ROSE 2010 - IEEE International Workshop on Robotic and Sensors Environments
Phoenix - Arizona, 15-16 October 2010

An Embedded Feature-Based Stereo Vision System for Autonomous Mobile Robots

Pinhas Ben-Tzvi, Xin Xu
Robotics and Mechatronics Lab
Department of Mechanical and Aerospace Engineering
George Washington University
Washington, DC 20052
bentzvi@gwu.edu

***Abstract*—Stereo vision systems have recently become an essential part for most autonomous mobile robots. There are several commercially available stereo vision products, which have been used for autonomous functions in mobile robot platforms. However, most of them are not suitable for compact sized robots. This paper presents an embedded stereo vision system that provides flexible baseline for robots of compact size. In terms of hardware, it provides baseline flexibility, and can be easily fitted into any robot. In terms of software, a feature based stereo vision algorithm is proposed to improve the real-time performance of the stereo vision system. The proposed featured-based method is evaluated by comparing it with the area-based method using standard test images. The results show that the computational time is significantly reduced (reduced by 60% in a relatively complex image). The proposed method is also implemented in real world environments in order to test its effectiveness.**

Keywords-embedded stereo vision system; correspondence; feature extraction;

I. INTRODUCTION

Recently, stereo vision systems have been widely used for autonomous robots. However, there are still many challenging problems in terms of implementation. The most heavily investigated topics include: stereo correspondence methods, real-time performance of stereo vision systems, and hardware implementation.

In terms of stereo correspondence algorithms, a number of algorithms have been proposed in the literature, such as: absolute differences, sum of squared differences, and sampling-insensitive absolute differences [1]. Hirschmuller and Scharstein [2] have recently compared several cost algorithms and concluded that the performance of cost algorithms depends on the stereo vision system type. In general, the stereo vision correspondence can be divided into three classes [3]:

- **Pixel-based methods**: intensity value (or its norm and square) is compared pixel by pixel. They are detail oriented and can obtain dense disparity map. However, these methods are usually computationally complex and sensitive to noise.
- **Area-based methods**: these methods compute block-by-block correspondence. They are less sensitive to noise compared to the previous class and can also obtain dense disparity map, but the accuracy is low in areas where disparities aren't continuous and smooth.
- **Feature-based methods**: these methods use features as matching descriptors, such as edges, lines, regions, and gradient peaks. The dense disparity map cannot be obtained. However, the computation cost is greatly reduced, and they are insensitive to noise.

A comprehensive evaluation and comparison of stereo vision algorithms has been done by Scharstein and Szeliski [4]. They have also established a test bed database for researchers conducting research on stereo vision to compare the performance of their algorithms. Also, they have proposed a general taxonomy for stereo vision systems. However, only algorithms pertaining to generating dense disparity maps are evaluated in [4], and feature-based methods have not been evaluated. In general, many researchers in this area [5–6] have adopted this taxonomy, which consists of the following four steps: a) matching cost computation; b) cost aggregation; c) disparity computation; and d) disparity refinement. Other preprocessing steps can be added to this taxonomy, such as calibration and rectification. A recent framework for stereo vision systems was proposed by Ng and Ganapathy [7], and another comparison paper was published by Brown et al. [8]. The authors compared a number of stereo vision algorithms and concluded that there are still challenges in real-time systems in terms of precision, reliability, and dense depth map.

In terms of system implementation, recently, many researchers have embedded stereo vision systems in real-time applications. Hariti et al. [9] have applied a stereo vision system in real traffic conditions using two cameras mounted on a vehicle. Sawasaki et al. [10] have embedded a stereo vision system into a mobile service robot to perform vision-based navigation in a building hallway.

Su et al. [11] and Mokri and Jamzad [12] have used what is called omni-directional stereo system, where the camera is mounted on the top of the robot, or any system of interest, to provide a top view of the environment using a reflective mirror. The problem with this type of systems is that it needs an extra height for the system, which might not be available. For example, in [10] the omni-directional camera is mounted at 750 mm height.

Kim et al. [13] proposed a cross-visual stereo vision system which tracks objects with specific color and measures the distance to the object by applying the trigonometric measurement method and the robot kinematics to the system. Two motors are used to control the cameras position to keep the object in the center of the image, so that the error caused

by distortion can be reduced. A drawback of this system is that it can only track objects by specific color, which limits its application.

The most recent research work in stereo vision falls under the area-based category; however, feature based stereo vision is still used especially in embedded systems. In [10], the authors have proposed a featured-based stereo vision system. They have computed the correspondence between selected features of the frontal view. The number of features selected to compute the 3D map was very small. Therefore, they were able to use larger kernel size. Also, Messom and Barczak [14] have proposed a feature-based algorithm to detect human faces for mobile robots interacting with humans.

The contributions of this paper are as follows: i) describe a flexible stereo vision system in terms of its baseline for mobile robots of compact size; ii) compare four different correspondence algorithms to identify and select the most suitable algorithm in terms of efficiency, iii) combine the winning correspondence method with feature extraction to improve real-time performance of the stereo vision system.

In section II, the current commercially available stereo vision systems are briefly reviewed. The design of the stereo vision system is briefly described in section III. Section IV briefly describes feature extraction, and four correspondence cost functions are compared. In section V, a feature-based stereo vision algorithm is proposed by combining feature extraction with the selected correspondence cost function from section IV. The feature-based and area-based stereo vision algorithms are compared in terms of effectiveness and efficiency. The proposed feature-based algorithm is tested using real world images, and the experimental results are shown in section VI.

II. COMMERCIAL STEREO VISION SYSTEMS

In this section, we investigate the possibility of integrating a commercially available stereo vision system for a compact robotic platform. There are several commercially available stereo vision systems, as listed in Table 1. But none of these products provide both the flexibility in terms of baseline (the distance between the centers of two cameras) and the adaptability for embedded systems.

The Bumblebee stereo vision system is widely used commercially. It requires PCI interface to connect the image processing board to a central computing unit. The PCI interface is rarely used in embedded systems because of the relatively larger size that the PCI card occupies. Moreover, most stereo vision systems have a fixed baseline, which limits the flexibility of integrating these stereo vision systems into robots of compact size. Although the nDepth's baseline is 6cm, it still requires a dedicated image processing card with PCI interface. The Videre's baseline is discretely flexible; however, it has the same interface problem as the nDepth. The DeepSea G2 system from TYZX Co. and SVS from Surveyor Company are both embedded stereo vision systems; however, the common problem with these products is that the base line is still too large for compact robots.

Based on the above mentioned analysis, the two main problems of the commercially available products are the interface and the baseline. Based on those observations, we designed a stereo vision system using cameras with standard USB 2.0 interface, which provides sufficient bandwidth and requires no extra space for additional convertor board hardware. Moreover, since the camera housing can be custom designed, the baseline can be easily changed to meet the space requirements of the robot design.

TABLE I. COMPARISON OF STEREO VISION SYSTEMS.

	Baseline (cm)	Interface
Bumblebee2	12	IEEE 1394, PCI
nDepth	6	IEEE 1394, PCI
Videre	6, 9, 15, 30	IEEE 1394, PCI
TYZX	22	10/100 Ethernet
Surveyor	10.75	UART/Wi-Fi

III. DESIGN OF EMBEDDED STEREO VISION SYSTEM

We constructed a stereo vision system with two CCD cameras (PointGrey Chameleon USB2.0) and a Single Board Computer (SBC), which has embedded Linux operating system (OS). The stereo vision algorithms were implemented in OpenCV 2.0 environment (open source software available for downloading [15]).

The principle of building this embedded stereo vision system is to provide modularity and flexibility in terms of hardware and software implementation, so that it can be easily fitted into any other robots.

Since typically a SBC is required for robots working autonomously for the purpose of decision making, we designed our stereo vision system based on a SBC board instead of a dedicated DSP. The SBC board used is ADLS15PC [16]. It has an Intel Atom Processor to provide sufficient computing resources. The stereo vision software running on this SBC can also be used for any other SBCs with limited or no changes depending on its OS.

Two off-the-shelf cameras with standard USB interfaces were chosen instead of using two CCD imaging sensors and making interfaces on our own. The frame rate of this camera is 30Hz and the resolution is 640X480. However, this fast frame rate is not required for the decision making program. Since the SBC will also be used for the decision making program, the main concern is how to reduce the computational workload to ensure the overall real-time performance. This problem is addressed later in the paper in Section VI by proposing a feature-based algorithm.

The described hardware components can be readily used, which make this design easily adoptable for other robots. As a case study, the mobile robot on which the stereo vision system is planned to be embedded into is shown in Fig. 1 [17-19].

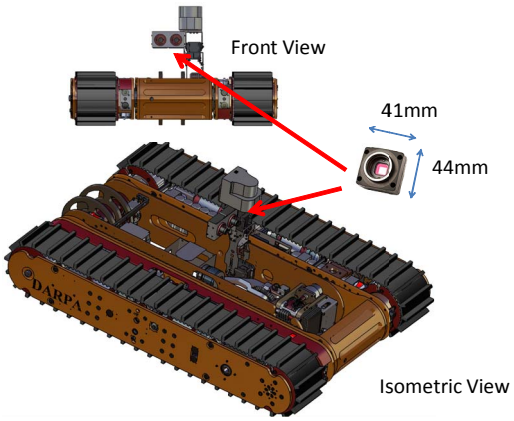


Figure 1. The mobile robot and the stereo vision cameras.

Sheet metal housing was designed to protect the cameras and rigidly fix their relative positions (Fig. 1). The cameras were aligned by using four dowel pins, and were fastened to the sheet metal housing with screws. This mechanism ensures that the cameras are aligned together stably. Even if there are small position errors when mounting the cameras, they can be detected and compensated through the calibration stage.

The proposed stereo vision system provides flexibility in terms of baseline. The distance between the two cameras can be adjusted to fit the design of the robot. In the current design, the baseline is 46mm, and the total width of the stereo vision system is 90 mm, as shown in Fig.1. None of the commercially available stereo vision systems mentioned above can be fitted into such a small space. Although small baseline (46mm) is correlated to short visible range (2.5 meters), since this stereo vision is used to detect obstacles in front of the robot when it is navigating and climbing, this range is sufficient for that purpose.

IV. A COMPARISON BETWEEN FOUR CORRESPONDENCE FUNCTIONS

To find the suitable correspondence cost function, four cost functions are implemented in un-optimized code and compared mainly in terms of computation time. The test images were obtained from the stereo vision testing images library [3]. Note that for better comparison, no post-filtering has been applied to the output of the four matching norms. The four functions are:

- Sum of the Absolute Differences (SAD)

$$\sum_{x=1}^{x=n}\sum_{y=1}^{y=n} abs[M_{l_{(x,y)}} - M_{r_{(x,y)}}] \qquad (1)$$

- Sum of the Square Differences (SSD)

$$\sum_{x=1}^{x=n}\sum_{y=1}^{y=n} [M_{l_{(x,y)}} - M_{r_{(x,y)}}]^2 \qquad (2)$$

- Maximum of Absolute Differences (MAD)

$$\max\,[M_{l_{(x,y)}} - M_{r_{(x,y)}}],\ 1 < x < n, 1 < y < n \qquad (3)$$

- Dot Product Norm (DPN)

$$n - \frac{\sum_{x=1}^{x=n}\sum_{y=1}^{y=n}[M_{l_{(x,y)}} \bullet M_{r_{(x,y)}}]}{\sum_{x=1}^{x=n}\sum_{y=1}^{y=n}[M_{l_{(x,y)}}^2 - M_{r_{(x,y)}}^2]} \qquad (4)$$

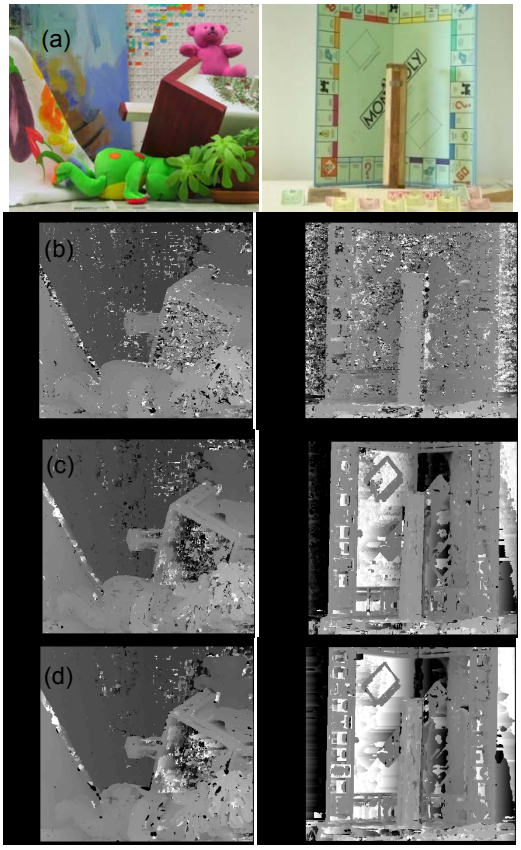
where M_l, M_r are $n \times n$ matrices from the right and left featured-extracted images, and x *and* y are integers.

The resulting disparity images are shown in Fig. 2. The computation time comparison is shown in Table 2.

As can be seen from Fig. 2, all 4 functions have noise in the textureless areas, and the DPN function has better accuracy compared to the other three functions. However, in terms of computation time, the DPN function is much slower than the other three functions. Moreover, feature extraction will only select feature points, which will eliminate the problem caused by textureless areas. Based on these considerations, the DPN function was not selected. There is no significant difference between the MAD, SAD and SSD functions in terms of time and accuracy – all of them can be used for feature-based method. In our experiments, the MAD function was used for the correspondence step.

TABLE II. SPEED COMPARISON OF FOUR COST FUNCTIONS.

Time(sec)	DPN	MAD	SAD	SSD
Teddy	80.157	26.516	28.563	28.766
Monopoly	89.173	37.688	38.516	40.453



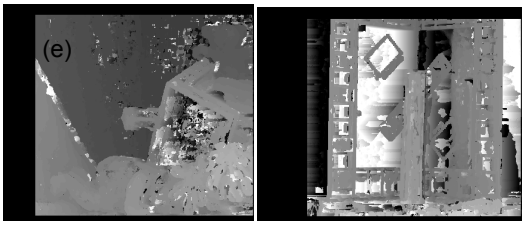


Figure 2. Comparison of different correspondence methods: (a) original left image; (b) dot product (c) max absolute difference; (d) sum of absolute difference; (e) sum of square of difference

V. FEATURE-BASED STEREO VISION ALGORITHM

A. Feature extraction

There are many types of features extraction algorithms. Although some algorithms are more robust, for example SIFT, they may not be able to extract enough features for correspondence. Since the feature extraction step is used for reducing points instead of matching, the algorithm that can extract enough features in any scenario should be selected. In this paper, Canny's feature extraction method [20] was used to collect a list of points on edges. The low threshold of Canny's algorithm was 50, and the high threshold was 150. The thresholds are carefully selected by comparing experimental results to avoid over-extraction (too many feature points) or under-extraction (too few feature points). Over-extraction will slow down the speed of the correspondence step, and under-extraction may not provide enough points for correspondence and therefore may cause correspondence to fail.

To provide more points for correspondence, a post processing step is used to extract larger edge areas (11x11 matrices around each feature points are selected to form a new featured image). The results shown in Fig. 3 represent good feature extraction.

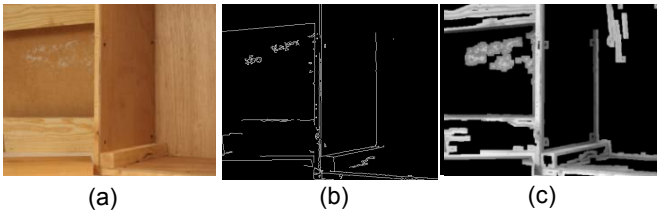


Figure 3. Canny's edge detection and post processing: (a) the original image; (b) Canny edge detection; (c) post-processed feature image

B. A comparison between feature-based and area-based stereo vision algorithms

The proposed feature-based stereo vision algorithm combines the feature extraction and the MAD function to obtain better real-time performance. In the original area-based stereo vision, the correspondence map between the right and the left image are computed on the whole image. Comparatively, in feature-based stereo vision algorithm, the correspondence map is computed only on the features.

According to the results shown in Fig. 4, the correspondence map for the images in case of the feature-based algorithm represents the objects clearly. The computation times are recorded in Table 3, with the feature extraction step computation time included. The computation time of feature-based method is reduced significantly compared to the area-based method. Even in Fig. 3(a), which is complex and contains many edges, the computation time still can be reduced by 60.3%. The results show that the feature-based stereo vision algorithm is more time efficient than the area-based algorithm. Although feature extraction somehow increases the computational burden, it can be compensated by the reduced number of points.

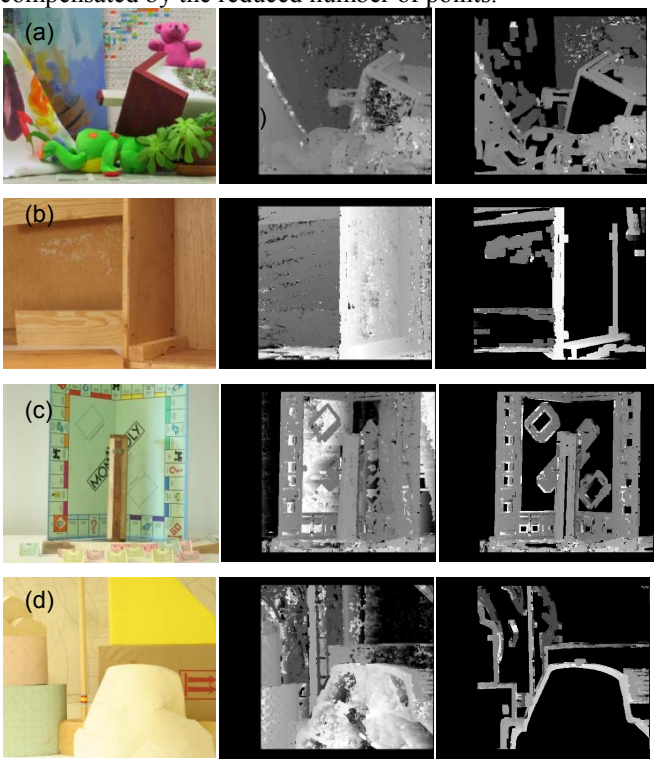


Figure 4. Feature-based stereo vision method: Left images are the original images, central images are the area-based disparity maps, and right images are the feature-based disparity maps - (a) teddy; (b) wood; (c) monopoly; (d) lampshade

TABLE III. A COMPARISON OF AREA-BASED AND FEATURE-BASED METHODS.

	Teddy	Wood	Monopoly	Lampshade
Area-based (sec)	26.5	16.6	37.7	16.1
Feature-based (sec)	10.5	3.5	7.3	2.9
Improvement (%)	60.3	78.9	80.6	82.2

VI. THE STEREO VISION SYSTEM IMPLEMENTATION

The proposed feature-based stereo vision algorithm is implemented in our stereo vision system to test its effectiveness in some preliminary real-world environments.

A. Stereo calibration

Thirteen groups of images of a 5x4 chessboard were used for stereo calibration. The calibration error is evaluated by

checking how closely the points in one image lie on the epipolar lines of the other image [21]. The dot product of the points with the epiline is computed in the undistorted image. In an ideal case, this value should be zero. The accumulated absolute distance forms the error. Since the actual baseline of stereo vision system is known, the calibration result can be checked by comparing the actual baseline with the computed result. The calibration error evaluation is summarized in Table 4.

TABLE IV. CALIBRATION ERROR.

Parameters	Experiment result	Ideal value
Calibration error	1.35	0
Computed Baseline (cm)	7.069	7

Calibration is also used for computing the parameters for the stereo vision algorithm, such as the rotation matrix, translation matrix, and the intrinsic matrix. The results of these parameters are recorded in .xml files during calibration.

B. *Rectification*

One example of rectification is shown in Fig. 5. The left and right cameras are not perfectly horizontally aligned. With the parameters obtained from calibration, images can be rectified (Fig. 5(b)) to make them horizontally aligned.

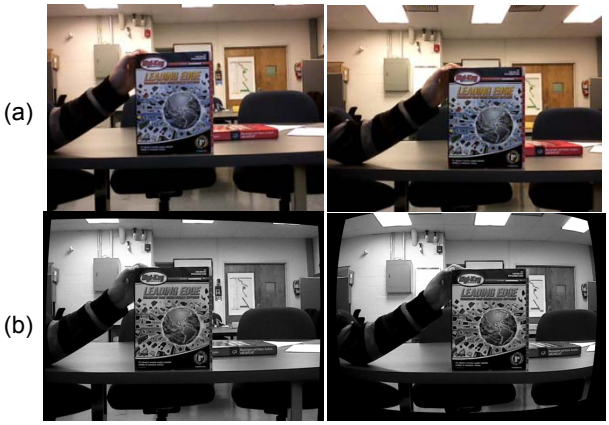


Figure 5. Recification: (a) original images; (b) rectified images

C. *Feature extraction*

The algorithm described in Section V was used to extract edges and corners based on the rectified images. The resulting image is shown in Fig. 6.

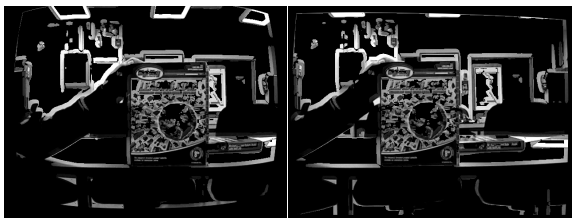


Figure 6. Feature extraction

D. *Results*

The correspondence search was performed for every five rows at the same time. The size of the search matrices, M_l and M_r is 3 by 3. The correspondence points are shown in Fig. 7. The circles denote the correspondence points. The corresponding feature points are found correctly, and can represent the shapes of the objects clearly. Compared to the original number of pixels, 307200, the proposed method significantly reduced the number of searched pixels to 63149. The computational burden is therefore reduced by ~80%.

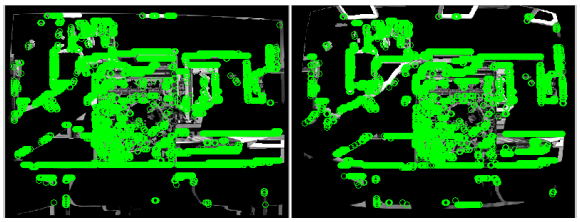


Figure 7. Correspondence points

The disparity map is shown in Fig. 8. The triangulation method described in Section III and the parameters obtained in the calibration step are used to compute the real depth map, as shown in Fig. 9. In the real depth map, the positions of the book features and the background are represented by 3D points, and the depth is represented by z dimension.

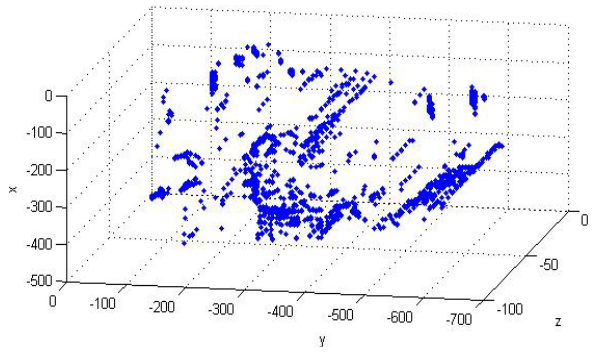


Figure 8. Disparity map

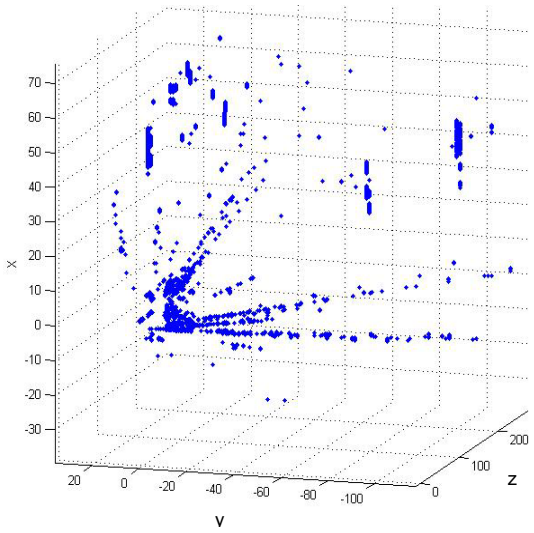


Figure 9. Depth map (cm)

VII. CONCLUSIONS

This paper proposed an embedded stereo vision system with flexible baseline that can be easily fitted into robots of compact size. The stereo vision parameters were obtained through a calibration program after the baseline is changed. A time-efficient feature-based algorithm was proposed to improve the real-time performance. Four correspondence cost functions were compared, and the time-efficient MAD cost function was selected for the correspondence step. The feature extraction was utilized to select the edges of the objects for stereo vision correspondence, and to further reduce the computation time. The proposed feature-based and the original area-based stereo vision algorithms were compared. The experimental results showed that with feature-based method, the computational time can be reduced significantly. Furthermore, some preliminary experimental results in real-world environments show the effectiveness of the proposed feature-based correspondence method.

In future work, this stereo vision system will be integrated into the second generation hybrid mobile robot platform that is currently being manufactured (model shown in Fig. 1). Among other unique functionalities, this robot has the capability to climb tall obstacles [18]. The proposed stereo vision system will be part of the sensor suite used to provide information about the objects and the environment surrounding the robot in order to implement various autonomous and semi-autonomous functions.

ACKNOWLEDGMENT

This work is supported by Defense Advanced Research Projects Agency (DARPA) under grant number R0011-09-1-0049. We also would like to acknowledge the support provided by the DARPA Program Manager, Ms. Melanie Dumas.

REFERENCES

[1] S. Birchfield, C. Tomasi, "A Pixel Dissimilarity Measure That is Insensitive to Image Sampling", *IEEE Trans on Pattern Analysis & Machine Intell*, Aug 1998, vol. 20(4), pp.401-406.

[2] H. Hirschmuller, and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching", *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp.1-8.

[3] H. Liu, "Stereo Matching Using The Discrete Wavelet Transform", *International Journal of Wavelets, Multiresolution and Information Processing*, 2007, vol. 5, no. 4, pp: 567–588

[4] D. Scharstein, R. Szeliski, "A Taxonomy & Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int. Journal of Computer Vision*, vol. 47, Apr-June 2002, pp: 7-42.

[5] D. Scharstein, *View Synthesis Using Stereo Vision*, Springer, Berlin / Heidelberg, 29 June 2003.

[6] D. Scharstein and R. Szeliski. "Stereo matching with nonlinear diffusion", *International Journal of Computer Vision*, 29 Nov 2004 , vol. 28, no. 2, pp:155–174.

[7] O-E Ng, and V. Ganapathy, "A novel modular framework for stereo vision", *IEEE/ASME Int. Conference on Advanced Intelligent Mechatronics*, 14-17 July 2009, pp.857-862.

[8] M.Z. Brown, D. Burschka, and G.D. Hager, "Advances in computational stereo", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Aug. 2003, vol.25, no.8, pp: 993-1008.

[9] M. Hariti, Y. Ruichek, and A. Koukam, "A Fast Stereo Matching Method For Real Time Vehicle Front Perception With Linear Cameras", *IEEE Intelligent Vehicles Symposium*, 9-11 June 2003, pp. 247-252.

[10] N. Sawasaki, M. Nakao, Y. Yamamoto, K. Okabayashi, "Embedded Vision System for Mobile Robot Navigation", *Proc of the 2006 IEEE Int Conf on Robotics and Automation*, Orlando, Florida. 15-19 May 2006, pp: 2693-2698.

[11] L. Su, C. Luo, and F. Zhu, "Obtaining Obstacle Information by an Omnidirectional Stereo Vision System", *2006 IEEE Int Conf on Information Acquisition*, 20-23 Aug. 2006, pp.48-52,

[12] Y. Mokri and M. Jamzad, "Omni-stereo vision system for an autonomous robot using neural networks", *Canadian Conference on Electrical and Computer Engineering*, Saskatoon, May 2005, pp: 1590-1593.

[13] I-H Kim, D-E Kim, Y-S Cha, K-H Lee, and T-Y Kuc, "An embodiment of stereo vision system for mobile robot for real-time measuring distance and object tracking", *Int Conf on Control, Automation & Systems* , 2007, pp:1029 – 1033.

[14] C.H. Messom, and A.L. Barczak, "Classifier & Feature Based Stereo for Mobile Robot Systems", *IEEE Instrumentation & Measurement Technology Conf Proc*, May 2008, pp: 997-1002.

[15] OpenCV 2.0, open source software. Available: http://opencv.willowgarage.com/wiki/, downloaded on Mar 2009.

[16] SBC board ADLS15PC Specifications Sheet. Available: http://www.adl-usa.com/ds/ADLS15PC.pdf, downloaded on May 2010.

[17] P. Ben-Tzvi, A.A. Goldenberg, J.W. Zu, "Articulated Hybrid Mobile Robot Mechanism with Compounded Mobility and Manipulation and On-Board Wireless Sensor/Actuator Control Interfaces", J. Mechatronics, Vol.20, No.6, pp. 627 – 639, 2010.

[18] P. Ben-Tzvi, "Experimental Validation and Field Performance Metrics of a Hybrid Mobile Robot Mechanism", J. Field Robotics, Vol.27, No. 3, pp. 250 – 267, 2010.

[19] P. Ben-Tzvi, A.A. Goldenberg, J.W. Zu, "Design and Analysis of a Hybrid Mobile Robot Mechanism with Compounded Locomotion and Manipulation Capability", Transactions of the ASME, J. Mechanical Design, Vol.130, pp. 1 – 13, 2008.

[20] J.F. Canny, "A computational approach to edge detection", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Nov 1986, vol. 8, no. 6, pp: 679-698.

[21] G. Bradski and A. Kaebler, *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media, Sebastopol, CA, 2008.