Evaluation of a Real-Time Kinetic Depth System

Gillian M. Hayes* and Robert B. Fisher

Department of Artificial Intelligence, University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, Scotland, United Kingdom
email: gmh@uk.ac.edinburgh.aifh

We describe a robot vision system which produces a depth map in real time by means of motion parallax or kinetic depth. A video camera is held by a robot which moves so that a given point in space is kept fixated on the centre of the camera's imaging surface. The optical flow is calculated in a Datacube MaxVideo system and a full-frame depth map is produced 12.5 times per second. Calculated depths show an average 10% discrepancy with measured depths over 7 nonconsecutive images.

For an observer fixating a point in space and moving perpendicularly to the direction of gaze, objects in front of the fixation point appear to move in the opposite direction and objects behind the fixation point appear to move in the same direction. The speed of apparent motion is proportional to the distance of the object from the fixation point. This phenomenon is known as kinetic depth. If the parameters of the observer's optical system and the details of the observer's motion and fixation point are known, then measurements of the speed and direction of apparent motion of an image point (optical flow) give the distance of the corresponding object point from the observer.

We have constructed a kinetic depth system which produces a depth map once every 80ms [1]. A robot, holding a video camera, moves along a fixating trajectory. The time and space derivatives of two consecutive video images are calculated in a Datacube MaxVideo image processing system and combined to give a 512 x 512 pixel depth map at half frame-rate.

It is planned to integrate such a depth-detection module into an existing assembly robot vision system, SOMASS [2], for the purpose of obstacle avoidance. Before moving the hand which is holding the camera to a particular 3D target point, the robot would execute a fixating motion about a suitable point and build up a depth map. Any depth points in the region of the depth map corresponding to the direction of the target point which are closer to the robot than the target point would indicate the presence of an obstacle.

The first step in this process is the construction of the kinetic depth system and the second step is to evaluate its sensitivity to system parameters. We describe the kinetic depth system here and give preliminary results of the evaluation.

THE KINETIC DEPTH SYSTEM

The work reported here was inspired by the Rochester Robot of Brown, Ballard et al. [3], which integrates many primate-like visual functions such as vergence, control of gaze direction, saccades, etc. as well as kinetic depth, into one robot head-eye system. Our system, which is modelled after its kinetic depth system, is shown in Figure 1. A Sony CCD video camera is attached to the gripper of an Adept robot. The robot is programmed to move the camera along a straight line perpendicular to the optical axis, changing the yaw angle during the motion so that the same point in space, the fixation point, is always imaged onto the centre of the camera's imaging surface. The optical flow is calculated in a Datacube MaxVideo image processing system and a depth map is produced at half frame-rate (12.5 times per second).

It is important to know how accurate the depth maps produced by such a system are, and how sensitive they are to the system parameters: robot velocity, fixation point position, camera focal length and pixel width. We have carried out experiments in which these parameters were varied in order to assess whether it will be possible to integrate this subsystem into the SOMASS assembly system.

* Supported during part of this work by an SERC studentship

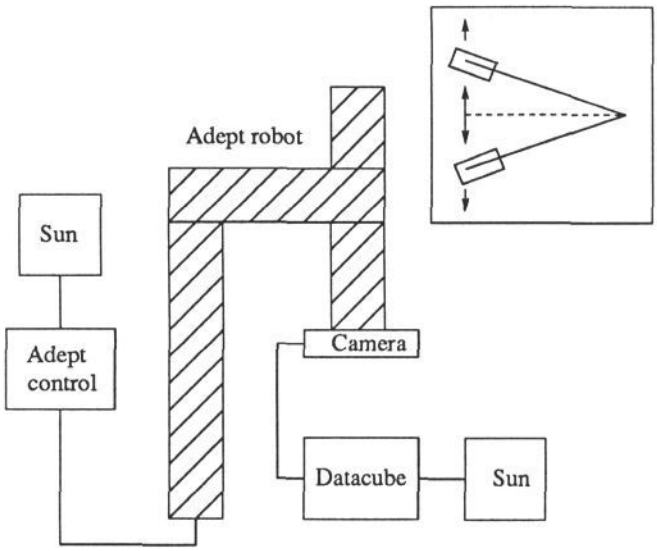

Figure 1: Schematic of the kinetic depth system. Inset shows fixating movement of camera

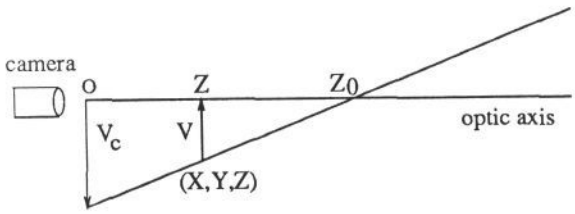

Figure 2: Geometry of kinetic depth

The depth map of an image can easily be calculated. A full treatment is given by Ballard and Ozcandarli [4]; we give the essential details here. From Figure 2 we see that if the camera is moved along a straight-line trajectory with velocity V_c, fixating at a point a distance Z_0 away, then some other point a distance Z away will appear to move with velocity V, where

$$V/V_c = -(Z_0 - Z)/Z_0$$

so that if we know V, V_c and Z_0, we can calculate Z. The robot is programmed to move the camera with known velocity about a known fixation point, i.e. we set V_c and Z_0 ourselves.

The velocity of the object point can be calculated from the optical flow in the image. If the object is at the point (X, Y, Z) in 3D space and has velocity (V_X, V_Y, V_Z), then the optical flow (u, v) at the corresponding point in the image is

$$u = -f(V_X/Z - XV_Z/Z^2), \quad v = -f(V_Y/Z - YV_Z/Z^2)$$

if the imaging plane is at $(0, 0, -f)$ where f is the focal length of the camera's lens. Restricting camera motion to small movements along the X-axis, so that $V_Y \simeq 0$, Z is almost constant and V_Z is very small (foveal approximation), and for $Z >> f$, we have $u \simeq -fV_X/Z$ and $v = 0$. Combining this equation with the brightness change constraint equation, $I_x u + I_t = 0$, where I_x and I_t are the partial derivatives of image intensity with respect to image x coordinate and time t [5], we obtain a simple equation for Z, the distance of the object from the camera:

$$Z = \frac{AI_x}{I_x + BI_t} \tag{1}$$

where

$$A = Z_0, \qquad B = -Z_0/fV_c.$$

This is the *kinetic depth equation.*

Thus, a depth map can be obtained if the time and space derivatives of the image intensity can be calculated. These derivatives and the depth Z are calculated in a Datacube MaxVideo image processing system, and the results, in the form of a false colour image with each pixel representing a depth value, can be displayed in real time on a colour monitor.

Figure 3 shows how this is done. The time derivative I_t at time t is produced by taking the difference of two consecutive images:

$$I_t(t, i, j) = [I(t, i, j) - I(t-1, i, j)]/\Delta t$$

for a pixel in column i and row j with Δt being the time between frames (40ms). This calculation is carried out using the subtraction function on the Datacube's MAXSP board. The space derivative I_x is produced for alternate images by convolution with a gradient mask on the VFIR board:

$$I_x(t, i, j) = [I(t, i+1, j) - I(t, i-1, j)]/2\Delta x$$

where Δx is the width of a pixel on the camera's imaging surface. The derivative images are circulated through the MAXSP board again where they are combined according to equation 1 in a two-input 6-bit look-up table to give

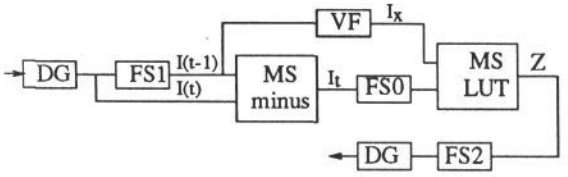

Figure 3: Vision implementation in the Datacube. The boards used are DG:digimax, FS:framestore, MS:maxsp, VF:vfir

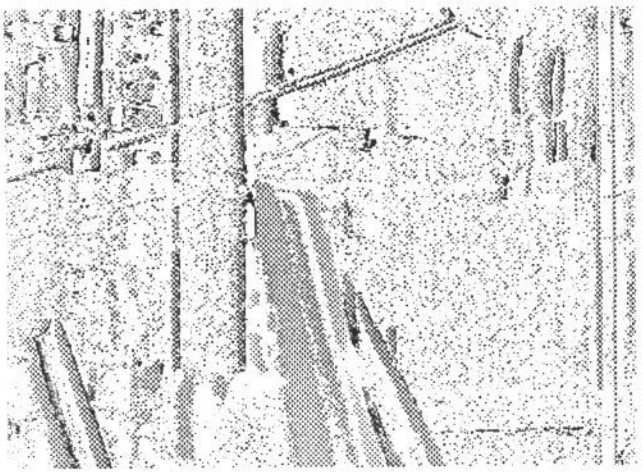

Figure 4: Depth map. Greyscale indicates depth: white is no data or 0cm, black is 255cm or further. The slanting stripes are the edges of inclined wooden strips resting on a table, the vertical stripes to the left of centre are the edges of a metal support post, and a cable crosses the top-left corner

depth values. With additional Datacube boards we would not need to recirculate the derivative images; we would be able to calculate the kinetic depth in one pass, and use an 8-bit look-up table, giving finer resolution for the calculated depths. The depth map is displayed in false colour on a monitor. A typical depth map is shown in Figure 4.

It can be seen that depth values are available in the regions where there is an appreciable intensity gradient, and that there are large, virtually noise-free, regions in between, where the computed depth value is zero. The space derivative of the image is particularly sensitive to point noise and it was necessary to reduce this by ignoring depth values calculated at those points where the intensity space derivative I_x is small.

Image feature	Computed depth (cm)	Actual depth (cm)	% difference
1	82	73	12
2	95	89	7
3	77	73	5
4	88	89	1
5	39	42	7
6	37	42	12
7	55	59	7
8	49	59	17
9	57	55	4
10	48	55	13
11	95	84	13
12	102	115	11
13	88	102	14

Table 1: Depth values

SYSTEM ANALYSIS

Table shows a comparison between computed and actual depth values for several objects. Each value is an average over 7 views of particular features in the image. The depth value of a view is computed as an average over nonzero depth values in an 8x8 window placed on the image feature; about 25% of points in a typical window give no depth information.

A depth value at any particular image point is often within 20% of the actual value, but can be up to 100% larger. The average depth value for a window was up to 60% away from the actual value, but was within 15% more than half the time. In a robotic obstacle avoidance system we would expect to calculate depths by averaging over several consecutive images; the depth values averaged over 7 images (not consecutive, and thus from noticeably different viewpoints) shown in Table are up to 20% away from the actual values, but often show less than 10% discrepancy.

These values can be compared with those of Skifstad and Jain [6], who calculated depth maps in an off-line fashion, displacing a camera a fixed distance between acquisition of each image. For camera displacements perpendicular to the optical axis, they measure (relative) depths which show a 35% discrepancy with the actual values for a sequence of images. (The relative depths were accurate only to a scale factor: the actual discrepancy may thus be larger or smaller.)

Previous experiments used a smaller robot (UMI RTX) which was not able to move smoothly in the $Z = 0$ plane

when loaded with the camera, and which jittered during the motion. Although the joint velocity of this robot was constant, the linear velocity of the gripper was not and an average value had to be used in calculations. The best depth values measured with this system showed a 30% discrepancy with the actual values, i.e. worse than the results above. Evidently some improvement in depth values was achieved by using a robot which could move with a constant linear velocity along the X-axis, in keeping with the simplifications in the mathematical treatment.

The velocity of the Adept is subject to a measurement error of 10%; velocities typically in the region of 12cm s^{-1} were used. The camera had a focal length of 16mm and a pixel width of 17.89μm and the results shown above are for a fixation distance of 85cm. It is to be expected that the system be sensitive to these parameters and also to image brightness, lighting conditions and the camera gain and we are investigating this.

Equation 1 was derived for a camera moving along a linear trajectory but it did not take into account the variation in Z_0 and Z that this would produce. Further experiments in which the camera moves along an arc will show how this affects the computed depth values. We also expect to see a pixel quantisation effect arising from the fact that changes in image intensity are continuous, but can only be measured pixelwise.

DISCUSSION

We have shown that a real-time kinetic depth system can produce reasonable depth values quickly and presented a different, less hardware-intensive implementation than that of Ballard and Ozcandarli [4].

It is clear from Table that the depth values our system computes are less accurate than those available from, say, stereo vision systems or laser stripers. However, a new depth map is produced every 80ms; using every image field instead of every frame would reduce this to 40ms, and with one extra Datacube board, this could be further reduced to 20ms. Apart from the necessity of moving the camera with a predictably linear velocity, the factors affecting the accuracy are still uncertain, although they will certainly include those mentioned in the previous section; we can calculate statistical errors in the data reduction and the discrepancy between actual and computed values, but do not yet know how large any systematic errors might be.

What accuracy do we require of our depth maps? This depends on the nature of the visual task. If the visual task is obstacle detection, which requires a simple yes/no answer (i.e. qualitative vision [7]), then estimates of depth which are quite inaccurate are sufficient for coarse robot motion. Moreover, a system which is designed to take such inaccuracy into account will be robust. If an object is to be approached and picked up by the robot, then more precise methods must be called into play, for example, the stereo visual servo system of Conkie and Chongstitvatana [8] which is also being integrated into the SOMASS system and which guides a robot hand to a visible target.

REFERENCES

1. **Gillian M. Hayes** *A Real-Time Kinetic Depth System* MSc Thesis, Department of Artificial Intelligence, University of Edinburgh, August 1989

2. **C. A. Malcolm and T. Smithers** *Symbol Grounding via a Hybrid Architecture in an Autonomous Assembly System* DAI RP 420, Department of Artificial Intelligence, University of Edinburgh, 1989

3. **Christopher M. Brown (editor): Dana H. Ballard, Timothy G. Becker, Christopher M. Brown, Roger F. Gans, Nathaniel G. Martin, Thomas J. Olson, Robert D. Potter, Raymond D. Rimey, David G. Tilley, and Steven D. Whitehead** *The Rochester Robot* Technical Report, Computer Science Department, University of Rochester, New York, August 1988

4. **Dana H. Ballard and Altan Ozcandarli** "Eye fixation and early vision: kinetic depth" In *Proceedings of the Second International Conference on Computer Vision, Tampa, FLA, December 5–8, 1988*, pp 524–531, IEEE Computer Society Press, 1988

5. **Berthold K. P. Horn and Brian G. Schunck** "Determining optical flow" *Artificial Intelligence*, Vol. 17 (1981) pp 185–203

6. **K. Skifstad and R. Jain** *Range Estimation from Intensity Gradient Analysis.* CSE-TR-02-88, Computer Science and Engineering Division, University of Michigan, 1988

7. **John Aloimonos** "Purposive and qualitative active vision" Paper presented at the 1st European Conference on Computer Vision, Antibes, France, April 23–26, 1990 and at the Workshop on Control of Perception in Active Vision, Antibes, France, April 27, 1990

8. **Alistair Conkie and Prabhas Chongstitvatana** "An uncalibrated stereo visual servo system" *Proceedings of the British Machine Vision Conference 1990, Oxford, September 24–27, 1990*