T-RA/6/5//38265

Using Camera Motion to Estimate Range for Robotic Parts Manipulation

David Vernon
Massimo Tistarelli

Reprinted from
IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION
Vol. 6, No. 5, October 1990

Using Camera Motion to Estimate Range for Robotic Parts Manipulation

DAVID VERNON, MEMBER, IEEE, AND MASSIMO TISTARELLI, MEMBER, IEEE

Abstract—A technique is described for determining a depth map of arts in bins using optical flow derived from camera motion. Simple rogrammed camera motions are generated by mounting the camera on ıe robot end effector and directing the effector along a known path. 'he results achieved using two simple trajectories, where one is along ıe optical axis and the other is in rotation about a fixation point, are etailed. Optical flow is estimated by computing the time derivative of a equence of images, i.e., by forming differences between two successive nages and, in particular, matching between contours in images that ave been generated from the zero crossings of Laplacian of Gaussian-ltered images. Once the flow field has been determined, a depth map is omputed utilizing the parameters of the known camera trajectory. 'mpirical results are presented for a calibration object and two bins of arts; these are compared with the theoretical precision of the tech-ique, and it is demonstrated that a ranging accuracy on the order of vo parts in 100 is achievable.

I. Introduction

AS ROBOT VISION matures, it is becoming increasingly desirable to extend its capabilities to include 3-D sens-ıg. A significant goal of this capability is to solve the in-picking problem in which a robot manipulator is required) identify and grasp an object jumbled in a bin of many such bjects. Although active sensing (and active triangulation in ıe form of light striping, in particular) has been popular in roviding range information, it has not yet been successfully mployed in bin picking. Furthermore, future robot vision pplications will require increasing robustness such as is romised by image understanding systems. A central tenet of nage understanding research is the necessity of inferring the -D structure of the imaged scene through the use of several ıutually redundant visual cues (such as stereopsis and visual ıotion [1]–[3]). One particularly useful paradigm for the eneration of these disparate cues is based on analysis of ero-crossing contours in Laplacian of Gaussian-filtered im-ges [4]–[6]: These contours represent the position and orien-tion of intensity discontinuities in the image [7], [8]. Al-ıough the coherent integration of information derived from ıch filtered images, along with other visual cues such as ıading, texture, and occlusion, is still in its infancy, progress being made, and it seems sensible to begin to deploy limited versions of this technology to industrial applications now, especially as hardware becomes available to implement the computationally expensive filtering stage. The research described in this paper endeavors to do just that while at the same time providing a pathway for future developments.

In particular, this paper describes the use of a single camera mounted on a robot end effector (describing a simple camera motion) to infer the depth of objects jumbled together in bins. Two types of camera motion are employed. The first describes a trajectory along the optical axis of the camera, whereas in the second, the camera is rotated about a fixation point. The optic flow field resulting from the first type of egocentric motion is very easy to compute because all flow vectors are directed radially outward from the focus of expansion (FOE), i.e., the center of the image [9]. For camera motion in rotation about a fixation point, the rotational component of optical flow can be determined directly from the known camera trajectory, and the direction of the translational component is also constrained by the camera motion. Knowing the direction of the flow vector, the magnitude of the visual motion is directly derived from a time-derivative of a sequence of images acquired at successive points along the camera trajectory.

Such use of a constrained camera motion is ideally suited to industrial environments because manipulator arm trajectories can be specified at will. Furthermore, the technique facilitates the future incorporation of more general camera motion and, eventually, the mutual integration of information derived from other passive visual sensing.

II. The Bin-of-Parts Problem and Range Estimation

The bin-of-parts, or bin-picking, problem is widely recognized as being one of the most difficult tasks in robotics. Although a considerable amount of effort has been expended by the robotics and computer vision community in an attempt to solve this problem, no general solution has yet been reported in the literature. Indeed, it is worth noting that, so far, only two broad approaches have been documented, which appear to provide any realistic bin-picking capabilities. Details of these approaches may be found in, for example, [10] and in [11]. The central requirement in the bin-of-parts problem is to be able to direct a robot manipulator to select, grasp, and remove an arbitrarily oriented part (or object) from a bin of many such objects. These objects will, in general, be jumbled together and will occlude one another significantly. Thus, in order for the robot manipulator to be able to grasp the object, it must be able to identify the

Manuscript received July 1, 1988; revised February 19, 1990. This work as supported by the European Strategic Program for Research and Devel-ɔment in Information Technology (ESPRIT) under Project P419: Image and ovement Understanding.

D. Vernon is with the Department of Computer Science, Trinity College, ublin, Ireland.

M. Tistarelli is with the Department of Computer, Communication, and /stems Science, University of Genova, Genova, Italy.

IEEE Log Number 9038265.

1042-296X/90/1000-0509$01.00 ©1990 IEEE

position and the orientation of the object, in spite of the fact that it would probably be partially hidden by one or more other objects in the bin. Such an identification of position and orientation, which is often referred to as the object pose, is an extremely daunting task, and it has been suggested that although the problem is not intractable, it is extremely difficult [13]. This very difficulty has given rise to two distinct schools of thought: one holding that it is not absolutely necessary to determine the pose of an object in the bin and that the part can be removed using more realistic techniques and the other adhering to the pursuit of the general pose estimation problem. The former approach owes much of its success to Kelley and his co-workers at the University of Rhode Island [11], [13]–[20], whereas the latter owes much to the research of Horn and Ikeuchi [10], [21], [22]. In either case, it is usually necessary to identify the range of the objects before attempting to grasp them and remove them from the bin. This paper is concerned with the extraction of such range data. One can characterise 3-D image acquisition systems on the basis of two criteria [26]–[28]: 1) whether they are active or passive devices and 2) whether they are triangulation or nontriangulation devices.

Active image acquisition systems explicitly utilize contrived illumination to accomplish the range estimation. Passive image acquisition systems use ambient illumination. Current trends in the acquisition of range data indicate a predeliction for active systems [23]–[25]. Active systems are, in general, more precise since they can be designed to suit a particular application and since they inherently allow (and require) much more control over the sensors and the data being sensed. Conversely, the motivation for researching entirely passive techniques for 3-D imaging is their adaptability. This arises from the exploitation and integration of numerous mutually redundant visual cues, e.g., shading, occlusion, motion due to both observer and camera, stereopsis, texture gradients, and focusing. These cues are capable of providing useful and reliable range information as well as measures of local surface orientation of objects to facilitate grasping by manipulators when integrated in coherent manner and especially when used with other passive range estimation techniques (e.g., see [10]). Paradoxically, the use of several redundant visual cues to provide the 3-D information is both the source of difficulty in building such a system and the basis for its robust nature. The motivation underlying the work described in this paper is to begin, in a small way, development of a purely passive system by using one limited version of one of the essential cues of this type of 3-D computer vision.

III. Inferring Depth from Camera Motion

In the controlled environment of a robot workcell with a camera mounted on the end effector "looking" into the bin, simple motion strategies can be adopted to determine the depth of the objects. Since general camera motion requires knowledge of six parameters

$$W = (W_x, W_y, W_z)$$

$$\omega = (\omega_x, \omega_y, \omega_z)$$

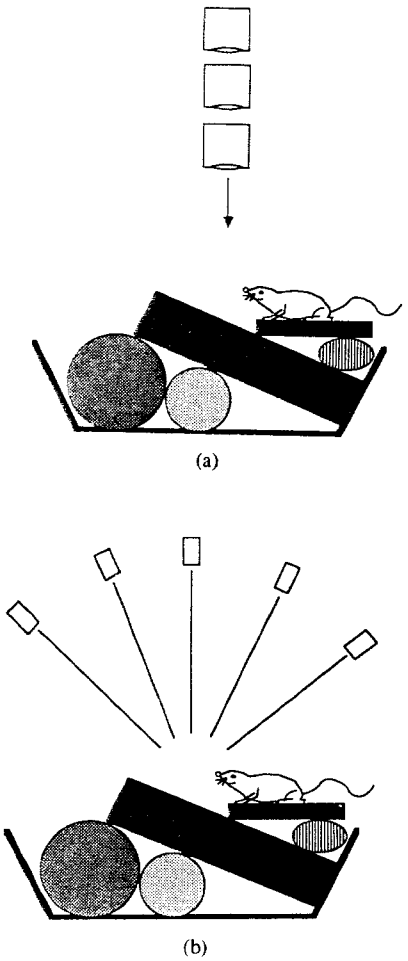


Fig. 1. (a) Translational motion along the optic axis; (b) rotational motion about a fixation point.

i.e., the translational and rotational components of camera velocity, constraining the camera motion greatly simplifies the computation of optical flow. In particular, two simple cases are empirically studied in this paper, and a general analysis of accuracy is given in Section V. In the first motion strategy, the camera is moved along the direction of the optical axis, which is the Z axis of the camera-centered coordinate system. In the second case, the camera is rotated about some fixation point on the optical axis but is some distance from the camera (see Fig. 1). Thus, the camera motion is now parameterized simply by W_z in the first case, and by W_x, W_z, and ω_y in the second. As we shall see in the remainder of this section, this greatly simplifies the computation of the optical flow. The rotational motion was chosen since this allows us to extend the baseline of camera movement while maintaining an approximately fixed field of view. A pure lateral translation of the camera is a viable alternative (e.g., see [37] and [38]), but in this case, the effective area in which the optical flow, and hence the range, can be computed is reduced to approximately the intersection of the fields of view of the camera in its initial and final position. Note, however, that in the case of the work cited in [37], this resulted in only a 6% reduction in the width of the effective field of view.

The magnitude of the optical flow that results from either

camera motion is unknown, but it is computable by differentiating the image sequence with respect to time. If the luminance intensity does not change with time (i.e., there are no moving light sources in the environment), the component of the orthogonal velocity vector for each image point along the direction of the local intensity gradient is given by [29]

$$v^{\perp} = -(\partial I/\partial t)/|\nabla I| \tag{1}$$

where ∂ indicates the partial derivative operator, and $|\nabla I|$ is the local intensity gradient.

The algorithm for computing depth can be summarized as follows:

a) Convolve the images with a Laplacian of Gaussian operator [8].
b) Extract the zero crossings, computing the local slope and orientation of each contour point.
c) Compute the difference between the convolution of successive frames of the sequence.
d) Compute the velocity component in the direction perpendicular to the orientation of the contour.
e) Compute the velocity along the contour using the known motion parameters (see below).
f) Search for the zero crossings of the second frame projected from the first frame in the direction of the velocity vector.
g) Compute the depth map from this optic flow.

These steps form the body of an iterative scheme that allows one to compute the optical flow of a sequence of images. The approach of Matthies *et al.* [37] utilizes Kalman filtering to temporally integrate and refine successive estimates of depth derived from image pairs, whereas the technique described in this paper exploits spatial tracking of features across a sequence of images to increase the baseline and, hence, increase the accuracy of the triangulation procedure when computing depth.

Since all flow computations are done at image contours only (having been extracted using the Laplacian of Gaussian operator), the amount of data to be processed is limited, and furthermore, the effects of noise are less pronounced.

The computation of $v^{\perp}$ (the orthogonal component of velocity) is based on a computation of the time derivative using a five-point approximation formula [30] according to the relationship described in (1).

The computation of the true velocity vector depends on the prior knowledge of the parameters of the camera motion: The distance of the camera at time T and $T + \Delta t$ from the fixation point; θ, which is the rotational angle of the camera around the Y axis; W_x and W_z, which are the components of the translational velocity of the camera along the X axis and the Z axis, respectively. W_x and W_z are defined with respect to the coordinate system of the camera at time T (see Fig. 2). Using basic trigonometric relations, we find

$$W_x = \frac{D_2 \sin\theta}{\Delta t} \tag{2}$$

$$W_y = 0 \tag{3}$$

$$W_z = \frac{D_1 - D_2 \cos\theta}{\Delta t} \tag{4}$$

where D_1 and D_2 are the distances of the camera from the fixation point at time ΔI and $T + \Delta t$, respectively.

These computed egomotion parameters are used to determine the true image velocity vector v. Note that v comprises two components v_t and v_r, where one is due to camera translation $W = (W_x, W_y, W_z)$, and the other is due to camera rotation $\omega = (\omega_x, \omega_y, \omega_z)$:

$$v_t = \left(\frac{xW_z - FW_x}{Z}, \frac{yW_z - FW_y}{Z}\right) \tag{5}$$

$$v_r = \left(\frac{xy\omega_x - (x^2 + F^2)\omega_y + y\omega_z}{F}, \frac{(y^2 + F^2)\omega_x - xy\omega_y - x\omega_z}{F}\right) \tag{6}$$

$$v = v_t + v_r \tag{7}$$

where F is the focal length of the lens, x and y are the coordinates of the point in the image plane at time T, and Z is the distance from the camera to the world point corresponding to image point (x, y).

For the constrained camera motion shown in Fig. 2, the camera translational velocity is given by (2)–(4), whereas the rotational velocity ω is $(0, \theta/\Delta t, 0)$. The image velocity components can then be written as

$$v_t = \left(\frac{x(D_1 - D_2\cos\theta) - FD_2\sin\theta}{Z\Delta t}, \frac{y(D_1 - D_2\cos\theta)}{Z\Delta t}\right) \tag{8}$$

$$v_r = \left(\frac{-(x^2 + F^2)\theta}{F\Delta t}, \frac{-xy\theta}{F\Delta t}\right). \tag{9}$$

In these two equations for v_t and v_r, the only unknown is Z (which is what we wish to determine). Thus, to determine v_t and v_r, and hence Z, we exploit the value of $v^{\perp}$, which is the orthogonal component of velocity, computed at an earlier stage. This can be accomplished directly by solving the attendant system of equations [30], [36] or by a geometrical construction [31]. The system of nonlinear equations to recover the direction of the image velocity v can be formulated in the following manner:

$$v_x = v_{tx} + v_{rx}$$

$$v_{rx} = \frac{-(x^2 + F^2)\theta}{F\Delta t}$$

$$v_y = v_{ty} + v_{ry}$$

$$v_{ry} = \frac{-xy\theta}{F\Delta t}$$

$$v_{tx} = \frac{x(D_1 - D_2\cos\theta) - FD_2\sin\theta}{Z\Delta t}$$

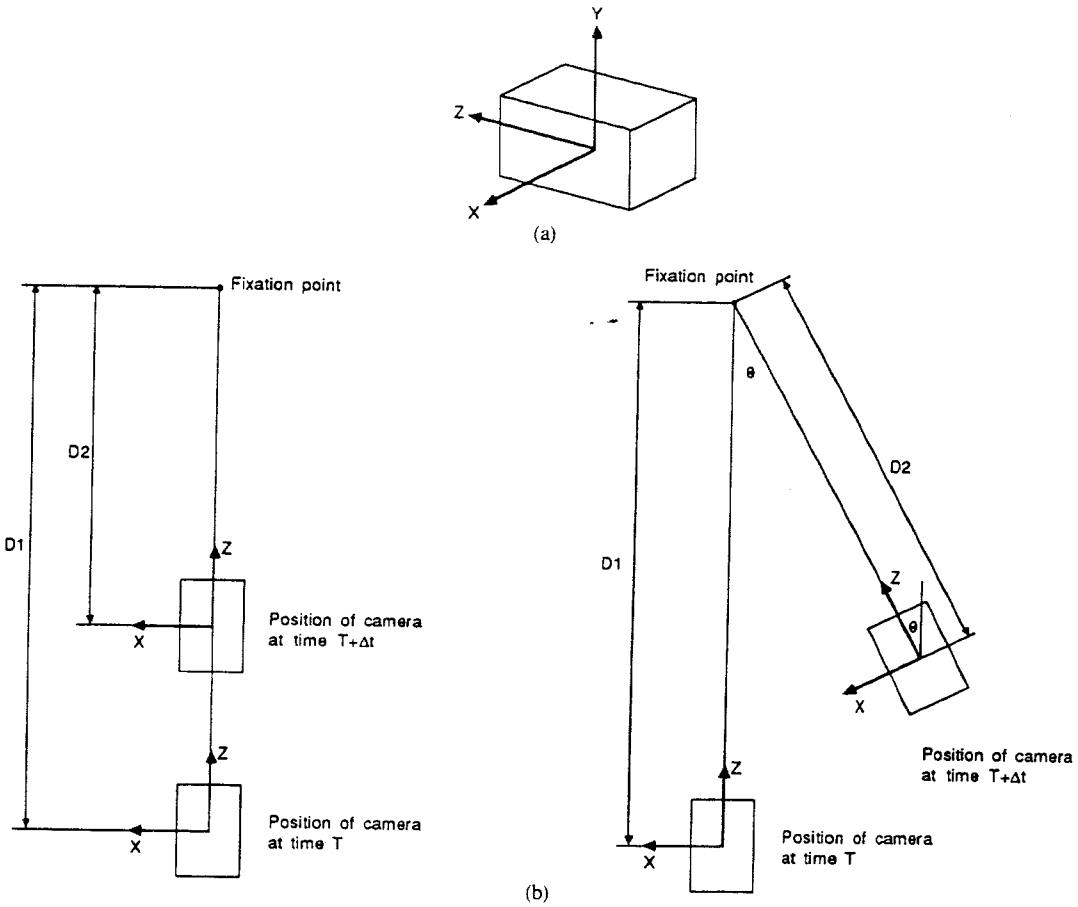


Fig. 2. (a) Camera coordinate system; (b) parameters associated with camera motion.

$$v_{ty} = \frac{y(D_1 - D_2 \cos\theta)}{Z\Delta t}$$

$$|v| = \frac{v_x}{\cos(\beta - \alpha)}$$

$$|v| = \frac{v^{\perp}}{\cos\alpha}$$

$$\frac{v_y}{v_x} = \tan(\beta - \alpha)$$

where the unknown terms are

$$v = (v_x, v_y), \alpha, Z$$

and the known terms are F, D_1, D_2, θ (measured from the camera); $v^{\perp}$, β (measured from the images); the image point coordinates (x, y) and the time span Δt. α is the angle subtended by the vectors v and $v^{\perp}$, β is the angle $v^{\perp}$ makes with the X axis, and v_{rx} and v_{ry} are the X and Y components of the vector v_r, which represents the component of the flow field due to the rotation of the camera.

The system can be solved, in closed form, for the direction of the velocity vector v:

$$\tan(\beta - \alpha) = \frac{(D_1 - D_2\cos\theta)(v^{\perp} - \cos\beta(yv_{rx} - xv_{ry})) - FD_2(\cos\beta v_{ry}\sin\theta)}{(D_1 - D_2\cos\theta)(v^{\perp} + \sin\beta(yv_{rx} - xv_{ry})) - FD_2(v^{\perp}\sin\theta - \sin\beta v_{ry}\sin\theta)}. \quad (10)$$

In the solution by geometrical construction, v is determined from the intersection of three straight lines derived from v_r (for which all terms are known), $v^{\perp}$ (which was computed previously), and the position of the FOE.

First, v_r defines the first line of the construction (refer to Fig. 3). Second, the position of the FOE defines the direction of v_t, since v_t is parallel to the line joining the FOE and the point (x, y) in question. Thus, the second line is parallel to v_t and passes though the point given by v_r (see Fig. 3). The coordinates of the FOE are given by

$$(x_{FOE}, y_{FOE}) = \left(\frac{FW_x}{W_z}, \frac{FW_y}{W_z}\right) \quad (11)$$

where W_x, W_y, and W_z are the known velocities of the camera in the x, y, and z directions, respectively [32].

Finally, we note that v is also given by the sum of the orthogonal component and the tangential component of velocity

$$v = v^{\perp} + v^{T}$$

Since these two vectors are orthogonal to one another and since $v^{\perp}$ is known, this relationship defines a third line through the point given by $v^{\perp}$ and normal to the direction of

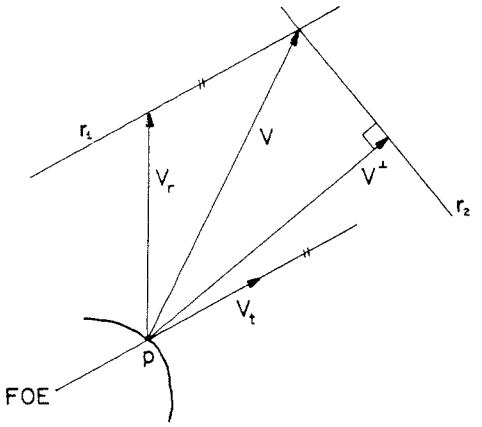


Fig. 3. Computation of true velocity v from $v^{\perp}$, v_t, and v_r at a point P on a zero-crossing contour.

$v^{\perp}$. Hence, v is given by the intersection of the second and the third lines (see Fig. 3).

Computing v in this manner and, in particular, computing $v^{\perp}$ using the five-point approximation, errors can still recorded in the final flow. A significant improvement can be achieved by performing a contour-to-contour matching between successive frames, along the direction of the flow vectors, tuning the length of the flow vectors to the correct size. The tracking procedure searches in the direction of the flow vector until the next contour is found: then, it searches in the direction of the new flow vector, and so forth, until the whole image sequence is processed. Although a small difference between successive frames is required to guarantee the accuracy in the computation of the orthogonal component $v^{\perp}$, a long baseline is required for the range measurement. For this reason, many images are considered, and the flow field obtained for a sequence of, say, five images is used for range computation; the flow vector from the first image to the last image is employed in the computation of range. We shall return to this point again when discussing the precision of the technique.

The depth, for each contour point, is computed by

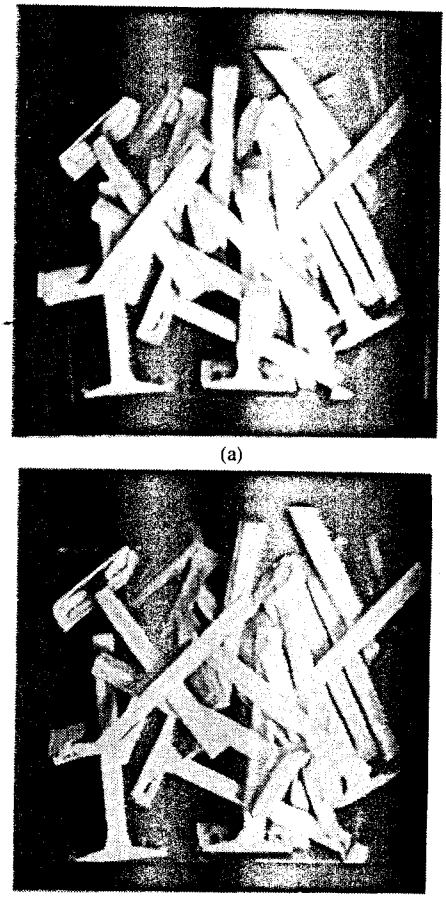
$$\frac{Z}{W_z} = \frac{D_f}{|V_t|} \tag{12}$$

where Z is the distance of the environmental point from the camera, D_f is the distance of the image point from the FOE, V_t is the translational component of the flow, and W_z is the camera velocity along the optic axis.

It is worth noting that (10) holds only for the contour points with a non-zero velocity; the camera velocity v is actively controlled to be different from zero.

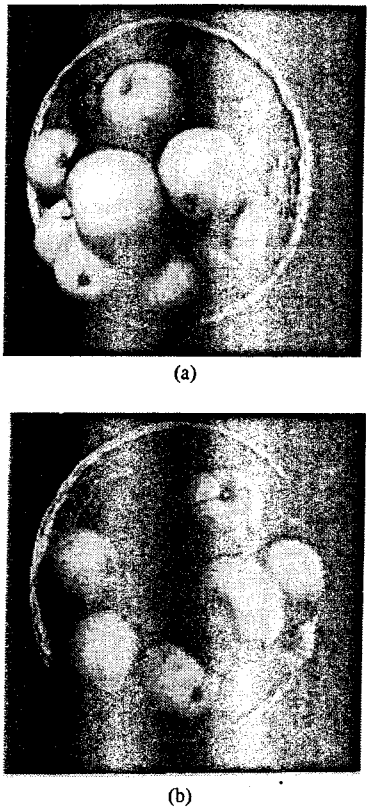
IV. Experimental Procedure and Results

In order to evaluate this approach to inferring the depth of objects, motion sequences of three different scenes were generated. These scenes contained a bin of disposable razors, a basket of fruit, and a white 45° cone with black stripes at regular intervals (see Figs. 4–6). To generate the linear ego-motion sequences, a Panasonic CCD camera was mounted on the end effector of a low-cost revolute manipula-



(a) (b)

Fig. 4. (a) First image of bin of razors; (b) last image of bin of razors.



(a) (b)

Fig. 5. (a) First image of basket of fruit; (b) last image of basket of fruit.

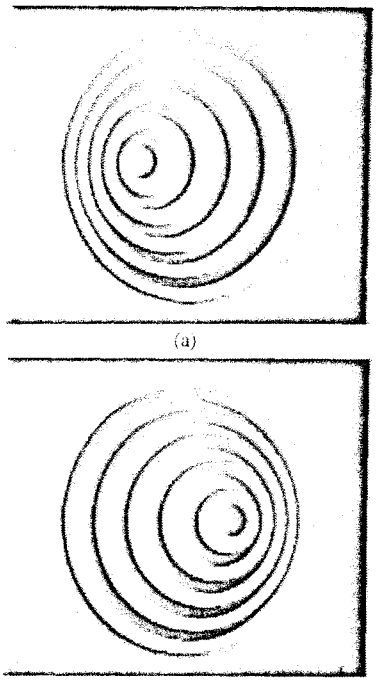


Fig. 6. (a) First image of cone; (b) last image of cone.

tor (a SmartArms 6R/600). The robot was programmed to position the end effector over the work surface, with the camera pointing directly down, and to move downwards along a vertical trajectory. An image of the scene was acquired at 20-mm intervals on this trajectory; a total of nine images were acquired in each motion sequence. To generate the motion sequence of the camera in rotation about a fixation point, a simple jig was constructed, and nine images were generated at 5° increments equally distributed about the vertical. (For the purposes of illustration, Figs. 4 through 20 depict the results of the rotational motion only. Complete quantitative summary of the results of both the translational and the rotational experiments are given in Figs. 22–27, and in Table I.)

Each of the constituent images in these image sequences were then convolved with a Laplacian of Gaussian mask (standard deviation of the Gaussian function = 4.0—see Figs. 7–9), and the zero-crossing contours were extracted. Since the Laplacian of Gaussian operator isolates intensity discontinuities over a wide range of edge contrasts, many of the resultant zero crossings are noisy. An adaptive threshold technique [33] was employed to identify these contours and to exclude them from further processing (Figs. 10–12 depict the super-imposed zero crossings of the first and last images).

The zero-crossings contour images and their associated convolution images were then used to generate six time derivatives; since the time derivative utilizes a five-point operator, the time derivative can only be estimated for images 3–7. The associated orthogonal component of velocity is then computed, followed by the true optical flow vectors. An extended flow field was then estimated by tracking the flow vectors from image 3 through images 4 and 5 to image 6 on a contour-to-contour basis, i.e., tracking a total of three images (see Figs. 13–15). For the sake of comparison, two depth

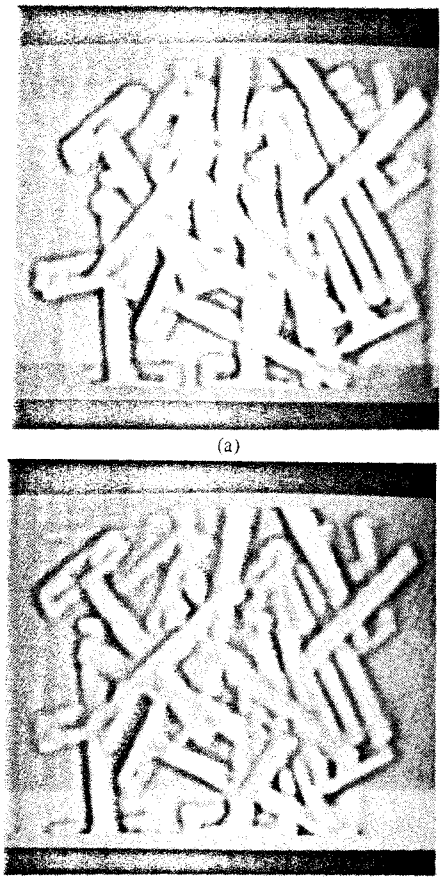


Fig. 7. Convolution of (a) first and (b) last image of bin of razors with a $\nabla^2 G$ mask ($\sigma = 4$).

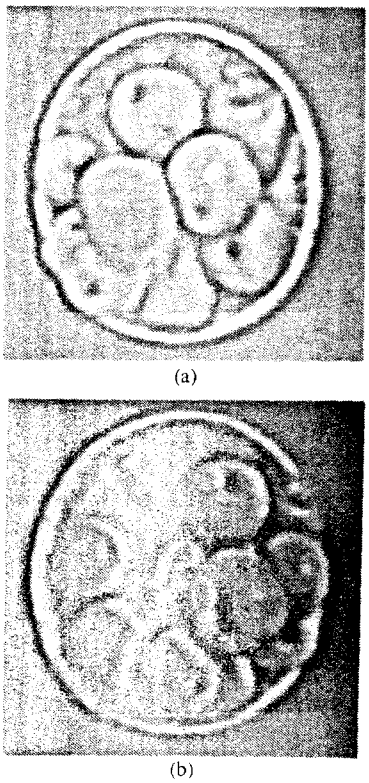
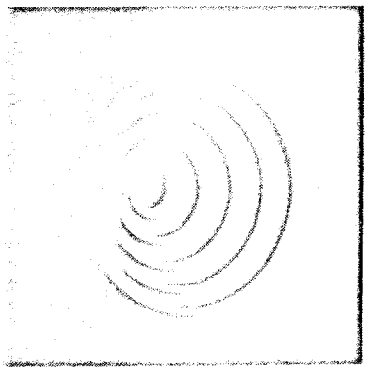
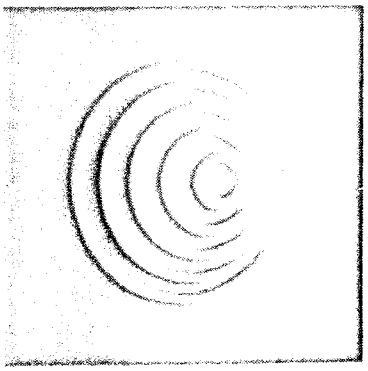


Fig. 8. Convolution of (a) first and (b) last image of basket of fruit with a $\nabla^2 G$ mask ($\sigma = 4$).



(a)



(b)

Fig. 9. Convolution of (a) first and (b) last image of cone with a $\nabla^2 G$ mask ($\sigma = 4$).

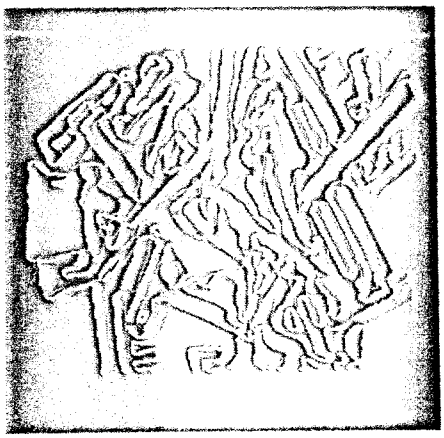


Fig. 10. Superimposed zero-crossings of the first and last images.

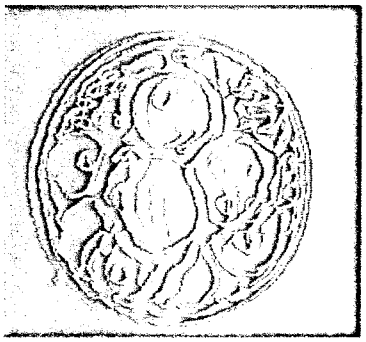


Fig. 11. Superimposed zero crossings of the first and last images.

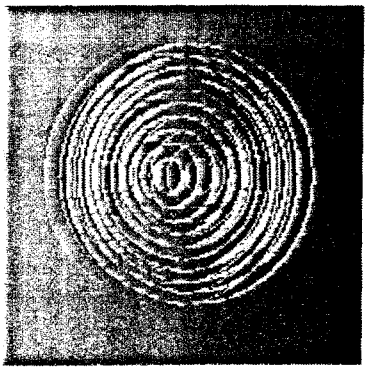


Fig. 12. Superimposed zero crossings of the first and last images.

images (representing the distance from the camera to each point on the zero-crossing contour) were generated for both scenes: one from the instantaneous flow vectors of image 3 (Figs. 16(a), 17(a), and 18(a)) and one from the tracked velocity vectors (Figs. 16(b), 17(b), and 18(b)). To better illustrate the depth value along each contour, the contour images are shown in side elevation. Thus, the relative depth of the contours is a function of the distance along the horizontal axis. Finally, a range image representing the range of all visible points on the surface was generated by interpolation (Figs. 19–21).

V. Analysis

It has been shown [30], [36] in an extension of an analysis of the precision of range estimation by motion [34] that the theoretical accuracy of computing depth from motion is given by[1]

$$|\delta Z| = \frac{|D_f W_z|}{|v_t|^2} |\delta v_t| \tag{13}$$

where D_f is the distance of the image point from the FOE or focus of contraction (FOC) on the image plane, W_z is the Z component of camera translational velocity, v_t is the image velocity due to camera translation, and $|\delta v_t|$ is the magnitude of the accuracy in measuring the displacement of the image point.

Assuming W_z and v_t to be constant throughout the considered time span of, for instance, N frames, and letting v_{t_0} be the translational velocity between two successive frames and W_{z_0} be the corresponding camera velocity along the direction of the Z axis, then substituting in (13), we obtain

$$|\delta Z| = \frac{|D_f N W_{z_0}|}{N^2 |v_{t_0}|^2} |\delta v_t|$$

$$= \frac{|D_f W_{z_0}|}{N} \frac{|\delta v_t|}{|v_{t_0}|^2}. \tag{14}$$

Thus, the precision in the computed depth is directly proportional to the considered number of frames N.

[1] In the case of a pure lateral translation of the camera in a direction normal to the focal axis, the distance to the FOE D_f is infinite, and the translational velocity W_z in the Z direction is zero. Equation (8) is, thus, singular. Recalling (5) and noting that W_z is now zero, we have $v_t = (-FW_x/Z, -FW_y/Z)$. Thus, $|v_t| = F/Z|W|$. Rearranging $Z = F|W|/v_t$ and hence $|\delta Z| = |\delta v_t| F|W|/|v_t|^2$.

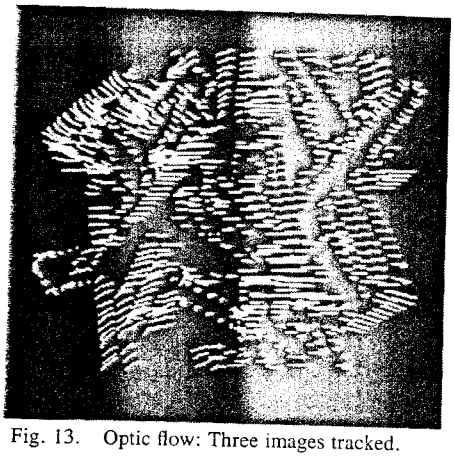


Fig. 13. Optic flow: Three images tracked.

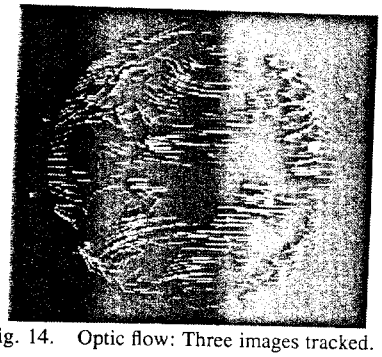


Fig. 14. Optic flow: Three images tracked.

The distance of the image point from the FOE is given by

$$D_f = \sqrt{(x - x_{FOE})^2 + (y - y_{FOE})^2}.$$

Using the equations for x_{FOE} and y_{FOE} given in (11), we have

$$D_f = \sqrt{\left(x - \frac{FW_x}{W_z}\right)^2 + \left(y - \frac{FW_y}{W_z}\right)^2}$$

and using the equations for W_x, W_y, and W_z given in (2)–(4)

$$D_f = x - \frac{FD_2 \sin\theta}{D_1 - D_2 \cos\theta}.$$

Assuming incremental time periods between images ($\Delta t = 1$), we have, from (4)

$$W_{z_0} = D_1 - D_2 \cos\theta.$$

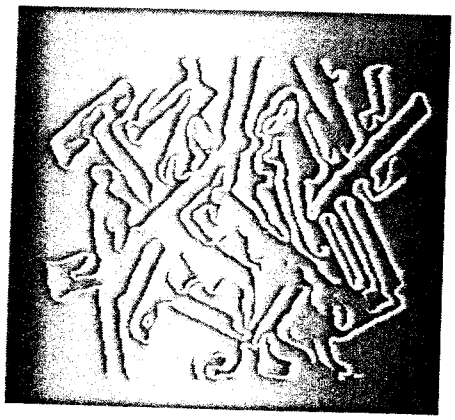
In addition, from (8)

$$|v_{t_0}| = \sqrt{\left(\frac{x(D_1 - D_2\cos\theta) - FD_2\sin\theta}{Z}\right)^2 + \left(\frac{y(D_1 - D_2\cos\theta)}{Z}\right)^2}.$$

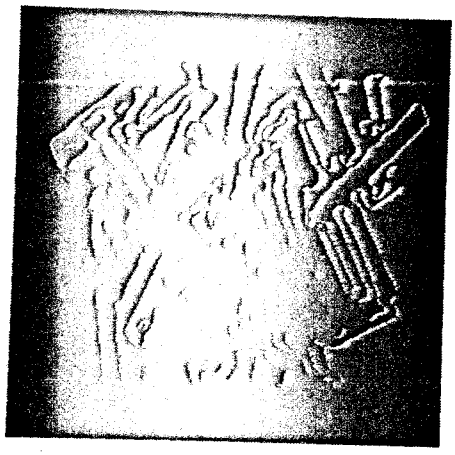
Finally, it is assumed that $|\delta v_t| = 1$, that is, the error in measuring the translational component of the flow vector is one pixel. Hence, (14) can be rewritten

$$|\delta Z| = \frac{\left(x - \frac{FD_2\sin\theta}{D_1 - D_2\cos\theta}\right)(D_1 - D_2\cos\theta)}{N\left(\left(\frac{x(D_1 - D_2\cos\theta) - FD_2\sin\theta}{Z}\right)^2 + \left(\frac{y(D_1 - D_2\cos\theta)}{Z}\right)^2\right)}. \quad (15)$$

Fig. 15. Optic flow: Three images tracked.

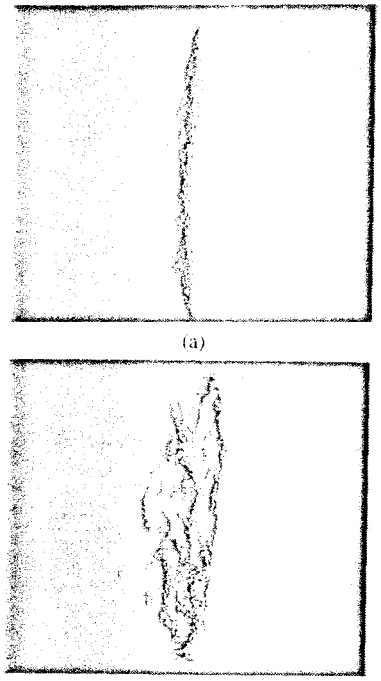


(a)



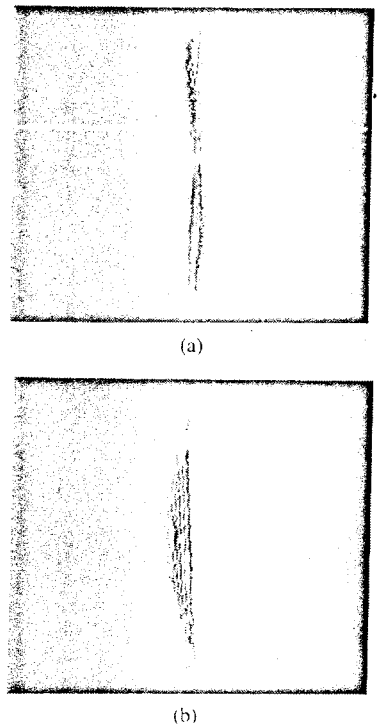
(b)

Fig. 16. (a) Depth contours (one image tracked); (b) depth contours (three images tracked).



(a) (b)

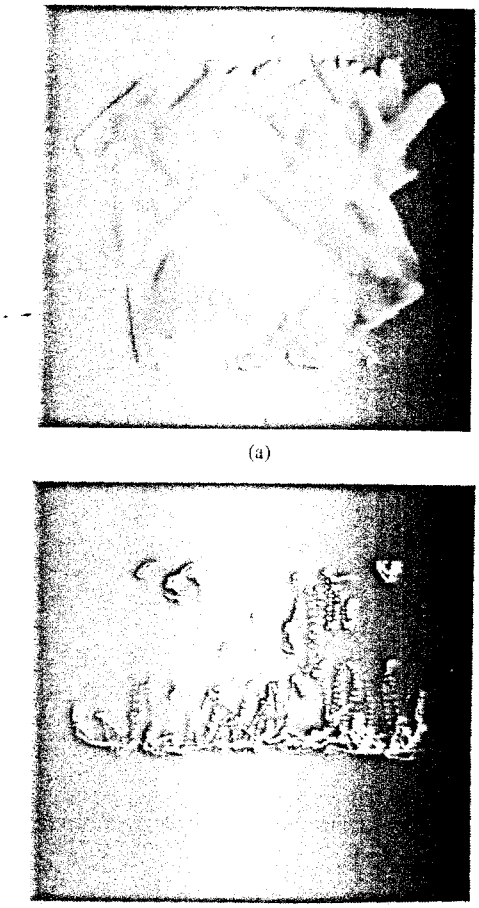
Fig. 17. (a) Depth contours (one image tracked); (b) depth contours (three images tracked).



(a) (b)

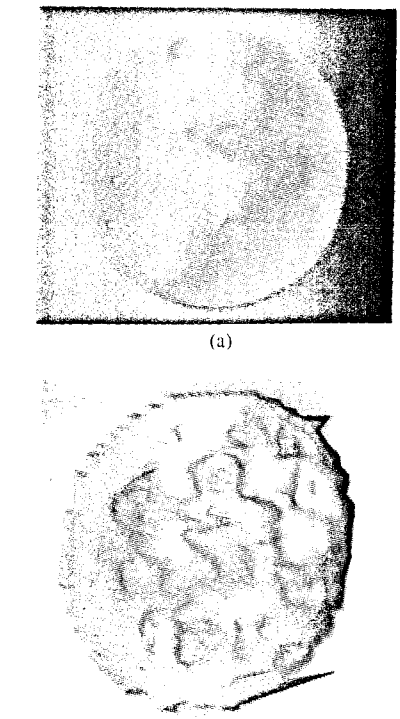
Fig. 18. (a) Depth contours (one image tracked); (b) depth contours (three images tracked).

Thus, for a given focal length F, which is a given initial distance to the fixation point D_1, the accuracy in the computation of range $|\delta Z|$ is a function of the position in the image (x, y), the angle of rotation θ of the camera about its Y axis, the actual range Z, and the baseline of the motion (which is a function of θ and D_2). This accuracy measure $|\delta Z|$, given equivalently by (11) and (15), is plotted in Fig.



(a) (b)

Fig. 19. (a) Range image; (b) perspective view of depth contours.



(a) (b)

Fig. 20. (a) Range image; (b) perspective view of range image.

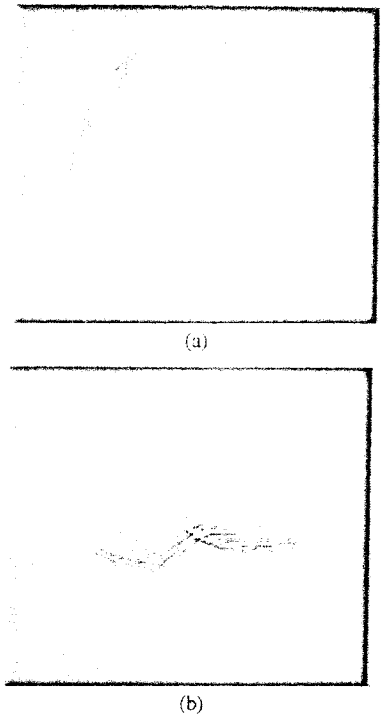


Fig. 21. (a) Range image; (b) perspective view of depth contours.

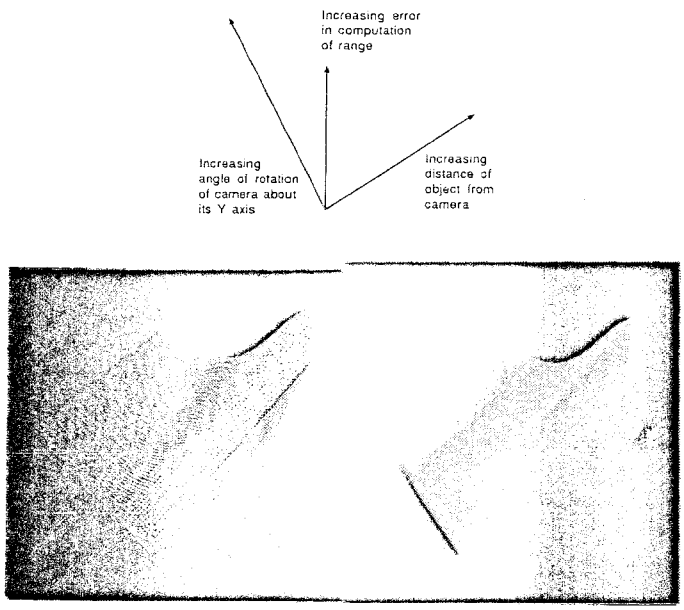


Fig. 22. Linear trajectory—Error in computation of range as a function of the distance Z of the object from the camera and as a function of the rotation θ of the camera about its Y axis, i.e., the motion configuration (refer also to Fig. 2).

22 for different values of the distance Z of the world point from the camera and for different motion trajectories, varying the rotation θ of the camera about its Y axis and maintaining constant the camera displacement and the number of frames. The error in computation is proportional to the elevation of the 3-D plot. It is evident that the error increases with Z. However, the variation in accuracy with the rotation of the camera is very interesting; beginning with an axial motion ($\theta = 0$), the error increases, at first, as θ increases and only begins to decrease again after the angle increases beyond

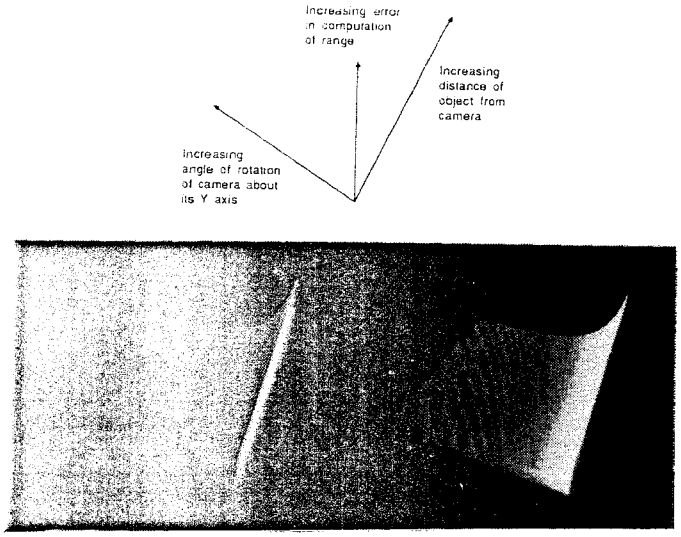


Fig. 23. Circular trajectory—Error in computation of range as a function of the distance Z of the object from the camera and as a function of the rotation θ of the camera about its Y axis, i.e., the motion configuration (refer also to Fig. 2).

$\theta = 7°$ (approximately). The error corresponding to a rotation of the camera of $\theta = 45°$ is shown at the extreme of the θ axis.

Since this graph assumes a *linear* baseline (and hence, does not exactly correspond to the experiment involving pure rotational motion described in this paper), the accuracy $|\delta Z|$, which is plotted against Z and θ as before but this time changing the baseline to create a circular trajectory by ensuring $D_1 = D_2$, is plotted in Fig. 23. In this case, the results are as expected; the error increases with Z and decreases with θ, although it is evident that the marginal increase in accuracy with increasing θ falls off quite quickly. Note that these graphs are plotted for an image point at coordinates (100, 100).

With regard to the experiments described above, the theoretical accuracy of the range computation $|\delta Z|$ at a point 500 mm distant from the camera and at image coordinates (128, 128) at the periphery of the field of view (since a resolution of 256 × 256 pixels was used throughout) in the case of linear egomotion along the focal axis is

$N = 1$: 58.42 mm
$N = 3$: 19.47 mm
$N = 5$: 11.68 mm

Similarly, in the case of rotational egomotion about the fixation point 500 mm distant from the camera, the theoretical accuracy is

$N = 1$: 12.63 mm
$N = 3$: 4.21 mm
$N = 5$: 2.52 mm..

In the experiments described in the previous section, the mean and standard deviation of the range measurements along each cone contour was computed. It should be noted well that to facilitate direct comparison of the results of the four types of motion, all the range results have been normal-

TABLE I
SUMMARY OF THE COMPUTED RANGE VALUES

Cone Radius	Actual Range	Translational Motion 1 Image Tracked		Translational Motion 3 Images Tracked		Rotational 1 Image Tracked		Rotational 3 Images Tracked	
		μ	σ	μ	σ	μ	σ	μ	σ
R_1	545.0	545.0	149.0	545.0	95.92	545.0	6.3	545.0	64.6
R_2	536.0	518.6	143.2	523.3	78.4	544.3	5.2	533.6	10.8
R_3	527.0	483.1	151.6	510.1	74.12	543.6	4.5	525.3	8.8
R_4	518.0	457.8	157.2	497.0	91.5	543.2	3.6	518.0	7.6
R_5	509.0	413.0	160.1	488.3	95.9	542.7	2.7	511.2	5.9
R_6	500.0	334.3	174.1	479.6	95.9	542.3	4.0	505.8	5.6
R_7	491.0	258.4	165.7	497.6	109.0	541.7	3.2	497.3	6.4
R_8	482.0	168.5	131.9	470.8	156.9	541.7	2.1	488.3	5.2
R_9	473.0	49.44	87.0	440.3	187.4	541.9	1.3	480.1	6.7
R_{10}	464.0	50.4	42.0	296.4	222.3	542.3	1.4	475.2	14.3

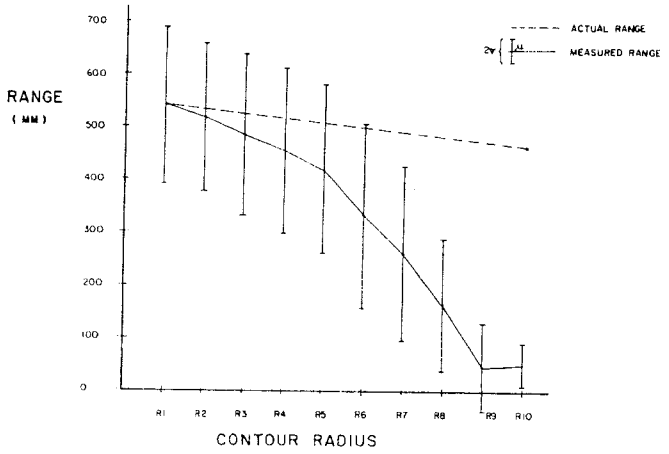


Fig. 24. Range: Linear motion along the optic axis (one image tracked).

ized so that the range of the furthest contour agrees with its theoretical (actual) value; this would be a normal calibration step in an industrial setup in any case. These results are tabulated for the four types of motion investigated (rotational and translational motion, one and three images tracked) (see Table I). These values are also plotted graphically in Figs. 24 through 27, where the true range is also depicted.

Referring to these figures and Table, it can be observed that in the case of linear motion along the optic axis (one image), the standard deviation of the range is approximately three times greater than the theoretical accuracy; the mean value deviates by a consistently increasing amount from the true range. However, the deviation is in the correct direction so that the relative depth of each contour is correct. These inaccuracies can probably be accounted for by the inaccurate location of the focus of expansion. It was assumed in the computation that the FOE was located at the center of the image, but any deviation from a true camera trajectory along the optical axis will invalidate this, introducing errors. Since only one image is being tracked in this case, a slight lateral movement of the camera results in asymmetrical optical flow and significantly increased variation in the computed range.

Tracking for three images provides, as expected, significantly better results. The mean range of each contour follows

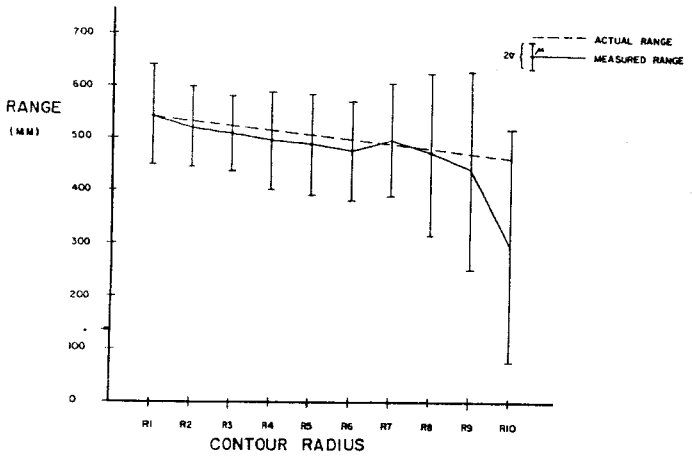


Fig. 25. Range: Linear motion along the optic axis (three images tracked).

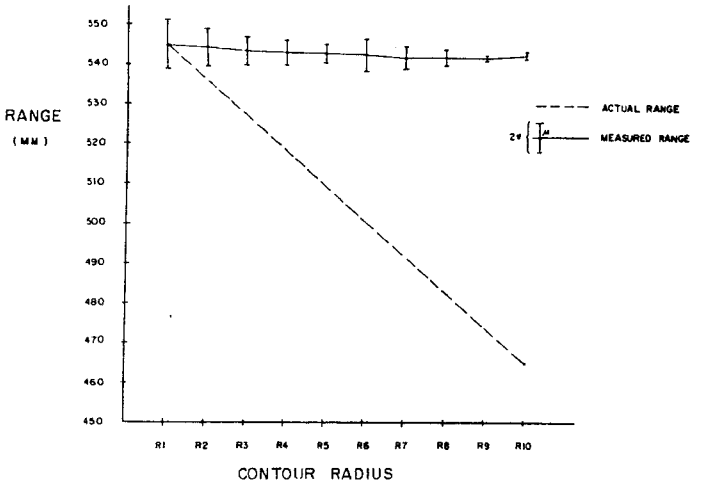


Fig. 26. Range: Rotational motion about a fixation point (one image tracked).

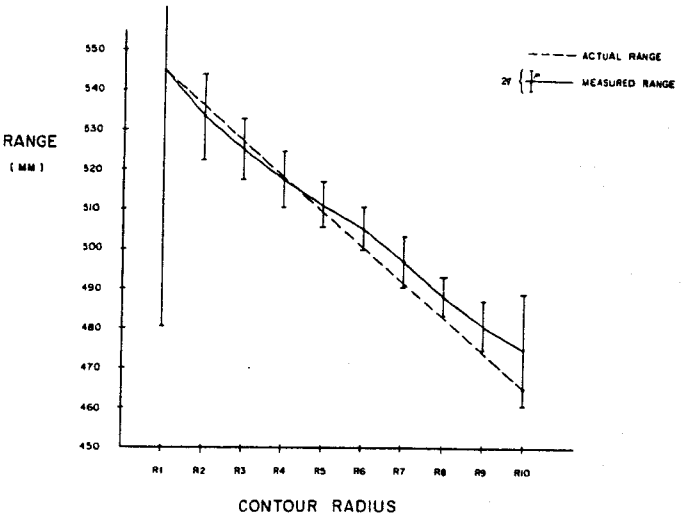


Fig. 27. Range: Rotational motion about a fixation point (three images tracked).

the true range quite well. Again, however, the standard deviation of the range values for any one contour is quite large (at least four times the theoretical accuracy).

In the case of rotational camera motion about a fixation point, we observe much improved results when tracked across

three images. Here, the mean computed contour range and the actual contour range concur very well. The computed ranges of the five largest cone contours are all within the theoretical accuracy, whereas the five smallest contours are typically computed to within 50% of the theoretical accuracy (i.e., the error is up to twice the theoretical accuracy). The standard deviation of the computed range values is extremely low; in most cases, it is in the order of the theoretical accuracy of the computation. In the two contours where this is not so, the deviation may be explained by anomolous range values at points on the contour where the direction of camera motion is parallel to the contour orientation, and hence, large errors in the computation of the orthogonal component $v^{\perp}$ result.

In tracking a single image, the computed range values are poor. The range values change very little, though at least the direction of change is generally correct, according valid relative depth to each contour. This probably due to the very small magnitude of the measured orthogonal component $v^{\perp}$.

In processing an increased number of frames, these results show a reduction in bias, i.e., the absolute error, for both the translational and rotational cases, a decrease in variance for the translational case, but an increase in variance for the rotational case. First, the bias is reduced since, with an increase in the baseline, the numerical accuracy of the depth computation increases. However, the unexpected increase in variance of computed depth in the rotational case is accounted for by the shape of the object or, rather, by the shape of the zero-crossing contours. Specifically, the orientation of a significant proportion of the concentric circular zero-crossing contours is identical to that of the direction of the optical flow vectors. The increased variance is due to the inherent ambiguity in matching these zero-crossing points when refining the vector length during the tracking phase of the algorithm. This results in consequent errors in the depth computation: see Figs. 18(b) and 21(b).

VI. Discussion

It has been shown that tracking across three images in an image sequence generated by a camera in rotation about a fixation point, it is possible to compute the range of objects to within at least 10 mm over a distance of 500 mm (i.e., 2% error) using rotational camera motion about a fixation point. Even better accuracy is achieved if one is prepared to aggregate the range value along a contour. Ninety percent of the mean ranges for the cone contours were computed to within 7.1 mm of their correct value, and 50% of them were computed to within 3.6 mm of their correct value. The variation in computed range values for the linear motion along the optic axis can be accounted for by small lateral deviations from the true trajectory. It is not clear whether this limitation is due to inherent sensitivity of this technique or whether it is due to poor positional control of the robot manipulator. In any case, the theoretical and practical accuracy of the rotational motion suggests it to be a clearly superior approach.

It should be noted, however, that for the other applications, the axial motion is an extremely interesting type of motion; it would be of particular use with autonomously guided vehicles as they proceed in a straight line toward a fixation point. Furthermore, one can improve the results obtained in this paper by choosing an alternative experimental setup and, specifically, by extending the baseline and the number of images used in computing the optical flow.

VII. Conclusions

This paper has described the successful use of passive vision, under the guise of analysis of simple optical flow based on two types of camera motion, to infer depth of objects in bins. It is important to note that the theoretical and practical accuracy achievable with this technique makes its deployment as a ranging technique for robot manipulators entirely feasible. The main advantage of the technique is that it relies only on ambient lighting to accomplish the range estimation. Without doubt, the technique will prove to be even more robust when augmented with other visual cues, and the task of integrating such motion with an analysis of stereo disparity is actively being pursued [39]. Thus, the approach represents a useful starting point from which a robust passive 3-D robot vision system, based on the mutual integration of several visual cues, can be developed.

All of the research described in this paper was carried out using a low-level computer vision development package (VIS [35]), and to achieve the required computational speed required of industrial vision systems, the system is currently being ported to a specially designed transputer-based multiprocessor system.

Acknowledgment

The authors wish to acknowledge the many helpful suggestions made by the reviewers of a previous version of the paper.

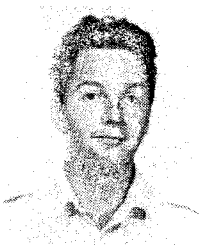
References

[1] M. Brady, "Computational approaches to image understanding," *ACM Comput. Surveys*, vol. 14, no. 1, pp. 3-71, 1982.
[2] T. O. Binford, "Survey of model-based image analysis systems," *Int. J. Robotics Res.*, vol. 1, no. 1, pp. 18-64, 1982.
[3] J. M. Tenenbaum, H. G. Barrow, and R. C. Bolles, "Prospects for industrial vision," SRI Int. Tech. Note 175, 1978.
[4] D. Marr, *Vision*. San Francisco: Freeman, 1982.
[5] W. E. L. Grimson, *From Images to Surfaces*. Cambridge, MA: MIT Press, 1981.
[6] E. C. Hildreth, *The Measurement of Visual Motion*. Cambridge, MA: MIT Press, 1983.
[7] D. Marr, "Early processing of visual information," *Philo Trans. Royal Soc. London*, vol. B275, pp. 483-524, 1976.
[8] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Royal Soc. London*, vol. B207, pp. 187-217, 1980.
[9] G. Sandini and M. Tistarelli, "Analysis of camera motion through image sequences," in *Advances in Image Processing and Pattern Recognition* (V. Cappellini and R. Marconi Eds.). New York: Elsevier Science, 1986, pp. 100-106.
[10] K. Ikeuchi, H. K.Nishihara, B. K. Horn, P. Sobalvarro, and S. Nagata, "Determining grasp configurations using photometric stereo and the PRISM binocular stereo system," *Int. J. Robotics Res.*, vol. 5, no. 1, pp. 46-65, 1986.
[11] R. B. Kelley, H. A. S. Martins, J. R. Birk, and J-D. Dessimoz, "Three vision algorithms for acquiring workpieces from bins," *Proc. IEEE*, vol. 71, no. 7, pp. 803-821, 1983.
[12] T. Sakata, "An experimental bin-picking robot system," in *Proc. 3rd. Int. Conf. Assembly Automat.*, 1982, pp. 615-626.

[13] J-D. Dessimoz, J. R. Birk, R. B. Kelley, A. S. Martins, and I. Chi Lin, "Matched filters for bin picking," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 686-697, 1984.
[14] R. B. Kelley, "Heuristic vision algorithms for bin picking," in Proc. 7th. Conf. Ind. Robot Technol. (Gottenburg, Sweden), 1984, pp. 599-610.
[15] R. B. Kelley, J. R. Birk, H. A. S. Martins, and R. Tella, "A robot system which acquires cylindrical workpieces from bins," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-12, no. 2, pp. 204-213, 1982.
[16] R. Kelley, J. Birk, J. Dessimoz, and R. Tella, "Acquiring connecting rod castings using a robot with vision and sensors," in *Proc. 1st. Int. Conf. Robot Vision Sensory Contr.*, 1981, pp. 169-178.
[17] R. Kelley, J. Birk, D. Duncan, H. Martins, and R. Tella, "A robot system which feeds workpieces directly from bins into machines," in *Proc. 9th. Int. Symp. Ind. Robotics*, 1979, pp. 309-355.
[18] J. R. Birk, R. B. Kelley, and L. Wilson, "Acquiring workpieces: Three approaches using vision," in *Proc. 8th. Int. Symp. Ind. Robot* (Stuttgart, West Germany), 1978, pp. 724-733.
[19] J. R. Birk, R. B. Kelley, and J-D. Dessimoz, "Visual control for handling unoriented parts," *Proc. SPIE*, vol. 281, pp. 169-175, 1981.
[20] J. R. Birk, R. B. Kelley, and H. A. S. Martins, "An orienting robot for feeding workpieces into stored bins," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-11, no. 2, pp. 151-160, 1981.
[21] B. K. P. Horn and K. Ikeuchi, "Picking parts out of a bin," AI Memo 746, MIT AI Lab, Cambridge, MA, 1983.
[22] K. Ikeuchi, "Determining attitude of object from needle map using extended Gaussian image," AI Memo 714, MIT AI Lab, Cambrdige, MA, 1983.
[23] P. J. Besl and R. Jain, "Three-dimensional object recognition," *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75-145, 1985.
[24] M. J. Magee, B. A. Boyter, C. Chien, and J. K. Aggarwal, "Experiment in intensity guided range sensing recognition of three-dimensional objects," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. PAMI-7, no. 6, pp. 629-637, 1985.
[25] R. Y. Wong and K. Hayrapetian, "Image processing with intensity and range data," *IEEE Comput. Soc. Conf. Patt. Recognition Image Processing* (Las Vegas, NV), 1982, pp. 518-520.
[26] W. D. M. McFarland and R. W. McLaren, "Problems in three-dimensional imaging," *Proc. SPIE*, vol. 449, pp. 148-157, 1983.
[27] W. D. M. McFarland, "Three dimensional images for robot vision," *Proc. SPIE*, vol. 442, pp. 108-116, 1983.
[28] G. Mooney and N. Murphy, "Three-dimensional computer vision for robotic assembly," Prelim. Rep. Nat. Inst. Higher Education (Dublin, Ireland), 1986.
[29] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, no. 1, pp. 185-204, 1981.
[30] G. Sandini and M. Tistarelli, "Integration of edge motion information," Rep. No. TK1-WP-1-D12, ESPRIT Project 419—*Image and Movement Understanding*, Comm. Euro. Commun., 1986.
[31] G. Sandini and M. Tistarelli, "Analysis of object motion and camera motion in real scenes," in *Proc. IEEE Int. Conf. Robotics Automat.* (San Francisco), 1986, pp. 627-632.
[32] D. T. Lawton, "Processing translational motion sequences," *Comput. Vision Graphics Image Processing*, vol. 22, pp. 116-144, 1983.
[33] D. Vernon, "On the properties of zero-crossing contours in Laplacian of Gaussian-filtered images," Tech. Rep. CSC-88-03, Dept. Comput. Sci., Trinity College, Dublin, Ireland, 1988.
[34] S. Bharwani, E. Riseman, and A. Hanson, "Refinement of environmental depth maps over multiple frames," in Proc. Workshop Motion Representation Anal., 1986, pp. 73-80, IEEE Comput. Soc.
[35] D. Vernon and G. Sandini, "VIS: A virtual image system for image understanding," *Software: Practice Experience*, vol. 18, no. 5, pp. 395-414, 1988.
[36] G. Sandini and M. Tistarelli, "Active tracking strategy for monocular depth inference from multiple frames," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 12, no. 1, pp. 13-27, 1990.
[37] L. Matthies, R. Szeliski, and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences," in *Proc. DARPA Image Understanding Workshop*, Apr. 1988, pp. 199-213.
[38] H. Baker and R. Bolles, "Generalising epi-polar plane image analysis on the spatio-temporal surface," in *Proc. DARPA Image Understanding Workshop*, Apr. 1988, pp. 1022-1030.
[39] E. Grosso, G. Sandini, and M. Tistarelli, "3D object reconstruction using stereo and motion," *IEEE Trans. Syst. Man. Cybern.*, vol. 19, no. 6, pp. 1465-1476, 1989.

David Vernon (M'82) received the B.A. and B.A.I. degrees in mathematics and engineering science in 1979 and the Ph.D. degree in computer science in 1985 from the University of Dublin, Trinity College, Dublin, Ireland.

He has been a lecturer at the Department of Computer Science, Trinity College, Dublin, since 1983, and he worked with Westinghouse Electric Inc. as a software engineer from 1979 to 1981. He is a Chartered Engineer and a director of the Vision and Sensor Research Unit in the Irish Advanced Manufacturing Technology program. His current research interests are focused on the development of visual servoing techniques for robotics and the investigation of autonomous systems and their relevance to computational models of visual perception.



Massimo Tistarelli (M'85) was born in Genoa, Italy, on November 11, 1962. He received the degree in electronic engineering "magna cum laude" from the University of Genoa, Italy.

Since 1984, he has worked on image processing and computer vision at the Laboratory of Robotics of the Department of Communication, Computer, and Systems Science of the University of Genova, where he is currently completing the Ph.D. degree. In 1986, he was a Research Assistant at the Department of Computer Science, Trinity College, Dublin, developing a system for the integration of low-level visual processes. In 1989, he was a Visiting Scientist at Thinking Machines Co., Cambridge, MA, where he developed parallel algorithms for dynamic image processing on the Connection Machine. His research interests include robotics, artificial intelligence, image processing, and computer vision, particularly in the area of three-dimensional and dynamic scene analysis.

Mr. Tistarelli is a member of the IEEE Computer Society and the Robotics and Automation Society.