An active multibaseline stereo system with real-time image acquisition

Sing Bing Kang, Jon Webb, C. Lawrence Zitnick,
and Takeo Kanade

September 1994

CMU-CS-94-167

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3891

This research was partially supported by the Advanced Research Projects Agency of the Department of Defense under contract number F19628-93-C-0171, ARPA order number A655, “High Performance Computing Graphics,” monitored by Hanscom Air Force Base. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARPA or the U.S. government.

Keywords: Parallel programming, Vision and Scene Understanding, Robotics, Digitization, Computer Vision.

Abstract

We describe our four-camera multibaseline stereo system in a convergent configuration and our implementation of a parallel depth recovery scheme for this system. Our system is capable of image capture at video rate. This is critical in applications that require three-dimensional tracking. We obtain dense stereo depth data by projecting a light pattern of frequency modulated sinusoidally varying intensity onto the scene, thus increasing the local discriminability at each pixel and facilitating matches. In addition, we make most of the camera view areas by converging them at a volume of interest. Results indicate that we are able to extract stereo depth data that are, on the average, less than 1 mm in error at distances between 1.5 to 3.5 m away from the cameras.

List of Figures

List of Tables

1 Introduction

Binocular stereo vision is a simple and flexible method by which three-dimensional (range) information of a scene can be obtained. Therefore, it is not surprising to find that stereo is a very active area of research [2]. The geometrical issues in stereo have also been well explored [6]. The primary drawback of stereo is the problem with image point correspondence (for a survey of correspondence techniques, see [5]). The trade-off between accuracy (which is aided by a wide baseline, or separation between the cameras) and ease of correspondence (which is simpler with a narrow baseline) has been mitigated using multiple cameras or camera locations. Such an approach has been termed *multibaseline stereo* [12].

Stereo vision is computationally intensive. Fortunately, the spatially repetitive nature of depth recovery lends itself to parallelization. This is especially critical in the case of multibaseline stereo with high image resolution and the practical requirement of timely extraction of data. A number of researchers have worked on fast implementation of stereo (e.g., [11], [13], [14]).

In this report, we describe our implementation of a depth recovery scheme implemented in iWarp for a four-camera multibaseline stereo in a convergent configuration. Our system is capable of image capture at video rate. This is critical in applications that require tracking in three dimensions (an example is [10]). One method to obtain dense stereo depth data is to interpolate between reliable pixel matches [8]. However, the interpolated values may not be accurate. We obtain accurate dense depth data by projecting a light pattern of sinusoidally varying intensity onto the scene, thus increasing the local discriminability at each pixel. In addition, we make the most of the camera view areas by converging them at a volume of interest. Experiments have indicated that we are able to extract stereo depth data that are, on the average, less than 1 mm in error at distances between 1.5 to 3.5 m away from the cameras.

We introduce the notion of an *active* multibaseline stereo for extraction of dense stereo range data in Section 2. The principle of multibaseline stereo is explained, and in addition, we justify our use of the camera system in a convergent configuration. In this section, we briefly describe our image acquisition system that enables us to capture intensity images at video rate (30 Hz). Before the camera system can be used, it must be calibrated; this procedure is described in Section 3.

Prior to depth recovery, we apply a warping operation called *image rectification* to the set of images as a preprocessing step for computational reasons; this warping operation is described in Section 4. Our implementation of the depth recovery algorithm is subsequently detailed in this section.

Finally, we present results of our experiments in Section 5, analyze the sources of error in our system in Section 6, and summarize our work in Section 7.

2 The active 4-camera system

Our multibaseline camera system is shown in Fig. 1. It comprises four cameras mounted on a plain metal bar, which in turn is mounted on a sturdy tripod stand; each camera can be rotated about a vertical axis and fixed at discrete positions along the bar. The four camera video signals are all synchronized by ganging the genlock signals.

Fig. 1 **The 4-camera system**

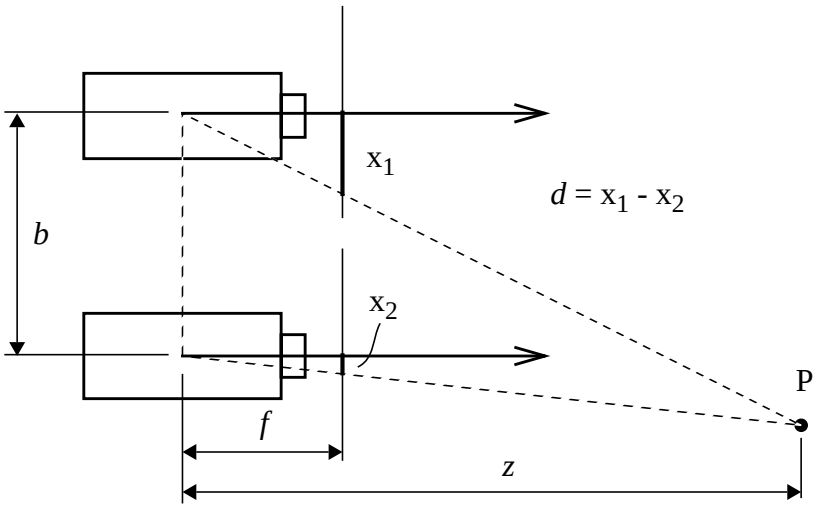
In addition to the camera, we use a projector to cast a pattern of sinusoidal varying intensity (active lighting) onto the scene. This notion of an *active multibaseline stereo* allows a denser depth map as a result of improved local scene discrimination and hence correspondence.

2.1 The principle of multibaseline stereo

In binocular stereo where the two camera axes are parallel, depth can easily be calculated given the disparity (the shift in position for corresponding points between the images). If the focal length of both cameras is *f*, the baseline *b* and disparity *d*, then the depth *z* is given by $z = f \cdot b / d$ (Fig. 2).

In multibaseline stereo, more than two cameras or camera locations are employed, yielding multiple images with different baselines [12]. In the parallel configuration, each camera is a lateral displacement of the other. From Fig. 2, $d = f \cdot b / z$ (we assume for illustration that the cameras have identical focal lengths).

For a given depth, we then calculate the respective expected disparities relative to a reference camera (say, the leftmost camera) as well as the sum of match errors over all the cameras. (An example of a match error is the image difference of image patches centered at corresponding points.) By iterating the calculations over a given resolution and interval of depths, the depth associated with a given pixel in the reference camera is taken to be the one with the lowest error.



From similar triangles,

$$\frac{d}{b} = \frac{f}{z}$$

Fig. 2 **Relationship between the baseline *b*, disparity *d*, focal length *f*, and depth *z***

The multibaseline approach has the advantage of reducing mismatches during correspondences due to the simultaneous multiple baselines. In addition, it produces a statistically more accurate depth value [12]. However, using multiple cameras alone does not solve the problem of matching ambiguity that occurs with smooth untextured object surfaces in the scene. This is the reason why the idea of using active lighting in the form of a projected pattern on the scene is important. The projected pattern on object surfaces in the scene helps in disambiguiting local matches in the camera images.

2.2 Why use a verged camera configuration?

The primary problem associated with a stereo arrangement of parallel camera locations is the limited overlap between the fields of views of all the cameras. The percentage of overlap increases with depth. The primary advantage is the simple and direct formula in extracting depth.

The parallel camera configuration is suitable for outdoor applications where accuracy is not of utmost importance while speed is (e.g., [13]). A problem with this configuration is the low percentage of overlap in the field of views of the cameras.

Verging the cameras at a specific volume in space is optimal in an indoor application where maximum utility of the camera visual range is desired and the workspace size is constrained and known *a priori*. Such a configuration is illustrated in Fig. 3. One such application is the tracking of objects in the Assembly Plan from Observation project [9]. The aim of the project is to enable a robot system observe a human perform a task, understand the task, and replicate that task using a robotic manipulator. By continuously monitoring the human hand motion, motion breakpoints such as the point of grasping and ungrasping an object can be extracted [10]. The verged multibaseline camera system can extend the capability of the sys-

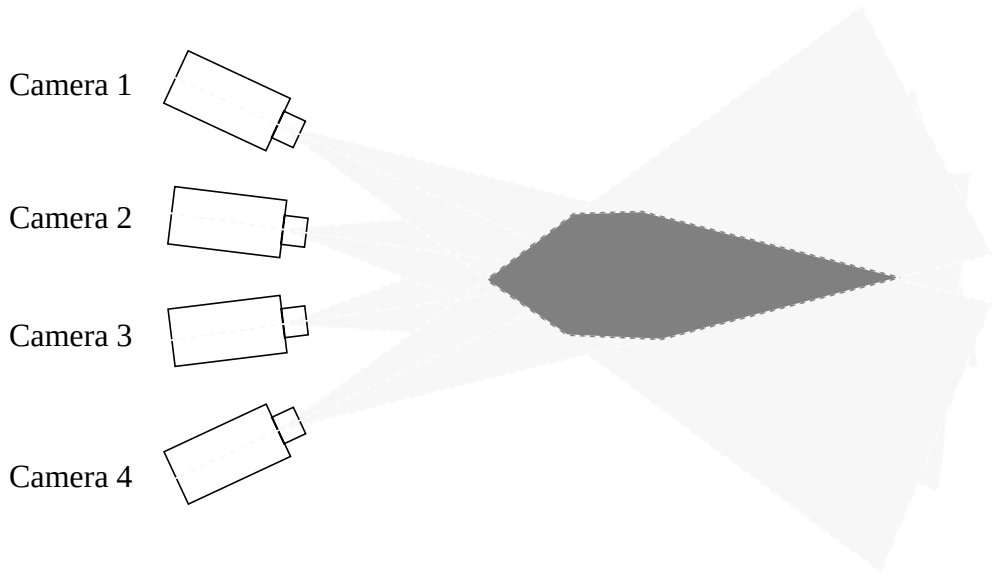


Fig. 3 **A verged camera configuration (dark shaded area is the common 3D space viewable from all cameras).**

tem to tracking the object being manipulated by the human. For this purpose, we require fast image acquisition (though processing is not as critical) and accurate depth recovery.

2.3 Video-rate image acquisition system

Our image acquisition system consists of the physical camera setup described earlier in this section, the video interface board, and the 8×8 matrix of iWarp cells (Fig. 4). Each iWarp component contains a 20 MFLOPS computation engine and low-latency (100-150 ns) communication engine for interfacing with other iWarp cells [3]. The existing iWarp system is an 8×8 torus of iWarp cells, half of which have 16 MB DRAMS per cell. The video interface, which is described in detail elsewhere [17], is connected directly to the iWarp cell through the memory interface; the digitized video data is routed and distributed at video rate to the DRAMs by taking advantage of iWarp's systolic design [4].

3 Camera calibration

Before data images can be taken and the scene depth recovered, we must first calibrate the camera configuration. Calibrating the camera configuration refers to the determination of the extrinsic (relative pose) and intrinsic (optic center offset, focal length and aspect ratio) camera parameters. The pinhole camera model is assumed in the calibration process. The origin of the verged camera configuration coincides with that of the leftmost camera.

A printed planar dot pattern arranged in a 7×7 equally spaced grid is used in calibrating the cameras; images of this pattern are taken at known depth positions (five in our case). An example set of images taken by the camera system is shown in Fig. 5.

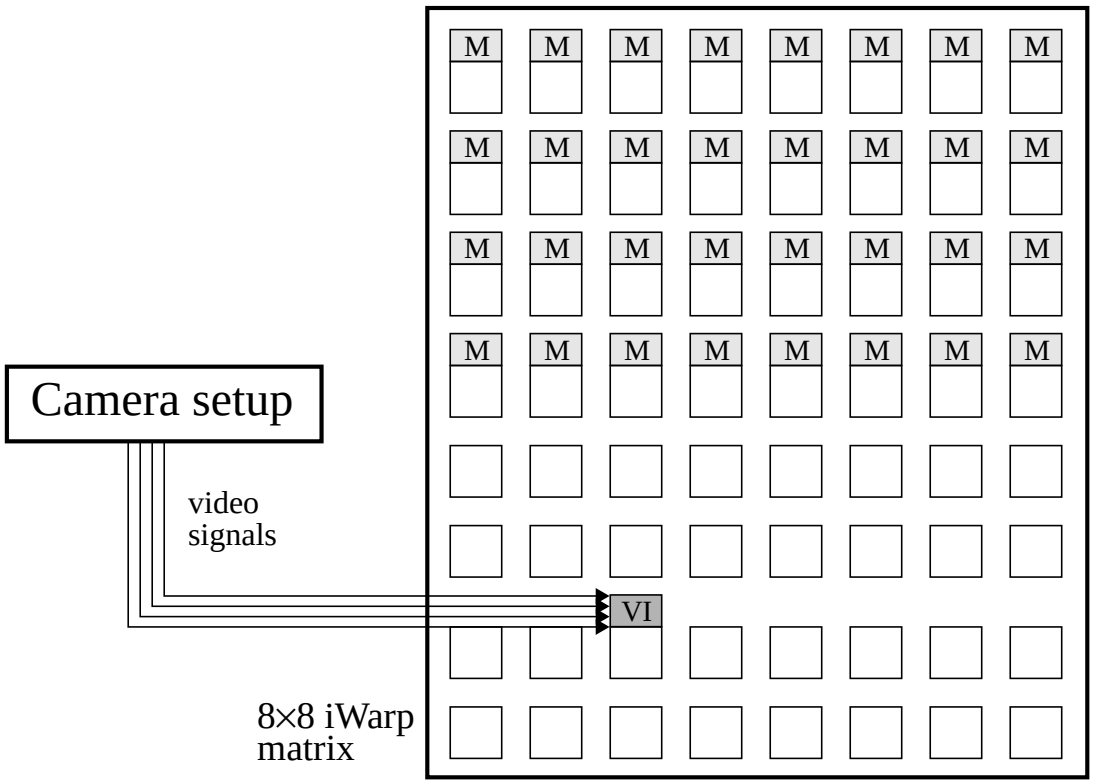
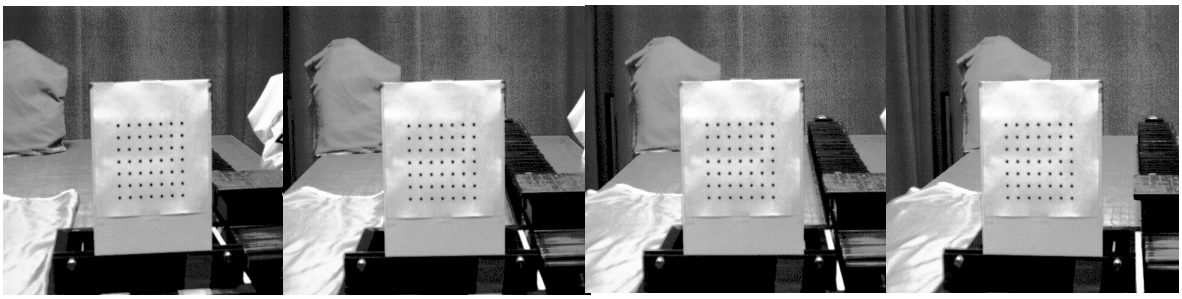


Fig. 4 **Block diagram of the image acquisition system. The shaded boxes labeled "M" indicate the 16M DRAMs connected to local iWarp cells while the shaded box labeled "VI" refers to the video interface connected to one of the iWarp cells.**

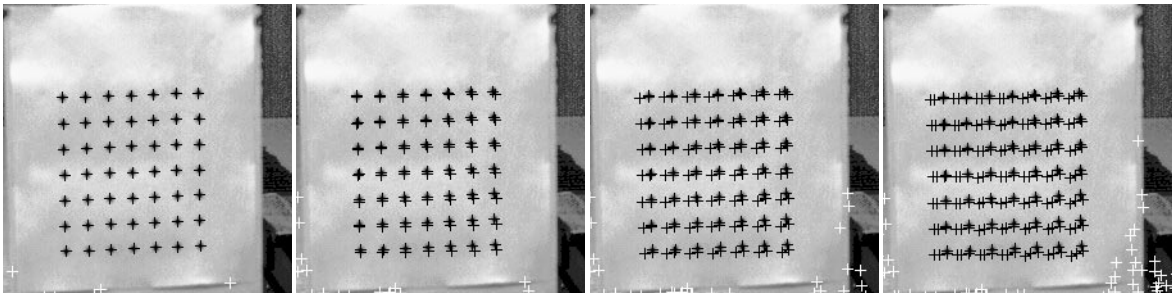


(a) (b) (c) (d)

Fig. 5 **Calibration images (equalized) taken from the convergent camera configuration ((a)-(d))**

The dots of the calibration pattern are detected using a star-shaped template with the weight distribution decreasing towards the center. The entire pattern is extracted and tracked from one camera to the next by imposing structural constraints of each dot relative to its neighbors, namely by determining the nearest and second nearest distances to another dot. This filters out wrong dot candidates, as shown in Fig. 6.

The simultaneous recovery of the camera parameters of all four cameras can be done using the non-linear least-squares technique described by Szeliski and Kang [16]. The inputs and outputs to this module are shown in the simplified diagram in Fig. 7. An alternative would be to use the pairwise-stereo calibration approach proposed by Faugeras and Toscani [7].



(a) (b) (c) (d)

Fig. 6 **Detecting and tracking the calibration points (only part of the image associated with Camera 1 is shown). The black +'s are the detected points while the white +'s are the spurious and rejected points: (a) Points detected in image of Camera 1; (b) Points detected in images of Cameras 1 and 2; (c) Points detected in images of Cameras 1, 2, and 3; (c) Points detected in all images.**

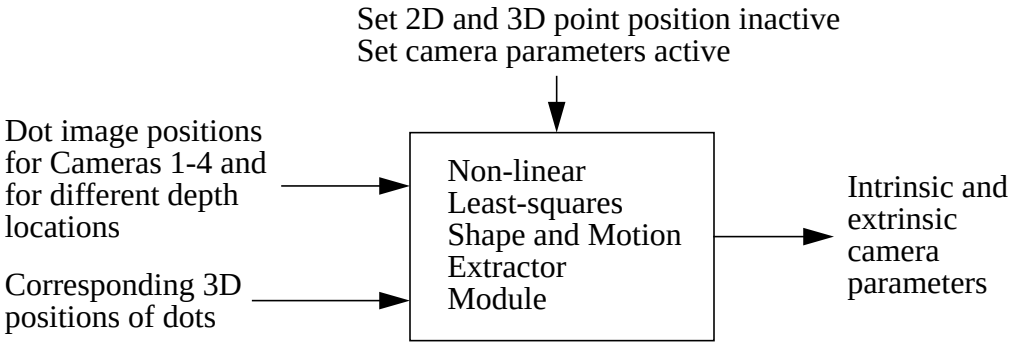


Fig. 7 **Non-linear least-squares approach to extraction of camera parameters**

4 Image rectification and depth recovery

If two camera axes are not parallel, their associated epipolar lines are not parallel to the scan lines. This introduces extra computation to extract depth from stereo. To simplify and reduce the amount of computation, *rectification* can be carried out first. The process of rectification for a pair of images (given the camera parameters, either through direct or weak calibration) transforms the original pair of image planes to another pair such that the resulting epipolar lines are parallel and equal along the new scan lines. Rectification is depicted in Fig. 8. Here $\mathbf{c}_1$ and $\mathbf{c}_2$ are the camera optical centers, Π_1 and Π_2 the original image planes, and Ω_1 and Ω_2 the rectified image planes. The condition of parallel and equal epipolar lines necessitates planes Ω_1 and Ω_2 to lie in the same plane, indicated as Ω_{12}. A point $\mathbf{q}$ is projected to image points $\mathbf{v}_1$ and $\mathbf{v}_2$ on the same scan line in the rectified planes.

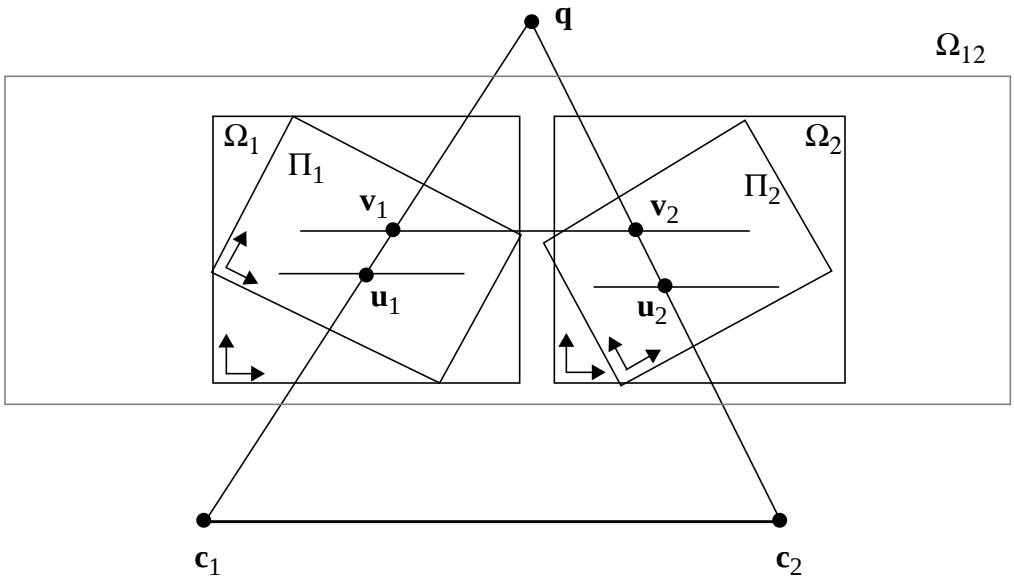


Fig. 8 **Image rectification**

A simple rectification method is described in [1]. However, the rectification process described there is a direct function of the locations of the camera optical centers. It is not apparent how the desirable properties of minimal distortion and maximal inclusion can be achieved with their formalism. We have modified their formalism to simplify the rectification mapping and adapt it to our situation.

Let the original 3×4 perspective transforms of two cameras be P_1 and P_2, where

$$P_j = \begin{bmatrix} \mathbf{p}_{j1}^T & p_{j14} \\ \mathbf{p}_{j2}^T & p_{j24} \\ \mathbf{p}_{j3}^T & p_{j34} \end{bmatrix}$$

The original perspective transform P_j is constructed from known camera parameters of the form

$$\tilde{\mathbf{u}}_j = \begin{bmatrix} f_j & 0 & 0 \\ 0 & a_j f_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_j & \mathbf{t}_j \end{bmatrix} \tilde{\mathbf{q}} = \begin{bmatrix} f_j & 0 & 0 \\ 0 & a_j f_j & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}_{j1}^T & t_{jx} \\ \mathbf{r}_{j2}^T & t_{jy} \\ \mathbf{r}_{j3}^T & t_{jz} \end{bmatrix} \tilde{\mathbf{q}} = \begin{bmatrix} f_j \mathbf{r}_{j1}^T & f_j t_{jx} \\ a_j f_j \mathbf{r}_{j2}^T & a_j f_j t_{jy} \\ \mathbf{r}_{j3}^T & t_{jz} \end{bmatrix} \tilde{\mathbf{q}} = P_j \tilde{\mathbf{q}}$$

where the tilde (~) above the vector indicates its homogeneous representation. $\mathbf{q}$ is the 3D point, $\mathbf{u}_j$ the image coordinate vector, f_j the focal length, a_j the aspect ratio, and R_j and $\mathbf{t}_j$ the extrinsic camera parameters. It is easy to see that the camera axis vector is $\mathbf{r}_{j3}$, and in the camera image coordinate system, the x- and y-directions are along $\mathbf{r}_{j1}$ and $\mathbf{r}_{j2}$, respectively.

Also, let M and N be the rectified perspective transforms, respectively, where

$$M = \begin{bmatrix} \mathbf{m}_1^T & m_{14} \\ \mathbf{m}_2^T & m_{24} \\ \mathbf{m}_3^T & m_{34} \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} \mathbf{n}_1^T & n_{14} \\ \mathbf{n}_2^T & n_{24} \\ \mathbf{n}_3^T & n_{34} \end{bmatrix}$$

Since perspective matrices are defined up to a scale factor, we can set both m_{34} and n_{34} to be unity. Accordingly, based on the analysis in [1], $\mathbf{m}_3 = \mathbf{n}_3$, $\mathbf{m}_2 = \mathbf{n}_2$, $m_{24} = n_{24}$, and from the constraint that $\mathbf{c}_1$ and $\mathbf{c}_2$ remain the optical centers,

$$\begin{aligned} \mathbf{m}_1^T\mathbf{c}_1 + m_{14} &= 0 \\ \mathbf{m}_2^T\mathbf{c}_1 + m_{24} &= 0 \\ \mathbf{m}_2^T\mathbf{c}_2 + m_{24} &= 0 \\ \mathbf{n}_1^T\mathbf{c}_2 + n_{14} &= 0 \\ \mathbf{m}_3^T\mathbf{c}_1 + 1 &= 0 \\ \mathbf{m}_3^T\mathbf{c}_2 + 1 &= 0 \end{aligned}$$

Let $\mathbf{d}_{12} = \mathbf{c}_1 - \mathbf{c}_2$. In a departure from [1], we choose the common rectified camera axis direction not only to be perpendicular to $\mathbf{d}_{12}$, but also to point in the direction between those of the unrectified camera axes (i.e., $\mathbf{r}_{13}$ and $\mathbf{r}_{23}$). This is done by first calculating

$$\mathbf{g} = \mathbf{r}_{13} + \mathbf{r}_{23}$$

We then find the nearest vector perpendicular to $\mathbf{d}_{12}$:

$$\mathbf{g}' = \mathbf{g} - \frac{\mathbf{g}^T\mathbf{d}_{12}}{\left\|\mathbf{d}_{12}\right\|^2}\mathbf{d}_{12}$$

Thus,

$$\mathbf{m}_3 = \mathbf{n}_3 = -\frac{\mathbf{g}'}{\mathbf{g}'^T\mathbf{c}_1} = -\frac{\mathbf{g}'}{\mathbf{g}'^T\mathbf{c}_2}$$

Determining $\mathbf{m}_2$ (and hence m_{24}) is similar, with the additional constraint that

$$\left\|\mathbf{m}_2\right\| = \left|\frac{a_1 f_1}{t_{1z}}\right|$$

Finally, $\mathbf{m}_1$ is determined from the relation

$$\mathbf{m}_1 = \tau(\mathbf{m}_2 \times \mathbf{m}_3)$$

τ (and hence $\mathbf{m}_1$ and m_{14}) is calculated based on the constraint

$$\left\|\mathbf{m}_1\right\| = \left|\frac{f_1}{t_{1z}}\right|$$

$\mathbf{n}_1$ and n_{14} are calculated in the same way, using the counterpart values of P_2.

As in [1], the homographies (or linear projective correspondences) that map the unrectified image coordinates to the rectified image coordinates are

$$H_1 = \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \begin{bmatrix} (\mathbf{p}_{12} \times \mathbf{p}_{13}) & (\mathbf{p}_{13} \times \mathbf{p}_{11}) & (\mathbf{p}_{11} \times \mathbf{p}_{12}) \end{bmatrix}$$

where

$$\tilde{\mathbf{v}}_1 = H_1 \tilde{\mathbf{u}}_1$$

$\tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{v}}_1$ are the homogeneous unrectified and rectified image coordinates, respectively, and

$$H_2 = \begin{bmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \\ \mathbf{n}_3^T \end{bmatrix} \begin{bmatrix} (\mathbf{p}_{22} \times \mathbf{p}_{23}) & (\mathbf{p}_{23} \times \mathbf{p}_{21}) & (\mathbf{p}_{21} \times \mathbf{p}_{22}) \end{bmatrix}$$

with

$$\tilde{\mathbf{v}}_2 = H_2 \tilde{\mathbf{u}}_2$$

$\tilde{\mathbf{u}}_2$ and $\tilde{\mathbf{v}}_2$ are similarly defined.

To recover depth from multibaseline stereo (specifically a 4-camera system) in a convergent configuration, we first rectify pairs of images as shown in Fig. 9.

There are two schemes which allows us to recover depth. The first uses all the homographies between the unrectified images and rectified images (namely H_{11}, H_{12}, H_{13}, H_{21}, H_{32}, and H_{43} in Fig. 10).

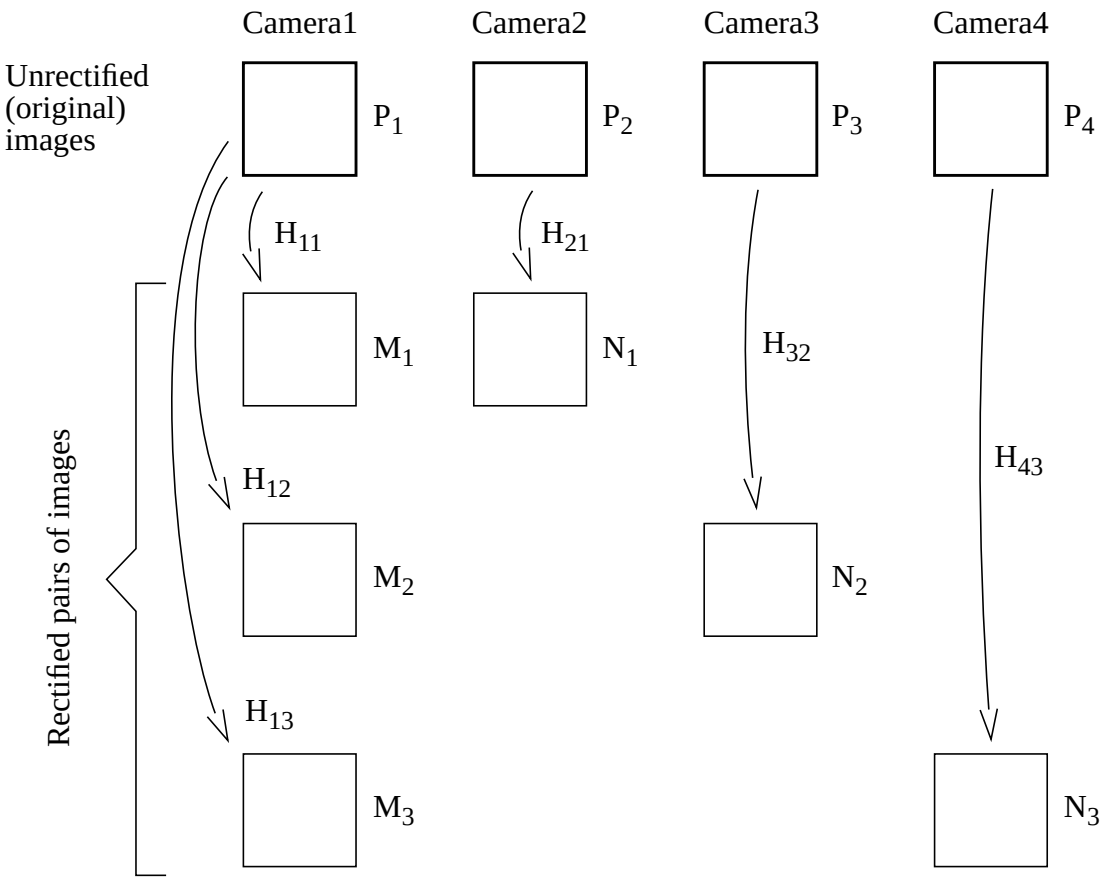


Fig. 9 **Image rectification scheme**

4.1 Direct approach for depth recovery

Subsequent to rectification, to recover depth, we first determine the corresponding location in the rectified image plane for the three pairs of cameras (Fig. 10). We wish to recover the 3D location **q** of the point corresponding to $\mathbf{u}_0$. **q** can be specified in the following form:

$$\mathbf{q} = \mathbf{c}_1 + \lambda \hat{\mathbf{d}}$$

where $\mathbf{c}_1$ is the optical center of the first ("reference") camera, $\hat{\mathbf{d}}$ is the unit vector in the direction from $\mathbf{c}_1$ to **q**, and λ is the depth of **q** from the reference camera optical center. If

$$\tilde{\mathbf{u}}_1 = \begin{bmatrix} \alpha_1 \\ \beta_1 \\ 1 \end{bmatrix}, \quad \tilde{\mathbf{v}}_j = \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{w}}_j = \begin{bmatrix} x'_j \\ y'_j \\ 1 \end{bmatrix} \quad \text{with} \quad y_j = y'_j$$

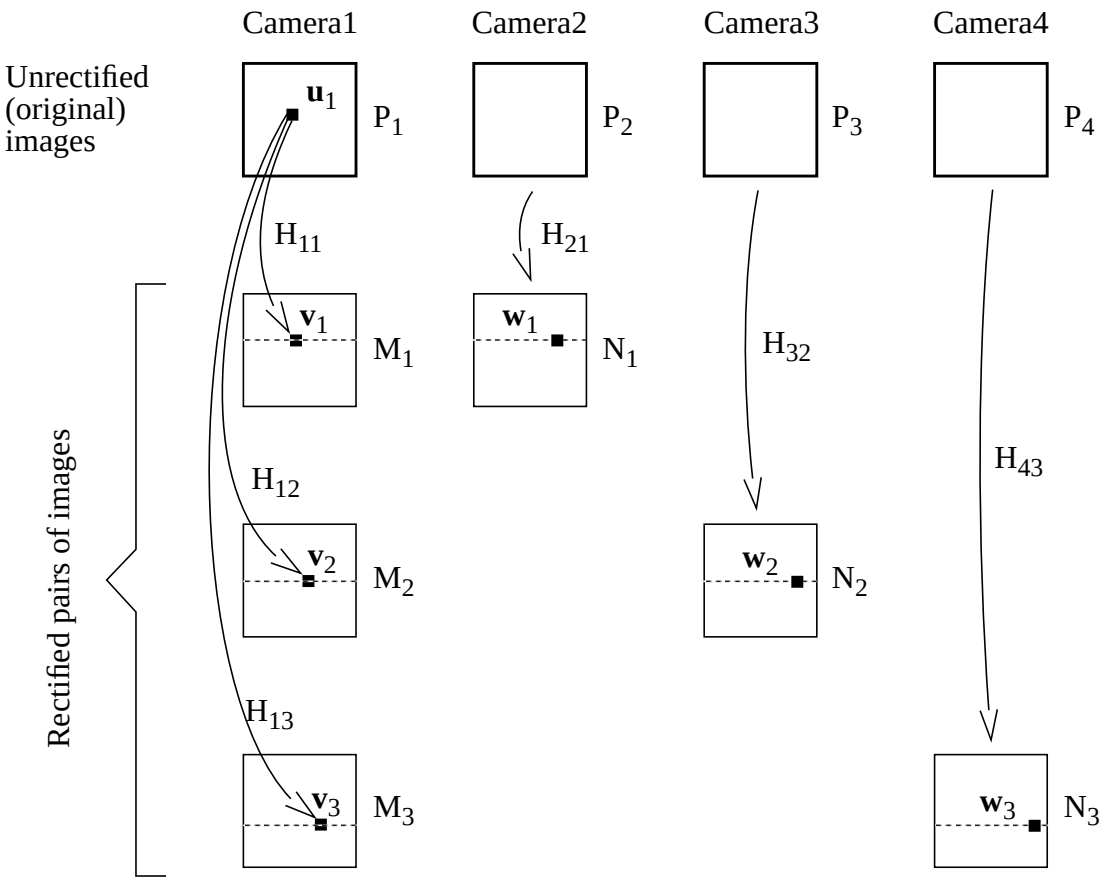


Fig. 10 **Recovering depth from multibaseline stereo after rectification**

then

$$\mathbf{u}_1 = \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{p}_{11}^T\left(\mathbf{c}_1 + \lambda\hat{\mathbf{d}}\right) + p_{114}}{\mathbf{p}_{13}^T\left(\mathbf{c}_1 + \lambda\hat{\mathbf{d}}\right) + p_{134}} \\ \dfrac{\mathbf{p}_{12}^T\left(\mathbf{c}_1 + \lambda\hat{\mathbf{d}}\right) + p_{114}}{\mathbf{p}_{13}^T\left(\mathbf{c}_1 + \lambda\hat{\mathbf{d}}\right) + p_{134}} \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{p}_{11}^T\hat{\mathbf{d}}}{\mathbf{p}_{13}^T\hat{\mathbf{d}}} \\ \dfrac{\mathbf{p}_{12}^T\hat{\mathbf{d}}}{\mathbf{p}_{13}^T\hat{\mathbf{d}}} \end{bmatrix}$$

since $P_1\mathbf{c}_1 = [0\ 0\ 0]^T$. So

$$(\alpha_1\mathbf{p}_{13} - \mathbf{p}_{11})^T\hat{\mathbf{d}} = 0$$

$$(\beta_1\mathbf{p}_{13} - \mathbf{p}_{12})^T\hat{\mathbf{d}} = 0$$

i.e.,

$$\hat{\mathbf{d}} \parallel (\alpha_1\mathbf{p}_{13} - \mathbf{p}_{11}) \times (\beta_1\mathbf{p}_{13} - \mathbf{p}_{12})$$

from which we get

$$\hat{\mathbf{d}} = \frac{\alpha_1(\mathbf{p}_{12}\times\mathbf{p}_{13}) + \beta_1(\mathbf{p}_{13}\times\mathbf{p}_{11}) + (\mathbf{p}_{11}\times\mathbf{p}_{12})}{\left\|\alpha_1(\mathbf{p}_{12}\times\mathbf{p}_{13}) + \beta_1(\mathbf{p}_{13}\times\mathbf{p}_{11}) + (\mathbf{p}_{11}\times\mathbf{p}_{12})\right\|}$$

To find the disparity, $\Delta_j = x'_j - x_j$, as a function of the projection transform elements, we first find the expressions for the rectified image coordinates (noting that $y_j = y'_j$):

$$\mathbf{v}_j = \begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{m}_{j1}^T\hat{\mathbf{d}}}{\mathbf{m}_{j3}^T\hat{\mathbf{d}}} \\ \dfrac{\mathbf{m}_{j2}^T\hat{\mathbf{d}}}{\mathbf{m}_{j3}^T\hat{\mathbf{d}}} \end{bmatrix} \quad \text{and} \quad \mathbf{w}_j = \begin{bmatrix} x'_j \\ y_j \end{bmatrix} = \begin{bmatrix} \dfrac{\mathbf{n}_{j1}^T\mathbf{c}_1 + n_{j14} + \lambda\mathbf{n}_{j1}^T\hat{\mathbf{d}}}{\lambda\mathbf{m}_{j3}^T\hat{\mathbf{d}}} \\ \dfrac{\mathbf{m}_{j2}^T\hat{\mathbf{d}}}{\mathbf{m}_{j3}^T\hat{\mathbf{d}}} \end{bmatrix}$$

Hence

$$\Delta_j = x'_j - x_j = \frac{\mathbf{n}_{j1}^T\mathbf{c}_j + n_{j14} + \lambda(\mathbf{n}_{j1} - \mathbf{m}_{j1})^T\hat{\mathbf{d}}}{\lambda\mathbf{m}_{j3}^T\hat{\mathbf{d}}}$$

By varying λ within a specified interval and resolution, we can calculate Δ_j's for the pairs of rectified images, and hence calculate the sum of matching errors (as in [13] with multiple parallel cameras). The depth is recovered by picking the value of λ associated with the least matching error.

4.2 A computationally more efficient approach for depth recovery

The method described above implies that we must calculate, at each point and for each depth, the corresponding points in all images. This requires projective transformations of all images to be performed for each depth value. There is a more computationally efficient way to recover depth. This stems from the following properties:

1. The two rectified planes fall on the same plane.
2. The line joining the two projection centers is parallel to this common plane.

Properties 1 & 2 (which are the necessary conditions for rectification) give rise to

3. The homography between the two rectified planes cannot be projective (since the scan lines on the rectified images are parallel, i.e., the corresponding rows at both rectified images are equal). This is true since the "projection" lines (the corresponding scan lines) meet at infinity.

From 3, the homography between rectified planes must then be at most a 2D affine transform, i.e., the last row of the homography matrix must be (0 0 1). This dispenses with the additional division by the z-component in calculating the corresponding matched point for a particular depth.

The scheme now follows that in Fig. 11. The matching is done using the homographies between *rectified* images K_1, K_2 and K_3 (which we term as *rectified homographies*). The rectified homographies can be readily determined as follows:

For a known depth plane ($z = d$), we can "contract" the 3×4 perspective matrix M (to the rectified plane) to a 3×3 homography G. For camera l, we have

$$M_l \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{l1} & \mathbf{p}_{l2} & \mathbf{p}_{l3} & \mathbf{p}_{l4} \end{bmatrix} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{l1} & \mathbf{p}_{l2} & d\mathbf{p}_{l3} + \mathbf{p}_{l4} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = G_l \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s_l \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix}$$

where $\mathbf{p}_{lj}$ is the jth column of M_l and $(u_l, v_l)^T$ is the projected image point in camera l. Similarly, for camera m,

$$M_m \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = G_m \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s_m \begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix}$$

Since the rectified planes are coplanar, $s_l = s_m$; hence

$$\begin{bmatrix} u_m \\ v_m \\ 1 \end{bmatrix} = \frac{1}{s_m} G_m \left(s_l G_l^{-1} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} \right) = G_m G_l^{-1} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix} = K_{lm} \begin{bmatrix} u_l \\ v_l \\ 1 \end{bmatrix}$$

Note that, due to rectification, $v_m = v_l$, and as explained earlier in this subsection, the bottom row of K_{lm} is (0 0 1). In other words, the projective transformations are reduced to affine transformations, reducing the amount of computation.

Depth recovery then proceeds in a similar manner as the direct approach described in the previous subsection.

4.3 An approximate depth recovery approach

In both approaches described earlier, for each depth, each pixel in the unrectified reference image has to be mapped $N_{cameras} - 1$ times to the respective rectified images (corresponding to the homographies H_{11}, H_{12}, and H_{13} in Fig. 11). We can work in the rectified image

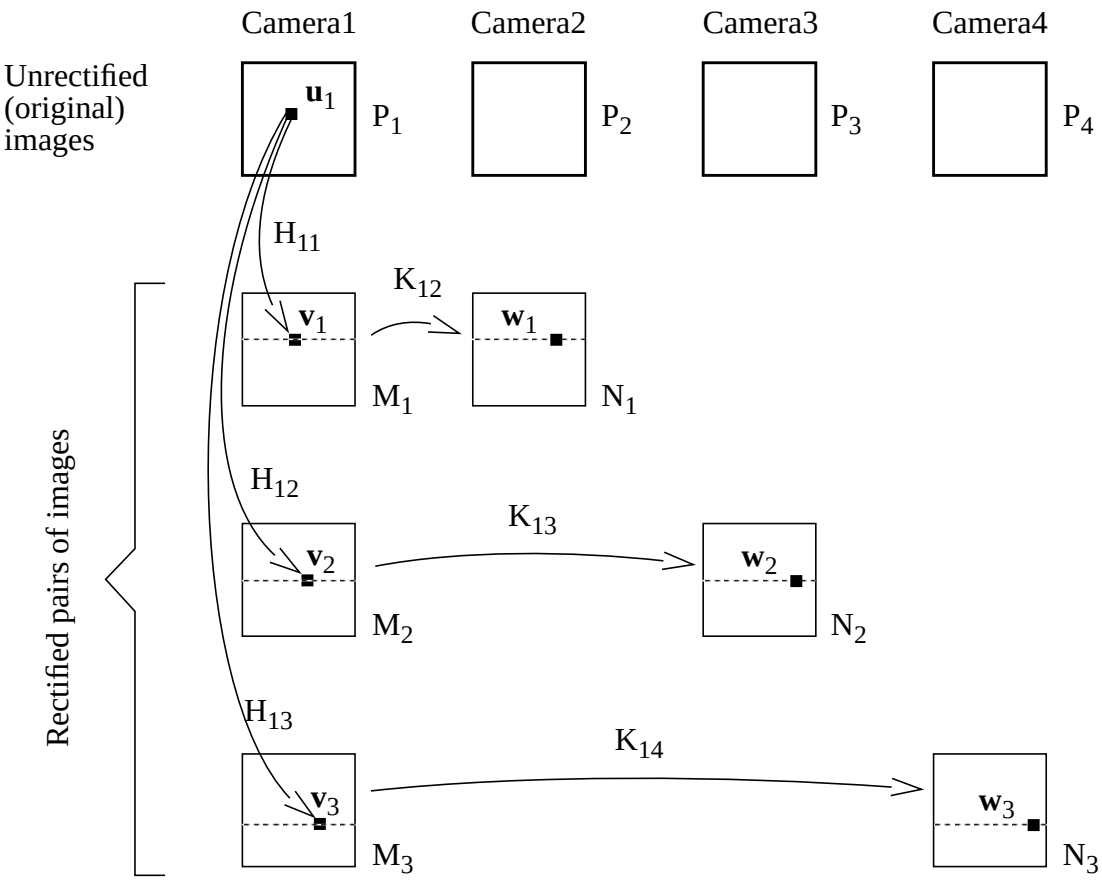


Fig. 11 **A computationally more efficient depth recovery scheme**

coordinates (say M_1), but this still requires mapping from M_2 to M_1 and M_3 to M_1 in the collection of match errors for each depth value. This means that we need to perform $(N_{\text{cameras}} - 2)\, N_{depth}$ sets of bilinear interpolations associated with image warping (where N_{depth} is the number of depth values and $N_{cameras}$ is the number of cameras).

In order to avoid the warping operations, we use an approximate depth recovery method. The matching is done with respect to the rectified image of the first pair. However, the rectified images N_2 and N_3 will not be row preserved relative to M_1 (Fig. 12). We warp rectified images N_2 and N_3 so as to preserve the rows as much as possible, resulting in N'_2 and N'_3 (Fig. 12). The errors should be tolerably small as long as the vergence angles are small. In addition, this effect should not pose a significant problem as we are using a local windowing technique in calculating the match error.

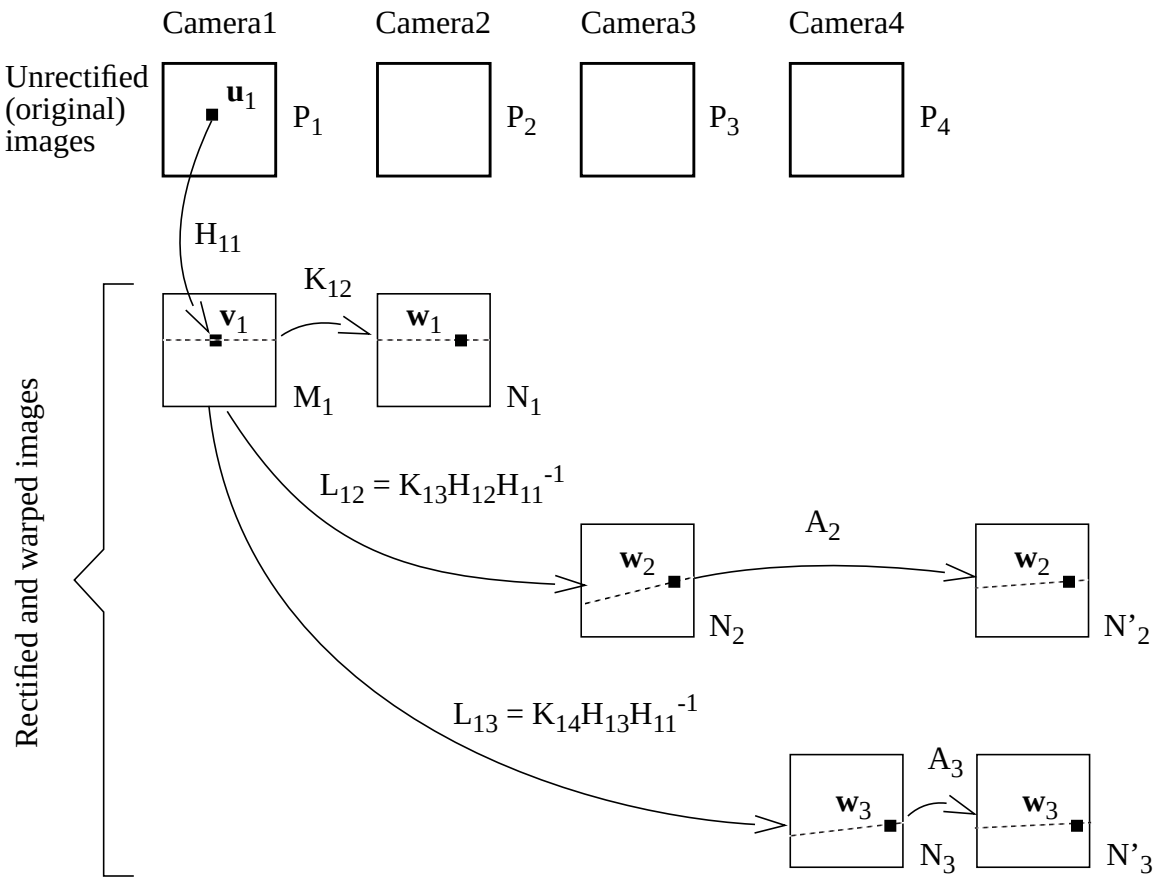


Fig. 12 **The approximate depth recovery scheme (compare this with Fig. 11)**

By comparing Fig. 12 with Fig. 11, we can see that the mapping from M_1 to N_2 is given by the homography $L_{12} = K_{13}H_{12}H_{11}^{-1}$. Similarly, the mapping from M_1 to N_3 is given by $L_{13} = K_{14}H_{13}H_{11}^{-1}$. The matrices A_2 and A_3 are constructed such that

$$\begin{bmatrix} c' \\ r \\ 1 \end{bmatrix} = A_2 L_{12} \begin{bmatrix} c \\ r \\ 1 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} c'' \\ r \\ 1 \end{bmatrix} = A_3 L_{13} \begin{bmatrix} c \\ r \\ 1 \end{bmatrix}$$

i.e., the resulting overall mapping is row preserving (*r* and *c* are the row and column respectively). In general, this would not be possible, unless all the camera centers are colinear; however, this is a good approximation for small vergence angles and approximately aligned

cameras. A_2 and A_3 are calculated from the following overconstrained relation using the pseudoinverse calculation:

$$A_j L_{1j}^{d_{min}} \begin{bmatrix} c_{min} & c_{min} & c_{max} & c_{max} \\ r_{min} & r_{max} & r_{min} & r_{max} \\ 1 & 1 & 1 & 1 \end{bmatrix} \Bigg| L_{1j}^{d_{max}} \begin{bmatrix} c_{min} & c_{min} & c_{max} & c_{max} \\ r_{min} & r_{max} & r_{min} & r_{max} \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 \\ r_{min} & r_{max} & r_{min} & r_{max} & r_{min} & r_{max} & r_{min} & r_{max} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{for } j = 1, 2.$$

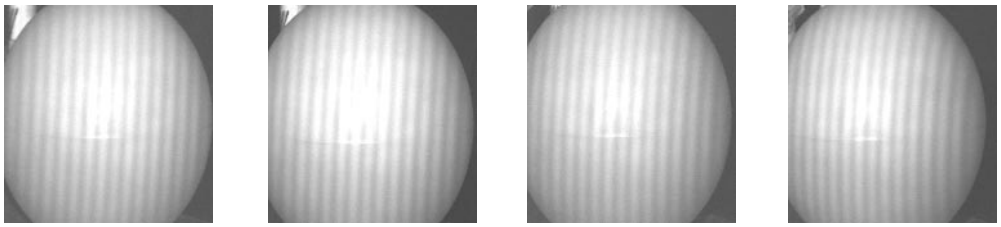
where $L_{1j}^{d_{min}}$ is associated with the minimum depth and $L_{1j}^{d_{max}}$ with the maximum depth, c_{min} and c_{max} are the minimum and maximum values of the image column, and r_{min} and r_{max} are the minimum and maximum values of the image row, respectively. X_i (i=1,...,8) are don't-care values. The symbol | is used to represent matrix augmentation.

This algorithm has been implemented in parallel using the Fx (parallel Fortran) language developed at Carnegie Mellon [15]. Fx, a variant of High Performance Fortran with optimizations for high-communication applications like signal and image processing, runs on the Carnegie Mellon-Intel Corporation iWarp, the Paragon/XPS, the Cray T3D, and the IBM SP2. The experiments reported in this paper were done on the iWarp.

5 Experimental results

In this section, we present results of our active multibaseline stereo system. As mentioned before, a pattern of sinusoidally varying intensity are projected onto the scenes to facilitate image point correspondence.

An example of a set of images (Scene 1) and the extracted depth image is shown in Fig. 13 and Fig. 14 respectively. The large peaks at the borders of the depth map are outliers due to mismatches in the background outside the depth range of interest.



(a) (b) (c) (d)

Fig. 13 **Views of the globe (Scene1) from the four cameras ((a)-(d))**

Another example (Scene 2) is shown in Fig. 15 with the recovered elevation map in Fig. 16. As can be seen from the elevation map, except at the edges of the objects on the scene, the data looks very reasonable.

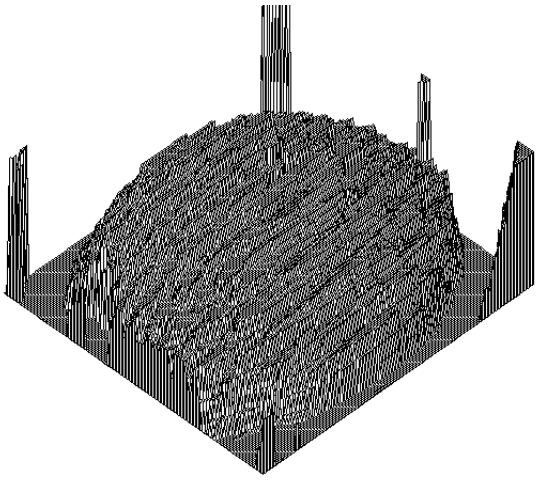
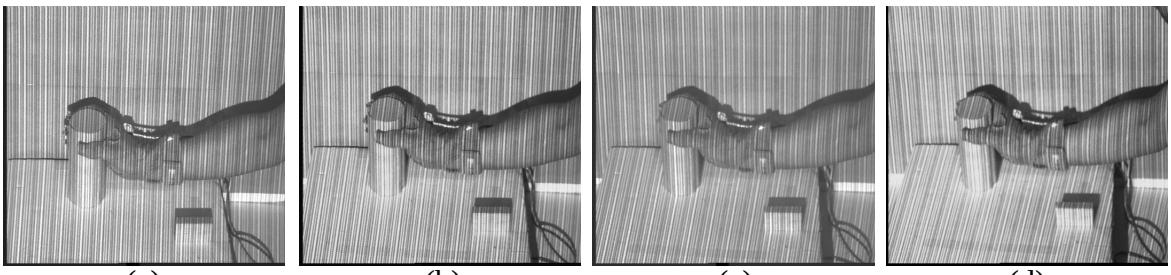


Fig. 14 **Recovered elevation map of Scene1**



(a) (b) (c) (d)

Fig. 15 **Views of Scene 2**

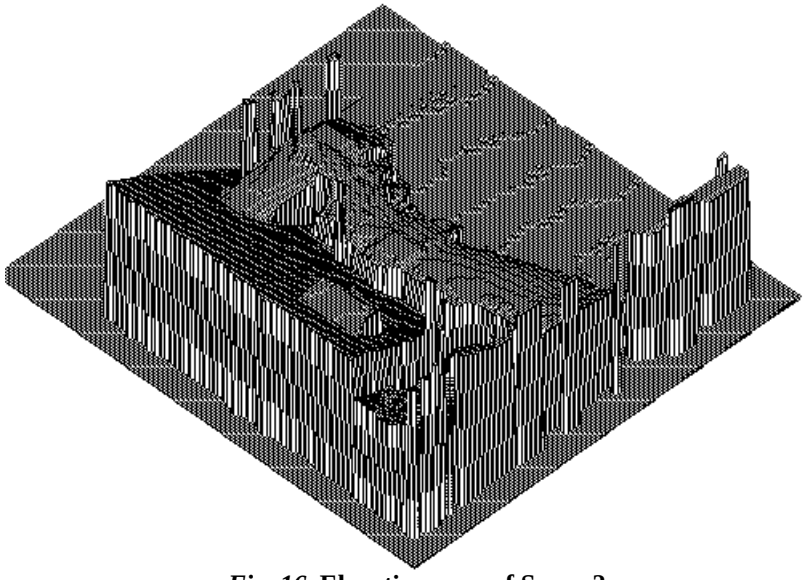
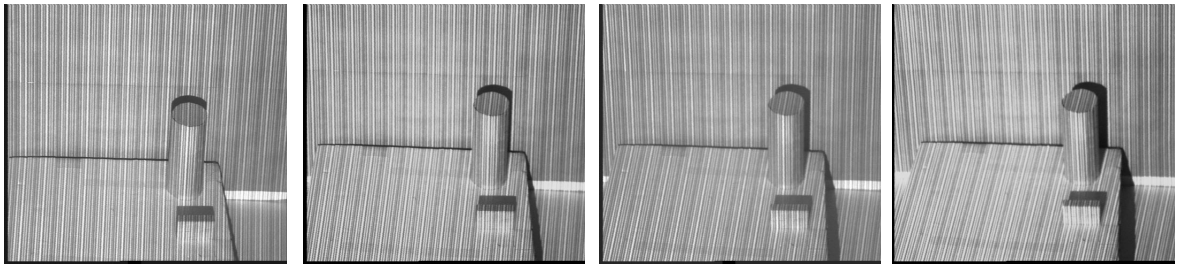


Fig. 16 **Elevation map of Scene 2**

For Scene 3 (Fig. 17), subsequent to depth recovery (Fig. 18), we fit the known models onto the range data using Wheeler and Ikeuchi's 3D template matching algorithm [18] to yield results seen in Fig. 19. Again the data looks very reasonable.



(a) (b) (c) (d)

Fig. 17 **The four camera views of Scene 3**

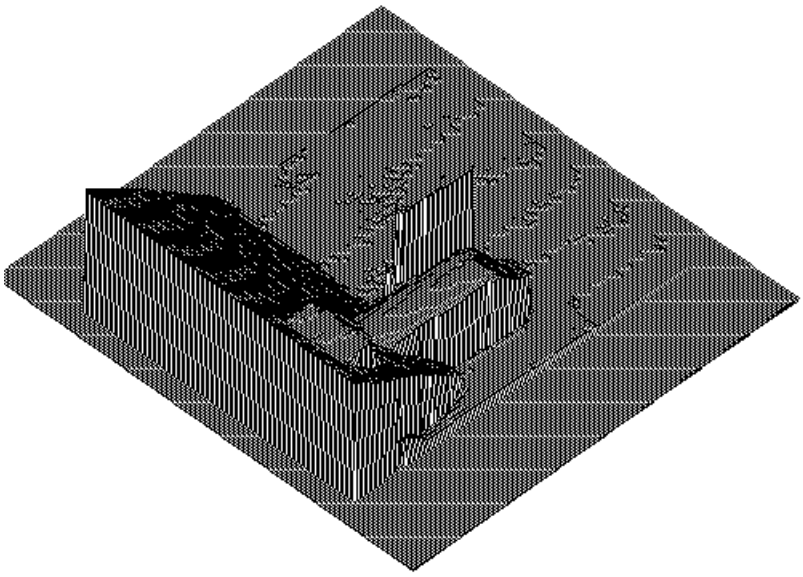


Fig. 18 **Extracted elevation map of Scene 3**

We have also performed some error analysis on some of the range data that were extracted from Scene 2. Fig. 20 show the areas for planar fit; Table 1 shows the numerical results of the planar fit. As can be seen, the average planar fit error is smaller than 1 mm (the furthest planar patch is about 1.7m away from the camera system). Fig. 21 depicts the error distribution of the resulting planar fit across the image (only on pixels on planar surfaces in the scene). The darker pixels are associated with lower absolute error in planar fitting.

We have also obtained stereo range data of a cylinder of known cross-sectional radius and calculated the fit error. In both scenes (with different camera settings), the cylinder is placed about 3.3 m away from the camera system.

As can be seen from Table 2, the mean absolute error of fit is less than 1 mm.

6 Observations on accuracy

We have exceeded one millimeter accuracy. Here we informally characterize the remaining sources of error in our system.

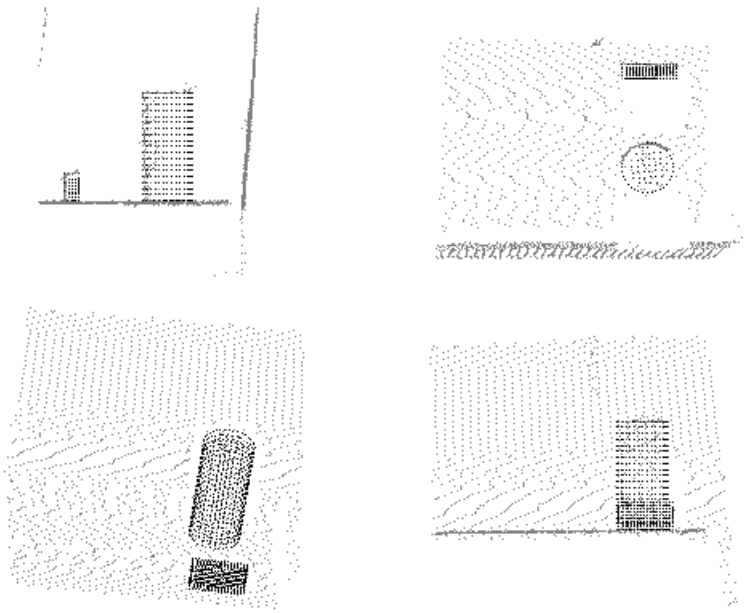


Fig. 19 **Recovered 3D points of Scene3 with fitted cylinder and box models (shown at four different viewpoints)**

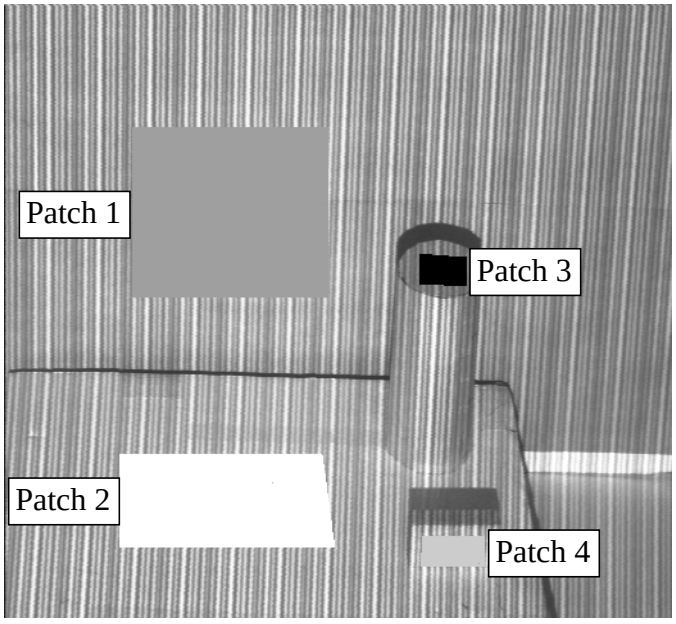


Fig. 20 **Sampled areas for planar fit.**

There are a number of sources of error in our system and in stereo generally:

1. The use of an active multibaseline approach reduces the chance of false matches, but they can still occur.
2. The fundamental assumptions of stereo are that the texture being viewed is unique over the search window, and that the surface is visible to and lies at the same angle to all camera optical axes. The former assumption is addressed by the active component of our sys-

Table 1 Results of fitting planes to selected patches in Scene2.

Patch #	Patch size (pixels)	plane equation: $p \cdot \hat{\mathbf{n}} = d$		Avg. |error| (μm)	Max. |error| (mm)	Std. dev. (μm)
		$\hat{\mathbf{n}}$	d (mm)			
1	20925	(0.012, 0.075, 0.997)	1746.8	550	2.24	400
2	12405	(0.009, 0.999, -0.003)	1119.6	420	1.91	310
3	993	(0.026, 0.999, 0.0240)	1023.8	520	2.97	420
4	1340	(-0.025, 0.019, 0.999)	1449.5	370	1.75	320

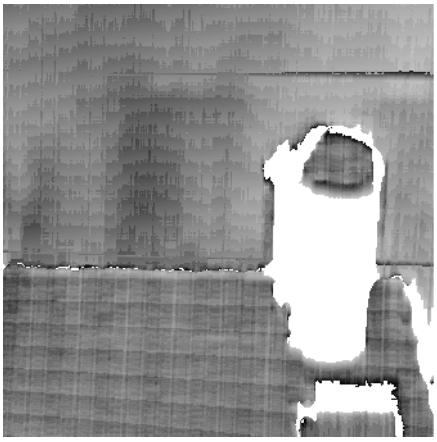
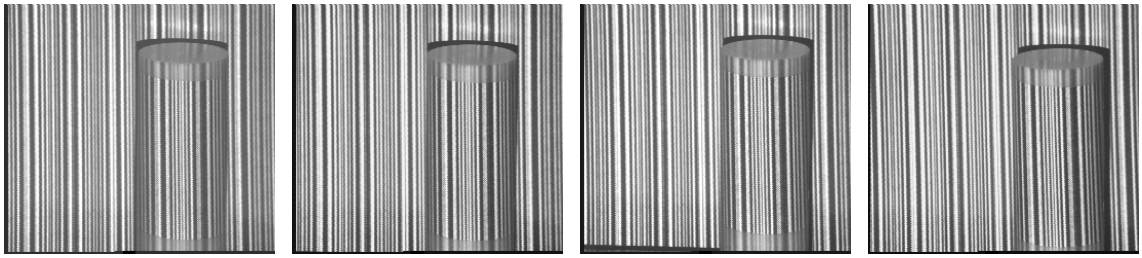


Fig. 21 **Plane fit error distribution for Scene2 (enhanced, planar surfaces only)**



(a) (b) (c) (d)

Fig. 22 **Four camera views of the first cylinder scene**

Table 2 Results of fitting cylinders

Cylinder scene #	Patch size (pixels)	Average |error| (μm)	Maximum |error| (mm)	Standard deviation (μm)
1	25200	640	4.35	540
2	35150	640	3.17	500

tem, but the latter is not and cannot be, except by placing the cameras as close together as possible (which reduces accuracy). The failure of this assumption is particularly evident at the boundaries of objects, where it is the cause of significant error.

3. Errors are possible during calibration, since the position of our calibration plate is adjusted by hand (limiting its accuracy in positioning to about 1 mm), and the dot pattern positions are not always found precisely.
4. We use a pinhole camera model, which will result in errors near the edge of the image, particularly with short focal lengths.
5. We make the approximation discussed in Section 4.3, which will result in errors when the camera optical centers are not colinear.

Of these, only the first seems to be a cause of significant error (the second also causes large error, but we deliberately omit it from our error analysis since it is fundamental to stereo). All of the large errors (more than 1 mm) are observed to be in regions where the projected pattern does not provide sufficient texture for a correct match.

We have attempted to reduce these errors by analysis and experimentation. Analysis shows that a frequency-modulated sine wave pattern, as used there, is a good choice since it does not require large dynamic range (our iWarp video interface has manually adjustable gain and offset controls, leading us to limit the dynamic range to avoid clipping). Also, a *randomly* frequency-modulated sine wave gives the best possible result, since the same pattern occurs twice in the search area with vanishingly small probability, theoretically eliminating the possibility of false matches. Experiments with randomly modulated patterns have shown that

- The lowest frequency of the sine wave (as seen in the image) must be higher than the width of the correlation match window.
- The highest frequency usable is constrained by the resolution of the camera and the focus control of the projector. Using a higher frequency than the maximum results in a gray blur and many false matches.

The trade-off between these two constraints involves optimizing the projector placement and focus, the camera resolution, the number of cameras, and the camera dynamic range.

In addition, many of the problems of false matches occur where the limited dynamic range of our video interface plays a role, particularly with dark surfaces or sufaces which lie at an oblique angle to the projector (so that no pattern appears in the image), or surfaces with specularities (so that clipping overwhelms the pattern). In these cases, we believe careful adjustment of the projector, including use of multiple projectors (since there is no particular constraint between the projector and camera in active stereo, this is easy to do), can serve to reduce these effects. The use of multiple patterns, either time-sequenced (taking advantage of our system’s ability to capture images at high speed) or color-sequenced (using color cameras) is also promising.

7 Summary

We have briefly described a 4-camera system that is capable of video rate image acquisition. It uses a software distribution approach which takes advantage of iWarp's systolic design. The four cameras are used in a converging configuration for more effective use of the camera view spaces. In addition, to recover dense stereo range data from each set of images, we project a sinusoidally varying pattern onto the scene to enhance local intensity discriminability. This results in the notion of *active* multibaseline stereo system.

We have also described in detail our implementation of the depth recovery algorithm which involves the preprocessing stage of image rectification. Our approximate depth recovery implementation was designed for reduced computation.

The results that we have obtained from this system indicated that the mean errors (discounting object border areas) are less than a millimeter at distances varying from 1.5 m to 3.5 m from the camera system. The performance of the system is thus comparable to a good structured light system, while allowing data to be captured at full video rate.

Active multibaseline stereo appears to be a promising addition to structured light imaging systems. It allows images to be captured at high speed and still have high spatial resolution. It allows great freedom in the relationship between the camera, the surface, and the light source, making it possible to manipulate these so as to get high accuracy in a wide variety of circumstances.

Acknowledgments

Many thanks to Bill Ross for his helpful information on setting up a multibaseline camera system. Tom Warfel built the real-time video interface board to the iWarp that made this project possible. We are also grateful to Luc Robert for interesting discussions on image rectification. Mark Wheeler helped in the analysis that led to the decision to verge the cameras.

References

[1] Ayache, N. and C. Hansen. *Rectification of images for binocular and trinocular stereovision.* in Proceedings of the 9th International Conference on Pattern Recognition. 1988. Rome, Italy.: p. 11-16.

[2] Barnard, S.T. and M.A. Fischler, *Computational stereo.* Computing Surveys, 1982. **14**(4): p. 554-572.

[3] Borkar, S., *et al. iWarp: An Integrated Solution to High-Speed Parallel Computing*. in Proceedings of Supercomputing '88. 1988. Orlando, Florida.: p. 330-339.

[4] Borkar, S., *et al. Supporting Systolic and Memory Communication in iWarp*. in Proceedings of the 17th International Symposium on Computer Architecture. 1990. Seattle, WA.: p. 70-81.

[5] Dhond, U.R. and J.K. Aggarwal, *Structure from stereo - A review.* IEEE Transactions on Systems, Man, and Cybernetics, 1989. **19**(6): p. 1489-1510.

[6] Faugeras, O.D., *Three-Dimensional Computer Vision: A Geometric Viewpoint*. 1993, MIT Press.

[7] Faugeras, O.D. and G. Toscani. *The calibration problem for stereo*. in Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. 1986: p. 15-20.

[8] Fua, P., *A parallel stereo algorithm that produces dense depth maps and preserves image features.* Machine Vision and Applications, 1993. **6**: p. 35-49.

[9] Ikeuchi, K. and T. Suehiro. *Towards an Assembly Plan from Observation.* in Proceedings of the IEEE International Conference on Robotics and Automation. 1992: p. 2171-2177.

[10] Kang, S.B. and K. Ikeuchi. *Determination of motion breakpoints in a task sequence from human hand motion.* in Proceedings of the IEEE International Conference on Robotics and Automation. 1994: p. 551-556.

[11] Matthies, L., *Stereo vision for planetary rovers: Stochasting modeling to near real-time implementation.* International Journal of Computer Vision, 1992. **8**(1): p. 71-91.

[12] Okutomi, M. and T. Kanade, *A Multiple-Baseline Stereo.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993. **15**(4): p. 353-63.

[13] Ross, B. *A practical stereo vision system.* in Proceedings of the IEEE International Conference on Computer Vision and Patterrn Recognition. 1993: p. 148-153

[14] Rygol, M., *et al.*, *A Parallel 3D Vision System,* in *Active Vision,* A. Blake and A. Yuille, Editors. 1992, MIT Press: Cambridge, MA. p. 239-261.

[15] Subhlok, J., *et al. Exploiting Task and Data Parallelism on a Multicomputer.* in Symposium on Principles and Practice of Parallel Programming. 1993: ACM SIGPLAN.

[16] Szeliski, R. and S.B. Kang, *Recovering 3D shape and motion from image streams using non-linear least squares.* Journal of Visual Communication and Image Representation, 1994. **5**(1): p. 10-28.

[17] Webb, J.A., T. Warfel, and S.B. Kang, *A Scalable Video Rate Camera Interface.* Tech. Rep. CMU-CS-94-166, Computer Science Department, Carnegie Mellon University, 1994.

[18] Wheeler, M.D. and K. Ikeuchi, *Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition.* in Proceedings of the 2nd CAD-Based Vision Workshop, 1994: p. 46-53.