Using the algorithms given in *section* 5 of [1], we can find ω_{R_z} for any given fixation patch size. Figure 2 shows that for small patch sizes (less than 50 × 50 pixels) the estimated value of ω_{R_z} is oscillating wildly and results in unacceptable ω_{R_z}'s. As the patch size increases, the estimated ω_{R_z} converges towards real value of rotation. Namely, for large patch sizes (say 100 × 100 pixels) the estimated rotation, -0.309 *degrees*, is very much near the real rotation, -0.3 *degrees*.

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, ω_{R_z}. Small patch size results in value for ω_{R_z} which is usually far larger than real values. This is possibly because in a small patch, small translations can be interpreted as large rotations. Figure 3 shows a hy-

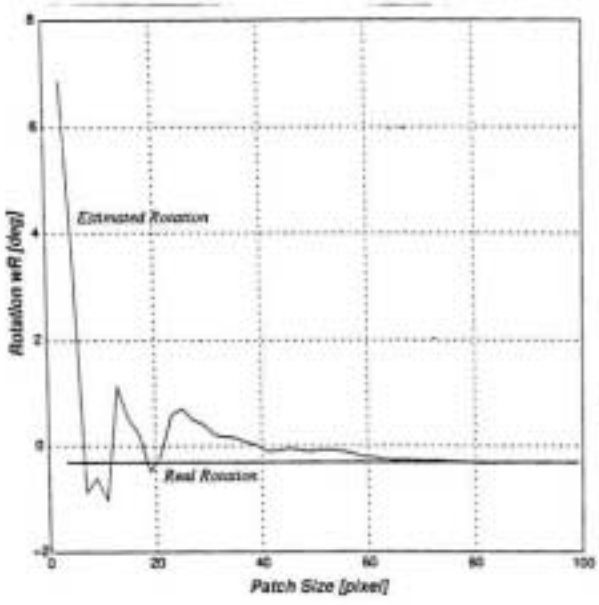

Figure 2: Estimated value of the component of rotation velocity about the fixation axis, ω_{R_z}, for different sizes of fixation patch. For large patch sizes, the estimated value of ω_{R_z} converges towards the real value of ω_{R_z}, -0.3 *degrees*.

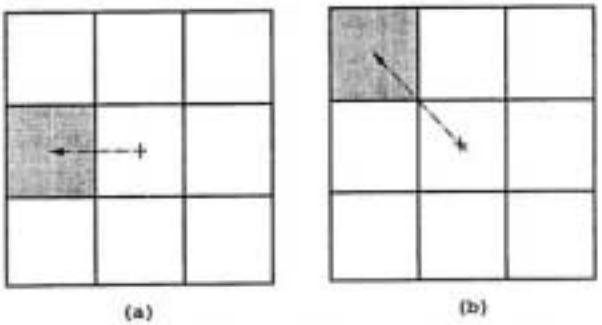

Figure 3: Using small fixation patch can result in wrong interpretation of a large rotation. In a patch of 3×3 pixel, a pixel height vertical translation can be seen as 45 *degrees* rotation which is not an acceptable answer at all, considering the finite motion between images.

pothetical situation where (*a*) and (*b*) are a sequence of a small 3×3 pixels patch. The real motion in this case is most likely a pixel height vertical translation. But if we try to interpret it as a rotation about the patch center we will end up with a 45 *degrees* of rotation which is not acceptable, considering the finite motion between images.

AUTONOMOUS CHOICE OF OPTIMUM FIXATION PATCH SIZE

The experimental results and explanations in the previous section show that relatively large patch sizes should be used in order to get a good estimate of 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 value for the fixation velocity because depth variations generally increase as the patch size increases. In this section, we will describe the solution to this problem.

Computing the Fixation Velocity

As shown in the previous section, we can find a good estimate for $\omega_{\mathbf{R}_o}$ using a relatively large patch but the corresponding fixation velocity estimate for such large patches is usually not reliable. Using only the acquired estimate of $\omega_{\mathbf{R}_o}$, we can write the motion field at any point (x, y) on a small fixation patch as

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

where (x_o, y_o) is the position of fixation point, and (u_o, v_o) is the fixation velocity that we are interested in [1]. Ideally, the BCCE must be satisfied at any point on the fixation patch as

$$\left(u_o + \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}(y - y_o)\right) E_x + \left(v_o - \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}(x - x_o)\right) E_y + E_t = 0 . \tag{3}$$

However, due to noise, the above equation does not necessarily hold at 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

$$I = \iint_P \left[\left(u_o + \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}(y - y_o)\right) E_x + \left(v_o - \frac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}(x - x_o)\right) E_y + E_t\right]^2 dx\,dy \tag{4}$$

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

$$\begin{bmatrix} \iint_P E_x^2\,dx\,dy & \iint_P E_x E_y\,dx\,dy \\ \iint_P E_x E_y\,dx\,dy & \iint_P E_y^2\,dx\,dy \end{bmatrix} \begin{pmatrix} u_o \\ v_o \end{pmatrix} = \begin{pmatrix} \iint_P \left[\dfrac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}((x - x_o)E_y - (y - y_o)E_x) - E_t\right] E_x\,dx\,dy \\ \iint_P \left[\dfrac{\omega_{\mathbf{R}_o}}{\sqrt{x_o^2 + y_o^2 + 1}}((x - x_o)E_y - (y - y_o)E_x) - E_t\right] E_y\,dx\,dy \end{pmatrix} \tag{5}$$

which we can solve for the two unknowns u_o and v_o. Note that $\omega_{\mathbf{R}_o}$ is known here.

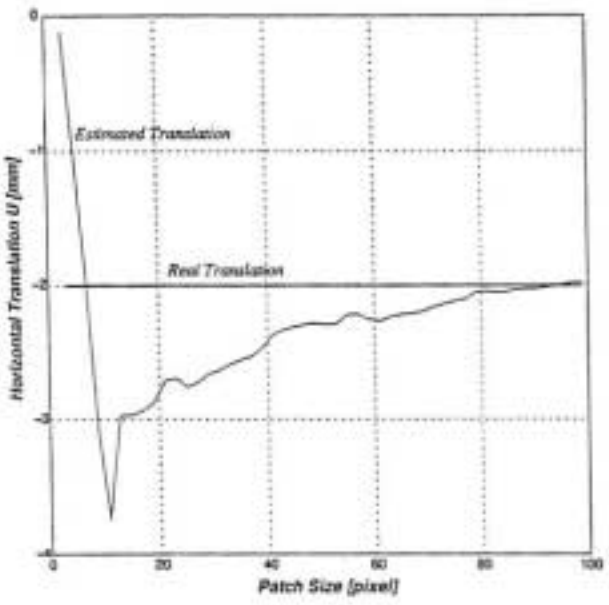

Figure 4: Estimated value of the horizontal component of translational velocity, along the X-axis versus the fixation patch size.

Figure 4 shows the estimated values of the horizontal translation $U\frac{d_{x_0}}{Z_0}$ for different sizes of fixation patch. It can be seen that U nicely converges towards the real horizontal tramslation, -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 there is no way of changing the fixation patch size by checking the computed values of fixation velocity. In order to solve this crucial problem, we should find an autonomous way of choosing an optimum fixation patch size.

Normalized Error

As shown before, for any given size of the fixation patch, we can find the fixation velocity components, u_0, v_0 and also the component of the rotational velocity about the fixation axis, $\omega_{\mathbf{R}_0}$, using a relatively large patch. Knowing these values, the image velocity (x_t, y_t) at any point (x, y) in the image plane is given by Equation 2. Ideally, for any given image point (x, y), the BCCE must be satisfied as

$$x_t E_x + y_t E_y + E_t = 0. \tag{6}$$

However, in practice we are dealing with real images which are usually noisy. As a result, the term $x_t E_x + y_t E_y + E_t$ is not zero. This term can be considered an error norm for the corresponding pixel. In a patch of size $p \times p$ pixels, we can add these error terms and define the *normalized* error as

$$e = \frac{\Sigma \left[x_t E_x + y_t E_y + E_t\right]^2}{p^2}, \tag{7}$$

This definition allows us to compare the performance of different patch sizes by studying the behavior of the normalized error e in response to the changes in the patch size p. This consideration may allow us to find an optimum patch size which results in minimum normalized error e. Figure 5 shows the corresponding normalized error as a function of fixation patch size. Usually, the normalized error first increases with patch size and reaches a peak. After dipping down, it increases again.

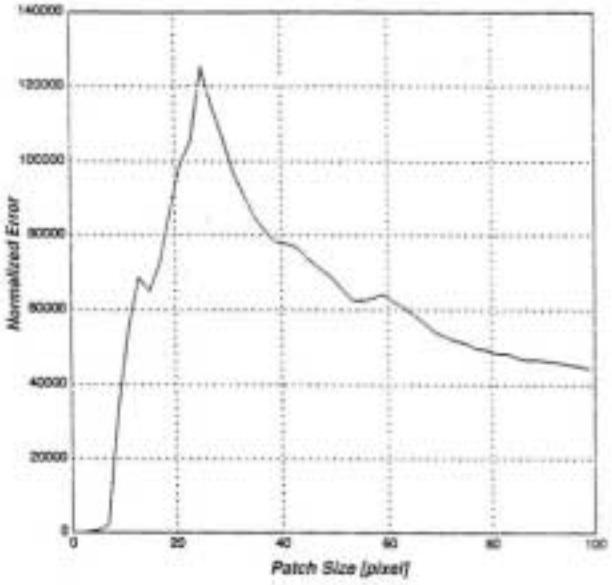

Figure 5: Estimated value of the normalized error e versus the fixation patch size.

This is because at first, the information in a small patch size is not enough to give a good estimate of the motion and this will cause the normalized error to grow. By increasing the patch size, we are increasing the amount of information available to the algorithms and this will give a better motion estimate and results in a smaller normalized error. As the patch size goes beyond some optimum size, the normalized error starts increasing. This is because usually at that stage, the depth variation in the patch increases and will give wrong fixation velocity estimates which in turn results in a large normalized error. Note that the second rising section of the normalized error plot is not shown in Figure 5 in order to keep the plots uniform in size.

As one might expect, the optimum fixation patch size depends on the patch texture which may vary from patch to patch. However, the general pattern of normalized error allows us to autonomously find an optimum fixation patch size which gives good estimates for the fixation velocity. The optimum patch size is the one that corresponds to the minimum normalized error after the first peak.

AUTONOMOUS CHOICE OF AN APPROPRIATE FIXATION POINT

In general, the 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 (0,0) makes the formulations simpler. However, in practice, one should take measures in choosing an appropriate fixation point.

Most significantly, the motion of the chosen point should be detectable in the image. To clarify this, we can consider a patch which has a uniform brightness. Choosing the center of such a patch as the fixation point will not be useful because this point may have moved and we will not be able to recover the motion only based on the information given on that patch.

For a small patch of uniform motion field, the least square method can be applied to the BCCE to obtain the following system of linear equations for the motion field (u, v) as

$$\begin{bmatrix} \iint_\rho E_x^2\,dxdy & \iint_\rho E_xE_y\,dxdy \\ \iint_\rho E_xE_y\,dxdy & \iint_\rho E_y^2\,dxdy \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -\iint_\rho E_tE_x\,dxdy \\ -\iint_\rho E_tE_y\,dxdy \end{pmatrix} \tag{8}$$

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

$$D = \left(\iint_\rho E_x^2\,dxdy\right)\left(\iint_\rho E_y^2\,dxdy\right) - \left(\iint_\rho E_xE_y\,dxdy\right)^2, \tag{9}$$

is not zero. As a result, we can define a good fixation point as a point whose corresponding patch has a nonzero $|D|$ which is the largest among all other possible choices. It is very easy to implement this criteria for autonomous choice of fixation point, and it works very well even on real noisy images.

ROTATION AXIS CALIBRATION

In our experiment, we have not explicitly applied any vertical translation (along Y-axis). However, Figure 6 shows a vertical translation of about -0.9 *mm*. This is mainly because the real rotation axis does not coincide with the optical axis[1]. At *CMU Imaging Laboratory*, the rotation mechanism is not set up to coincide the Z axis of rotation with the optical axis.

To clarify this, we should mention that in motion vision, the assumption is that the rotation axis passes through the origin of the viewer centered coordinate system. As a result, in this experiment, our algorithms only give the rotation about the optical axis Z.

[1]In general, mounted CCD at an angle may also cause such kinds of errors. But it is not the case here because the inaccuracy of motion has happened only in the vertical direction.

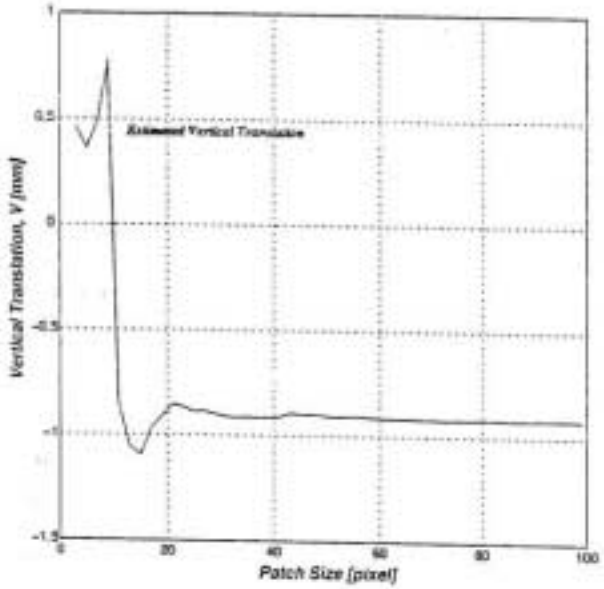

Figure 6: Estimated value of the horizontal component of translational velocity, along the X-axis, versus the size of fixation patch.

rithms only give the rotation about the optical axis Z. According to the basic kinematics, the compensating translation which results from shifting the rotation axis is given by $\mathbf{V}_0 = -\omega \times \mathrm{B}$. Where B is a vector from a point on the shifted rotation axis to a point on the real rotation axis. In our special case, $\mathbf{V}_0 = -(\omega \hat{z}) \times (b\hat{x})$. In this experiment, $\mathbf{V}_0 = -0.9\hat{y}\ mm$ and $\omega = -0.3$ *degrees*. As a result, 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 total calibration of the rotation axis which is parallel to the optical axis in any camera system arrangement. In order to find the real axis of rotation, the following steps should be taken:

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

2- Compute $\omega_{\mathbf{R}_z}$ by applying the algorithms given in section 5 of [1] using a relatively large patch.

3- Find the translational motion (u_0, v_0) at the principal point using the Equation 5.

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

$$\begin{cases} b_x = -\dfrac{v_0 f}{Z_0\, \omega_{\mathbf{R}_z}} \\ b_y = +\dfrac{u_0 f}{Z_0\, \omega_{\mathbf{R}_z}} \end{cases} \qquad (10)$$

where Z_0 is depth at the principal point, and f is the focal length. As a result, the real rotation axis is parallel to the optical axis and intersects the image plane at point (b_x, b_y).

CONCLUSIONS

The experimental results presented here show that the fixation velocity and the component of rotational velocity about the fixation axis ($\omega_{\mathbf{R}_z}$) can be accurately computed using only the information from a small patch around the fixation point. The corresponding optimum patch size in our experiment is about 100×100 pixels which results in f field of view of about 2×2.4 degrees. Recovery of fixation velocity and $\omega_{\mathbf{R}_z}$ is one of the most important part of fixation method. Obtaining such good results while using only a small patch of real image insures the feasibility of the fixation method. This is especially true if we consider that the nominal focal length and nominal pixel size are used in the computations.

Our goal and on going work is to make a stable autonomous 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 the parameters. Fixation offered a general direct method and this paper presented a technique for autonomous choice of an optimum fixation patch size and an appropriate fixation point location. These results open the road for the implementation of a fully autonomous motion vision system.

The method described for the the calibration of the real rotation axis offers a simple solution to an important implementation problem. This problem

can result in considereable error in the motion estimates if it is not detected and compensated for.

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.

REFERENCES

1. 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).
2. 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, early (1992).
3. B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificail Intelligence*, 17, (1981), 185-203.
4. E. C. Hildreth, "The Measurement of Visual Motion," MIT Press, Cambridge, Mass., (1984).
5. B. K. P. Horn and E. J. Weldon Jr, "Direct Method for Recovering Motion," *Inter. J. of Computer Vision*, 2 (1988), 51-76.
6. S. Negahdaripour and B. K. P. Horn, "Direct Passive Navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1) Jan (1987), 168-176.
7. 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).
8. 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, Kia wash Island, May 7-9 (1986), pages 23-29.
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) Jan (1990) 13-27.
10. W. B. Thompson, "Structure-from-Motion by Tracking Occlusion Boundaries," Biological Cybernetics, 62 (1989) 113-116.