Sensitivity of Structure From Motion

R. Dutta and M.A. Snyder

COINS TR90-103

October 1990

Sensitivity of Structure from Motion[1]

R. Dutta M. A. Snyder

Computer and Information Science Department
University of Massachusetts at Amherst
Amherst, Massachusetts 01003

Abstract

This paper examines the robustness of correspondence–based approaches to structure from motion. Unlike earlier studies it is proven in an *algorithm–independent* way, that small absolute errors in image displacements cause absolute errors in rotational motion parameters significant enough to lead to large relative errors in the determination of environmental depth. Even if the motion parameters are known a priori through sophisticated navigation systems, small errors in image displacements still lead to large errors in depth for environmental points whose distance from the camera is greater than a few multiples of the the total translation of the camera. Furthermore, it is shown that the widely used instantaneous velocity approximation to the image displacements leads to poor recovery of structure. In order to clarify the many issues on robustness that are raised in this paper, a new depth determination algorithm is developed and applied to dynamic image sequences of natural outdoor scenes.

[1]This research has been supported by the Defense Advanced Research Projects Agency under RADC contract F30602-87-C-0140 and Army ETL contract DACA76-89-C-0017.

1 Introduction

A large number of mathematically elegant approaches for determining structure from motion have relied on some form of point or feature correspondences between two or more perspective views [1-10]. These correspondence–based approaches take advantage of the image displacements induced by egomotion. Most such methods match a large number of points or features in two temporally separated images and quantitatively measure the image displacements. A consistent set of motion parameters is then determined to explain these displacements. Once the motion parameters have been determined, the depth of environmental points can be found by using their individual image displacements. In a few algorithms (e.g. [6]) optimal values of environmental depths and motion parameters are found concurrently.

In spite of the extensive research that has been conducted on this problem, robust depth determination under a wide variety of scene structures (especially in an outdoor environment) has remained elusive. Naturally, this has led to investigations into the reason for such failures. Algorithm–specific errors have been explored by Tsai and Huang [11], Barron [12], and Fang and Huang [9]. General methodologies for computing the extent of precision of 3–D Parameters in the case of translational motion have been explored by Snyder [13]. Large shifts in the Focus of Expansion have been shown to occur in the case of approximate translational motion when small rotations have been ignored [14,15]. Weng, Huang, and Ahuja [10] have given some qualitative conditions for better structure recovery, based primarily on simulations. Adiv [16] has characterized and demonstrated situations where inherent ambiguities exist in the interpretation of noisy flow fields. Verri and Poggio [17] have argued for using only qualitative properties of optical flow by showing that the motion field and the optical flow are in general different except under special

conditions. In addition to the above there are a large number of other serious studies of this problem (see the review in [18]). However, what is lacking is a comprehensive algorithm-independent study of the issues associated with the correspondence–based structure from motion problem. The present paper is an attempt to remedy this situation. After doing so by dealing with the theoretical issues, we formulate an algorithm which finds depths of environmental points in dynamic image sequences of outdoor scenes with about 8% average error.

The key problem is to show, in an algorithm–independent way, the effect of all relevant quantities on the computation of environmental depth. The computation of environmental depth is important because it leads to the recovery of the three–dimensional structure of the environment from the two-dimensional image. We start by using the small rotation image displacement equations and show mathematically, as well as experimentally, that small absolute errors in the determination of rotational motion parameters cause large relative errors in depth computation unless the image displacements are really very large. This is proved in an algorithm–independent way and has the immediate consequence that motion algorithms must precisely estimate rotations. However, we show that small absolute errors in the image displacements themselves give rise to rotational errors significant enough to give large errors in depth. The proofs show that the occurrence of the large errors is ubiquitous. We therefore conclude from this *theoretical* analysis that is in general difficult to secure robust depth of environmental points. We emphasize that in contrast to our algorithm–independent theoretical approach the conclusions of the earlier studies have either been based on qualitative frameworks or have been algorithm–dependent.

In addition to the above, we explore the direct relationship between small errors in the image displacement and computed relative depth. This is necessary because of suggestions that sophisticated navigation systems can give motion parameters almost exactly. It is

shown that even if the motion parameters were known exactly, small errors in computing the magnitude of the correct image displacement generally cause large errors in depth.

As stated earlier all the above analyses are done with the small rotation image displacement equations. Even though, it is common to approximate the exact image displacement by using the instantaneous velocity of brightness patterns [1,6,7,19] we do not do this because in the study of robustness issues it is a good strategy not to make any approximations as far as possible. If this is not possible then we should consider the resultant effect of all approximations made. It should be noted that under realistic conditions, quantitative depth recovery demands that rotation of a vehicle be kept small. This is because large rotations give rise to such huge image displacements that image correspondence becomes virtually impossible to establish. Besides, rotations by themselves cannot help us determine depth. Translations on the other hand are usually regulated in such a manner that they facilitate the establishment of correspondence in the scene. Large translations are required if the depths of the environmental points are large whereas small translations are required if the depths of the environmental points are small. Hence, the assumption of small rotations and arbitrary translation is a realistic one in practical situations. On the other hand we shall show that the approximation of the image displacement by using the instantaneous velocity of brightness patterns is not a suitable one for realistic scenarios.

We will refer to the common technique in which the exact image displacement is approximated by using the instantaneous velocity of brightness patterns as the *velocity formalism*, and to the small rotation displacement equation technique as the *displacement formalism*. We show mathematically that the use of the velocity formalism of the motion equations can potentially cause a very large relative error in depth for a large number of environmental locations. In addition, we provide experimental results under realistic scenarios to show that on average the velocity formalism performs poorly when compared to the

displacement formalism. That it performs worse is not unexpected because after all it is an approximation. What is unexpected is the extent to which it performs badly. This part of our work should be viewed in the context of the large number of papers using the velocity formalism to try to recover structure from motion. Also, it gives additional support to our claim that the displacement formalism, and not the velocity formalism, should be used to analyze fundamental issues in robustness. It should be noted that quite often the image displacements computed by the velocity formalism deviate significantly from the true image displacements.

Even though there are a multitude of problems associated with correspondence–based structure from motion it might still be possible to make it work in certain situations. We attempt this in sections 8 and 9 by refining the motion parameter estimates given by other algorithms. The refining is done by minimizing a new error measure which uses normalized absolute deviation in directional depths. We present experimental results clarifying the various issues involved for natural outdoor scenes with precisely known quantitative ground truth for depths.

The outline of out paper is as follows. We begin by stating the small rotation image displacement equations then use them to show that small absolute errors in rotation cause large errors in depth. This is followed by the proof that small image displacement errors causes large absolute errors in rotational parameters, leading to large errors in depth. After this we investigate the direct relationship between errors in image displacements and errors in depth when the motion parameters are known exactly. Next, we show both theoretically and experimentally how poorly the velocity formalism can perform. This is followed by experimental results showing how rotational errors affect depth computation. We then present a new algorithm for depth determination and analyze the experimental results for a natural scene. Finally, we make concluding statements based on our theoretical and

experimental results.

2 Depth from Image Displacement under Egomotion

In this section we give the equations for the displacement formalism for computing depth from motion. We assume a right handed coordinate system fixed with respect to the camera as shown in Figure 1. Let us also assume that the right hand rule is used for rotations. We consider the case where the camera is undergoing motion.

As can be seen from Figure 1 the environmental point P, with world coordinate (X, Y, Z), is projected onto point p, in the image plane with image coordinates (x, y). Let f be the focal length of the camera, and denote by $\vec{T} = (T_1, T_2, T_3)$, $\vec{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$ the translational and rotational rigid motion of the camera (This implies that $P' = RP + T$ where R is the rotation matrix and P' is the new position of P after undergoing rigid motion).

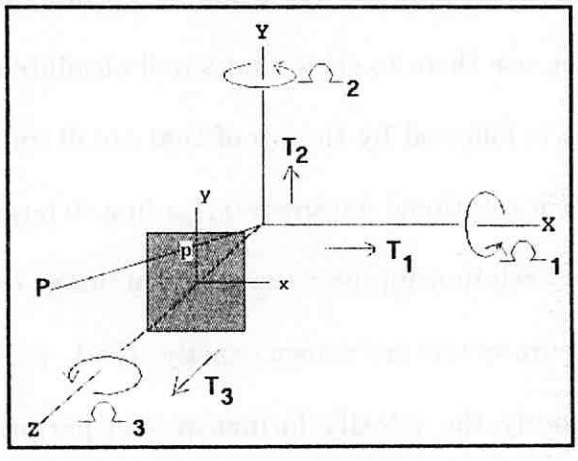

Figure 1: **Coordinate System**

We shall be working with the small rotation motion equations. For small rotations [2] $sin(\theta) \approx \theta$ and $cos(\theta) \approx 1$.

For simplicity we define the following abbreviations:

$$
\begin{aligned}
I &= 1 + \frac{\Omega_2 x}{f} - \frac{\Omega_1 y}{f} \\
J &= \left(\frac{\Omega_1 xy - \Omega_2 x^2}{f}\right) - \Omega_2 f + \Omega_3 y \\
K &= \left(\frac{\Omega_1 y^2 - \Omega_2 xy}{f}\right) + \Omega_1 f - \Omega_3 x \\
\alpha &= -fT_1 + xT_3 \\
\beta &= -fT_2 + yT_3.
\end{aligned}
$$

It should be noted from the abbreviations that I, J, and K are functions of rotational parameters whereas α and β are functions of translational parameters. With the above abbreviations the image displacement, $\vec{l}$ induced by the motion of the camera is given by

$$\vec{l} = u\,\hat{x} + v\,\hat{y} \equiv (u, v) \tag{1}$$

where $\hat{x}$ and $\hat{y}$ are unit vectors along the x-axis and y-axis respectively, and [3]

$$u = \frac{J + (\alpha/Z)}{I - (T_3/Z)} \tag{2}$$

$$v = \frac{K + (\beta/Z)}{I - (T_3/Z)}. \tag{3}$$

[2] With this approximation, for small angles the errors in the sine and cosine of θ are negligible. For example even if $\mid \theta \mid = 0.100^c$ (the superscript c denotes radians), which is as large as 5.7°, the relative error in $sin(\theta)$ is 0.167% and in $cos(\theta)$ is 0.500%. It should be noted that even though we are considering small angles in the derivations, the absolute errors in rotation we shall be considering will sometimes have a least value that is roughly 600 times smaller. Besides, it will be shown later that this really does not matter.

[3] Various equivalent versions can be found in the literature [9,20]. Usually most derivations proceed by assuming f to be 1. For convenience of implementation we have not made this assumption.

In addition, we choose units in which $|\vec{T}| = 1$, so that

$$T_1^2 + T_2^2 + T_3^2 = 1. \tag{4}$$

The depth Z of an environmental point P can be determined from either equation (2) or equation (3). We denote by Z_x the depth determined from equation (2) and by Z_y the depth determined from equation (3). Hence,

$$Z_x = \frac{T_3 u + \alpha}{Iu - J} \tag{5}$$

$$Z_y = \frac{T_3 v + \beta}{Iv - K}. \tag{6}$$

It should be noted that it is possible to write the depth in terms of the displacement vector $\vec{l}$ and the motion parameters by using equations (1), (2) and (3). Since this becomes rather cumbersome to manipulate for analyzing errors, we shall often (but not always) work with Z_x and Z_y separately. When we use Z_x the conclusions are based using the x-component of the image displacement and when we use Z_y the conclusions are made using the y-component of the image displacement. At this point it should be noted that algorithms should always try to satisfy the constraint of non-negative depths. The depth is of course expressed in units of the total translation, $|\vec{T}|$.

It is often convenient to measure image distances in pixels. The focal length measured in pixels given by

$$f = \frac{N}{2} \cot\left(\frac{FOV}{2}\right) \ pixels \tag{7}$$

where the image dimension is (N *pixels* $\times$ N *pixels*) and FOV is the field of view of the camera. We shall be using a 256×256 image with a 45° field of view often in the paper. For

our convenience let us call this the 'standard' image. The focal length for the 'standard' image is computed to be 309 pixels with equation (7).

With these preliminaries in mind we proceed to analyze the effect of various quantities on environmental depth in the next few sections.

3 Effect of Small Rotational Errors on Depth

In this section we prove mathematically that small absolute errors in computing rotations causes very large relative errors in depth.

We begin by using equation (5) and taking the partial derivative with respect to the rotation around the Y-axis, Ω_2 get [4]

$$\frac{\delta Z_x}{\delta \Omega_2} = -Z_x \left(\frac{xu}{f} + \frac{x^2}{f} + f\right) \left[\left(1 + \frac{\Omega_2 x}{f} - \frac{\Omega_1 y}{f}\right) u - \left(\frac{\Omega_1 xy - \Omega_2 x^2}{f} - \Omega_2 f + \Omega_3 y\right)\right]^{-1}. \tag{8}$$

This can be rewritten as

$$\frac{\delta Z_x}{Z_x} = -\left(f^2 + x^2 + xu\right) \left[fu + \Omega_2 xu - \Omega_1 yu - \Omega_1 xy + \Omega_2 x^2 + \Omega_2 f^2 - \Omega_3 yf\right]^{-1} \delta \Omega_2. \tag{9}$$

From equation (9)

$$\frac{\delta Z_x}{Z_x} = -G(x,y)\, \delta \Omega_2, \tag{10}$$

where

$$G(x,y) = \left(f^2 + x^2 + xu\right) \left[fu + \Omega_2 xu - \Omega_1 yu - \Omega_1 xy + \Omega_2 x^2 + \Omega_2 f^2 - \Omega_3 yf\right]^{-1}. \tag{11}$$

[4] There would have been higher order terms in the expression on the right hand side of equation (8) if we had considered higher order terms in the expansion of sines and cosines of angles. However, it can be shown that for small rotations of the camera the higher order terms contribute negligibly when compared to the terms that we have considered.

Equation (9) gives the relative error in depth, due to a small error in the rotational parameter Ω_2, at coordinate (x, y) in the image.

In section 3.1 we discuss the errors at the center of the image and in section 3.2 we derive an approximate expression for the error at all locations in the image. We begin by choosing for illustration the center of the image because the principal objects of interest are frequently projected there. Also, in applications using tracking it is a commonly used point. However, it is insufficient to show that the large errors occur only in a specific position and therefore we also search for a general solution.

3.1 Error at the center of the image

At the center of the image ($x = 0, y = 0$) the relative error in depth is found by substituting 0 for both x and y in equation (9):

$$\frac{\delta Z_x}{Z_x} = -\left(\frac{f}{u + \Omega_2 f}\right)(\delta\Omega_2). \tag{12}$$

Equation (12)indicates that we are not just considering very small rotational errors in *small* rotations because what is important is the sum of u and $\Omega_2 f$. If $(u + \Omega_2 f)$ is small, the large magnitude of Ω_2 by itself will not be able to help to reduce the relative error in depth. In any case, for practical motion analysis, the rotations cannot be very large because they would induce such large image displacements that correspondence algorithms would be unable to handle them.

In the following subsections we consider three cases:

1. $\Omega_2 \approx 0$

2. $\Omega_2 \approx -\frac{u}{f}$

3. Ω_2 is a random variable, centered at 0 with standard deviation σ.

Of the three cases the third requires some clarification. Often the true value of the rotational parameter of the moving camera is not known. That is to say there is an uncertainty (not error) associated with it. Hence this is modelled as a random variable. When we measure Ω_2 there is some error $\delta\Omega_2$, which is the difference between the experimental and theoretical values (i.e. ground truth) of Ω_2. It should be noted that we choose the distribution of Ω_2 so that it is similar to the distributions usually encountered for the true values of the rotational parameters of a moving camera [5]. On the other hand, $\delta\Omega_2$ is the error which a motion algorithm makes in finding the true Ω_2.

3.1.1 Ω_2 is close to 0

When $\Omega_2 \approx 0$ equation (12) can be written as (considering only absolute values of the error)

$$\frac{\delta Z_x}{Z_x} \approx \left(\frac{f}{|\, u \,|}\right)(|\, \delta\Omega_2 \,|) \tag{13}$$

Equation (13) gives the relative error in computed depth with respect to the absolute error in the computation of the rotation around the Y-axis. Note that since the image displacement, $|\, u \,|$ is necessarily much less than the focal length of the camera (recall that f is typically several hundred pixels), even a small error in Ω_2 results in a large relative error in depth.

In Figure 2 the results of using equation (13) to find the relative error in depth are

[5] Vehicle (not camera) rotations measured by a land navigation system on the Autonomous Land Vehicle (ALV) [21,22] strongly suggest a bell shaped distribution for the rotation R of the ALV around the axis perpendicular to its base when it is moved approximately straight ahead. Since the ALV is 16000 pounds in weight, eight wheel powered, and hydrostatically driven, it is expected that the spread of the angles would be very small on an almost level terrain. However actual data from a 30 frame sequence collected on almost level ground has the spread of R at about 2.4°. For smaller, lighter vehicles and uneven terrain the spread is likely to be even more. In our theoretical analyses, whenever we assign specific values to the parameters for illustration, we choose the spread in such a way that it becomes apparent that we are seeking some kind lower bound on the errors.

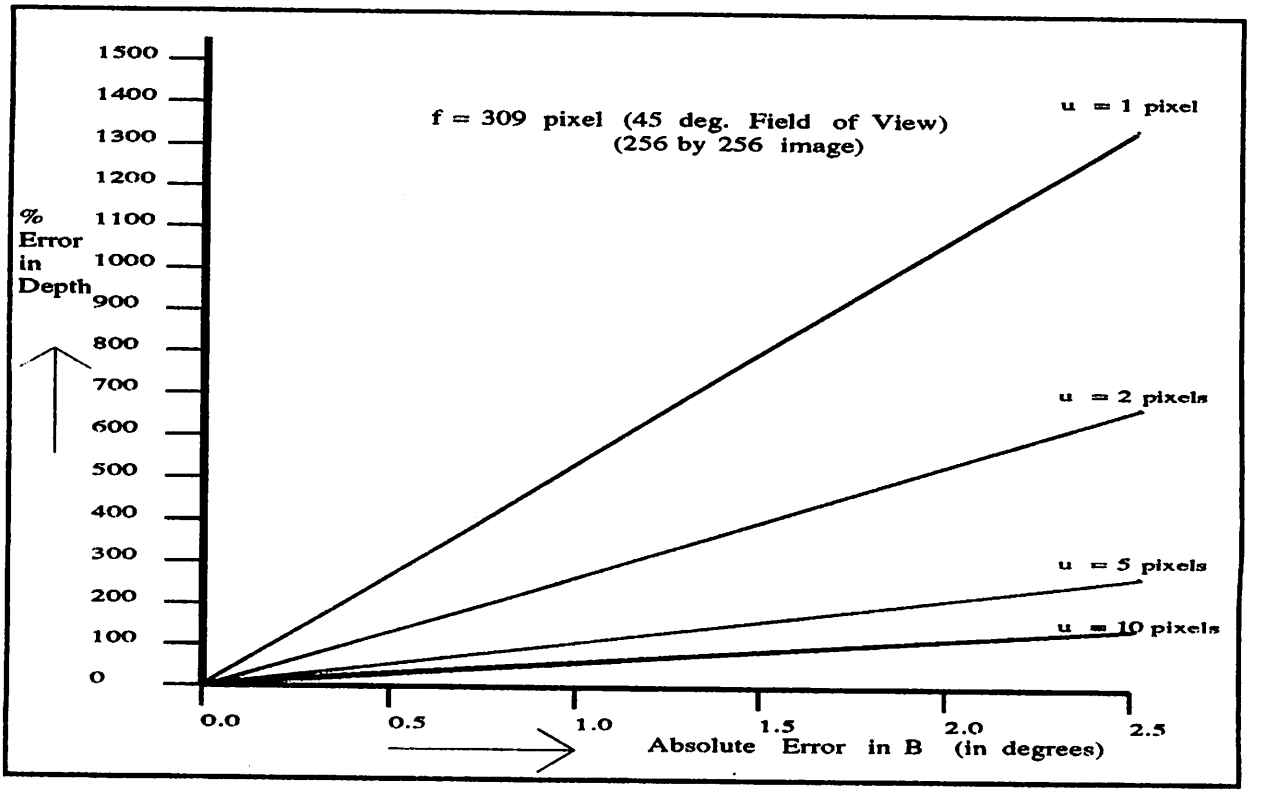

Figure 2: **Relative Error in Depth**

shown. It is remarkable to see that we get over 100% error in depth estimation when $\delta\Omega_2$ is as small as 0.5° and the displacement vector, u is 2 *pixels*. We also note that even when $\delta\Omega_2$ is 0.1° and the displacement vector, u is 2 pixels we get a 27% relative error in depth. Hence the absolute error in finding the rotational parameter Ω_2 must be very small indeed.

3.1.2 Ω_2 is nearly equal to (-u/f)

From equation (12) we see that when $\Omega_2 = -\frac{u}{f}$ the relative error in depth is infinite. This result should be treated with caution since we have originally ignored (Ω_2^2) and higher order terms in in the Taylor series expansions of sine and cosine of Ω_2. However, as shown earlier, for small angles the errors in the sine and cosine of Ω_2 are negligible. Hence we can

safely say that for a 256 $\times$ 256 image with a field of view of 45° the relative error in depth can become enormous for many combinations of values of the displacement vector, u and the rotational angle Ω_2 (e.g. $u = 5$ pixel and $\Omega_2 = -0.016\ rad. = -1°$). When Ω_2 is nearly equal to $(-u/f)$, this means that the image displacement u is nearly equal to $(-\Omega_2 f)$. Qualitatively speaking the image displacement in this case is solely due to rotation and hence depth cannot be determined. As a corollary the larger the contribution of rotation to the image displacement the larger the error in depth.

3.1.3 Ω_2 is a Random Variable

Let Ω_2 be a random variable. Then we find the probability that the relative error in depth, ξ is less than $\mid k \mid$.

With a normal distribution for Ω_2 it is not possible to get a closed form solution for the relative error in depth. Therefore we consider a different probability density function, $q(\Omega_2)$ for Ω_2 which we call the Modified Inverse Square Distribution, given by

$$q(\Omega_2) = \frac{M}{\pi} \frac{1}{1 + M^2 \Omega_2^2}. \tag{14}$$

From Figure 3 it can be seen that the modified inverse square distribution looks similar to the normal distribution. However it lacks some of the nice properties of the latter. The spread of the normal curve is controlled by the standard deviation, σ whereas the spread of the modified inverse square curve can be controlled by adjusting $\frac{1}{M}$ (i.e. large $\sigma \sim small\ M$). For our purposes either distribution for Ω_2 would be fine.

The relative error ξ in depth, is a function of the random variable Ω_2. Using equation (12) we find that

$$\xi = \frac{1}{a\Omega_2 + b} \tag{15}$$

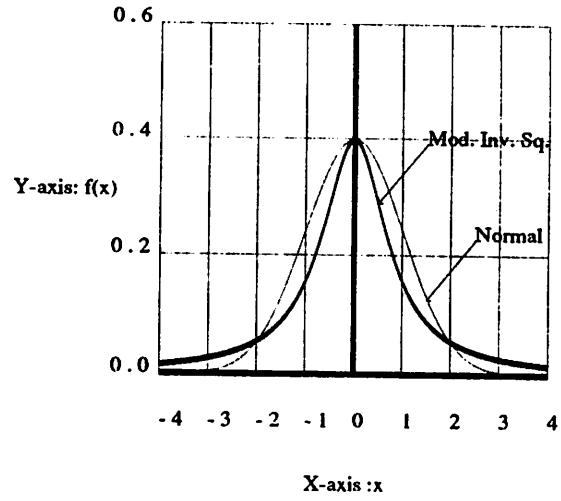

Figure 3: **Normal and Modified Inverse Square density functions.** Normal Distribution shown has mean = 0 and standard deviation = 1. Modified Inverse Square Density has M = 1.25

where

$$a = -\left(\frac{1}{\delta\Omega_2}\right), \qquad b = -\frac{u}{f\delta\Omega_2} \tag{16}$$

We can solve analytically for the the probability $P(\xi \leq \mid k \mid)$ that the relative error ξ in depth is less than $\mid k \mid$. This is shown in equation (17)

$$P(\xi \leq \mid k \mid) = \frac{1}{\pi}\left[tan^{-1}\frac{(a^2 + M^2b^2)k + M^2b}{M \mid a \mid} + tan^{-1}\frac{(a^2 + M^2b^2)k - M^2b}{M \mid a \mid}\right] \tag{17}$$

Table 1 has been constructed using equation (17)with $M = 100$. This choice of M gives a 0.94 probability that $\mid \Omega_2 \mid \leq 0.1$ *rad.*, a 0.87 probability that $\mid \Omega_2 \mid \leq 0.05$ *rad.* , and a 0.5 probability that $\mid \Omega_2 \mid \leq 0.01$ *rad.*. This kind of distribution for Ω_2 is a good model for a vehicle which is trying to move without much rotation.

From Table 1 it can be concluded that even with very small errors in rotation the relative errors in depth are very high. As an example it can be computed from the table that with an absolute error in rotation of 0.10° for displacement vectors of 5 pixel length, in 78 out of 100 cases the relative error in depth will be more than 5% , and in 45 out of

Table 1: **Probability(Relative error in depth, $\xi \leq |\ k\ |$) using Modified Inverse Square distribution with no Taylor series approximations.** The results of this table are strictly valid only for the center of the image. M = 100; 256 × 256 *image*; Field of View = 45°.

$\delta\Omega_2 = 0.50°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
1 pix.	0.01	0.04	0.07	0.18	0.34	0.56
2 pix.	0.01	0.04	0.07	0.18	0.36	0.61
5 pix.	0.01	0.04	0.08	0.22	0.55	0.83
10 pix.	0.01	0.04	0.08	0.47	0.87	0.95

$\delta\Omega_2 = 0.10°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
1 pix.	0.04	0.18	0.34	0.63	0.80	0.90
2 pix.	0.04	0.18	0.36	0.69	0.84	0.92
5 pix.	0.04	0.22	0.55	0.87	0.94	0.97
10 pix.	0.04	0.47	0.87	0.96	0.98	0.99

$\delta\Omega_2 = 0.05°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
1 pix.	0.07	0.34	0.56	0.80	0.90	0.95
2 pix.	0.07	0.36	0.61	0.84	0.92	0.96
5 pix.	0.08	0.55	0.83	0.94	0.97	0.98
10 pix.	0.08	0.87	0.95	0.98	0.99	1.00

$\delta\Omega_2 = 0.01°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
1 pix.	0.34	0.80	0.90	0.96	0.98	0.99
2 pix.	0.36	0.84	0.92	0.97	0.98	0.99
5 pix.	0.55	0.94	0.97	0.99	0.99	1.00
10 pix.	0.87	0.98	0.99	1.00	1.00	1.00

100 cases the relative error in depth will be more than 10%, and in 13 out of 100 cases the relative error will be more than 25%. For smaller image displacements like 1 or 2 pixels the results are not very good even with absolute errors in rotation of 0.01° (e.g. with an image displacement of 1 pixel, in 1 out of 5 cases the relative error is more than 5% and in 1 out of 10 cases the relative error is more than 10%).

3.1.4 Conclusions

We can safely conclude from the above analysis that small errors in estimating rotations can cause very large relative errors in depth computation. The above conclusion is strictly true for error in rotations around the x and y axes. This is because analogous results can be obtained for Ω_1 by considering Z_y. Hence the absolute error in the estimation of rotational parameters, particularly Ω_1 and Ω_2, has to be very small indeed. It should be noted that at the center of the image Ω_3 (rotation around the Z-axis) does not cause any error in depth.

3.2 General solution by approximations with Taylor Series

In section 3.1 we found the relative error in depth at the center of the image with no approximations except for ignoring second and higher order terms in the expansions of sine and cosine of small angles. It is seen from the results of section 3.1 that for small displacements the relative error in depth is unusually high. In this section we try to give somewhat more general results. It should be noted that this is much harder because in order to arrive at good approximations we have to take into account the feasible range of various parameters in typical scenes. In fact the results of this section ideally apply only in the case of relatively large image displacements.

Let $\Psi = \frac{x}{f}$ and $\Upsilon = \frac{y}{f}$. Then the function $G(x,y)$ in equation (11) can be written as

$$G(x,y) = F(\Psi,\Upsilon) = \left(1 + \Psi^2 + \frac{u}{f}\Psi\right)\left[\frac{u}{f} + \Omega_2\frac{u}{f}\Psi - \Omega_1\frac{u}{f}\Upsilon - \Omega_1\Psi\Upsilon + \Omega_2\Psi^2 + \Omega_2 - \Omega_3\Upsilon\right]^{-1} \tag{18}$$

We expand $F(\Psi,\Upsilon)$ in a Taylor series T_0 about $(\Psi = 0, \Upsilon = 0)$ and ignore terms involving second or higher order derivatives of F. This approximation means that we are omitting $O((\frac{x}{f})^2)$, $O((\frac{y}{f})^2)$ and higher order terms. This is a very good approximation in the central region of any image. Even in the periphery of almost all commonly collected images it is a very good approximation [e.g. with a Field of View of 45° for a 256×256 image, $f = 309$ *pixels*, and so $(\frac{x}{f})^2$ and $(\frac{y}{f})^2$ are about 0.17 at most, in any part of image]. Within T_0 there are some expressions which are functions of Ω_2. We expand these functions of Ω_2 in a Taylor series about $\Omega_2 = 0$ and ignore terms with second or higher order derivatives. The larger the value of u when compared to $\Omega_2 f$ the better the approximation. In any case, we had earlier omitted $O(\Omega_2^2)$ and higher terms while approximating $cos(\Omega_2)$ and $sin(\Omega_2)$ in the rotation matrices. We also omit terms which are of order greater than or

equal to $O(PQ)$ where $P \in \{\Omega_1, \Omega_2, \Omega_3\}$ and $Q \in \{\Psi, \Upsilon\}$ since they are of the second order or more in smallness. We obtain

$$F(\Psi,\Upsilon) = \left(1+\Psi^2+\frac{u}{f}\Psi\right)\left[\frac{u}{f}+\Omega_2\frac{u}{f}\Psi-\Omega_1\frac{u}{f}\Upsilon-\Omega_1\Psi\Upsilon+\Omega_2\Psi^2+\Omega_2-\Omega_3\Upsilon\right]^{-1}$$

$$\Rightarrow G(x,y) = (f^2+x^2+xu)\left(fu+\Omega_2 xu-\Omega_1 yu-\Omega_1 xy+\Omega_2 x^2+\Omega_2 f^2-\Omega_3 yf\right)^{-1}$$

$$\approx \frac{f}{u}-\frac{f^2}{u^2}\Omega_2+\frac{x}{f} \tag{19}$$

For a 256×256 image $\frac{x}{f}$ is strictly less than or equal to 0.4 whereas $\frac{f}{u}$ is about 15.4 even with a displacement, u of 20 pixels. Hence we can omit $\frac{x}{f}$ in equation(19) with negligible error. It should noted that u is much greater than $\Omega_2 f$.

Therefore, we can write

$$G(x,y) = (f^2+x^2+xu)(fu+\Omega_2 xu-\Omega_1 yu-\Omega_1 xy+\Omega_2 x^2+\Omega_2 f^2-\Omega_3 yf)^{-1} \approx \frac{f}{u}-\frac{f^2}{u^2}\Omega_2 \tag{20}$$

Using equations (9) and (20) we then can express the realtive error ξ in depth as

$$\xi = \frac{\delta Z_x}{Z_x} \approx \left[-\frac{f}{u}+\left(\frac{f}{u}\right)^2\Omega_2\right]\delta\Omega_2 \tag{21}$$

Note that equation (21) is also an approximation to equation (12) under the conditions that the magnitude of the displacement vector, u is greater than the magnitude of $\Omega_2 f$. Hence under these conditions the results of the previous section can be extended to a large part of the image. However, for ease of analysis we shall consider equation (21) rather than equation (12).

We shall consider the following cases for Ω_2: 1) Ω_2 has the modified inverse square distribution defined earlier; 2) Ω_2 has normal distribution.

3.2.1 Ω_2 with Modified Inverse Square Distribution

The modified inverse square distribution has already been defined in section 3.1.3.

Let

$$c = \left(\frac{f}{u}\right)^2 \delta\Omega_2 \tag{22}$$

$$d = -\frac{f}{u}\delta\Omega_2 \tag{23}$$

Since relative error ξ in depth is a linear function of the random variable Ω_2 [from equation (21)] it is relatively easy to arrive at the following result

$$P(\xi \leq \mid k \mid) = \frac{1}{\pi}\left[tan^{-1}\frac{M(k+d)}{\mid c \mid} + tan^{-1}\frac{M(k-d)}{\mid c \mid}\right] \tag{24}$$

Equation (24) gives the probability that the relative error ξ in depth will be less than $\mid k \mid$.

Table 2 has been constructed using equation (24). The value of M used is 100, the rationale for this being the same as in section 3.1.3. It should be noted that this table does not contain results for image displacements less than 10 pixels in magnitude as the approximations used are not valid for small displacements.

3.2.2 Ω_2 with Normal Distribution

Consider Ω_2 to be a continuous random variable following a normal distribution with mean μ_2 and standard deviation σ_2. Let us represent this information symbolically as

$$\Omega_2 \sim N(\mu_2, \sigma_2) \tag{25}$$

where

$$N(\mu_2, \sigma_2) = \frac{1}{\sigma_2\sqrt{2\pi}}\int_{-\infty}^{\infty} exp\left(\frac{-(x-\mu_2)^2}{2{\sigma_2}^2}\right) dx. \tag{26}$$

Table 2: **Probability(Relative error in depth, $\xi \leq \mid k \mid$) using Modified Inverse Square distribution with Taylor Series approximations.** M=100; 256×256 image; Field of View = 45°

$\delta\Omega_2 = 0.50°$						
displ. (abs val)	Relative error in depth, k					
	1%	5%	10%	25%	50%	100%
10 pix.	0.01	0.03	0.07	0.38	0.86	0.94
15 pix.	0.01	0.04	0.10	0.82	0.95	0.98
20 pix.	0.01	0.04	0.14	0.93	0.97	0.99
30 pix.	0.01	0.05	0.75	0.97	0.99	0.99
40 pix.	0.01	0.08	0.94	0.99	0.99	1.00

$\delta\Omega_2 = 0.10°$						
displ. (abs val)	Relative error in depth, k					
	1%	5%	10%	25%	50%	100%
10 pix.	0.03	0.38	0.86	0.96	0.98	0.99
15 pix.	0.04	0.82	0.95	0.98	0.99	1.00
20 pix.	0.04	0.93	0.97	0.99	0.99	1.00
30 pix.	0.05	0.97	0.99	1.00	1.00	1.00
40 pix.	0.08	0.99	0.99	1.00	1.00	1.00

$\delta\Omega_2 = 0.05°$						
displ. (abs val)	Relative error in depth, k					
	1%	5%	10%	25%	50%	100%
10 pix.	0.07	0.86	0.94	0.98	0.99	0.99
15 pix.	0.10	0.95	0.98	0.99	1.00	1.00
20 pix.	0.14	0.97	0.99	0.99	1.00	1.00
30 pix.	0.75	0.99	0.99	1.00	1.00	1.00
40 pix.	0.94	0.99	1.00	1.00	1.00	1.00

$\delta\Omega_2 = 0.01°$						
displ. (abs val)	Relative error in depth, k					
	1%	5%	10%	25%	50%	100%
10 pix.	0.86	0.98	0.99	1.00	1.00	1.00
15 pix.	0.95	0.99	1.00	1.00	1.00	1.00
20 pix.	0.97	0.99	1.00	1.00	1.00	1.00
30 pix.	0.99	1.00	1.00	1.00	1.00	1.00
40 pix.	0.99	1.00	1.00	1.00	1.00	1.00

From equation (21) the relative error ξ in depth is a function of the random variable Ω_2. Hence by using simple statistical theory [23] we find that ξ is also normally distributed

$$\xi \sim N\left(\frac{f^2}{u^2}\delta\Omega_2\mu_2 - \frac{f}{u}\delta\Omega_2 \ , \ \frac{f^2}{u^2}\delta\Omega_2\sigma_2\right). \tag{27}$$

Now, let,

$$\mu_\xi = \frac{f^2}{u^2}\mu_2\,\delta\Omega_2 - \frac{f}{u}\delta\Omega_2 \tag{28}$$

$$\sigma_\xi = \frac{f^2}{u^2}\delta\Omega_2\sigma_2. \tag{29}$$

To find the probability that the absolute value of the relative error ξ in depth is less than k we have to compute

$$P(\textit{relative error in depth}, \xi \leq \mid k \mid) = \frac{1}{\sigma_\xi\sqrt{2\pi}}\int_{-k}^{k} exp\left(\frac{-(x-\mu_\xi)^2}{2{\sigma_\xi}^2}\right) dx \tag{30}$$

Table 3: **Probability(Relative error in depth, $\xi \leq |k|$) using normal distribution for Ω_2 ($\mu_2 = 0$, $3\sigma_2 = 0.1$ *rad.*) with Taylor series approximations.** 256×256 image; Field of View = 45°

$\delta\Omega_2 = 0.50°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
10 pix.	0.02	0.09	0.18	0.44	0.79	1.00
15 pix.	0.02	0.12	0.25	0.72	1.00	1.00
20 pix.	0.02	0.11	0.83	0.95	1.00	1.00
30 pix.	0.00	0.10	1.00	1.00	1.00	1.00
40 pix.	0.00	0.16	1.00	1.00	1.00	1.00

$\delta\Omega_2 = 0.10°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
10 pix.	0.09	0.44	0.79	1.00	1.00	1.00
15 pix.	0.12	0.72	1.00	1.00	1.00	1.00
20 pix.	0.11	0.95	1.00	1.00	1.00	1.00
30 pix.	0.10	1.00	1.00	1.00	1.00	1.00
40 pix.	0.16	1.00	1.00	1.00	1.00	1.00

$\delta\Omega_2 = 0.05°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
10 pix.	0.18	0.79	1.00	1.00	1.00	1.00
15 pix.	0.25	1.00	1.00	1.00	1.00	1.00
20 pix.	0.31	1.00	1.00	1.00	1.00	1.00
30 pix.	0.63	1.00	1.00	1.00	1.00	1.00
40 pix.	0.97	1.00	1.00	1.00	1.00	1.00

$\delta\Omega_2 = 0.01°$						
displ.	Relative error in depth, k					
(abs val)	1%	5%	10%	25%	50%	100%
10 pix.	0.79	1.00	1.00	1.00	1.00	1.00
15 pix.	1.00	1.00	1.00	1.00	1.00	1.00
20 pix.	1.00	1.00	1.00	1.00	1.00	1.00
30 pix.	1.00	1.00	1.00	1.00	1.00	1.00
40 pix.	1.00	1.00	1.00	1.00	1.00	1.00

To compute the probabilities we use the well known error function (probability integral) $erf(x)$ [24] to rewrite equation (30) as

$$P(\xi \leq |\ k\ |) = \frac{erf(\frac{k-\mu_\xi}{\sigma_\xi\sqrt{2}}) - erf(\frac{-k-\mu_\xi}{\sigma_\xi\sqrt{2}})}{2} \tag{31}$$

There is no analytic form for $erf(x)$. However, with the help of statistical tables $P(\xi \leq |\ k\ |)$ can be computed. The results are shown in Table 3. It can be seen from the tables that even for fairly large displacements, rotational errors of the order of 0.1° or more makes it unlikely that the relative error in depth is less than 5 %. Note again that because of the approximations involved, the results of this section are not valid for small image displacements.

3.2.3 Conclusions

From this approximate analysis we can conclude that even with large image displacements we can still get moderately high relative error in depth at all positions in most images,

unless the absolute error in rotation is really quite small. We have already seen in section 3.1 that for small image displacements the relative error in depth can be very high indeed, even with very small errors in rotation.

4 Effect of Small Errors in Image Displacements on Rotational Parameters

In section 3 we found that small absolute errors in rotational parameters leads to large errors in depth. However, the question might be posed as to whether it is indeed possible to measure rotations to a very high degree of precision by any algorithm. In this section we show that it is very hard to determine rotations to the precision required in section 3 and hence it is difficult for any algorithm to determine depths. .

To show this let us consider the center (0,0) of the image. Then using equation (2) we can write the rotational parameter Ω_2 as a function of the x-component of the image displacement, u

$$\Omega_2 = -u\,(\frac{1}{f})\,(1 - \frac{T_3}{Z}) - \frac{T_1}{Z}. \tag{32}$$

To first order in the uncertainty δu, the error $\delta\Omega_2$ in Ω_2 is given by

$$|\,\delta\Omega_2\,| \;=\; |\,-\frac{1}{f}(1 - \frac{T_3}{Z})\,|\,|\,\delta u\,|\,. \tag{33}$$

Since $(T_3/Z) \ll 1$ for almost all scenes, we can write equation (33) as

$$|\,\delta\Omega_2\,| \approx \frac{1}{f}\,|\,\delta u\,|\,. \tag{34}$$

Quantization of the image plane implies that δu is at least of order 1 pixel. In this case, for a 256×256 image with a field of view of 45°, $\delta\Omega_2$ can be calculated from equation

(34) to be 0.19°. Even if we did not make the assumption that $(T_3/Z) \ll 1$ and let (T_3/Z) be $\frac{1}{4}$, which is a large translation indeed for most motion algorithms, $\delta\Omega_2$ is still 0.14° under the above conditions. In section 3 we have seen that even 0.14° absolute error in rotation can cause difficulties in depth determination both at the origin and elsewhere unless the image displacements are really very high. Hence small absolute errors in image displacements cause significant absolute errors in rotational parameters. The assumptions for our conclusion are : (a) image displacements are not measured to subpixel precisions; (b) f is of the order of hundreds of pixels or less.

In this context it should be mentioned that equation (34) suggests that the effect of uncertainties in image displacements on rotational motion parameters can be reduced by having a larger focal length. However it is known [16] that a large field of view is necessary to facilitate the determination of motion parameters. However, for the same image resolution a large field of view means that the focal length is small. Hence it does not seem likely that this problem can be surmounted.

5 Effect of Small Displacement Errors on Depth

From sections 3 and 4 we conclude that unless the image displacements are extremely large (of the order of at least 20 pixels), it is not possible to compute depth robustly with the correspondence-based approach. Recently, some researchers have tried to avoid the issue of computing correspondence-based egomotion parameters by incorporating sophisticated navigation systems into their vehicle. However, this section shows that small absolute errors in image displacement can cause significant errors in depth even if the motion parameters are known exactly. Obviously, imprecisely known motion parameters would only increase these errors.

Taking partial derivative of equation (1) with respect to Z, we find that

$$\left|\frac{\delta \vec{l}}{\delta Z}\right| = \frac{1}{(IZ - T_3)^2}\sqrt{(\alpha I + JT_3)^2 + (\beta I + KT_3)^2}. \tag{35}$$

This implies that

$$\frac{|\,\delta Z\,|}{Z} = \frac{(IZ - T_3)^2}{Z}\frac{1}{\sqrt{(\alpha I + JT_3)^2 + (\beta I + KT_3)^2}}\,|\delta \vec{l}|\,. \tag{36}$$

Equation (36) expresses the relative error in depth as a function of depth, the motion parameters, and the uncertainty in the magnitude of the displacement vector.

To understand equation (36), let us consider the most frequently attempted situation in motion – a vehicle moving straight ahead along the optical axis with no rotations [6]. Of course, this is an idealized situation and is not realizable in practice without good instrumentation. Anyway, in this case the only non–vanishing motion parameter is $T_3 = 1$. Hence, equation (36) can be rewritten as

$$\hat{\xi} = \frac{|\,\delta Z\,|}{Z} = \frac{(Z-1)^2}{Z}\frac{1}{\sqrt{x^2 + y^2}}\,|\delta \vec{l}| \tag{37}$$

To illustrate the relative error in depth given by equation (37) for a 256×256 image with field of view $= 45°$, we have in Figure 4 used shading to indicate regions of the image in which the relative error $\hat{\xi}$ in depth lies in certain ranges, at a uniform depth of $Z = 10$ (i.e., the depth is ten times the total translation). We have considered that the uncertainty in the magnitude of the displacement vector is the minimum of just 1 pixel (digitization

[6] Although approximate motion of this type is frequently used in experimental setups this causes the image displacements to be the smallest. For depth determination it is sometimes better for the camera to move sideways rather than ahead. In some sense at least binocular stereo is better than this kind of motion for depth determination.

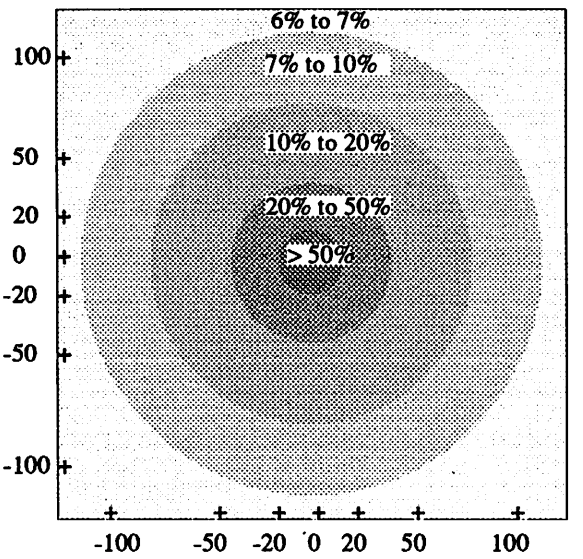

Figure 4: **Relative Error in depth marked in different regions of a 256×256 image. Uncertainty in displacement vectors = 1 pixel; Field of view = 45°; The only camera motion is translation along the optical axis and $(T_3/Z) = 0.1$.**

uncertainty) [7]. From the figure it can be concluded that the relative error in depth can be rather high for a very large portion of the image with environmental points just ten times as far away as the total translation. Measurements to subpixel accuracy and higher resolution images can often reduce the error as can be observed by comparing the errors in Figure 4 and Figure 5.

6 Errors caused by the Velocity Formalism

The velocity formalism [6,7,19] is used widely in motion analysis as an approximation to the displacement formalism. This simplifies the rather long expressions that occur in the displacement formalism. In the velocity formalism the equation for the approximate image displacement $(\breve{u}, \breve{v})$ is given by

[7]Subpixel accuracy is possible in some cases through interpolation. However even in such cases it is rather hard to *guarantee* that the error in all situations will be less than 1 pixel. In any case even with a 0.5 pixel uncertainty in the magnitude of the displacement vector the errors shown would just be halved.

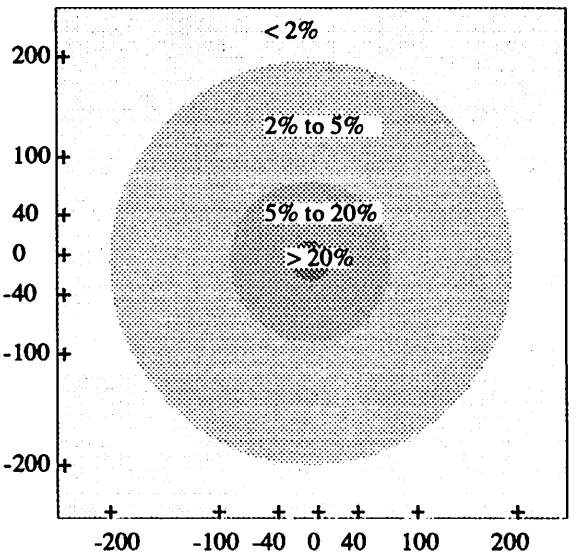

Figure 5: Relative Error in depth marked in different regions of a 512 × 512 image. Uncertainty in displacement vectors = 0.5 pixel; Field of view = 45°; The only camera motion is translation along the optical axis and $(T_3/Z) = 0.1$.

$$(\breve{u}, \breve{v}) = (u_t, v_t) + (u_r, v_r) \tag{38}$$

where

$$u_t = (\alpha/Z) \tag{39}$$

$$v_t = (\beta/Z) \tag{40}$$

$$u_r = J \tag{41}$$

$$v_r = K \tag{42}$$

In this section we first prove that the velocity formalism can lead to a very large relative error in the recovery of depth for a large number of environmental locations. Then we give experimental evidence to show how much worse the velocity formalism performs on average, when compared to the displacement formalism.

6.1 Theoretical Possibility of Large Errors in the Velocity Formalism

From equations (38), (39) and (41) we can get the computed depth, Z_x as

$$Z_x = \frac{\alpha}{\breve{u} - J} \tag{43}$$

Similarly from equations (38), (40) and (42) we can get computed depth Z_y as

$$Z_y = \frac{\beta}{\breve{v} - K} \tag{44}$$

Since the image displacement computed by the velocity formalism is not the true displacement we have denoted it by $(\breve{u}, \breve{v})$. The true image displacement is given by the displacement formalism and is (u, v). Hence we can write equations (43) and (44) as

$$Z_x = \frac{\alpha}{u - J} \tag{45}$$

$$Z_y = \frac{\beta}{v - K} \tag{46}$$

Without loss of generality we consider only Z_x.

We see that the velocity formalism computes depth to be Z_x whereas the real depth is Z. The relative error E in depth computation is given by

$$E = \frac{Z - Z_x}{Z}, \tag{47}$$

which when combined with the expression for Z_x from Equation (45) gives

$$E = \frac{Z - \frac{\alpha}{u-J}}{Z} \tag{48}$$

Using equation (2) to substitute for u in the above equation we find

$$E = 1 - \frac{\alpha\,[I - (T_3/Z)]}{\alpha + J(1-I)Z + JT_3} \tag{49}$$

It is easy to see from equation (49) that there is an infinite relative error E in computed depth when

$$\alpha + J[(1-I)Z + T_3] = 0, \tag{50}$$

which upon substituting for J and I becomes

$$-fT_1 + xT_3 + \left(\frac{\Omega_1 xy - \Omega_2 x^2}{f} - \Omega_2 f + \Omega_3 y\right)\left[\left(\frac{\Omega_1 y}{f} - \frac{\Omega_2 x}{f}\right)Z + T_3\right] = 0. \tag{51}$$

Equation (51) as the condition under which we get an infinite relative error in depth. It is obvious that for any given set of motion parameters and focal length there are potentially a large number of combinations of image position and true depth which satisfy equation (51). We shall just give an example to show that even under the conditions under which the so called velocity equation is supposed to be an approximation to the true motion equation there could be an infinite error in depth.

Consider the following parameters

- Focal Length, $f = 309$; Field of view $= 23.4°$;
- Image Coordinates $(x, y) = (5, 60)$;
- Motion parameters are:

  $(T_1, T_2, T_3, \Omega_1, \Omega_2, \Omega_3) = (0.10, 0.30, 0.95, 0.015\ rad., -0.012\ rad., 0.012\ rad.)$;
- True depth, $Z = 1589$ units;
- Total translation $= 1$ unit.

We get an infinite error in depth (ignoring numeric errors) with the above values for the parameters.

From the above example we see that even when the commonly accepted conditions [6] for the near equivalence of the the velocity and displacment formalisms are satisfied viz.

1. Rotations are small (less than 1° about each axis)

2. Field of view of the camera is small (23.4°)

3. (T_3/Z) is small (0.0006)

we can still get infinite relative error in depth using the velocity formulation. Note that (x, y) is nowhere close to the "Focus of Expansion $(fT_1/T_3, fT_2/T_3) = (31, 98)$" [8] in the image. Furthermore there are indeed an infinite number of such sets of parameters where the relative error in depth can be very large. It should also be noted that the net effect of using the velocity formalism is to approximate the denominator on the right hand side of equations (2) and (3) as 1. Equations (2) and (3), as we know, state how the image displacements are computed with the displacement formalism.

6.2 Experiments on Velocity Approximation

Section 6.1 showed that under certain conditions there exists the possibility of very large relative errors in depth (compared to the displacment formalism) when the velocity formalism is used. In light of this, experimental results comparing the relative errors in depth for the velocity and displacement formalism are shown in Table 4 (Further details are given in Appendix A.).

From the experimental results in Case (1) it can be concluded that in general approximate motion equations based on the velocity formalism can give rise to about 10% to 30%

[8] as regard the velocity formulation

Table 4: **Comparison of the velocity and displacement formalisms.** The entries in the tables show the mean and standard deviation of the percentage error in the computation of environmental depth by the two methods.
$f = 309$ pixels ; i.e. field of view = 45°.
Total translation = 1 unit. T_3 is computed from T_1 and T_2.
50 random environmental points projected on a 256×256 grid are considered.
u, v (flow vectors) computed till 4 places after decimal (rounded).
depth range refers to the range in the depth of environmental points.

Case 1 Unrestricted known motion

$(T_1, T_2, T_3) = (0.100, -0.100, 0.990)$; $(\Omega_1, \Omega_2, \Omega_3) = (0.3°, -0.7°, 0.4°)$.

Depth Range	Displacement formalism		Velocity formalism	
Lower - Upper	Average	Std. Dev.	Average	Std. Dev
5 - 10	0.00	0.00	17	19
11 - 15	0.00	0.01	13	23
16 - 20	0.00	0.00	12	25
31 - 40	0.00	0.01	11	28
91 - 100	0.01	0.06	13	39
991 - 1000	0.07	0.15	29	97

Case 2 Moving straight ahead with no rotation. Only $T_3 = 1$ is nonzero.

Depth Range	Displacement formalism		Velocity formalism	
Lower - Upper	Average	Std. Dev.	Average	Std. Dev
5 - 10	0.00	0.00	14	3.7
31 - 40	0.00	0.01	2.8	0.21

Case 3 Motion in image plane. Both formalisms become identical here.

$(T_1, T_2, T_3) = (0.707, 0.707, 0.000)$; $(\Omega_1, \Omega_2, \Omega_3) = (0.0°, 0.0°, 0.5°)$.

Depth Range	Displacement formalism		Velocity formalism	
Lower - Upper	Average	Std. Dev.	Average	Std. Dev
5 - 10	0.00	0.00	0.00	0.00

relative error in computed depth just by virtue of the approximation and nothing else. The exact displacement equations for motion rarely give any error (except at very distant points because of very small displacement vectors) if u and v are computed accurately enough. It should be noted that the major contributory factor to the error is not just (T_3/Z) because even for small (T_3/Z)'s the error is significant. The experiments support our conclusion

made in section 6.1 that the velocity formalism can indeed cause large errors. Case (2) isolates the effect of approximating only (T_3/Z) because Ω_1 and Ω_2 are both known to be zero. The error in depth for the velocity formalism is somewhat lesser in this case. In Case (3) the two formalisms are identical and hence give the same results. However the motion is quite restrictive for this to occur (no forward motion and no rotations around the x and $y-$axis). From the results it is apparent that the velocity formalism is a bad approximation even with a field of view as small as 45°.

7 Experiments on the Effect of Rotational Errors

In section 3 we theoretically studied the effect on depth recovery of small errors in rotational parameters. In order to study experimentally the effect of small perturbations of rotational parameters we generated true image displacements at various image locations with the help of the known motion parameters and known depths at these locations. Then from the image displacements we computed depth at each of the locations with the motion parameters perturbed from their true value and found the error in depth determination. That is to say we knew the motion parameters apriori so we compared the depths obtained by using the right motion parameters with those obtained by using wrong motion parameters. The wrong motion parameters were obtained by perturbing the right motion parameters. The results are shown in Table 5.

It can be seen from Table 5 that at smaller depths errors in rotational parameter have less effect. However even when depths are in the range of 5 to 20 units and total translation is 1 unit we get over 100% error when the rotational parameters are changed by 0.5°. From the table it looks as if only rotational error of the order of 0.01° can be reasonably safely permitted even if the depths are small.

Table 5: **Relative error in depth vs. Error in rotational parameters**
$f = 309$ pixels ; i.e. field of view = 45 deg.
Total translation = 1 unit
500 random points on a 256×256 grid
u, v (flow vectors) computed till 4 places after decimal (rounded)
T_3 is computed from T_1 and T_2.

Case 1 Depths in range 5 to 200 units

Error in $\Omega_1, \Omega_2, \Omega_3$ (deg.)			% Rel. Error	
$\delta\Omega_1$	$\delta\Omega_2$	$\delta\Omega_3$	Average	Std. Dev.
0.5000	0.5000	0.5000	240	1600
0.1000	0.1000	0.1000	1800	40000
0.0100	0.0100	0.0100	37	220
0.0010	0.0010	0.0010	3.9	33
0.0001	0.0001	0.0001	0.28	0.98
Pure Translation				
-0.3000	0.7000	-0.4000	150	520

Case 2 Depths in range 5 to 20 units

Error in $\Omega_1, \Omega_2, \Omega_3$(deg.)			% Rel. Error	
$\delta\Omega_1$	$\delta\Omega_2$	$\delta\Omega_3$	Average	Std. Dev.
0.5000	0.5000	0.5000	150	440
0.1000	0.1000	0.1000	45	390
0.0100	0.0100	0.0100	3.4	12
0.0010	0.0010	0.0010	0.34	1.3
0.0001	0.0001	0.0001	0.02	0.12
Pure Translation				
-0.3000	0.7000	-0.4000	180	880

8 Depth Determination by Normalizing Absolute Deviation in Directional Depths.

In the earlier sections we have discussed the limitations of the correspondence–based approaches to depth determination. Even though the analysis presents a rather gloomy outlook for the success of depth determination algorithms the problem still remains important. Hence, we shall develop a depth determination algorithm which attempts to solve this problem within the theoretical limitations we have discussed. The major steps of our

algorithm are described below.

Correspondence stage - In the correspondence stage we determine interest points in the first frame by using the Moravec operator. Then the hierarchical correlation matching approach of Anandan [25] is used to determine the image displacements of these points with respect to the temporally displaced frame. Match reliabilites are also generated.

Rough motion parameter determination - Once the image displacements have been found we determine the rough motion parameters (to be refined later) by a direct minimization of the deviation between the actual and predicted image displacements using the method proposed by Adiv [6]. Adiv's algorithm uses the velocity formulation of the motion equations and, according to our analysis in section 6 can give only approximate results. However it is quite fast (a couple of minutes on a VAX-750) and hence it is used for finding the rough motion parameters (Further details are given in Appendix B.).

Precise motion parameter determination - Our theoretical analysis has showed that depth determination is very sensitive to errors in rotational parameters. Algorithms should try to reduce this error as much as possible. Hence, after obtaining a rough estimate of the motion parameters we must apply a *good optimizing criterion* to refine the estimate. The procedure is described below.

From equations (5) and (6) we can see that there are two ways of determining depth for a particular set of motion parameters and image displacement. Z_x is the depth that is determined from the x-component of the image displacement and Z_y is the depth that is determined from the y-component of the image displacement. Hence, one of the ways of refining the motion parameter estimates is to minimize the following

$$\sigma = \frac{\sum_{i=1}^{n} \frac{U(Z_{x_i}, Z_{y_i})}{\sqrt{Z_{x_i}^2 + Z_{y_i}^2}}}{n} \tag{52}$$

where

$$\begin{aligned} \mho(x,y) &= |\, x+y \,| \quad if\ x \leq 0\ and\ y \leq 0 \\ &= |\, x-y \,| \quad otherwise \end{aligned} \tag{53}$$

and n is the total number of displacement vectors. Z_{x_i} and Z_{y_i} are the depths computed by using the x and y-components respectively of the i^{th} displacement vector and the hypothesized values of the motion parameters.

Unlike other approaches this optimization criterion allows us to avoid the rather difficult problem of eliminating the depths from the error functions (which has to be done somehow when the deviation between the actual and predicted image displacements has to be minimized). We can call the error function σ, given by equation (52) as the *normalized absolute deviation in directional depths.* The nice property of σ is that it varies between 0 and $\sqrt{2}$. The lower the value of σ the better the minimization.

Qualitatively this error function tries to minimize the difference in the depths computed by the x and y-components of the image displacements. In a noiseless image with exact motion parameters the two would obviously be equal. Note that when any of the depths are negative the contribution to the error is quite high. Obviously this error function could be used to determine the rough motion parameters too. However, since there already exist reasonably quick methods (although search-based) like that of Adiv [6] to do this, we do not use the error function σ, to compute the rough motion parameters. Rather, after the rough motion parameters have been determined we use simulated annealing [26,27] around the neighborhood of the the rough motion parameters to get the true motion parameters. Since, having a good idea of the neighborhood speeds up the simulated annnealing process considerably (about 4 minutes in our experiments on a VAXstation-3100), we use the estimates of approximate motion parameters obtained in the previous stage as the starting

point for the annealing process. Hence unlike most other annealing algorithms our annealing part of the algorithm is not slow (Further details on annealing are given in Appendix C.).

Depth determination Once the motion parameters have been found we recover the depth at every point, Z_μ by computing

$$Z_\mu = \frac{Z_x + Z_y}{2} \tag{54}$$

It fairness it should be stated that Z_x and Z_y could have different errors associated with them because of the digitization process. From the practical point of view we find that this constrains the motion to be towards the scene if the algorithm has to work well. However, the nice part is that it is possible to arrive at an estimate of the *reliability* of Z_μ by noting the difference of Z_x and Z_y from each other. The reliability ζ is defined as

$$\zeta = \frac{\mho(Z_x, Z_y)}{\sqrt{Z_x^2 + Z_y^2}}. \tag{55}$$

The reliability scale varies from 0 to $\sqrt{2}$ with 0 being the best reliability. Qualitatively, we are testing how well the depths given by the two components of the displacement vectors match.

9 Experimental Results on an Outdoor Scene

In sections 3 – 5 we discussed the limitations of the correspondence-based approaches to depth determination. Following this in section 8 an algorithm was designed for determining depth. In this section we present the results obtained by applying our algorithm on a pair of temporally separated images [9].

[9] For the image sequence we have used, extensive experimental results with several other motion algorithms can be found in [14,15].

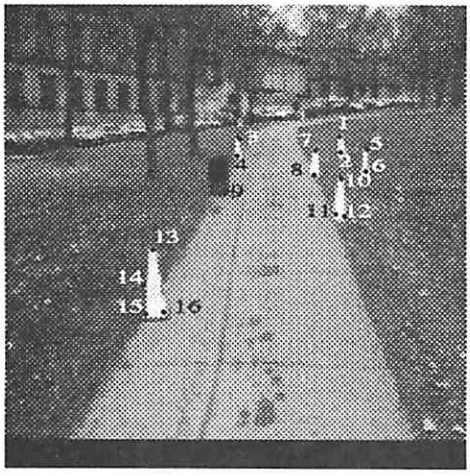

Figure 6: **First frame of the Sequence.** Points with known depth are labeled.

The first image of the sequence we used is shown in Figure 6 (More details can be found in Appendix D.). It shows the environmental points whose depths had been quantitively measured. Correspondences were established for 100 points between the first and the second frame.The displacement vectors are shown in Figure 7. After estimating the rough motion parameters with Adiv's algorithm we used simulated annealing to determine the precise motion parameters. The experimental results for motion parameters obtained by using the image displacements shown in Figure 7 are shown in Table 6. It is seen that the refined motion parameters differ from the rough motion parameters (as much as 0.13° for Ω_3 and 0.05° for Ω_2).

The experimental results from this algorithm for the depth of environmental points, found by using the motion parameters of Table 6 are shown in Table 7. It is noted that in general the results of depth determination do improve after the refining stage. The average error in depth reduces from 12.9% to 7.6% after our algorithm is applied. While it might be thought interesting to emphasize the average figure for the relative error in depth it would be misleading to do so because it would be an image dependent quantity related

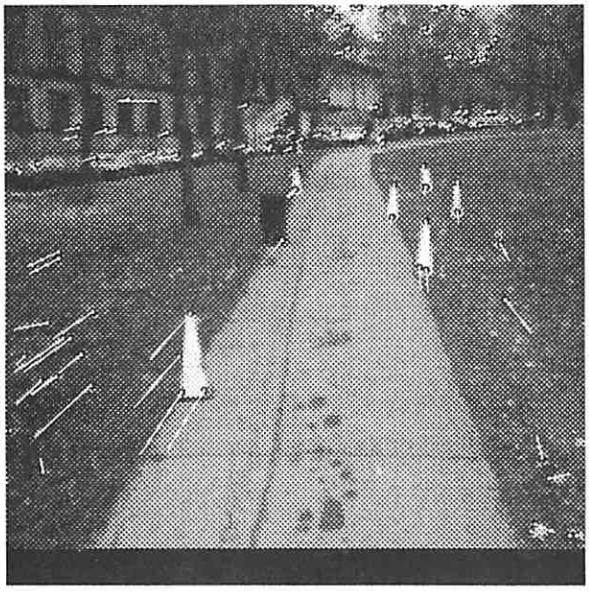

Figure 7: **Displacement vectors for first and third frame shown superimposed on the first frame.**

Table 6: **Motion Parameters obtained between Frame 1 and Frame 3.** (T_1, T_2, T_3) is the unit translational vector and $(\Omega_1, \Omega_2, \Omega_3)$ are the rotational components in degrees.

	Before Refining	After Refining
T_1	0.09	0.063
T_2	0.25	0.247
T_3	0.96	0.967
Ω_1	-0.19	-0.22
Ω_2	0.39	0.44
Ω_3	-0.30	-0.17

to the points chosen and the objects present. Based on experimentation, for objects like cones a *very rough* estimate of 8% relative error in depth in scenes like this can be given. In general the relative error is greater for larger depths. The exact figure for the relative error is less important than the insights we can get into the problem by analyzing the results in detail. For one thing, the value of σ after minimization is 0.47 (More details can

Table 7: **Depth Values of some environmental points.** The depths are in feet.

Object	pts.	Experimental depths (feet)		
		pref-ref	post-ref	true depth
cone1	1	66	77	76
	2	67	79	76
cone2	3	61	65	76
	4	61	66	76
cone3	5	50	56	56
	6	51	59	56
cone4	7	59	69	56
	8	46	58	56
can	9	44	46	46
cone5	10	31	38	36
	11	31	41	36
	12	32	41	36
cone6	13	18	22	21
	14	18	22	21
	15	19	22	21
	16	19	22	21

be found in Appendix E.). This is extemely high when compared to noiseless synthetic inputs, where $\sigma \approx 0$ after minimization. This means that there are a large number of noisy displacement vectors.

Also, Table 7 does not present the whole truth. The results are no doubt spectacular for the environmental points with quantitatively known depths but when we look at the depths of other environmental points in Figure 8 we see quite a few scattered negative values and a number of obviously wrong values (this can be qualitatively determined by looking at the scene). Of course, while looking at Figure 8 we should ignore the depths at the periphery of the image because the peripheral points may not be present in both frames. By looking at the fontsize (i.e. size of the digits) [10] of the depth values in Figure 8 we can conclude that the confidence ζ, is indeed successful in some cases to give very low confidences to erroneous depths (as for example in the bottom right of the image). Of course other factors like length of displacement vectors and image displacement confidences can also be considered. If we considered length of vectors most of the points at the top of

[10]The larger the fontsize in the picture the more reliable the depth of the corresponding point (i.e. the smaller is ζ).

the image would be less reliable as their displacements are small.

It should be noted that although large translations and hence large image displacements give reasonable depths of some points, it is not possible to do so in situations where the correspondences cannot be found well. The corners of cones are easy to match, but other features are much harder to match. Without other knowledge this will cause difficulties for autonomous navigation. The depth of 22 feet shown in the top left region of Figure 8 is so erroneous that it might lead to the conclusion that there is a nearby object. Occlusion could cause severe correspondence errors. Note further that in this scene we have been able to move four feet forward on the road because there are hardly any interest points in the image at nearby locations. If the Moravec operator had responded to nearby points, the corresponding image displacements would have been extremely high and most correspondence algorithms would have found matching difficult. Note also how the use of the more exact equations for image displacements have led to a large change in the computed depths after refining. This supports our contention that the relative error in depth is affected greatly by small errors in image displacements.

Having taken into account the theoretical limitations of correspondence algorithms the results on the natural scene seem very good. Obviously, there is enough scope for speedup through the design of more elegant algorithms. However, the possibility of getting much better depth through correspondence seems remote. Of course, it is possible to restrict the problem even further by considering feature-rich domains and using lines, areas and other higher level structures to compute depth.

10 Conclusion

We have analyzed of the important factors affecting the computation of environmental depth in correspondence–based methods. Our study of how an actual method works on a natural scene has provided us with additional insight. Based on all this we arrive at the

Figure 8: **Depths found at all Moravec points in the first frame.** A few points have been deleted or shifted slightly for clarity. The better the reliability (ζ) of the depth values the larger the size of the font in which the depth is typed. Ground truths are known for the depths typed in black.

following conclusions :

[**Effect of small rotations** -] Small absolute errors in rotations (even of the order of 0.1°) cause very large relative errors in depth determination. This result is algorithm-independent. The errors can be reduced by increasing the magnitude of the image displacements (but this has to be balanced by the fact that this makes correspondence more difficult). Also, one could try to find the motion parameters very precisely as we have done in our experiment with the natural scenes.

[**Effect of errors in image displacements on rotational parameters** -] Small absolute errors in image displacement cause significant (enough to cause large errors in depth) absolute errors in rotational parameters. Although a larger focal length can help in reducing the error, a larger focal length implies a smaller field of view (if the image resolution is kept constant) and this makes precise determination of motion parameters difficult.

[**Effect of errors in image displacement on depth** -] Even if the exact motion parameters are known, depth determination is sensitive to the the error in image displacement except for very nearby points. Hence, the solution is to have large camera translations.

[**Effect of the velocity approximation** -] The velocity formalism cannot be used to recover depth of environmental points robustly even when the motion parameters and image displacements are known. To recover precise depths when the motion parameters and image displacements are known the true displacement equations should be used.

[**Wrong matches, occlusion, interest points** -] Occlusion is common in natural scenes. As it can be seen from our results, correspondence–based methods find this very difficult to handle. Of course, we hasten to add that this applies strictly only to our correspondence algorithm. However, correspondence algorithms in the literature [28,29] cannot do this well either. There are other reasons for wrong matches, including the presence of noise and the inability of any algorithm to handle all situations.

[Time and space complexity] From our experiments with natural scenes it is clear that good depth determination is computationally time consuming. The algorithm takes a couple of hours on a VAX/750. The correspondence stage takes up over 90% of this time. Even with a fine–grained multiprocessor like the Connection Machine or a coarse–grained multiprocessor like the Sequent the correspondence stage still takes of the order of ten to fifteen minutes. The algorithms can be sped up somewhat at the expense of robustness. However, without hardware implementations it seems difficult to achieve the goal of real-time motion analysis. Our theoretical analysis implies that results can improved with high resolution images. However high resolution images would increase both the amount of space and time required to solve the problem.

Our results emphasize the point that depth computations are highly sensitive to image displacement measurements. In a related sense, these results imply that efforts to incorporate hardware to find motion parameters in moving vehicles will not determine depth accurately unless these devices have a very small absolute error in rotation (with sophisticated navigation systems this seems feasible) *and* the image displacements are computed with very small error. With correspondence–based methods and currently available hardware, robust depth determination in unrestricted domains does not seem feasible. However, as our experimental results on natural scenes show, under somewhat restrictive conditions it is possible to obtain good depth results.

Acknowledgements - Comments by Prof. Allen R. Hanson, Prof. Thomas S. Huang, Rakesh Kumar, Dr. John Oliensis, Prof. E. M. Riseman, and Harpreet Sawhney proved to be of great help to the authors.

A Details of the Experiments on Velocity Approximation

The procedure to compare the velocity formalism with the displacement formalism in section 6 was as follows:

1. Program *P1* generated the file *F1* which contained

   (a) 50 random image coordinates (x, y) in a 256×256 grid.

   (b) Random depths (within a given range) at the above image coordinates.

2. File *F1* together with the focal length, motion parameters $(T_1, T_2, \Omega_1, \Omega_2, \Omega_3)$ and total translation were given as input to program *P2*. Program *P2* used the displacement (i.e. exact) equations (Equations 2 and 3) to produce a file *F2* with image displacements, (u, v) at the above image coordinates. Note - the image displacements are rounded to 4 places after the decimal point.

3. Program *P3* uses File *F2* together with the known motion parameters and computes the depths using both the displacement formalism and the velocity formalism. There are two depths generated (one using u and the other using v) in each case for each image point. The relative error in depth is computed by using the original depth information in file *F1* for both cases. The depth was recomputed using the displacement formalism as a control in order to be certain that u and v had been kept sufficiently precise for depth determination. Actually in the case of real correspondence algorithms it is unlikely that u and v could be given this precisely. Since the experimental results shown in Table 4 indicate that the average percentage error in depth was virtually zero in all cases with the displacement formalism, it can be stated that 4 places after decimal is more than enough for u and v as far as our experiments go.

B Rationale for using Adiv's algorithm

To provide a rationale for using Adiv's algorithm for determining rough motion parameters we must give a short review of some some of the existing methods for depth determination In the absence of well–established terminology we assign some of these methods reasonably descriptive names. We must warn that there is some overlap between the methods and that the citations are representative rather than exhaustive. In any case some of the widely used practical techniques are as follows.

E-Matrix method - This was initially formulated by Longuet-Higgins, and independently by Tsai and Huang [2]. In this approach a 3×3 matrix called the E-matrix is defined in terms of motion parameters. Linear equations relating image coordinates to elements of the E-matrix are established. It has been experimentally found that this method is very susceptible to noise [11]. Overdetermined systems, least squares solution etc. can be investigated [10] to provide some improvements.

Flow based optimization - A form of this approach was proposed by Bruss and Horn [7]. Adiv [6] extended the method considerably and provided a working system with good error measures. Functionally the method can be described as trying to minimize the difference between the actual displacements in the image and the computed displacement based on the hypothesized values of the motion parameters. The difficulty is that the equations which arise are non–linear. Hence various approximations have to be made [20] to arrive at reasonably good solutions. It is mathematically a rather difficult problem, either with the velocity or with the displacement formulation, and a multi–resolution search strategy is the answer which Adiv provides. In general a method like this can only be evaluated on the basis of its performance on natural scenes. Extensive experimentation on natural images reported in [14] shows that Adiv's algorithm is generally effective to some extent.It has been claimed by [30], among others, that most methods for motion estimation do

not consistently work well in the presence of noise or when there is no initial first guess to within 10 percent of the final values. By doing a multi–resolution search over a unit sphere with a good optimizing criterion the above problems are to some extent removed. Of course, multi–resolution search is time consuming and takes several minutes with 100 points on a 256×256 image on VAX/750's in a single user mode. In general it can be said that the method does work on natural scenes if the correspondence is good.

Intersecting pairs of rays - Using sets of pairs of rays from two projection centers to points in the scene the relative orientation (i.e. the motion parameters) of one camera with respect to the other can be determined. Horn gives a method to do this [4] through iterative improvements, with no restrictions on rotational parameters. Again there is only empirical evidence to indicate that this converges well. The error term to be minimized is formulated on the basis of the coplanarity of the intersecting rays from the two camera positions and the baseline between the two positions.

The above is not an exhaustive list. There are other approaches and extensions like Kalman–filtering, depth from focus, stereo–motion, photometric–motion etc. that can also be used. Nevertheless, there are few well–documented results on reasonably difficult natural scenes. This paper tries to fill the gap by presenting clear results on a natural scene with known ground truth.

However, one thing that is apparent is that whenever constraints are simplified to arrive at linear equations the ultimate results for depth are not very good. When complex constraints are formulated giving rise to non–linear equations doubts are cast on the final solution since there is no mathematical way to prove that the final solution is indeed the correct one. The constraint of minimizing the difference between actual and computed flow is a very good one indeed. A multi–resolution search strategy with a good constraint is probably the best that can be done short of the computationally time consuming and virtually impossible exhaustive search. In view of the above we use the velocity formulation

of Adiv [6] to determine the coarse motion parameters.

C Simulated Annealing

Figure 9 gives the details of the simulated annealing algorithm used for estimating the precise motion parameters. The initial temperature, alpha and beta parameters for simulated annealing can be determined quite easily by finding out the rough range of difference in error values between neighboring perturbations of the motion parameters. A few minutes on a VAX/3100 in a single user mode is usually enough to compute the final motion parameters with simulated annealing (for roughly 100 interest points). While we would have liked to have exhaustively searched the neighborhood it is not feasible because there are five parameters that have to be solved for. This results in an $(O(n^5))$ algorithm where n is the number of values of each parameter that we have to search exhaustively. Without prior knowledge of the error surface, annealing is a relatively reliable and quick method when the approximate solution is already known. This is a further reason for using Adiv's algorithm to generate rough motion parameters.

Of course, even if we did an exhaustive search, rather than annealing, around the neighborhood of the rough motion parameters there is still no guarantee that the global minimum will be found. This is because there is always the possibility that the rough motion parameters are totally wrong or that the neighborhood is not large enough. In some sense our error measure directly tries to optimize a depth based measure derived from the displacement formalism. Therefore, we choose this in our refining stage as opposed to doing a finer search with the same error measure which gave us the coarse motion parameters. We should remember that just being a good error measure is not enough - the error surface should also be amenable to the determination of the global minima in natural scenes without spending too much time. Simulated annealing is very good at

Figure 9: **Simulated Annealing algorithm for finding precise motion parameters**

```
Procedure Simulated-Annealing-for-Precise-Motion-Parameters;
 soln := Rough-Motion-Parameters ;
 T := t0 ;initial temperature
 iterations := i0 ;number of iterations of inner loop initially
 REPEAT
    REPEAT
    newsoln := perturb(soln);
    IF (cost(newsoln) ¡ cost(soln)) or (random ¡ f(newsoln,soln,T))
    THEN soln = newsoln;
    UNTIL the inner loop is repeated iterations times;
 T := alpha * T;
 iterations := beta * iterations;
 UNTIL timeup
```

The following should be noted:

- *random* is a real uniformly generated pseudo-random number between 0 and 1 generated at each iteration.
- *cost(x)* where x is the set of motion parameters is computed by using Equation (52).
- *f(newsoln,soln,T)* is given by

$$exp(\frac{(cost(soln) - cost(newsoln))}{T}) \tag{56}$$

- *perturb(x)* finds a neighborhood of the set of motion parameters by changing them slightly.
- *alpha* is a constant that lies in the open interval (0,1).
- *beta* is a constant that is at least 1.
- *timeup* is the amount of computer time we wish to spend for solving the problem.

solving problems where the error surface is not known in advance. As will be evident later on from the depths that we get on natural scenes, our error measure seems to work well in the neighborhood around the correct solution. On noiseless synthetic data we get exact depth.

D Characteristics of the Images used

For our experiments we used a sequence of twenty images. The images were collected with the Carnegie Mellon NAVLAB [31]. The first image of the sequence is shown in Figure 6. The vehicle was made to move in an approximately straight line such that the distance between frames was 2 feet. The field of view of the camera was 45° and we worked on 256×256 images. In order to determine the ground truth for environmental objects, traffic cones were placed at measured distances. The traffic cones have well-distinguishable corners and hence image displacements for these points can be obtained somewhat more easily. The cones were placed at distances ranging from 21 to 76 feet with respect to the first frame of the sequence. Obviously with the total movement of the vehicle of 40 feet, some of the cones disappeared from the scene in later frames. Figure 6 shows the environmental points whose depths had been quantitatively measured. It should be noted that even though in some sense the set up has been contrived by putting traffic cones it is in general quite complex because of the presence of large homogeneously textured regions like road and grass, and the occlusion of the distant buildings through trees.

It was found experimentally that the total translation of two feet between frames was inadequate for getting large enough image displacements for the cones. Our theoretical proofs of section 3 suggest that large image displacements are necessary to counteract small errors in rotational parameters. Hence experiments were conducted by considering alternate frames. A four foot translation which is almost fully along the optical axis means that even in the first frame (T_3/Z) varies roughly between $\frac{1}{4}$ to $\frac{1}{20}$ for enviromental points with quantitatively known depth. Note that the theoretical analysis of section 5 shows that this kind of motion where the camera is moving approximately straight ahead is rather difficult to handle.

Figure 10: **Error as reduced by simulated annealing in the precise motion parameter determination phase.** The error value more or less stabilized after the iterations shown in the figure.

iter.	*error*	*iter.*	*error*
0	0.59190	21	0.57518
44	0.56434	69	0.54037
96	0.53635	125	0.52144
156	0.51783	190	0.51342
227	0.50392	267	0.49763
310	0.49903	357	0.50635
408	0.49072	464	0.49001
525	0.48509	592	0.48015
665	0.48082	745	0.48186
832	0.47839	927	0.47931
1031	0.47648	1145	0.47815
1270	0.47767	1407	0.48162
1557	0.47886	1721	0.48107
1901	0.47579	2098	0.47497
2314	0.47740	2551	0.47539
2811	0.47520	3096	0.47579
3490	0.47497	3753	0.47767
4131	0.47539	4546	0.47539

E Experiments with Simulated Annealing

The perturbation for the rotational angles in simulated annealing was 0.01° and for the translational parameters was 0.001. The annealing was done for 10000 iterations with (a) initial temperature = 60 ; (b) alpha = 0.92 ; (c) beta = 1.1. As can be seen from Figure 10 the error reduces from about 0.59 to 0.47. The maximum allowed deviation from the rough motion parameters is 0.1 in the case of translational parameters and 0.5° in the case of rotational parameters

References

[1] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In *Image Understanding 1984*, pages 179–193. Ablex Publishing Corporation, 1984.

[2] R. Tsai and T. S. Huang. Uniqueness and estimation of 3-d motion parameters of rigid bodies with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 13–27, January 1984.

[3] O. Faugeras, F. Lustman, and G. Toscani. Motion and structure from motion from point and line matches. *International Conference on Computer Vision*, pages 25–34, 1987.

[4] B. K. P. Horn. Relative orientation. *Proceedings DARPA Image Understanding Workshop*, pages 826–837, 1988.

[5] J. W. Roach and J. K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 554–562, November 1980.

[6] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 384–401, July 1985.

[7] A. Bruss and B. K. P. Horn. Passive navigation. *Computer Vision Graphics and Image Processing*, pages 3–20, January 1983.

[8] H. C. Longuet Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, pages 133–135, September 1981.

[9] J. Q. Fang and T.S. Huang. Solving three dimensional small-rotation motion equations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 253–258, June 1983.

[10] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 451–476, May 1989.

[11] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of 3-d motion parameters and surface structures of rigid objects. In *Image Understanding 1984*, pages 135–171. Ablex Publishing Corporation, 1984.

[12] J. L. Barron, A. D. Jepson, and J. K. Tsotsos. The sensistivity of motion and structure computations. *Proceedings Sixth National Conference on Artificial Intelligence*, pages 700–705, 1987.

[13] M. A. Snyder. The precision of 3-d parameters in correspondence-based techniques: The case of uniform translational motion in a rigid environment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 523–528, May 1989.

[14] R. Dutta, R. Manmatha, E. Riseman, and M. Snyder. Issues in extracting motion parameters and depth from approximate translational motion. *Proceedings DARPA Image Understanding Workshop*, Volume 2:pages 945–960, April 1988.

[15] R. Manmatha, R. Dutta, E. Riseman, and M. Snyder. Issues in extracting motion parameters and depth from approximate translational motion. *Proceedings IEEE Workshop on Visual Motion, Irvine, Calif.*, pages 264–272, March 1989.

[16] Gilad Adiv. Inherent ambiguities in recovering 3-d motion and structure from a noisy flow field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 477–489, May 1989.

[17] A. Verri and T. Poggio. Motion field and optical flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 490–498, May 1989.

[18] J. K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proceedings of the IEEE*, pages 917–935, August 1988.

[19] B. K. P. Horn. *Robot Vision.* The MIT Press, 1986.

[20] Gilad Adiv. *Interpreting Optical Flow.* PhD thesis, University of Massachusetts at Amherst, 1985. COINS TR 85-35.

[21] M. Turk, D. Morgenthaler, K. Gremban, and Martin Marra. Vits - a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 342–361, May 1988.

[22] R. Dutta, R. Manmatha, L. R. Williams, and E. M. Riseman. A data set for quantitative motion analysis. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 159–164, June 1989.

[23] John Rice. *Mathematical Statistics and Data Analysis.* Wadsworth and Brooks, 1987.

[24] J. Spanur and K. Oldham. *An Atlas of Functions.* Hemisphere Publishing Corporation, 1987.

[25] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, Volume 2:pages 283–310, 1989.

[26] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, May 1983.

[27] S. Nahar, S. Sahni, and E. Shragowitz. Simulated annealing and combinatorial optimization. *23rd Design Automation Conference*, pages 293–299, 1986.

[28] A. Bandopadhay and R.Dutta. Measuring image motion in dynamic images. *Proceedings of the IEEE Conference on Motion: Representation and Analysis*, pages 67–72, May 1986.

[29] C. L. Fennema and W. B. Thompson. Velocity determination in scenes containing several moving objects. *Computer Vision Graphics and Image Processing*, Volume 9:pages 301–315, 1979.

[30] C. Jerian and R. Jain. Polynomial methods for structure from motion. *International Conference on Computer Vision*, pages 197–206, December 1988.

[31] C. Thorpe, S. Shafer, T. Kanade, and the members of the Strategic Computing Vision Lab. Vision and navigation for the carnegie mellon navlab. *Proceedings DARPA Image Understanding Workshop*, pages 143–152, February 1987.