NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-769

A System for Extracting 3-Dimensional Measurements From a Stereo Pair of TV Cameras

(NASA-CR-147149) A SYSTEM FOR EXTRACTING 3-DIMENSIONAL MEASUREMENTS FROM A STEREO PAIR OF TV CAMERAS (Jet Propulsion Lab.) 19 p HC $3.50 CSCL 14E

N76-24546

Unclas

G3/35 28205

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 15, 1976

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-769

A System for Extracting 3-Dimensional Measurements From a Stereo Pair of TV Cameras

Y. Yakimovsky

R. Cunningham

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 15, 1976

PREFACE

The work described in this report was performed by the Science Data Analysis Division of the Jet Propulsion Laboratory.

CONTENTS

FIGURES

ABSTRACT

Obtaining accurate three-dimensional (3-D) measurement from a stereo pair of TV cameras is a task requiring camera modeling, calibration, and the matching of the two images of a real 3-D point on the two TV pictures. A system which models and calibrates the cameras and pairs the two images of a real-world point in the two pictures, either manually or automatically, was implemented at JPL. This system is operating and provides three-dimensional measurements resolution of ±1 mm at distances of about 2 m.

I. INTRODUCTION

Extracting 3-D measurement (X, Y, Z) from a stereo pair of two-dimensional images is a task important in various applications. It is used in orthodentistry (Refs. 1 and 2) as well as in various automatic control and robot systems. The use of a computer as a component in a system which measures 3-D position of feature points has been tried before (Refs. 3-7). The computer solves the system equations, finds and stores the cameras' calibration parameters, and pairs the points in the two images. Our present system produces accurate 3-D measurement from on-line TV images. We have obtained this result by using two rigidly connected solid state TV cameras as sensors (GE TN-2000, which is a charge injection device TV camera) with highly linear 50-mm lenses. This results in a stable and linear two-camera system. In addition, we used an accurate and flexible camera calibration scheme and a linear camera model.

The pairing of points in the two images is done either automatically, by the use of a correlation algorithm, or manually by an operator. The correlation algorithm works successfully due to accurate calibration and provides good matching which, in turn, provides accurate 3-D measurements. The following describes the camera modeling, the camera's calibration scheme, the algorithm which pairs stereo images of a point correlation, and the equations which solve for the real-world position of the point whose two images in the two pictures were paired.

II. THE CAMERA MODEL

The light sensor in the CID cameras (Fig. 1) is a rectangular area containing a two-dimensional array of 188×244 light sensitive elements. The video output (the TV picture) of each camera is digitized so that the picture appears to the computer (in our case, SPC-16/85 with a 64K 16-bit core) as a two-dimensional array 188×244 of 8-bit numbers. The elements of that array are indexed by (i, j) where $0 \le i \le 243$, $0 \le j \le 187$. This array is called the gray level array and the values of these numbers correspond to the brightness of the image.

The calibration parameters allow matching of each element (i, j) of the image with a ray $\vec{C} + \lambda \cdot \vec{R}(i,j)$ $0 \le \lambda \le \infty$ in the real 3-D world. So that if a real-world point $\vec{P}$ is imaged on picture element (i, j), it must be on that ray; that is, $\vec{P} = \vec{C} + \lambda \cdot \vec{R}(i,j)$ for some $\lambda \ge 0$. We assume that the cameras are geometrically linear, an assumption which is sufficient, considering the linear sensor array and the high quality lens that is used. The assumption of linearity means that there are $\vec{C}_1$, $\vec{H}_1$, $\vec{A}_1$ and $\vec{V}_1$ for the first camera and $\vec{C}_2$, $\vec{H}_2$, $\vec{A}_2$, and $\vec{V}_2$ for the second camera such that if a real-world point $\vec{P}$ is imaged on image coordinates (I_1, J_1) in the first camera and on image coordinate (I_2, J_2) in the second camera, then:

$$I_1 = \frac{\left(\vec{P} - \vec{C}_1, \vec{H}_1\right)}{\left(\vec{P} - \vec{C}_1, \vec{A}_1\right)} \tag{1}$$

$$J_1 = \frac{\left(\vec{P} - \vec{C}_1, \vec{V}_1\right)}{\left(\vec{P} - \vec{C}_1, \vec{A}_1\right)} \tag{2}$$

$$I_2 = \frac{\left(\vec{P} - \vec{C}_2, \vec{H}_2\right)}{\left(\vec{P} - \vec{C}_2, \vec{A}_2\right)} \tag{3}$$

$$J_2 = \frac{\left(\vec{P} - \vec{C}_2, \vec{V}_2\right)}{\left(\vec{P} - \vec{C}_2, \vec{A}_2\right)} \tag{4}$$

The semantic meaning of these parameter vectors is as follows: $\vec{C}_1$ and $\vec{C}_2$ are the positions of the focal center of the first and the second camera correspondingly, measured in the external coordinate system. (Hence, $\vec{P} - \vec{C}$ is a vector from the focal center of the camera towards $\vec{P}$). $\overrightarrow{A_1}(\overrightarrow{A_2})$ is a unit vector in the direction in which the first (second) camera is pointed. It is thus the direction of the symmetry axis of the lens as measured in the external coordinate system.

$\vec{H}_1(\vec{H}_2)$ is called the horizontal vector of the first (second) camera. $\overrightarrow{H_1}$ and $\vec{H}_2$ are not unit vectors and are not perpendicular to $\vec{A}$.

$\vec{V}_1(\vec{V}_2)$ is called the vertical vector of the first (second) camera. It is not a unit vector and it is not necessarily perpendicular to either $\vec{A}_1(\vec{A}_2)$ or $\vec{H}_1(\vec{H}_2)$.

The meaning of the $\vec{H}$ and $\vec{V}$ vectors is defined by Eqs. 1-4. $\vec{H}_1 - (\vec{H}_1, \vec{A}_1)\,\vec{A}_1$ is a vector in the real-world direction of the line of the sensor elements for which (j = 0), i = 0,..., 187 in the first camera. These elements produce the horizontal line on the first image. $\vec{V}_1 - (\vec{V}_1 \vec{A}_1)\vec{A}_1$ is a vector in the direction of the sensor elements on the vertical line (i=0) j=0,...,243 in the sensor array. $\vec{H}_2, \vec{V}_2$ have the identical meaning in the second camera.

III. THE CALIBRATION SYSTEM

The calibration process attempts to find the parameters $\vec{C}$, $\vec{A}$, $\vec{H}$ and $\vec{V}$ of both cameras. This is done as follows.

A set of n known points $\vec{P}_1$, ---, $\vec{P}_n$ in the real world are imaged on the camera, and the picture coordinate on which they are images (i_1, j_1) --- (i_n, j_n) are obtained and stored. In our case, it is done by moving the robot arm in front of the camera (Fig. 1) and taking pictures of the arm. A pattern recognition algorithm finds automatically a feature point of the arm in each picture. The arm is calibrated so that the actual coordinate $\vec{P}_m$ of the m-th feature point in the real world is known. The (i_m, j_m), which is the image of $\vec{P}_m$, on the screen, is found by the pattern recognition system. This set of matches of $\vec{P}_m$ with (i_m, j_m) is used to solve for $\vec{C}$, $\vec{A}$, $\vec{H}$ and $\vec{V}$ by substituting $\vec{P}_m$, i_m, j_m into Eq. (1). This yields

$$i_m = \frac{(\vec{P}_m - \vec{C}, \vec{H})}{(\vec{P}_m - \vec{C}, \vec{A})} \tag{5}$$

which is equivalent to

$$i_m \cdot (\vec{P}_m - \vec{C}, \vec{A}) = (\vec{P}_m - \vec{C}, \vec{H})$$

or

$$(\vec{P}_m - \vec{C}, i_m \cdot \vec{A} - \vec{H}) = 0$$

let

$$\vec{P}_m = \begin{Bmatrix} P_{m,1} \\ P_{m,2} \\ P_{m,3} \end{Bmatrix}, \quad \vec{C} = \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \end{Bmatrix}, \quad \vec{A} = \begin{Bmatrix} A_1 \\ A_2 \\ A_3 \end{Bmatrix}, \quad \vec{H} = \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \end{Bmatrix},$$

and let

$$C_H = (\vec{C}, \vec{H}) \text{ and } C_A = (\vec{C}, \vec{A})$$

Then we get the following linear system of n- equations in the 8 unknowns A_1, A_2, A_3, H_1, H_2, H_3, C_A, C_H:

$$(P_{m,1} \cdot i_m) A_1 + (P_{m,2} \cdot i_m) A_2 + (P_{m,3} \cdot i_m) A_3 - P_{m,1} \cdot H_1 - P_{m,2} \cdot H_2 - P_{m,3} \cdot H_3 - i_m \cdot C_A + C_H = 0 \tag{6}$$

where

$$m = 1, \cdots, n$$

This linear system of n equation by 8 variables is reduced to (n - 4) equations of 3 variables by setting $A_1 = 1$ and setting all coefficients of H_1, H_2, H_3 and C_H to 0 by combining 5 equations at a time. We solve for the optimal A_2, A_3, C_A by the standard optimization approach to the solution of an n x m linear system $S\vec{X} = \vec{B}$ where S is an n by m matrix and n > m. We use the fact that the $X_0 \epsilon R^m$ that satisfies

$$\min_{X \epsilon R^m} \|S \cdot \vec{X} - \vec{B}\|^2 = \|S \vec{X}_o - \vec{B}\|$$

is the same as the $\vec{X}_o$, which solves

$$S^T S \vec{X}_o = S^T \vec{B}.$$

Then $(1, A_2, A_3)$ is scaled to one $\left(\|\vec{A}\| = 1\right)$ and C_H, H_1, H_2, H_3 are computed from the original n equations system after the known $\vec{A}$ and C_H are substituted in the equation. Similarly, C_V, V_1, V_2, V_3 are computed by using Eq. (2) as follows:

$$j_m = \frac{\left(\vec{P}_m - \vec{C}_1, \vec{V}\right)}{\left(\vec{P}_m - \vec{C}, \vec{A}\right)} \tag{7}$$

Next, $\vec{C} = (C_1, C_2, C_3)$ is computed by solving the system:

$$C_A = C_1 \cdot A_1 + C_2 \cdot A_2 + C_3 \cdot A_3$$

$$C_H = C_1 \cdot H_1 + C_2 \cdot H_2 + C_3 \cdot H_3$$

$$C_V = C_1 \cdot V_1 + C_2 \cdot V_2 + C_3 \cdot V_3$$

Note that C_A, C_H, and C_V are known at this point.

After $\vec{C}$, $\vec{A}$, $\vec{H}$, and $\vec{V}$ were computed, the differences

$$\left| i_m - \frac{\left(\vec{P}_m - \vec{C}, \vec{H}\right)}{\left(\vec{P}_m - \vec{C}, \vec{A}\right)} \right| \text{ and } \left| j_m - \frac{\left(\vec{P}_m - \vec{C}, \vec{H}\right)}{\left(\vec{P}_m - \vec{C}, \vec{A}\right)} \right|$$

are computed. These values supply information as to the adequacy of the computed camera model; e.g., how much the computed (i, j) deviate from the actual (i_m, j_m). At present, the deviation does not exceed errors anticipated from the cameras' resolution coupled with the anticipated errors resulted from errors in the arm calibration parameters which are used to compute the feature point position (the $\vec{P}_m$'s).

IV. PROJECTING PICTURE ELEMENTS INTO THE REAL WORLD

The cameras' calibration parameters are used to project picture points onto the real world. A real point $\vec{P}$ imaged on point (I_1, J_1) in the first camera will satisfy the relation in Eq. (1) and Eq. (2). These equations can be inverted as follows to solve for possible $\vec{P}$ where I_1 and J_1 are given.

Eq. 1

$$I_1 = \frac{\left(\vec{P} - \vec{C}_1, \vec{H}_1\right)}{\left(\vec{P} - \vec{C}_1, \vec{A}_1\right)}$$

Eq. 2

$$J_1 = \frac{\left(\vec{P} - \vec{C}_1, \vec{V}_1\right)}{\left(\vec{P} - \vec{C}_1, \vec{A}\right)}$$

which is equivalent to

$$(\vec{P} - \vec{C}_1, \vec{H}_1 - I_1\vec{A}_1) = 0 \quad \text{and} \quad (\vec{P} - \vec{C}_1, \vec{V}_1 - J_1\vec{A}_1) = 0$$

Hence, $\vec{P}-\vec{C}_1$ is perpendicular to both $\vec{H}_1-I_1\vec{A}_1$ and $\vec{V}_1-j_1\vec{A}_1$ and therefore

$$\vec{P}-\vec{C}_1 \parallel (\vec{V}_1-J_1\vec{A}_1) \times (\vec{H}_1-I_1\vec{A}_1)$$

or

$$\vec{P} = \vec{C}_1 + \lambda \cdot \vec{R}_1(I_1, J_1), \qquad \lambda > 0$$

where $\vec{R}_1(I_1 J_1) \parallel (\vec{V}_1-J_1\vec{A}_1) \times (\vec{H}_1-I_1\vec{A}_1)$ and $\vec{R}_1(I_1, J_1)$ is a unit vector.

In other words, given that a point $\vec{P}$ is imaged on (I_1, J_1) in the first camera, we know that $\vec{P}$ must lie on the real-world line L_1:

$$\underline{L_1} = \vec{C}_1 + \lambda \cdot \vec{R}_1(I_1, J_1), \qquad \lambda < +\infty$$

where

$$\vec{R}_1(I_1, J_1) \parallel (\vec{V}_1 + J_1 \vec{A}_1) \times (\vec{H}_1 + I_1\vec{A}_1)$$

and $\vec{R}_1\ (I_1, J_1)$ is normalized to 1.

Usually we know the bounds on the distance of $\vec{P}$ from the focal center of the cameras (in our case, a point typically will be at a distance varying from 0.5 m in front of the camera to $(+\infty)$. Two points are computed on the line $\vec{C}_1 + \lambda \cdot \vec{R}_1(I_1 J_1)$, one 0.5 m in front of the first camera and one at infinity. These two points are projected on the image of the second camera using Eqs. (3) and (4). The near one on $T_n = (I_n, J_n)$ and the far one on $T_f = (I_f, J_f)$. The point $\vec{P}$ itself will be imaged somewhere on the straight line in the two-dimensional picture connecting the two points T_n and T_f. The exact image of $\vec{P}$ on that line in the second image is found manually (using a cursor) or automatically (using the correlation algorithm described in the next section). Once the (I_2, J_2) which are the picture coordinates on which $\vec{P}$ is imaged in the second camera are found, the position of $\vec{P}$ can be computed.

(I_2, J_2) define a line in space,

$$\underline{L_2} = \vec{C}_2 + \lambda \vec{R}_2(I_2, J_2),$$

by the same mechanism which L_1 was defined. The point $\vec{P}$ is computed to be the 3-D point for which the sum of its distances to the two lines is minimized. We do not use the intersector of the two lines because it may not exist due to numerical errors in calibration. That is, the computed P will be a three-dimensional point which satisfies:

$$\min_{\vec{X} \in R^3} \left(\|\vec{X} - \vec{C}_1\|^2 - \left(\vec{X} - \vec{C}_1, \vec{R}_1(I_1, J_1)\right)^2 + \|\vec{X} - \vec{C}_2\|^2 - \left(\vec{X} - \vec{C}_2, \vec{R}_2(I_2, J_2)\right)^2 \right) = \|\vec{P} - \vec{C}_1\|^2 - \left(\vec{P} - \vec{C}_1, \vec{R}_1(I_1, J_1)\right)^2$$

$$+ \|\vec{P} - \vec{C}_2\|^2 - \left(\vec{P} - \vec{C}_2, \vec{R}_2(I_2, J_2)\right)^2.$$

The reader is reminded that

$$\|\vec{X} - \vec{C}\|^2 - (\vec{X} - \vec{C}, \vec{R})^2$$

where $\|\vec{R}\| = 1$ is the distance between $\vec{X}$ and the line $\vec{C} + \lambda \vec{R}$.

Hence, the problem of extracting the 3-D coordinates of a point is reduced to finding and matching the two images of that point on the two video pictures.

V. THE STEREO CORRELATION ALGORITHMS

This section deals with the problem of matching the two images of a real-world point $\vec{P} = (X, Y, Z)$, which is in the field of view of both cameras. The method used is correlation of grey levels in the right and left images.

Let $T_1 = (I_1, J_1)$ be the image of $\vec{P}$ in the right camera and $T_2 = (I_2, J_2)$ be the image of $\vec{P}$ in the left camera. As described above, T_1 and T_2 define rays from the right and left cameras to the point $\vec{P}$. The intersection of these rays gives the coordinates (X, Y, Z) of the point $\vec{P}$.

The problem of correlation is to find the points T_1 and T_2 which are images of the same point $\vec{P}$.

In the current implementation, T_1 is selected with a cursor on the Ramtek display of the right image (see Fig. 3). A set of N points called a <u>mask</u> is selected from a small window centered at T_1. Currently, two types of masks are used. One is a set of concentric diamonds $D_0, D_1, \ldots, D_k$ where $D_0 = T_1$ and

$$D_i = \left\{(I, J): \left|I_1 - I\right| + \left|J_1 - J\right| = d_i\right\} i = 1, \ldots, k$$

Typical values for d_i are $d_i = 1$, $d_2 = 2$, $d_3 = 4$, $d_4 = 8$, (N = 61) (see Fig. 4).

The second type of mask (see Fig. 5) consists of 4 line segments defined by an integer k as follows:

(1)	horizontal	$(I_1 - k, J_1)$ to $(I_1 + k, J_1)$
(2)	vertical	$(I, J_1 - k)$ to $(I, J_1 + k)$
(3)	45°	$(I_1 - k, J_1 - k)$ to $(I_1 + k, J_1 + k)$
(4)	-45°	$(I_1 - k, J_1 + k)$ to $(I_1 + k, J_1 - k)$

A typical value of k is 8 (N = 65).

In either case, the mask representation is generalized as two sequences of N displacements ΔI_i and ΔJ_i where $i = 1, \ldots, N$. Thus, each point m_i of the mask can be expressed as $(I+\Delta I_i, J+\Delta J_i)$ with the mask centered at an arbitrary point (I, J).

With the mask centered at T_1, the right image is sampled to find the grey level X_k at each point m_k of the mask. These values are stored as a third N element array.

To find T_2, a search is initiated along a line segment L_2 in the left image. L_2 is determined, as mentioned above, by projecting the end points of a segment in the real 3-D world on the ray $\vec{C}_1 + \lambda \cdot \vec{R}(I_1, J_1)$ $0 \le \lambda \le \infty$ onto the left image. This real-world segment is chosen according to the context of the point $\vec{P}$. For instance, if $\vec{P}$ is the centroid of a rock to be picked up by the manipulator, then $\vec{P}$ is approximately 2 meters from the camera center, so the endpoints on the real world segment will be taken as the points 1.5 m and 2.5 m from the right camera on R_1.

For each point T_k on L_2, the following happens:

(a) With the mask centered at $T_k = (I_k, J_k)$, the left image is sampled to find the grey level Y_i at each point $m_i = (I_k+\Delta I_i, J_k+\Delta J_i)$ of the mask on the left image.

(b) The correlation coefficient C_k is computed by

(1)

$$C_k = \frac{\sum_{i=1}^{N} (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{N} (X_i - \overline{X})^2 \sum_{i=1}^{N} (Y_k - \overline{Y})^2}}, \qquad -1 \le C_k \le 1$$

where

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N}, \quad \overline{Y} = \frac{\sum_{i=1}^{N} Y_i}{N}$$

An equivalent form of (1) which requires the lesser amount of computations is

$$(2) \qquad \frac{\sum_{i=1}^{N} X_i Y_i - \left(\sum_{i=1}^{N} X_i \cdot \sum_{i=1}^{N} Y_i\right) \Big/ N}{\sqrt{\left(\sum_{i=1}^{N} X_i^2 - \left(\sum_{i=1}^{N} X_i\right)^2 \Big/ N\right) \cdot \left(\sum_{i=1}^{N} Y_i^2 - \left(\sum_{i=1}^{N} Y_i\right)^2 \Big/ N\right)}}$$

The X_i's remain constant while T_1 is fixed and we search for the appropriate T_2. Hence, we can maximize expression (3) in the search instead of expression (2) and save some compute time:

$$(3) \qquad \frac{D \cdot |D|}{N \cdot \sum_{i=1}^{N} Y_i^2 - \left(\sum_{i=1}^{N} Y_i\right)^2}$$

where

$$D = \sum_{i=1}^{N} Z_i \cdot Y_i - S_x \cdot \sum_{i=1}^{N} Y_i$$

$$S_x = \sum_{i=1}^{N} X_i$$

$$Z_i = N \cdot X_i$$

S_x and the Zi's which depend only on T_1 are maintained through the search and do not have to be recomputed. Hence the time spent on each correlation is consumed almost exclusively in computing

$$\sum_{i=1}^{N} Y_i, \sum_{i=1}^{N} Y_i^2, \text{ and} \sum_{i=1}^{N} Z_i Y_i$$

for each candidate T_k.

The correlation coefficient C_k will be equal to 1 if there is a perfect linear equivalence between X_1 and Y_1 for $1 \leq i \leq N$, and less than 1 otherwise. Thus, the point T_2 should correspond to the maximum value of C_k. Since there may be several points with grey level distributions similar to that around the actual point of interest T_1, additional steps are taken to increase the probability of finding the correct point T_2.

If the values of C_k are plotted as a function of T_k, the curve might look like that of Fig. 6.

In Fig. 6, T_2 should correspond to a local maximum on the curve. As the C_k are generated, a record is kept of the (at most) four greatest local maxima which occur, as well as the absolute maximum. The correlation at each maximum point is recomputed using a new mask consisting of all points in a 15×15 square. Then T_2 is taken as the point which gives the highest correlation with this mask.

VI. MASK SELECTION

For the correlation method to be effective, there must be significant information in the grey levels around the point T_1 covered by the mask so that the correlation value at T_2 will be significantly greater than at other points in the neighborhood of T_2. Therefore, before correlation on both images is started, two tests are applied to T_1 to select the proper size mask (Ref. 7).

The first involves comparing the variance V_m of the grey levels in the mask at T_1 to the noise level variance V_C of the camera. If $V_m < 3 \cdot V_c$, the point is considered unacceptable for correlation because the area in the mask is too homogeneous and there is not sufficient information to discriminate between points.

The second test is autocorrelation. This involves computing the correlation of a mask centered at T_1 with masks centered at points in the neighborhood of T_1 in the same (right) image. The neighborhood used is the line segment $(I_1 - 4, J_1)$ to $(I_1 + 4, J_1)$. If the correlation at the neighborhood points is significantly less than 1, then correlation with the left image proceeds. A successful autocorrelation test usually implies the existence of a local maximum on the correlation curve (Fig. 6) at the desired match point T_2 in the left image.

If either of these tests fails, the mask is expanded in size until the area covered by the mask exceeds the boundaries of the homogeneous region containing the point T_1. At this point, the variance and autocorrelation tests will be satisfied. The mask is expanded K-1 times by factors of 2, 3, . . . , K times the original size, until an acceptable mask is found. Presently the value of K is 7. If an acceptable mask is not found after K-1 expansions, no correlation will be attempted.

REFERENCES

1. S. Berkowitz and J. Krischen, Quantitative Analysis of Cleft Palate Casts a Geometric Study, The Cleft Palate Journal, Vol. 11, No. 2, April, 1974.
2. S. Berkowitz, Stereophotogrammetric Analysis of Casts of Normal and Abnormal Palates, American Journal of Orthodontists, July, 1971.
3. Levine, M. D., D. A. O'Handley and G. M. Yagi, Computer Determination of Depth Maps, Computer Graphics and Image Processing (1973), 2.
4. Lynn H. Quan, Computer Comparison of Pictures, Stanford University Artificial Intelligence Lab., AIM-144, 1971.
5. Yoram Yakimovsky, "Extracting Depth Information From a Stereo Pair," in Proceedings of the Milwaukee Symposium on Automatic Control, 1974, Milwaukee, Wisconsin, March 28-30, 1974.

6. March Jo-Hannah, Computer Matching of Areas in Stereo Images, Stanford University AI Artificial Intelligence Lab., 239, 1974.

7. Hans Moraveck, informal written and oral communication, Final Report, Visual Navigation Project, Stanford University, Artificial Intelligence Lab., Jan. 1975 (JPL supported research).

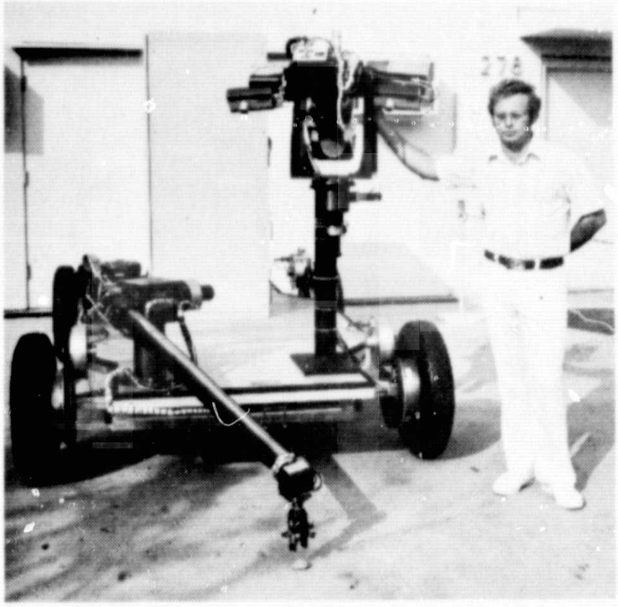

Fig. 1. The hardware configuration: the two cameras and the arm

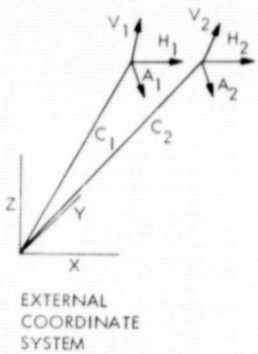

Fig. 2. The linear model of the camera pair

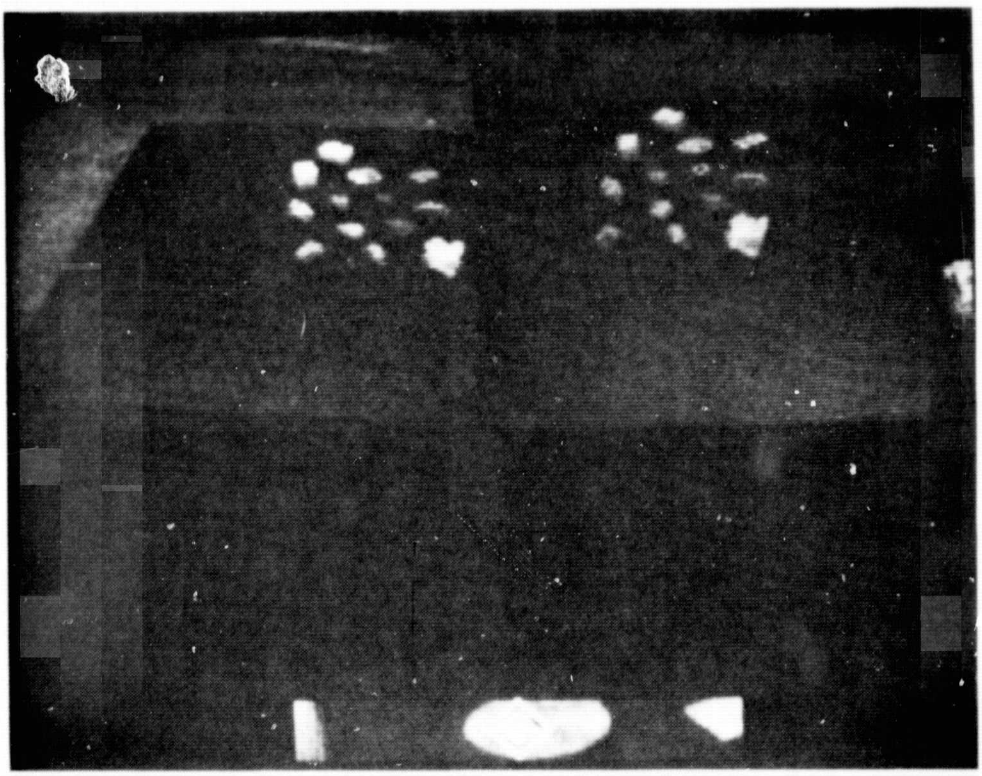

Fig. 3. Two-image stereo display with cursor overlay on a pair of matching points

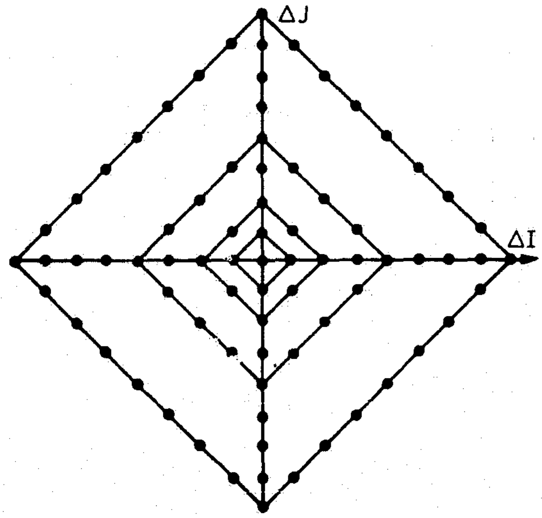

Fig. 4. Diamond mask

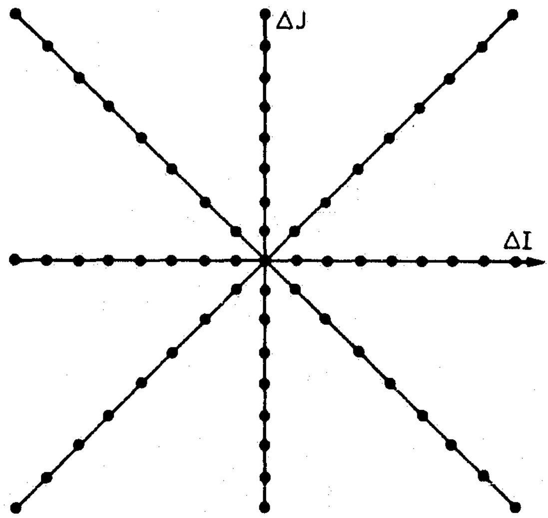

Fig. 5. Four-line segment mask

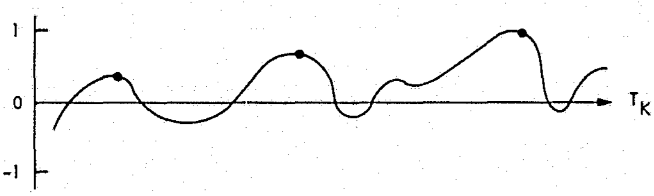

Fig. 6. Values of C_k plotted as a function of T_k

NASA–JPL–Coml., L.A., Calif.