3D Interface for Easy Operation of Disaster Management Robot Teleportation

Venushka Thisara Dharmasiri
Informatics Institute of Technology
57, Ramakrishna Road
Colombo-06, Sri Lanka
00600
p.dharmasiri@my.westminster.ac.uk

Thilak Chamindra
Informatics Institute of Technology
57, Ramakrishna Road
Colombo-06, Sri Lanka
00600
chaminh@westminster.ac.uk

ABSTRACT

It is difficult to understand the surrounding around a robot from local cameras in a disaster situation since the disaster environment could change rapidly and it could cause a large mental load for human operators in that kind of environment. Human operator needs to have a clear understanding about the height and the distance to control a robot. In addition to that this paper proposes to be developed using a multicore computer with low computational time with low memory consumption. Main contribution is made using none of precisely calibrated cameras, blue screen, special sensor device or special hardware. This paper discusses computer vision based 3D simulation application for a disaster management robot. We also performed an experiment using Flea2 camera dataset. This has got seven hundred frames for a view. All the frames captured under medium illumination level. All the frames we achieved are 3D views of every frame. Mainly our proposed approach is capable of taking multiple cameras feeds (2D feeds) and project a 3D cam feed to the computer screen. This identifies and matches common pixels of images and those matched images are used for 3D video processing and manipulation of 3D stream based on controllers.

Categories and Subject Descriptors

1.4.1 [**Digitization and Image Capture**]: Imaging geometry

1.4.6 [**Segmentation**]: Edge and feature detection

General Terms

Algorithms, Performance

Keywords

3D Interface, RANSAC, SGM, Fronto parrel, Harris detector, SIFT, FLANN matcher, Epipolar geometry

*IWAIT'*15, Oct. 8–10, 2015, Aizu-Wakamatsu, Japan.

1. INTRODUCTION

The existing system is based on the mobile robot. Cameras are mounted at various angles. Ground control and mobile robot communicates in a blind way. Ground control has one screen for one camera view. Sometimes a large screen is divided into views. The human operator can monitor all camera feeds and take decisions according to the environment further; in general situations robots are mounted with two cameras. One camera is a self- rotating camera.

Human operators are capable of viewing more than one screen and rotate the camera and take decisions within a very limited time period. But in a disaster environment, time is the most valuable factor. Most of the times in disaster situations, the environment changes dynamically & rapidly. Then controlling a mobile robot would be a challenging task. In that kind of a situation the human stress & fatigue will increase and it may cause large mental load for a human operator. This problem occurs when viewing various video feeds at the same time. If human operator had a clear understanding about the height and distance to control a robot, the impact on the operator wouldn't be the same. To overcome that issue our proposed solution is to develop a 3D Interface for the disaster management robot.

3D applications can be divided into hardware oriented & software oriented. Each approach has strong points and defects.

Most of the Hardware based approaches are based on sensors. [1, 2, 3, 4]. Following limitations can be identified in hardware based approaches. Uses special sensor nodes which are expensive and sensor integration, fusion takes time. On the other hand, software based approaches are based on vision based algorithms. SLAM concept [5] which is used for constructing the 3D depth map of an unknown environment or update a map within the known environment. Sato [6] has used multiple fish eye cameras to create bird eye to increase the scope of the view. Some approaches are not good under various lighting conditions [7]. Strictly calibrated, fixed position and focuses are used by Matsuyama[8]. This method is difficult to extend in physical resources.

SPS-StFI method [9] uses semi global block matching to construct a semi dense depth map on the reference image. This method says that a pixel is connected within the stereo field, if the semi global block matching does not return an estimate for that pixel. Defined unary cost of a depth in a pixel is the average of the costs of the flow & stereo matching. Under this method they developed two cost functions for stereo & motion pair if does not return an estimate for that pixel.

2. SYSTEM OVERVIEW

To develop the 3D algorithm, we use photos which are taken from 3D objects and convert those into 2D environment. This paper finds the relationships between those images (2D images) and recreates the 3D scene back.

This system uses a faster implementation. After acquiring a frame of two images, the system finds the correspondence between right & left images. To do that the proposed approach is key point based matching method. Detecting key points and extract the key points are more important. To extract the key points, scale and meaningful neighborhood is considered.

To extract precise image correspondence, Scale invariant Feature tracker (SIFT) is a well-known method for image matching [10]. Each extracted key point has a descriptor value. These descriptor values are matched using fast approximate nearest neighbor search method. (FLANN) most of the key points have a nearest descriptor value. By using FLANN matcher these key points can be identified. It helps to remove false matches.

Changing octaves count increases the computation time and key points. To get the correct octaves count, image size and resolution needs to be taken. The proposed solution's octaves count is kept as three. Increasing the octave number causes detection of both small and large size features of an image.

Images are taken around the scene. According to matched points, image rectification is applied. Most of the researchers use camera insentric parameters and excentric parameters based on the perspective transformation to image rectification. This paper approach is based on uncelebrated cameras. In order to perform it, fundamental matrix [11, 12] is calculated which is 3X3 matrix which determines corresponding points of images. *x* and *x'* are corresponding points in stereo image pair. *Fx* describes a line on the corresponding point x' on the other image. That means, for all pairs of corresponding points holds as follows

$$x'^T F x = 0 \text{ ------------------- } \mathbf{(1)}$$

To calculate the fundamental matrix, RANSAC algorithm is used. RANSAC algorithm takes inliers and do not take outliers as fundamental points and takes optimal fitting points. Which consider equals or less than eight best points to calculate the fundamental matrix. This ensures that selected points are not affected by noise.

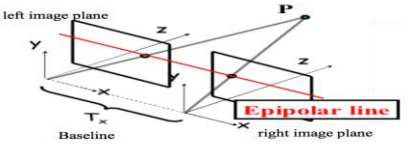


Figure 1: After rectifying two images

After finding the corresponding points of images, perspective transformation is applied to output images according to points of input images which is called as rectification.

The image transformation is used under epipolar geometry. This ensures that number of geometric relations between the 2D points and their projection onto the 3D image that lead to constraint's between the image points.

By using rectified images, disparity map is created. Disparity refers to the distance between two corresponding points of the common object images. Every pixel contains the disparity /distance value for that pixel. In this disparity map the brighter shades represent more shifts and lesser distance from the point of view (camera). The darker shades represent a lesser shift and therefore greater distance from the camera.

The Semi-Global Matching stereo method (SGM) [18] is one of the most frequently used depth cost estimation technique for block-matching because it is lightweight and easy to implement, which is a great advantage in real-time applications with limited memory and computing power. Which helps perform correlation methods to correction methods simplicity which assumes that all pixels within the window have the same distance from the camera. For an example, Slanted surfaces and abrupt changes, as caused by depth discontinuities, which helps to avoid result in wrongly including non- corresponding image parts to make a disparity map calculate.

For Right Camera,

$$cx = f\frac{x}{z} \qquad y_l = f\frac{y}{z} \text{ ------------------ } \mathbf{(2)}$$

For Left Camera,

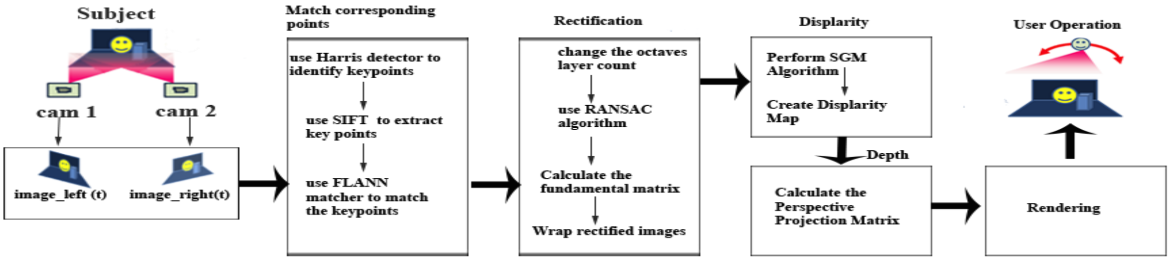
$$cx' = \mathrm{f}\,\frac{x - Tx}{z} \qquad y_r = f\frac{y}{z} \text{ -------------- } \mathbf{(3)}$$

Perspective transformation matrix from uncelebrated cameras is a research filed. We used a solution which is based on fronto parral assumption.

cx, *cy* takes as the principal point. *x, y* are image coordinates of input images. *cx* is the image center of *x* & *cy* is the image center of y. *f* is the focal length of camera. *Tx* is the difference between two cameras or translation between two cameras.

Left camera y and right camera y has the same coordinate. By changing cx & *cx*' It is possible find the depth of pixel. In order to find the depth of pixel perspective matrix is used (*Q*).

$$Q = \begin{bmatrix} 1 & 0 & 0 & -cx \\ 0 & 1 & 0 & -cy \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/Tx & (cx - cx')/Tx \end{bmatrix} \text{ -------------------- } \mathbf{(4)}$$



Subject
cam 1
cam 2
image_left (t)
image_right(t)
Match corresponding points
use Harris detector to identify keypoints
use SIFT to extract key points
use FLANN matcher to match the keypoints
Rectification
change the octaves layer count
use RANSAC algorithm
Calculate the fundamental matrix
Wrap rectified images
Displarity
Perform SGM Algorithm
Create Displarity Map
Depth
Calculate the Perspective Projection Matrix
User Operation
Rendering

Figure 2: Scematic diagram of 3D interface for disaster management robot teleportation

3. EXPERIMENTS

Table 1: Specification sheet one

Machine	Apple MacBook Pro
CPU	2.4 GHz Intel Core i5
Memory	8 GB 1600 MHz DDR3
Graphics	Intel Iris 1536 MB
Data set	technique of [13], published in [14, 15](Art, Books, Dolls, Laundry, Moebius, Reindeer, Computer, Drumsticks, Dwarves)

This section presents performance evaluation. In order to perform evaluation, larger width pixels and height pixels image are reduced by 1, 1.5, 2.0, 2.5… etc. Take the width axis and measure the algorithms computational time. Small images such as 32X 24 types cannot be taken for experimental use which does not help to identify the points. Small images have fewer details which may cause the lack of matching points.

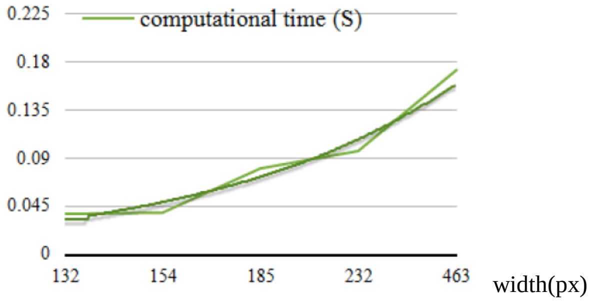


Figure 3: Specification dataset one computational time

The image points matching key points change the octaves count (pyramid layer) from which the key point has been extracted. Which Can be changed and obtain the best octave count. Changing the octaves count may cause high computational time and outliers. Outlier algorithms then take much time to take inliers by considering the facts experimentally. Octaves count is kept under the value of three. Less octaves count causes low matching points. Few matching points directly effects to the 3D model of the system. Low matching points are gradated incomplete 3D scene.

Table 2: By changing octaves count of a stereo frames

Octaves count	Computational time (S)	3D effect
2	0.108	Very bad
2.1 - 2.2	0.104 – 0.105	Very bad
2.3 - 2.8	0.105-0.108	Fair
2.9 - 3	0.107	Good
3.1 - 3.9	0.102-0.1107	Bad
4 - 4.2	0.11	Fair
4.3 - 4.4	0.106-0.109000	Bad
4.5 -5	0.109 -0. 111	Fair
5-10	0.107-0.113	Very bad

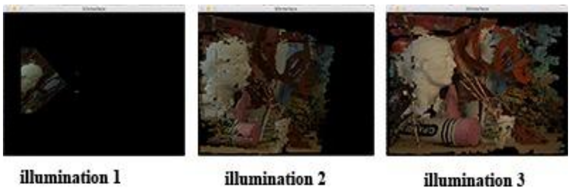


Figure 4: illumination levels

An algorithm is checked using various light conditions illumination level 1, illumination level 2 and illumination level 3. Light condition directly affects the accuracy of the 3D model.

Changing the light condition causes the reduction of good matching points of the images. It causes low computational time.

Low lighting conditions are tested using experimented data set. Under illumination 1 to 3 with exposure level 0. Tested images (figure 4)

The algorithm is checked against the light intensity values.

The Computational times of above illuminations are as follows,

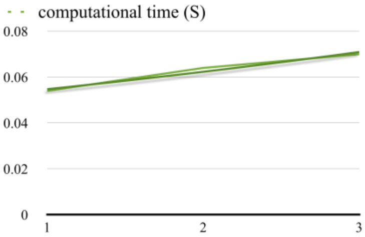
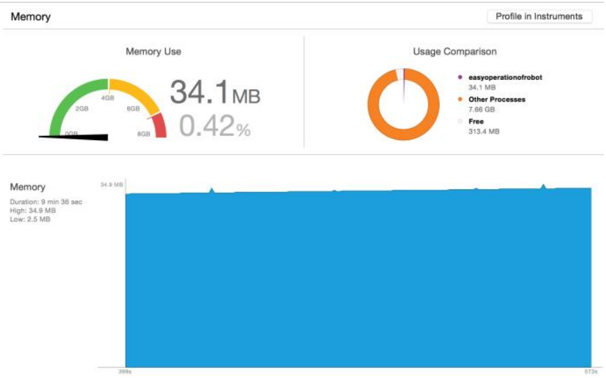


Figure 5: Computational time of figure 4

Table 3: Specification sheet two

Machine	Apple MacBook Pro
CPU	2.4 GHz Intel Core i5
Memory	8 GB 1600 MHz DDR3
Graphics	Intel Iris 1536 MB
Camera	3×PointgrayFlea2 IEEE1394b
Image size	320×240QVGA,RGB888Color
Camera separation	From 10 cm to 50 cm, optional
Camera separation	From 2 m to 4 m, optional



34.1 MB
0.42%

easyoperationofrobot 34.1 MB
Other Processes 7.66 GB
Free 313.4 MB

Memory
Duration: 9 min 36 sec
High: 34.9 MB
Low: 2.5 MB

Figure 6: Memory consumption of specification sheet two

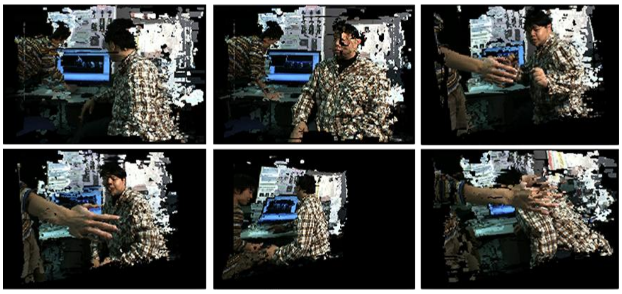


Figure 7: 3D Effect of Specification sheet two

4. CONCLUSION

In this proposed solution, in order to obtain the 3D model baseline of the stereo should be measure by the user. In a robotic environment, it can be planted to know the length. So baseline can be obtained easily.

Most of the real world images are non-linearly deformed when compared with those who captured at a different time or in a different viewpoint. Therefore in order to increase the accuracy of proposed solution it needs to have strict matching. In order to perform, the dense based matching is more suitable instead of using key point based matching. Daisy descriptor is one of the leading dense descriptors which matches images accurately and within less time when compared to SIFT, SURF descriptors.

SIFT uses Key points; Key points are good for analyzing the correspondence of many scenes which are significantly different. Scenes with a small displacement or with a tiny difference can use Dense SIFT or Daisy. Further development needs dense analysis rather than sparse analysis most of the cases. In a disaster situation, cameras can be damaged. The known base line length may change under various circumstances. The factorization method [16, 17] doesn't need the baseline length of cameras. That method needs multiple views (greater than two views) and it assumes affine space instead of the projection space. By using the factorization method, it is possible to calculate the depth of each matched points. Further, above approaches helps to develop a good 3D interface with low memory consumption.

5. ACKNOWLEDGEMENTS

We would like to thank Professor Kaitaro nurse & Associative professor Yuchi Yaguchi (Univercity of Aizu , Japan) wonderful cooperation and helping us dataset and valuable comments throughout the research.

6. REFERENCES

[1] Kuntze, H., Frey, C. W., Tchouchenkov, I., Staehle, B., Rome, E., Pfeiffer, K., Wenzel, A. & Wollenstein, J. Seneka - sensor network with mobile robots for disaster management. Homeland

[2] Joochim, C. & Roth, H. Development of a 3D mapping using 2D/3D sensors for mobile robot locomotion. Technologies for Practical Robot Applications, 2008. Tepra 2008. IEEE International Conference on, 10-11 Nov. 2008 2008. 100-105.

[3] Bruemmer, D. J., Boring, R. L., Few, D. A., Marble, J. L. & Walton, M. C. "I call shotgun!": an evaluation of mixed-initiative control for novice users of a search and rescue robot. Systems, Man and Cybernetics,

[4] 2004 IEEE International Conference on, 10- 13 Oct. 2004 2004. 2847-2852 vol.3.

[5] Baker, M., Casey, R., Keyes, B. & Yanco, H. A. Improved interfaces for human- robot interaction in urban search and rescue. Systems, Man and Cybernetics, 2004 IEEE International Conference on, 10-13 Oct. 2004 2004. 2960-2965 vol.3.

[6] Cui, Y., Schuon, S., Thrun, S., Stricker, D. & Theobalt, C. 2013. Algorithms for 3D shape scanning with a depth camera. IEEE Trans Pattern Anal Mach Intell, 35, 1039-50.

[7] Sato, T., Moro, A., Sugahara, A., Tasaki, T., Yamashita, A. & Asama, H. Spatio-temporal bird's-eye view images using multiple fish-eye cameras. System Integration (SII), 2013 IEEE/SICE International Symposium on, 15-17 Dec. 2013 2013. 753-758.

[8] Xiaojun, W., Takizawa, O. & Matsuyama, T. Parallel Pipeline Volume Intersection for Real-Time 3D Shape Reconstruction on a PC Cluster. Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on, 04-07 Jan. 2006 2006. 4-4.

[9] Matsuyama, T. Exploitation of 3D video technologies. Informatics Research for Development of Knowledge Society Infrastructure, 2004. ICKS 2004. International Conference on, 1-2 March 2004 2004. 7-14.

[10] K. Yamaguchi, D. Mcallester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. ECCV 2014.

[11] D. Lowe, "Distinctive image features from scale invariant keypoints," IJCV, vol. 60, no. 2, pp. 91–110, 2004.

[12] Q.T. Luong and T. Vieville, "Canonic Representations for the Geometries of Multiple Projective Views," Computer Vision and Image Understanding, vol. 64, no. 2, pp. 193-229, 1996.

[13] Bartoli, A. & Sturm, P. 2004. Nonlinear estimation of the fundamental matrix with minimal parameters. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26, 426-432.

[14] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light.In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.

[15] D. Scharstein and C. Pal. Learning conditional random fields for stereo.In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, June 2007.

[16] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching.In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), Minneapolis, MN, June 2007.

[17] Triggs, B. Factorization methods for projective structure and motion. Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on, 18-20 Jun 1996 1996. 845-851.

[18] Vidal, R., Soatto, S. & Sastry, S. S. A factorization method for 3D multi-body motion estimation and segmentation. Proceedings of the Annual Allerton Conference ON Communication Control and Computing, 2002. Citeseer, 1626-1635.

[19] Hirschmuller, H. Accurate and efficient stereo processing by semi-global matching and mutual information. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 20-25 June 2005 2005. 807-814 vol. 2.