OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

An Algorithm for Reconstructing Three Dimensional Images from Overlapping Two Dimensional Intensity Measurements with Relaxed Camera Positioning Requirements

Siranee Nuchitprasitchai[1], Michael C. Roggemann[2], Timothy C. Havens[3]

([1,2,3]Department of Electrical and Computer Engineering, Michigan Technological University, USA
snuchitp@mtu.edu, mroggema@mtu.edu, and thavens@mtu.edu)

ABSTRACT : This paper proposes and demonstrates an algorithm to generate three-dimensional (3-D) reconstructions using images from a stereo vision of two dimensional (2-D) surveillance camera without calibration. In the surveillance of public environment, the cameras are not set up for a binocular stereo system for a 3D reconstruction, but here they can be used when there is an overlapped scene. When the field of view of multiple cameras overlap the potential exists for computing the 3-D location of surfaces in the overlapping regions of the images. In this paper we apply the Scale Invariant Feature Transform (SIFT), the Random Sample Consensus (RANSAC), and the Sum of Absolute Differences (SAD) to reconstruct 3-D image from two overlapping images. The camera parameters and the geometry of the cameras are known; however, they do not correspond to conventional stereo image measurements. The process consists of two steps: image preparation and 3-D reconstruction. Image preparation involves rescaling, rectifying, and finding the corresponding points between the left and the right stereo images. The SIFT and the RANSAC algorithm are applied to find the difference of object size between the images and then to rescale and rectify the images. The corresponding points on the two images are found with a block matching method using the SAD technique. For 3-D reconstruction, a set of prototype geometric equations is introduced to calculate the 3-D locations (X, Y, Z) for each corresponding point. This algorithm for 3-D reconstruction was evaluated using different camera geometries, and using different objects. The results show that the target dimension estimated from the 3-D images has a small Root-Mean-Square-Error (RMSE) as compared to the actual dimension of the target.

Keywords: *3D Reconstruction, Scale Invariant Feature Transform, and Sum of Absolute Differences*

I. INTRODUCTION

Security has become more important in both private and public areas. Camera surveillance systems are widely used for security purposes [1-5]. In order to cover the area of interest, there are often multiple cameras present that have overlapping fields of view. These digital images of the same scene can be used to extract three-dimensional (3-D) information of the objects in the overlapping fields of view, such as the height of the person, the size of an object in that scene, or object distance [6-9]. 3-D image reconstruction from sets of two-dimensional (2-D) images using stereo vision has been an area of active research for many decades, and has been applied in many fields, such as medical imaging [10], robot navigation [11], image analysis [12], machine vision [13], and architecture [14]. In most cases, the geometry of the stereo cameras and the scene are carefully controlled to make the processing straightforward. In this geometry, the spatial scale of the two images is guaranteed to be the same, and the stereo reconstruction problem is straightforward [15]. In stereo vision, disparities between corresponding points in the two images can be found by using the following techniques: block-matching [16], gradient-based optimization [17], feature matching [18], dynamic programming [19], graph cuts [20], and belief propagation [21]. These techniques have been successfully demonstrated, and are used in commercially available products when the camera and the target geometries can be controlled.

In this paper, we propose a triangulation method based on the SIFT algorithm as a means of expanding the range of camera geometries from which 3-D information can be extracted. Our camera geometries are more flexible compared to standard stereo vision [22]. The optical axes of the cameras do not need to be parallel and

the cameras do not need to have the same distance from object. The 3-D reconstruction process consists of two steps: preparing the images and reconstructing the 3-D image. For the first step, the Scale Invariant Feature Transform (SIFT) [23] is applied to rescale and rectify the images. Some candidate matching points output by SIFT are incorrect, and including them in subsequent processing has negative effects on the 3-D image reconstruction. These incorrect matching points (outliers) are eliminated by using the RANdom SAmple Consensus (RANSAC) algorithm [24]. RANSAC is an iterative method to create a mathematical model fit to remove outlier data. In the next step, the Sum of Absolute Differences (SAD) block matching technique [25, 26] is used to find the corresponding points between the left and the right images. Triangulation-based geometric equations are used to calculate the 3-D location of each corresponding point. This 3-D data may be used to extract detailed shape information of objects in the scene. A comparison of the 3-D information with measurements of the target shows that the result is accurate to within small errors on the order of a few centimeters. The errors are evaluated more completely in the experimental section.

The remainder of the paper is organized as follows. Image preparation and triangulation-based geometric 3-D reconstruction regarding the proposed geometry, our approach for calculating the 3-D object point locations, is presented in Section 2. Experimental results showing 3-D image reconstructions and the errors between the actual size of the object and the measured size of the object are presented in Section 3. Conclusions are discussed in Section 5.

II. IMAGE PREPARATION AND TRIANGULATION-BASED GEOMETRIC 3-D RECONSTRUCTION

Our approach is to extracting 3-D information from 2-D overlapping images taken by two cameras that do not need to be on the same baseline, and do not need to be parallel like the standard stereo vision [22]. The cameras can also be rotated around the axes and have the different distance from the object. The different distances from the object results in that the camera positions may have z-axis displacement from each other; so, the cameras do not need to be on the epipolar line as in conventional stereo vision. For example, the left camera can be closer to the object than the right camera, or vice versa. The 3-D model is created by finding pixels in one 2-D image that can be identified as originating from the same point in another 2-D image. This is referred to as the correspondence problem [27] in stereo reconstruction. To solve the correspondence problem, 2-D images need to be prepared using SIFT, RANSAC, and SAD. This preparation is now explained.

A. Image rescaling and rectification

We employ a camera model based on the single thin lens camera. The single thin lens camera [22] describes the mathematical relationship between the 3-D object points and the image points. In the thin lens model, the rays of light emitted from a point travels along paths through the lens, converging at a point behind the lens. In geometric optics, a ray passing through the center of a thin lens is called the chief ray, which is not deflected by the lens. The image is inverted in the image plane. Fig. 1 shows a chief ray in the thin lens camera model, except that the image plane is moved to the front of the lens instead of behind it, and in this case the image is not inverted. The perspective model explains the projection of an object point at location P to the point P', where it is imaged as defined by a chief ray traced from P to P' through the center of the lens.

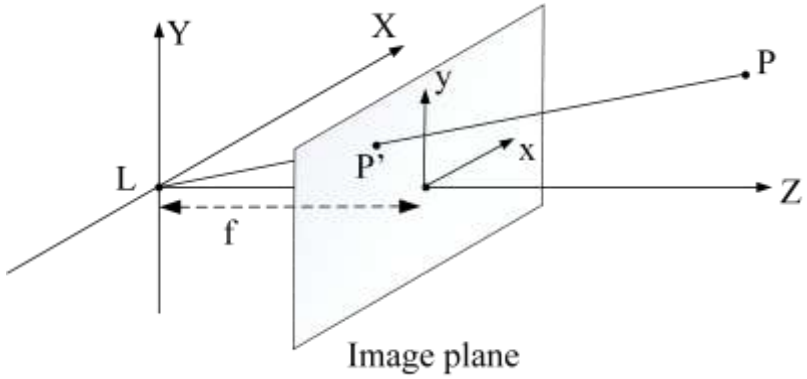


Fig. 1. The equivalent of single thin lens camera geometry

We apply the equivalent of single thin lens camera geometry from Fig. 1 to the geometry shown in Fig. 2. The left and the right camera positions have z-axis displacement as shown in Fig. 2a when both optical axis of cameras are parallel and in Fig. 2b when the optical axis of the left camera is rotated around the y-axis; therefore, the target objects in the left and the right images have different scales and aspects.

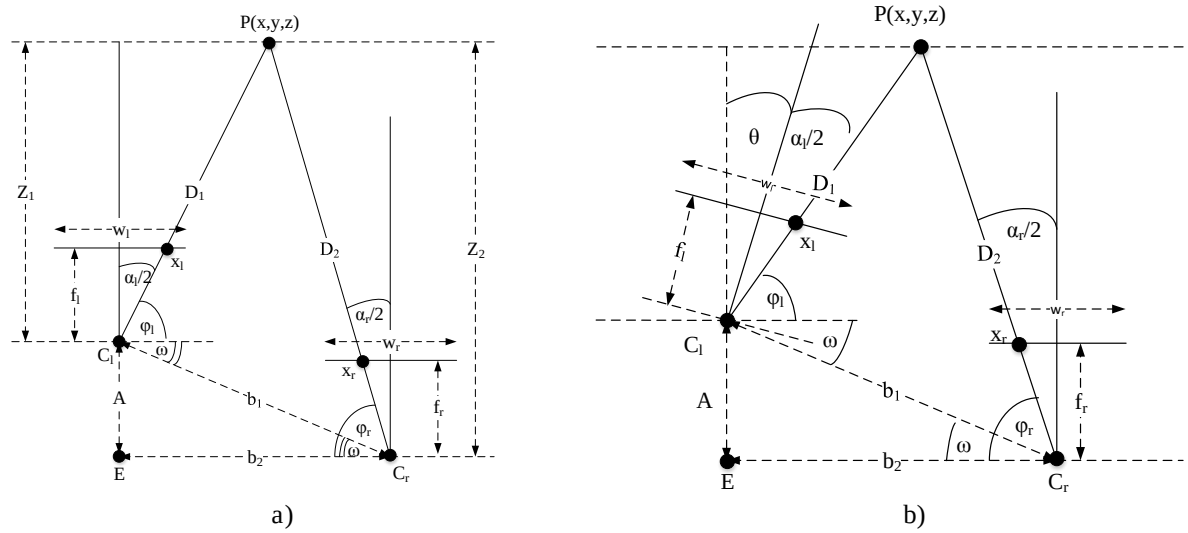


Fig. 2. Triangular geometry (a) both optical axis of cameras are parallel (b) the optical axis of the left camera is rotated around the y-axis

Here, the SIFT algorithm [23] is applied to find the matching points between stereo images when a pair of stereo images have different translation and scales. However, there can be many incorrect matching points or outliers in the result of SIFT that will cause problems for the 3-D reconstruction unless some means of correcting for this effect is implemented. Here we use the RANSAC [24] algorithm to eliminate incorrect matching points after using SIFT. 2-D intensity from the left and the right images saved in 2-D array are used in the SIFT algorithm to find the matching points and then the matching point information is filtered by the RANSAC algorithm to eliminate the outliers. The remain matching point information is used to rescale one of the images. After the pair of images have the same scale, the SIFT and the RANSAC algorithms are used again to find correct matching points between the images. The new matching point information is used to rectify one of the images. In this paper, we used SIFT and RANSAC code from an open source library called VLFeat [28]. Now both left and right images are ready to find the corresponding pixel in the next step.

B. Sum of Absolute Difference algorithm

From a rescaled and rectified image pair, we acquire corresponding points by employing a block-matching algorithm using the SAD algorithm [25, 26]. The SAD value is computed by

$$\underset{x_l,y_l \in SR}{\operatorname{argmin}} SAD(x_l, y_l, x_r, y_r) = \sum_{j=\left(-\frac{B-1}{2}\right)}^{\left(\frac{B-1}{2}\right)} \sum_{i=\left(-\frac{B-1}{2}\right)}^{\left(\frac{B-1}{2}\right)} \left| I_l(x_l + i, y_l + j) - I_r(x_r + i, y_r + j) \right|, \quad (1)$$

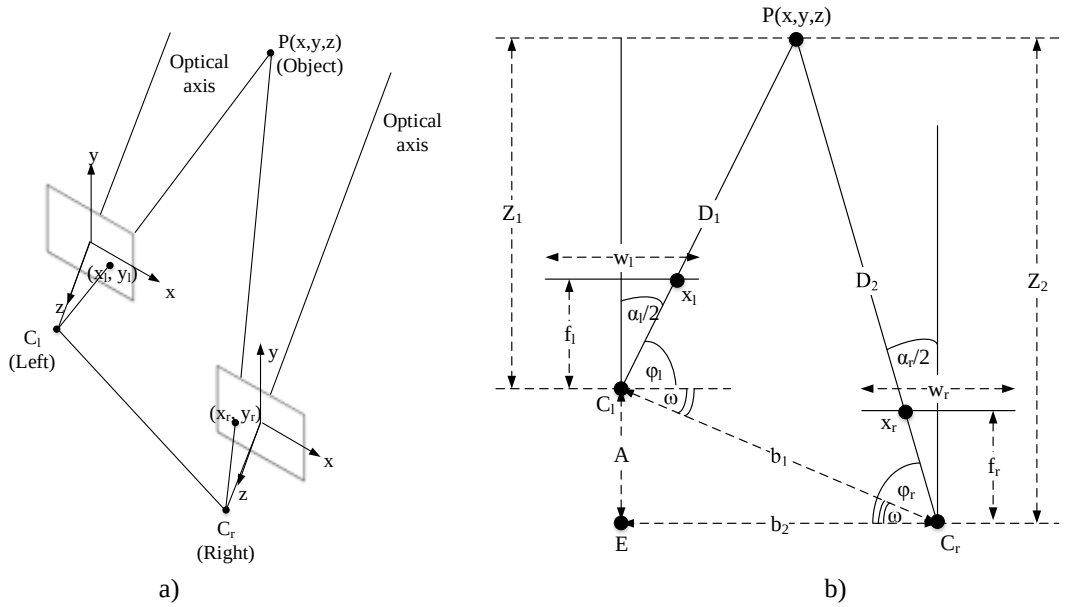
where B is the block size, SR is the search region, (x_l, y_l) is the candidate corresponding pixel in the left image, (x_r, y_r) is the interested pixel in the right image, and I_l and I_r are the pixel intensities in the left and the right images, respectively.

In Equation (1), SAD is calculated by taking the absolute difference between each pixel in a square block of certain size around the pixel of interest in the right image (reference image) and finding the corresponding pixel within the square block in the left image, while moving along the corresponding scan line or the search region. There should be only one best pair of corresponding points between the left and the right images that are determined when the value of SAD is minimum over the search region. When each pair of corresponding points between the left and the right images is found, the 3-D object point will be calculated as described in the next step until cover all corresponding points.

C. Depth from triangulation

The typical stereo vision system [22] is set up with two cameras positioned parallel to each other, observing an object placed along the axis perpendicular to a line connecting the cameras, and centred between the cameras. In this case, the standard stereo vision geometry yields a straightforward result for finding the 3-D object points from stereo images. However, in a security system, the camera positions are relaxed. It will not always be in the

parallel position, and the target will not always be on a line bisecting the cameras. In these cases, the requirements of the standard stereo vision geometry will not be satisfied. Here, we propose a new approach for reconstructing a 3-D image from a pair of cameras that are not parallel, while some parts of the images overlap with each other. Our method for calculating a set of 3-D object point positions is presented here with the geometry shown in detail in Fig. 3.



a) b)

Fig. 3. (a) Camera Positions (b) Triangular geometry

In Fig. 3b, the optical axes of both cameras are parallel but the camera positions have z-axis displacement. The left camera is closer to the object than the right camera. Each best pair of corresponding points between the left and the right images from the last step will be used to calculate each 3-D object point here. The angle between the interest point and the camera position in x-axis in the left and in the right images, φ_l and φ_r, are calculated by

$$\varphi_l = \frac{\pi}{2} - \frac{\alpha_l}{2} \quad \text{and} \quad \varphi_r = \frac{\pi}{2} - \frac{\alpha_r}{2}\ , \tag{2}$$

where α_l and α_r are the angle between the optical axis and the interested point in the left and in the right images as calculated by

$$\frac{\alpha_l}{2} = \tan^{-1}\left(\frac{d_l}{2*f_l}\right) \quad \text{and} \quad \frac{\alpha_r}{2} = \tan^{-1}\left(\frac{d_r}{2*f_r}\right), \tag{3}$$

where f_l and f_r are the focal length of the lens of the left and the right camera, and d_l and d_r are the size in the left and the right image from the middle of the image to the interested point as calculated by

$$\frac{d_l}{2} = (x_l - \frac{w_l}{2}) \times \vartheta_l \quad \text{and} \quad \frac{d_r}{2} = (\frac{w_r}{2} - x_r) \times \vartheta_r\ , \tag{4}$$

where x_l and x_r are the points of interest in the left and the right image that represents point P of the object, w_l and w_r are the width of the left and the right image size, and ϑ_l and ϑ_r are the pixel size of the left and the right image.

In another case of the relaxed camera position, the cameras are moved arbitrarily as shown in Fig. 4. From Fig. 3b, when the left camera is rotated θ degrees clockwise around the y-axis, the triangular geometry would be changed as shown in Fig. 4a. The additional geometric considerations to accommodate this situation are as follows.

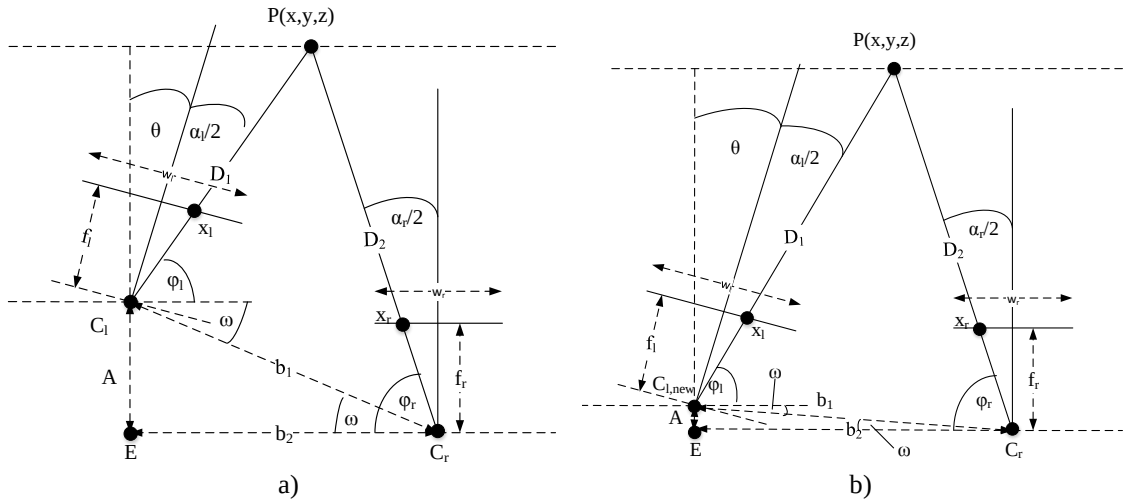


Fig. 4. Triangular geometry used in calculations after rotating the left camera (a) Before rescaling the image (b) After rescaling the image

From Fig. 4b, after rescaling, the new left camera position $C_{l,new}$ is calculated by

$$C_{l,new} = T_2 * R_y * T_1 * C_{l,old}, \tag{5}$$

where $C_{l,new} = \begin{bmatrix} C_{l,new,x} \\ C_{l,new,z} \\ 1 \end{bmatrix}$, $C_{l,old} = \begin{bmatrix} C_{l,old,x} \\ C_{l,old,z} \\ 1 \end{bmatrix}$, and R_y is the rotation matrix around the y-axis with a rotation of θ degrees, T_1 is the translation matrix from the original left camera position to the origin, and T_2 is the translation matrix from the origin back to the origin of the left camera position.

For each best pair of corresponding points, a 3-D object point is calculated. When the left camera is rotated θ degrees, Equation (2) needs to be altered to

$$\varphi_l = \frac{\pi}{2} - \frac{\alpha_l}{2} + \theta \quad \text{and} \quad \varphi_r = \frac{\pi}{2} - \frac{\alpha_r}{2}, \tag{6}$$

where φ_l and φ_r are the angle between the interest point and the camera position in the left and in the right images, α_l and α_r are the angle between the optical axis and the interested point in the left and in the right images as calculated from Equations (3-4), and θ is the degree of the left camera rotation.

D_1 and D_2 are calculated by

$$D_1 = (b_1 \times \sin(\beta_r))/\sin\phi, \tag{7}$$
$$D_2 = (b_1 \times \sin(\beta_l))/\sin\phi, \tag{8}$$

where ϕ is calculated by

$$\phi = \pi - \beta_l - \beta_r, \tag{9}$$

and $\beta_l = \varphi_l + \omega$ and $\beta_r = \varphi_r - \omega$. The quantity ω is calculated by

$$\omega = \sin^{-1}((A \times \sin(\pi/2))/b_1), \tag{10}$$

where $b_1 = \sqrt{A^2 + b_2^2}$; $A = 0$ after rescaling because both camera positions became parallel in virtual scene.

The X and Z information for object points are calculated by

$$D_1 = \sqrt{(C_{lx} - X)^2 + (C_{lz} - Z)^2}, \tag{11}$$
$$D_2 = \sqrt{(C_{rx} - X)^2 + (C_{rz} - Z)^2}. \tag{12}$$

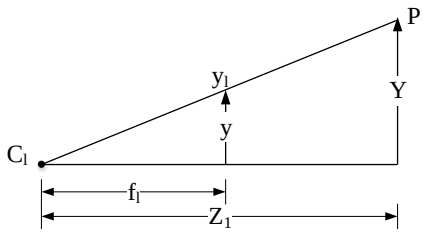


Fig. 5. Height triangular geometry

From Fig. 5, the Y information for each object point from the left camera is calculated by

$$Y = (y/f_l)\times Z, \quad (13)$$
$$y = ((h/2)\text{-}y_l)\times\vartheta_l, \quad (14)$$

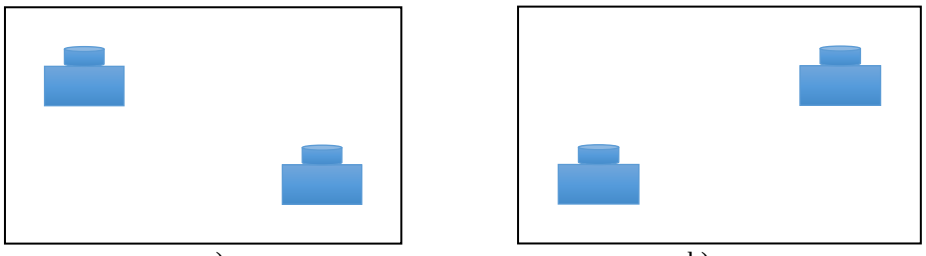
where f_l is the focal length of the left camera, Z is the depth value from point P of the object to C_l, h is the height of the left image size (height by width), y_l is the image point in the left image that represents the location of P, and ϑ_l is the pixel size of the left image. The Y information for the object point can be calculated by using the parameters of the right camera in the same way.

The object point (X, Y, Z) calculation is repeated until all corresponding pixels are calculated. Finally, the 3-D images were displayed from the set of 3-D object points using a 3D scatter plot in MATLAB.

III. Experimental Results

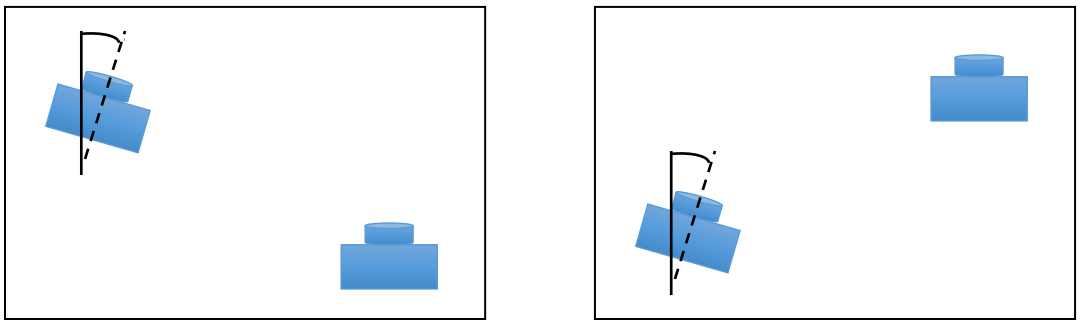
In this section we describe experiments to demonstrate 3-D image reconstruction using the geometry described in Section 3. The cameras used in this study were two identical 1394a Firefly MVs, with an image size of 480-by-640 (height-by-width), pixel size ϑ=6 μm with square pixels, and a focal length of 16.6 mm. The left and the right camera positions had z-axis displacement as shown in Fig. 3 when both cameras were parallel and in Fig. 4 when one camera was rotated around the y-axis; therefore, the target objects in the left and the right images had different scales and aspects. In order to rescale and to rectify the target object in the image, the SIFT algorithm was used to find a set of matching points between the left and the right gray scale images. Color images needed to be converted to be gray scale images before using the SIFT algorithm. The output of SIFT was passed on to RANSAC to find and to exclude outliers from the matching set originally generated by SIFT. Next, a set of 3-D point positions in object space was calculated for each pixel between two rectified images. To find the corresponding pairs, a block-matching algorithm with the SAD in Equation. (1) was used with a 67x67 block size and ± 15 pixels of search region size. After the corresponding points were found in the left image (x_l, y_l) and in the right image (x_r, y_r), the object points (X, Y, Z) were calculated by using Equations. (2) -(14). Finally, the 3-D image was reconstructed from the set of 3-D object points.

To setup the experiment, there were two different conditions for the cameras' settings. The first condition is that both cameras remained parallel in the z-axis, whereas the second condition is that the left camera was rotated θ degrees clockwise around the y-axis and the right camera remained the same. For each condition, the left camera was both moved forward and backward compared to the right camera's position. In the four camera settings, the right camera position was referenced at C_r= $(0, 0, 0)$ and both cameras were positioned at the same height. The four experiment setups are shown in Fig. 6.



a) b)

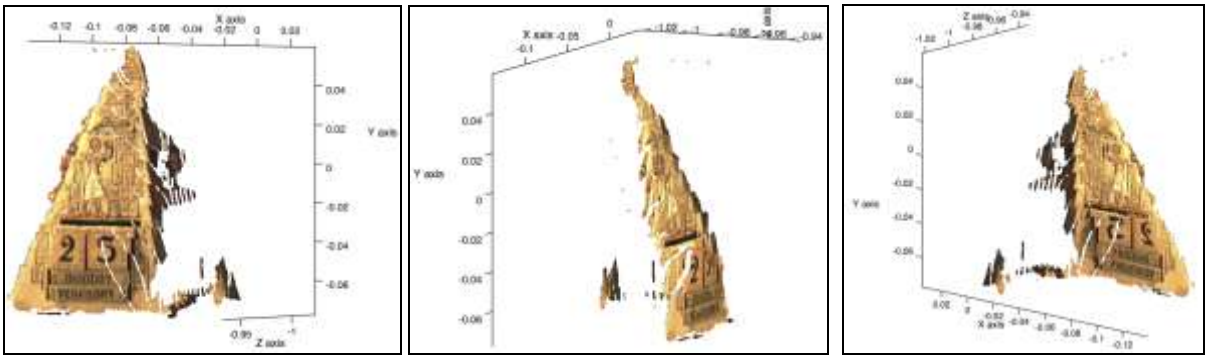
Fig. 6. Four experiment setups from top view
(a) Both cameras were parallel in the z-axis and the left camera was moved forward.
(b) Both cameras were parallel in the z-axis and the left camera was moved backward.



c) d)

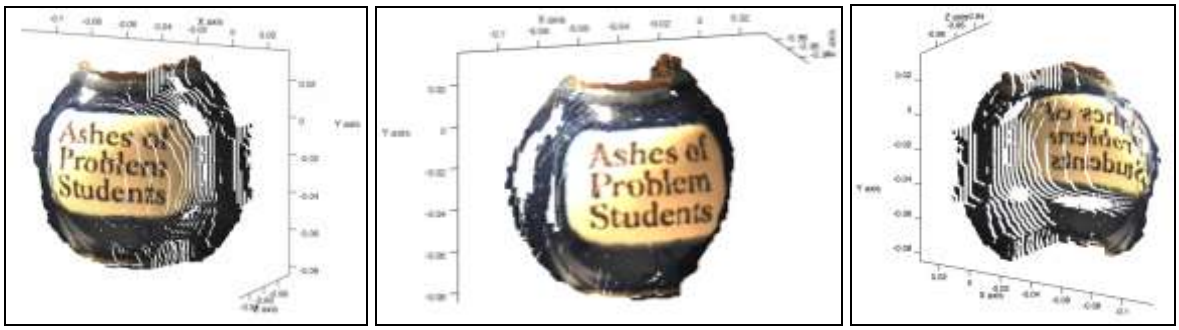
Fig. 6 (cont.). Four experiment setups from top view
(c) The left camera was rotated 7 degrees clockwise around the y-axis and was moved forward.
(d) The left camera was rotated 4.5 degrees clockwise around the y-axis and was moved backward.

The 3-D images were reconstructed from each pair of images taken from all for camera scenarios. Images were taken of five different objects in four experiments. The five objects were a jar, a fox, two dolls, an engine model, and a pyramid. There were two examples where the 3-D images were looked at from multiple viewpoints as shown in Figs. 7 and 8. All 3-D image reconstructions created from 2-D images of the five objects in different conditions and cases are shown in Figs. 9-12. The error between the object actual size and 3-D image reconstructions for all cases of different objects are calculated and shown in Table 1.



a) b) c)

Fig. 7. 3-D pyramid image from different viewpoints



a) b) c)

Fig. 8. 3-D jar image from different viewpoints

Figures 7 and 8 showed that the result of reconstruction of the pyramid and jar from different viewpoints had discontinuous surfaces because of the quantization noise when the cameras captured the real world objects into the pixels of the digital images. These pixels could not represent the continuity of the surface of the objects. This is similar to when converting analog to digital.

Fig. 9. Setup one: both cameras were parallel in the z-axis and the left camera was moved forward.

Fig. 10. Setup two: both cameras were parallel in the z-axis and the left camera was moved backward.

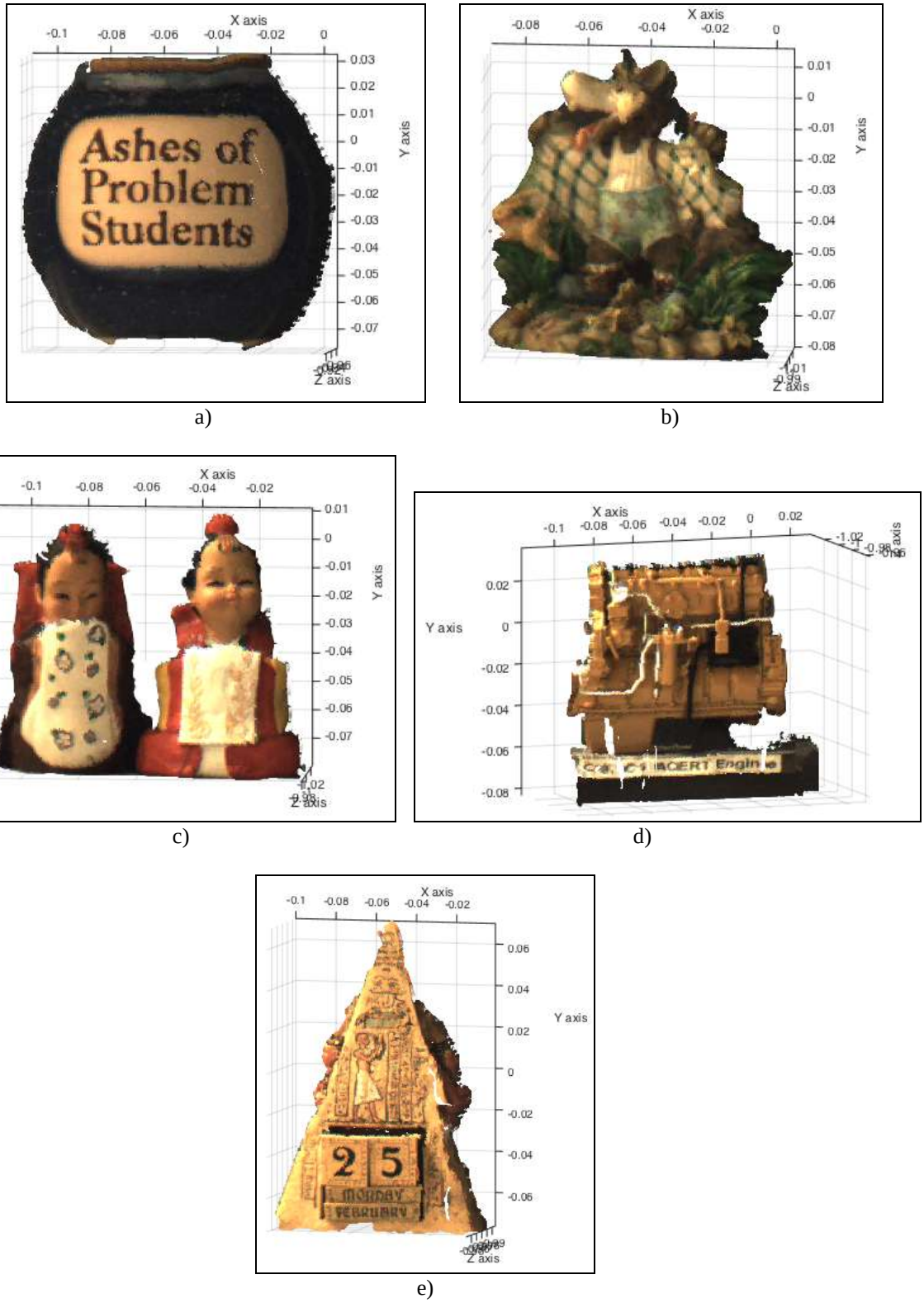


Fig. 11. Setup three: the left camera was rotated 7 degrees clockwise around the y-axis and was moved forward.

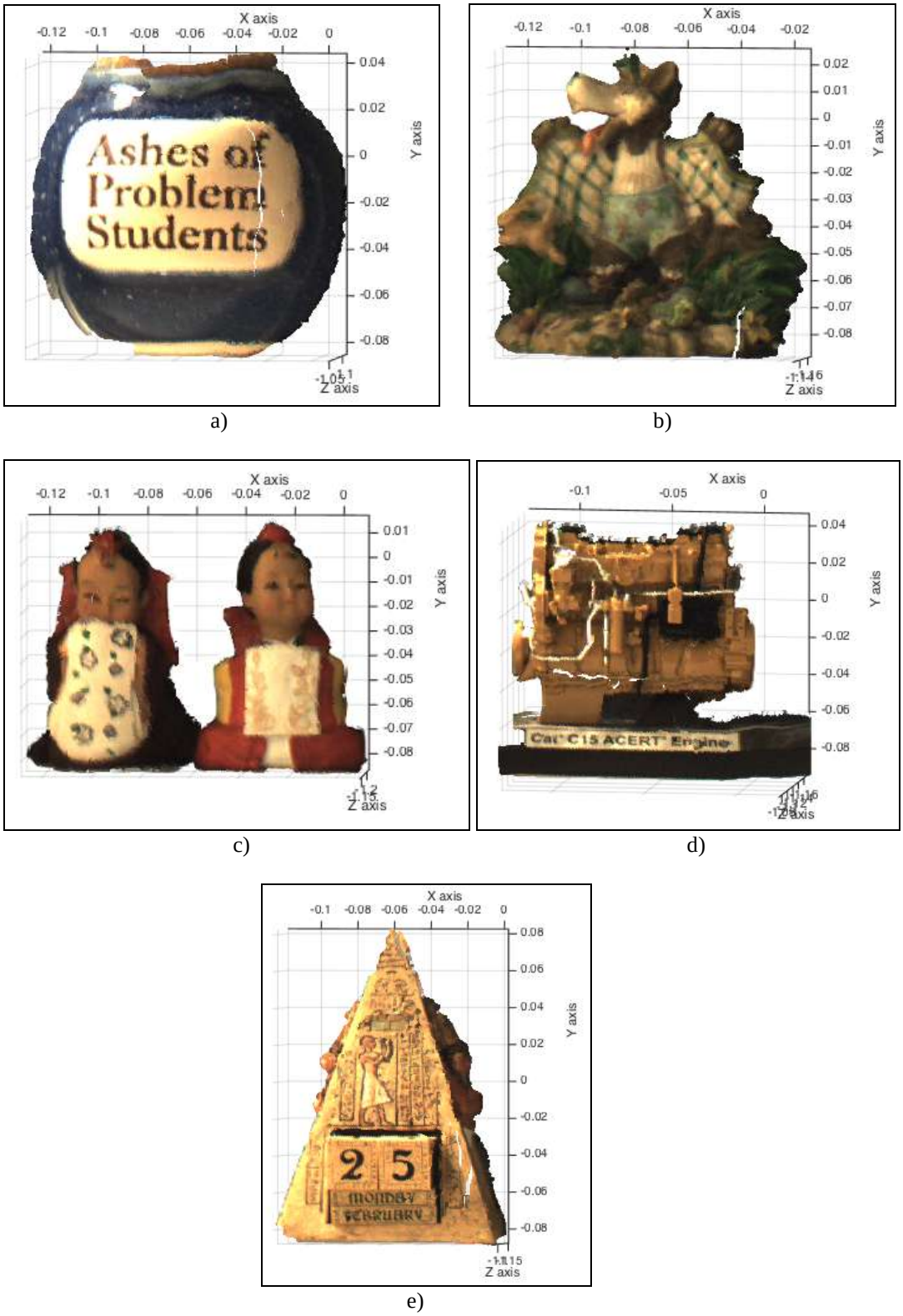


Fig. 12. Setup four: the left camera was rotated 4.5 degree clockwise around the y-axis and was moved backward.

It can be noticed from Figs. 7-12 that the 3-D images contain enough quality 3-D information to represent one side of the actual object. Table 1 shows the error values between the actual object size and 3-D image size for height and width for all conditions and cases. Experiments were performed to evaluate the prototype 3-D geometry algorithm by using RMSE. Table 2 shows RMSE measurement of object sizes between the actual size and the 3-D image size. There are some errors in 3-D reconstructions because there are incorrect matching points during block matching process. This can be attributed to errors in a search region—if the intensity of pixels is about the same, they will give similar results for SAD that lead to high probability of generating the incorrect matching points.

TABLE1. THE ERROR VALUES BETWEEN THE ACTUAL OBJECT SIZE AND THE 3-D IMAGE SIZE (UNIT:CM)

Object			First condition				Second condition			
Name	Size		First case		Second case		First case		Second case	
			3-D image	Error	3-D image	Error	3-D image	Error	3-D image	Error
Pyramid	Height	16.5	13.9	2.6	13.76	2.74	16	0.5	14	2.5
	Width	10.5	10	0.5	9.05	1.45	11	-0.5	9.7	0.8
Fox	Height	10.5	8.78	1.72	8.9	1.6	10	0.5	9.6	0.9
	Width	10	9.8	0.2	9	1	11.5	-1.5	9.5	0.5
Engine	Height	9.5	8.76	0.74	8.98	0.52	7.6	1.9	9.21	0.29
	Width	11.5	10.24	1.26	9.7	1.8	9.4	2.1	10.96	0.54
Doll	Height	9.5	8.6	0.9	8	1.5	9.75	-0.25	8.89	0.61
	Width	6	6.1	-0.1	5.48	0.52	6.4	-0.4	6	0
Jar	Height	12	10.37	1.63	9.76	2.24	11.9	0.1	10.13	1.87
	Width	12	12	0	11	1	11	1	11.4	0.6

TABLE 2. QUALITATIVE RESULTS FOR THE 3-D IMAGES FOR ALL CASES (UNIT:CM)

	First condition		Second condition	
	First case	Second case	First case	Second case
RMSE	1.249	1.588	1.106	1.12

IV. CONCLUSION

In this study, we proposed a triangular geometry to calculate 3-D information objects. This set of equations were used with processed images when the two cameras had x and z-displacement shift, and when one camera was rotated around the y-axis. Therefore, a set of 3-D object points could be calculated.

The findings of the study showed that the 3-D information captured in this manner has enough quality to represent one side of the actual object. The RMSE between the actual size and the measured 3-D image result in the first case when the left camera was moved backward for both conditions are less than the second case when the left camera was moved forward. The average RMSE is equal to 1.265. The results indicated that our set of prototype geometric equations could be used to calculate the 3-D information that can build a 3-D image with high reliability.

REFERENCES

[1] Robert-Inacio, F., A. Raybaud, and E. Clement, *Multispectral target detection and tracking for seaport video surveillance.* Proceedings of the IVS Image and Vision Computing New Zealand, 2007: p. 169-174.
[2] Hampapur, A., et al., *Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking.* Signal Processing Magazine, IEEE, 2005. 22(2): p. 38-51.
[3] Valera, M. and S.A. Velastin. *Intelligent distributed surveillance systems: a review*. in *Vision, Image and Signal Processing, IEE Proceedings-*. 2005. IET.
[4] Bird, N., et al. *Real time, online detection of abandoned objects in public areas*. in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. 2006. IEEE.
[5]Singh, A., et al. *An abandoned object detection system based on dual background segmentation*. in *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*. 2009. IEEE.
[6] Cucchiara, R. *Multimedia surveillance systems*. in *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*. 2005. ACM.
[7] Calderara, S., et al. *Entry edge of field of view for multi-camera tracking in distributed video surveillance*. in *Advanced Video and Signal Based Surveillance, 2005. AVSS 2005. IEEE Conference on*. 2005. IEEE.
[8] Mustafah, Y.M., A.W. Azman, and M.H. Ani. *Object Distance and Size Measurement Using Stereo Vision System*. in *Advanced Materials Research*. 2013. Trans Tech Publ.
[9] Nedevschi, S., et al. *High accuracy stereo vision system for far distance obstacle detection*. in *IEEE Intelligent Vehicles Symposium*. 2004.
[10] Stoyanov, D., et al., *Real-time stereo reconstruction in robotically assisted minimally invasive surgery*, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*. 2010, Springer. p. 275-282.
[11] Murray, D. and J.J. Little, *Using real-time stereo vision for mobile robot navigation.* Autonomous Robots, 2000. 8(2): p. 161-171.
[12] Kim, H., S.-j. Yang, and K. Sohn. *3d reconstruction of stereo images for interaction between real and virtual worlds*. in *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium on*. 2003. IEEE.
[13] Suhr, J.K., et al., *Automatic free parking space detection by using motion stereo-based 3D reconstruction.* Machine Vision and Applications, 2010. 21(2): p. 163-176.
[14] Pollefeys, M., et al., *Detailed real-time urban 3d reconstruction from video.* International Journal of Computer Vision, 2008. 78(2-3): p. 143-167.
[15] 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.
[16] Tao, T., J.C. Koo, and H.R. Choi. *A fast block matching algorthim for stereo correspondence*. in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*. 2008. IEEE.
[17] Lucas, B.D. and T. Kanade. *An iterative image registration technique with an application to stereo vision*. in *IJCAI*. 1981.
[18] Venkateswar, V. and R. Chellappa, *Hierarchical stereo and motion correspondence using feature groupings.* International Journal of Computer Vision, 1995. 15(3): p. 245-269.
[19] Park, C.S. and H.W. Park, *A robust stereo disparity estimation using adaptive window search and dynamic programming search.* Pattern Recognition, 2001. 34(12): p. 2573-2576.
[20] Boykov, Y., O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2001. 23(11): p. 1222-1239.
[21] Sun, J., N.-N. Zheng, and H.-Y. Shum, *Stereo matching using belief propagation.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2003. 25(7): p. 787-800.
[22] Hartley, R. and A. Zisserman, *Multiple view geometry in computer vision*. 2003: Cambridge university press.
[23] Lowe, D.G., *Distinctive image features from scale-invariant keypoints.* International journal of computer vision, 2004. 60(2): p. 91-110.
[24] Fischler, M.A. and R.C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.* Communications of the ACM, 1981. 24(6): p. 381-395.
[25] Bhaskaran, V. and K. Konstantinides, *Image and video compression standards: algorithms and architectures. 1997*. Kluwer Academic Publishers, Norwell, MA, USA.
[26] Vassiliadis, S., et al. *The sum-absolute-difference motion estimation accelerator*. in *Euromicro Conference, 1998. Proceedings. 24th*. 1998. IEEE.
[27] Scharstein, D. and R. Szeliski, *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.* International journal of computer vision, 2002. 47(1-3): p. 7-42.
[28] Fulkerson, A.V.a.B. *VLFEAT: An Open and Portable Library of Computer Vision Algorithms*. 2008; Available from: http://www.vlfeat.org/.