

# RAFI - A STEREO VISION BASED AUTONOMOUS MOBILE AREA MAPPING ROBOT WITH 16 DOF

Md. Shafayat Hossain, Ahmedullah Aziz, and Mohammad Wahidur Rahman

Department of Electrical and Electronic Engineering, Bangladesh University of  
Engineering and Technology, Dhaka, Bangladesh  
rumi3.1416@gmail.com

## ABSTRACT

*RAFI is a self-driven mapping robot with 16 Degrees-of-Freedom (DOF). Its software development, structural and electrical design and mechanical construction are presented in this paper. RAFI moves with wheels having differential type motion with 2DOF. The head has 3 DOF sufficient for volumetric mapping by rotating the head in different directions and both hands have 5 DOF to empower its grip and carrying. An 8-bit microcontroller is used to control all the motors. 4 Ultrasonic-rangefinders have been used for obstacle-detection and avoidance which are also interfaced with the same microcontroller. Its head is equipped with two identical webcams for stereoscopic vision for generating 3D map of the field of view after generating disparity map. To optimize the computational speed and mapping accuracy images of 640×480 resolution are 85% compressed and dynamic programming with image pyramiding by quad-pyramid without sub-pixel estimation is pursued.*

## KEYWORDS

*Area Mapping, DOF, Dynamic programming, Image pyramiding, Sub-pixel Estimation.*

## 1. INTRODUCTION

This paper focuses on the architectural design, mechanical construction and software development of an autonomous mobile robot-‘RAFI’, powered by stereo vision which is capable of mapping an unknown area. It is also capable of self-localizing and avoiding obstacles. It has two hands for grabbing objects of different shapes and sizes.

RAFI is equipped with two regular webcams in the head for 3D vision. It has two hands, each having 5 degrees of freedom for grabbing objects. It analyzes data output from the ultrasonic rangefinders to avoid obstacles while moving. The head has 3 degrees of freedom. Being equipped with proper object-matching algorithm, it can recognize objects. It can sense insufficiency of light and can use its own light-source for mapping (if needed). Stereo vision algorithms are implemented in MATLAB [1], run within Microsoft Visual Studio 2010 [2] with OpenCV [3] environment.

This paper is segmented into three parts. In the first portion, software development and implementation sequence is discussed. Then comes the System description which covers the architectural and mechanical construction details. Finally circuit design is detailed, covering the electrical system for proper control of the mechanical system.

## 2. PREVIOUS WORKS

Thrun et al. [4] developed the museum tour-guide robot MINERVA that employs EM to learn its map and Markov localization with camera mosaics of the ceiling in addition to the laser scan

occupancy map. 3D mapping has also been deployed by Thrun et al. [5]. The Monte Carlo Localization method based on the CONDENSATION algorithm was proposed in [6]. Hayet *et al.* [7] extracted and recognized visual landmarks for mobile robot navigation. Planar quadrangular landmarks are extracted from images and homography rectification is applied to obtain an invariant representation for the principal component analysis (PCA) learning stage. Kosecka *et al.* [8] employed gradient orientation histograms to capture the essential appearance information. A Learning Vector Quantization technique is applied to obtain sparser representations by selecting prototype vectors which best cover the class. During the recognition phase, new images are classified using a nearest neighbor test. Smith et al. developed stochastic map estimating the spatial relationships, uncertainty and inter-dependency. One of the appealing aspects of a hybrid metrical/topological approach to mapping and localization [9], [10] is that uncertain state estimates need not be referenced to a single global reference frame. Gutmann and Konolige [11] proposed a real-time method to reconstruct consistent global maps from dense laser range data. The techniques of scan matching, consistent pose estimation and map correlation are integrated for incrementally building maps, finding topological relations and closing loops.

Bosse et al. [12] proposed a hybrid approach by using a graph where each vertex represents a local frame (a local environment map) and each edge represents the transformation between adjacent frames. Loop closing is achieved via an efficient map matching algorithm. Kuipers et al. [13] presented a hybrid extension to the spatial semantic hierarchy, using metrical SLAM methods to build local maps of small-scale space while topological methods are used to represent the structure of large-scale space. Their method creates a set of topological map hypotheses and can handle multiple nested large-scale loops. Our approach also makes use of submaps, but differs from these works as we build 3-D submaps and our map also allows global localization to recover from localization failure.

Stereo vision has been used for mobile robot navigation using stereo correspondence and Kalman filtering [14]. Stephen Se et al. proposed vision based simultaneous localization and mapping by tracking SIFT (Scale Invariant Feature Transform) features [15]. Our approach is to make 3D submaps tracking SURF features and recover depth.

### **3. GENERATING 3D VIEW USING STEREO VISION**

Stereo vision is the process of recovering depth from camera images by comparing two or more views of the same scene. Simple binocular stereo uses only two images, typically taken with parallel cameras that are separated by a horizontal distance known as the "baseline" The output of the stereo computation is a disparity map (which is translatable to a range image) which tells how far each point in the physical scene was from the camera.

MATLAB and Microsoft Visual Studio 2010 with OpenCV are used to compute the depth map between two rectified stereo images. Firstly, basic block matching, which is the standard algorithm for high-speed stereo vision in hardware systems [8] is used. But, it has not shown satisfactory accuracy in generating 3D view. Then dynamic programming is applied to improve accuracy, and image pyramiding is done to improve speed. Combining dynamic programming, image pyramiding and sub-pixel accuracy shows slightly better 3D view but becomes computationally more expensive.

#### **3.1. Rectification**

Stereo image rectification projects images onto a common image plane in such a way that the corresponding points have the same row coordinates. This process is useful for stereo vision, because the 2-D stereo correspondence problem is reduced to a 1-D problem. As an example, stereo image rectification is often used as a pre-processing step for computing disparity or creating anaglyph images. Here, the rectification is computed of two un-calibrated images

without using the camera intrinsic. There is an obvious offset between the images in orientation and position. The goal of rectification is to transform the images, aligning them such that corresponding points will appear on the same rows in both images.

Step 1: Read the stereo image pair (Fig.1)

Step 2: Generate correspondence, points of interest are collected between two images and potential matches are found out using Speeded-Up Robust Features (SURF) descriptor [16], [17]. (Fig.2)

Step 3: Find Putative Point Correspondences using Sum of absolute differences (SAD) metric. Some outliers are still present (Fig.3 (a)).

Step 4: Remove outliers using geometric constraint by approximating the transformation using an affine transform thus, eliminate a substantial number of outliers. Since the underlying transformation between the images is non- planar, a large distance threshold has been set for computing the affine transform (Fig.3 (b)).

Step 5: Remove outliers using epi-polar constraint which is satisfied by the correctly matched points only (Fig.4).

Step 6: Rectify Images using projective transformations,  $t_1$  and  $t_2$ . The overlapping area of the rectified images are finally cropped (Fig.5).



(a)

(b)

Figure 1. (a) Image from left camera (b) Image from right camera (After flipping dimension)



(a)

(b)

Figure 2. Inlier points in both images (a) Left image (b) Right image

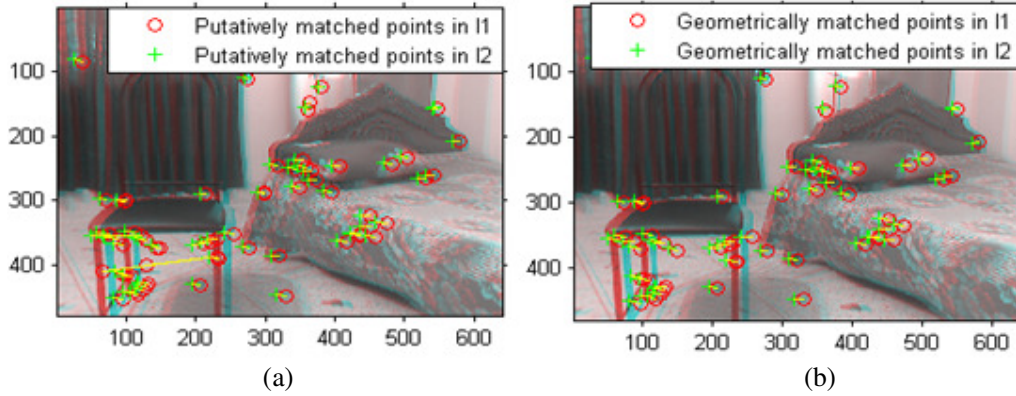


Figure 3. (a) Putatively matched points and (b) Geometrically matched points

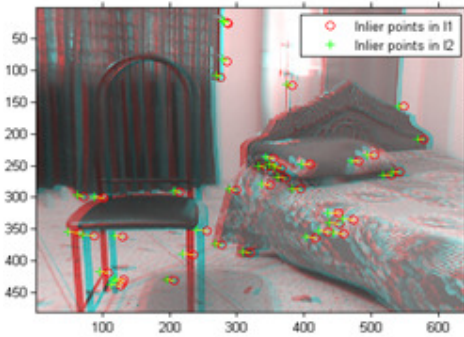


Figure 4. After removing outliers using epipolar constraint



Figure 5. Rectified Stereo image (Red-left and Cyan-right)

### 3.2. Basic Block Matching

For every pixel in the right image, the 7-by-7-pixel block is extracted around it and searched along the same row in the left image for the block that best matches it. The pixel's location in the first image is searched in a range of pixels around, and the sum of absolute differences (SAD) is used to compare the image regions. Only search over columns is required because the images are rectified. We use the Template Matcher System object to perform this block matching between each block and the region of interest.

The left color image  $F_L$  and the right color image  $F_R$  may be represented in the RGB color space as

$$F_L(i,j) = (R_L(i,j) G_L(i,j) B_L(i,j)) \quad (1)$$

And

$$F_R(i,j) = (R_R(i,j) G_R(i,j) B_R(i,j)) \quad (2)$$

$$SAD_{color}(x,y,\Delta) = \frac{1}{n \cdot m} \sum_{i=-k}^{i=k} \sum_{j=-k}^{j=k} (|R_R(x+i, y+j) - R_L(x+i+\Delta, y+j)| + |G_R(x+i, y+j) - G_L(x+i+\Delta, y+j)| + |B_R(x+i, y+j) - B_L(x+i+\Delta, y+j)|) \quad (3)$$

$$Disparity, D = \min_{|\Delta| \leq \Delta_{max}} \{SAD_{color}(x, y, \Delta)\} \quad (4)$$

Correct shape of the stereo image can be recovered using basic block matching, but when no prominent image features are present noisy patches appear. Then the matching process is subject to noise since each pixel chooses its disparity independently of all the other pixels. The depth map is saturated to have only positive values. In this case, the stereo cameras are near perfectly parallel, so the true disparities have only one sign. Thus this correction is valid. Generated disparity map and 3D view are shown in Fig.6, 7.

### 3.3. Sub-pixel Estimation

The disparity estimates returned by block matching are all integer-valued, so the depth map exhibits contouring effects where there are no smooth transitions between regions of different disparity. This can be ameliorated by incorporating sub-pixel computation into the matching metric. The minimum cost and the two neighboring cost values are also considered here as the disparity besides the location of the minimum cost. We fit a parabola to these three values, and analytically solve for the minimum to get the sub-pixel correction.

### 3.4. Dynamic Programming

Basic block matching creates a noisy disparity image which can be improved by introducing a smoothness constraint. Basic block matching chooses the optimal disparity for each pixel based on its own cost function alone. Now, a pixel is allowed to have a disparity with possibly sub-optimal cost for it locally. This extra cost must be offset by increasing that pixel's agreement in disparity with its neighbors. In particular, each disparity estimate is constrained to lie with  $\pm 3$  values of its neighbors' disparities, where its neighbors are the adjacent pixels along an image row. The problem of finding the optimal disparity estimates for a row of pixels now becomes one of finding the "optimal path" from one side of the image to the other. To find this optimal path, block matching metric is used as the cost function and constrain the disparities to only change by a certain amount between adjacent pixels. This is a problem that can be solved efficiently using the technique of dynamic programming [18], [19].

Dynamic programming is applied to each row individually. But it introduces errors of its own by blurring the edges around object boundaries due to the smoothness constraint. Also, it does nothing to smooth "between" rows, which is why a striation pattern now appears on the left side foreground chair. Despite these limitations, the result is significantly improved, with the noise nearly completely removed, and with many of the foreground objects being better reconstructed. Generated disparity map and 3D view are shown in Fig.8, 9.

### 3.5. Image Pyramiding

While dynamic programming can improve the accuracy of the stereo image, basic block matching is still an expensive operation, and dynamic programming only adds to the burden. One solution is to use image pyramiding and telescopic search to guide the block matching [20], [21]. With the full-size image, we had to search over a  $\pm$ pixel range to properly detect the disparities in the image. If the image is down-sized by a factor of two, however, this search could have been reduced to pixels on an image a quarter of the area, meaning this step would cost a factor of 8 less. Then the disparity is estimated from this down-sized operation to seed the search on the larger image, and therefore we only need to search over a smaller range of disparities. A four-level image pyramid is used here. The Pyramid and Geometric Scaler System objects have been used. The disparity search range is only pixels at each level, making it over 5x faster than basic block matching based computation. Yet the results compare favorably.

The disparities  $D(s+1)$  at level  $(s + 1)$  can be derived from the disparities  $D(s)$  of the preceding level  $(s)$  by applying a modified block matching algorithm to the image of level  $(s+1)$ . The search space for the disparity of each block at level  $(s+1)$  is derived from the disparity of the

corresponding block at level (s) by a tolerance factor  $D_T$ . This parameter defines the width  $D_\Delta$  of the reduced search space  $[D_{min}, D_{max}]$  and controls the smoothness of the disparity map.

$$D_\Delta(s) = 2^{s-1} \cdot D_T \quad (5)$$

$$D_{min}(s) = \begin{cases} D(0) - D_\Delta(s) & \text{for } s = 1 \\ D_{min}(s-1) - D_\Delta(s-1) & \text{for } s > 1 \end{cases} \quad (6)$$

$$D_{max}(s) = \begin{cases} D(0) + D_\Delta(s) & \text{for } s = 1 \\ D_{max}(s-1) + D_\Delta(s-1) & \text{for } s > 1 \end{cases} \quad (7)$$

### 3.6. Combined Pyramiding and Dynamic Programming

Finally dynamic programming is merged with image pyramiding, where the dynamic programming is run on the disparity estimates output by every pyramid level. The results compare well with the highest-quality results we have obtained so far, and are still achieved at a reduced computational burden versus basic block matching. Generated disparity map and 3D view are shown in Fig.10, 11. Sub-pixel estimation is introduced for better accuracy and generated disparity map and 3D view for this case are shown in Fig.12, 13.

### 3.7. Back-projection

Re-running basic block matching, we achieve the result below where the contouring effects are mostly removed and the disparity estimates are correctly refined. This is especially evident along the walls.

With a stereo depth map and knowledge of the intrinsic parameters of the camera, it is possible to back project image pixels into 3D points [22], [23].

One way to compute the camera intrinsics is with the MATLAB Camera Calibration Toolbox [24] from the California Institute of Technology(R). Such a tool will produce an intrinsics matrix, K, of the form:

$$K = \begin{bmatrix} \text{focal\_length\_x} & \text{skew\_x} & \text{amera\_center\_x} \\ 0 & \text{focal\_length\_y} & \text{camera\_center\_y} \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Computed Camera intrinsic matrix

$$K = \begin{bmatrix} 822 & 0 & 346.04963 \\ 0 & 822 & 293 \\ 0 & 0 & 1.000 \end{bmatrix} \quad (9)$$

This relates 3D world coordinates to homogenized camera coordinates via:

$$[x_{camera} \ y_{camera} \ z_{camera}]^T = K \cdot [x_{camera} \ y_{camera} \ z_{camera}]^T \quad (10)$$

With the intrinsics matrix, each image pixel can be back projected into a 3D ray that describes all the world points that could have been projected onto that pixel on the image. This leaves the distance of that point unknown to the camera. This is provided by the disparity measurements of the stereo depth map as:

$$Z_{world} = \frac{\text{focal\_length}}{\text{disparity}} (1 + \text{stereo baseline}) \quad (11)$$



Unit less pixel disparities cannot be used directly in this equation. Also, if the stereo baseline (the distance between the two cameras) is not well-known, it introduces more unknowns. Thus this equation is transformed into a general form:

$$Z_{world} = a + \frac{b}{disparity} \quad (12)$$

We solve for the two unknowns via least squares by collecting a few corresponding depth and disparity values from the scene and using them as tie points.

### 3.8. Results

We solve for the two unknowns via least squares by collecting a few corresponding depth and disparity values from the scene and using them as tie points.

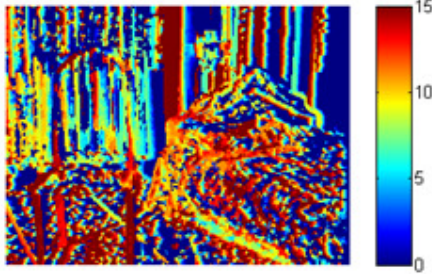


Figure 6. Depth-map from basic block matching. Noisy patches are present

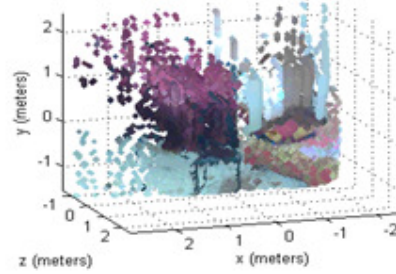


Figure 7. Generated 3D map from basic block matching

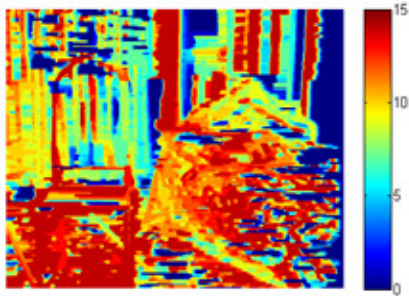


Figure 8. Depth map after dynamic programming

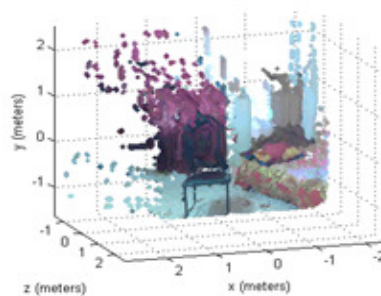


Figure 9. Generated 3D map from dynamic Programming

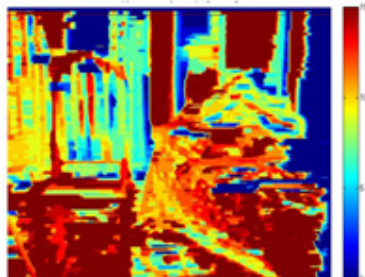


Figure 10. Depth-map after 4 level pyramid with dynamic programming

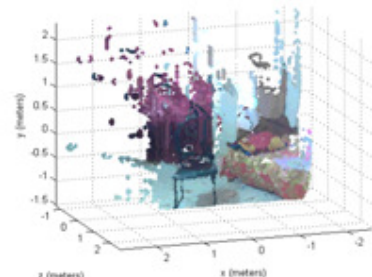


Figure 11. Generated 3D map from 4 level pyramid with dynamic programming

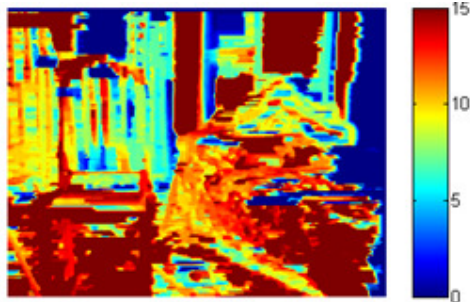


Figure 12. Depth-map after Pyramid with dynamic programming and sub-pixel accuracy

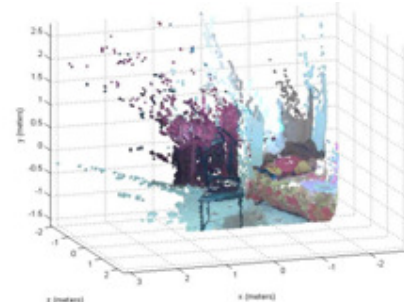
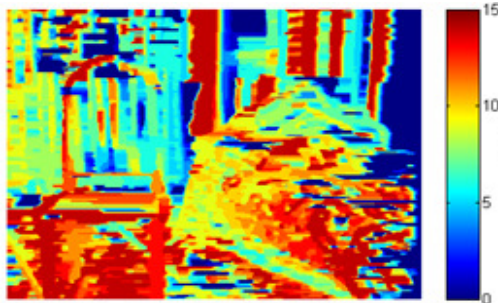
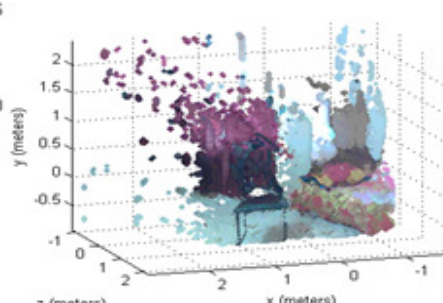


Figure 13. Generated 3D map from dynamic programming and sub-pixel accuracy

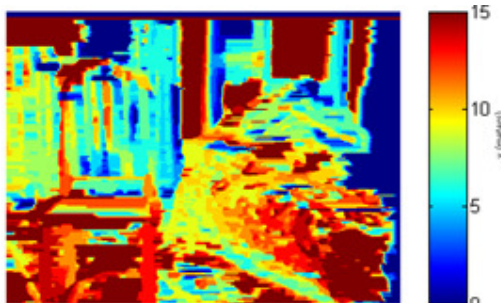


(a)

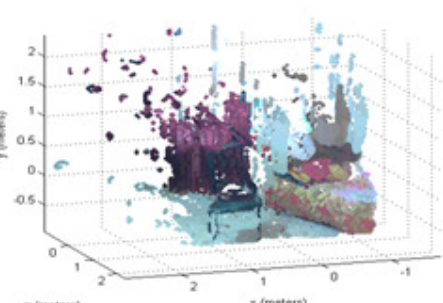


(b)

Figure 14. (a) Depth-map (b) 3D map from dynamic programming after resizing to 85%

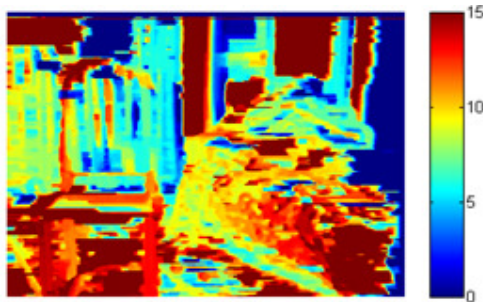


(a)

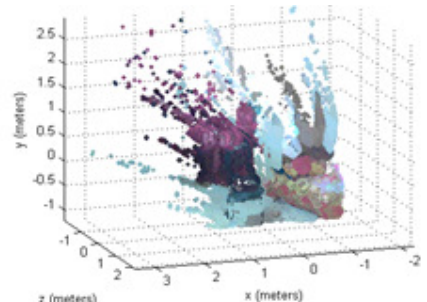


(b)

Figure 15. (a) Depth-map (b) 3D map from 4 level pyramid with dynamic programming after resizing to 85%



(a)



(b)

Figure 16. (a) Depth-map (b) 3D map from dynamic programming and sub-pixel accuracy after resizing to 85%



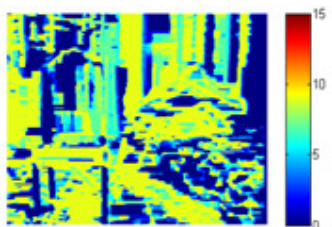


Figure 17. Depth-map after 2 level pyramid with dynamic programming

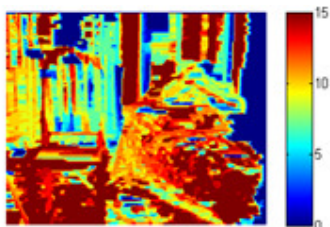


Figure 18. Depth-map after 3 level pyramid with dynamic programming

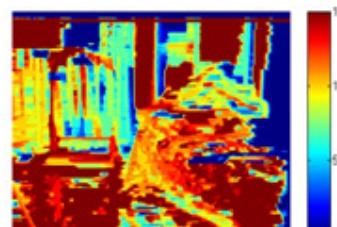


Figure 19. Depth-map after 5 level pyramid with dynamic programming

Table 1. Time requirement for different approaches in generating 3D view

Hardware	Algorithm	No resize	85% resize
Intel Core2Duo 2.66GHz  4GB RAM	Basic block matching	123s	-
	Dynamic programming	121.64s	87s
	4 level pyramid with dynamic programming	140s	95s
	Pyramid with dynamic programming and sub-pixel accuracy	348.30s	281s

After resizing the images into 85%, computation time is decreased significantly but the 3D view is considerably satisfactory (Fig.14-16). If the compression is below 85%, computation time decreases. But the problem is, 3D view becomes blurred and the object seems deformed.

With the increase in the number of pyramids in image pyramiding, computation time increases. 3 and higher level pyramids give satisfactory disparity map. But, 2 level pyramiding shows poor disparity map (Fig.17-19). For our purpose, we choose 4 level pyramiding.

### 3.9. Embedding the Mapping Algorithms in RAFI

Rafi is equipped with a laptop where mapping algorithms are implemented enabling it to generate 3D maps of an unknown area taking series of images with two camera and moving avoiding the obstacles. Mechanical and electrical designs are made suitable for the purpose.

## 4. MECHANICAL CONSTRUCTION

The skeleton of the robot is of cubic shape large enough for placing a laptop on and PCBs within. The cubic-shape box is built with low-cost partex board which is light-weight, strong and durable. For placing the hand and head, a platform is developed using strong but light-weight Aluminum pipe. The total structure without laptop does not weigh more than 8 Kg.

#### 4.1. Wheel setup

Two wheel are coupled with two different gear-motors with same operating-characteristics to facilitate differential drive (Fig.20). An omnidirectional wheel is placed in front for proper weight-balance of the robot.

#### 4.2. Hand Design & Construction

Both of the hands have 5 DOF powered by servo-motors (Fig.21). The hands are built with ultra-durable and light-weight Aluminium sheet and pipe. The shoulders have 2 DOF (Fig.22) and the grips have 3DOF (Fig.23) to facilitate human-like grip using opposable thumb.

#### 4.3. Head Design

The head provides placement of two Logitech c310 HD webcams [25] side-by-side for stereo-vision which enables it to generate 3-D map (Fig. 24). To detect insufficiency of light an LDR is placed and for self-sufficient illumination an LED is kept on the head. For fast movement and weight issues, thin and lightweight Acrylic and Aluminium sheet are used for construction of the head. The webcams are bound to be in fixed position with respect to each other and for this purpose a thin but durable frame is designed to hold the webcams.



Figure 20. Wheel setup for differential control

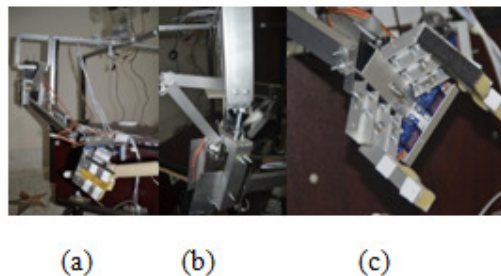


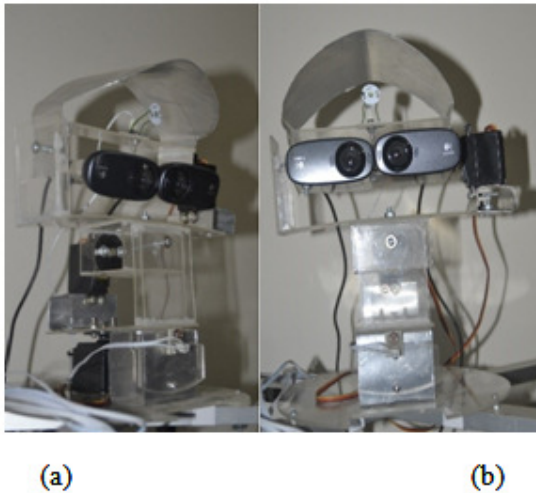
Figure 21. Hand construction (a) Total hand (b) Shoulder (c) Grip



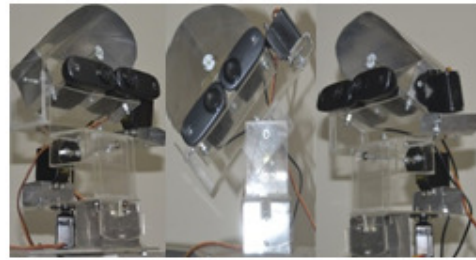
Figure 22. 2 DOF shoulder with different motions



Figure 23. 3 DOF grip showing different movements



**Figure 24. Structure of the Head (a) 3-D view  
(b) front view**



**Figure 25. 3 DOF of the head showing different movements**



**Figure 26. 1 DOF camera at the belly**

The head has 3DOF powered by servo-motors for proper area-mapping (Fig.25).

As backup for the ultrasonic rangefinders a small camera is attached with a servo-motor in the front side of the robot to provide information about any impediments lying in front so that the robot may avoid collision on the course (Fig.26).

## **5. ELECTRICAL DESIGN**

For wheel-rotation, gear-motors are used. Servo-motors are used in the hands and head for providing motion in different directions. Ultrasonic rangefinders are used for getting information about the impediments on the path. All the motors are controlled from laptop using serial communication with the ATmega16 [26] microcontroller. This section covers the total electrical part of the design. Two 3-cell Li-polymer battery packs have been used to power-up the electrical circuits.

### **5.1. Gear-Motor Control**

For high voltage and high current drive of the gear-motors from ATMEGA16 microcontroller, L-298 IC [27] has been used which is a dual-bridge controller for motor drive and can be controlled by sending PWM from the microcontroller into its Enable pin. It supports bi-directional motor-drive with about 46 volt and 3.5 Ampere. Diode-protection using 1N4007 [28] has been deployed for protecting the motor driver from back electromotive force.

### **5.2. Serial Interfacing**

Serial Interfacing is used for communication between Laptop and the microcontroller. It is to be noted that, Serial communication is more efficient than parallel one. ATmega 16 supports Universal Asynchronous serial Receiver and Transmitter (UART) communication [26]. The MAX232 converter [29] allows the microcontroller to communicate with the Laptop using a standard serial cable and the RS232 serial COM port. In order for the Laptop and the microcontroller to communicate successfully, some logic level shifting and translation is necessary. The MAX232 converter IC converts 0 and 5 volt Transistor -Transistor Logic (TTL) signals to -12 and 12 volt RS-232 serial COM port signals. As USB to RS-232 converter is affordable and locally available, system can be seamlessly used in laptops without built-in serial

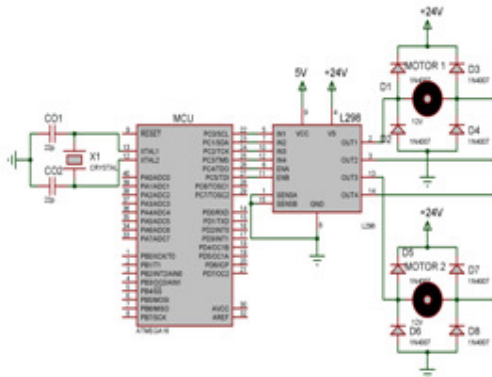


Figure 27. Circuit Diagram for Gear-motor control using L-298

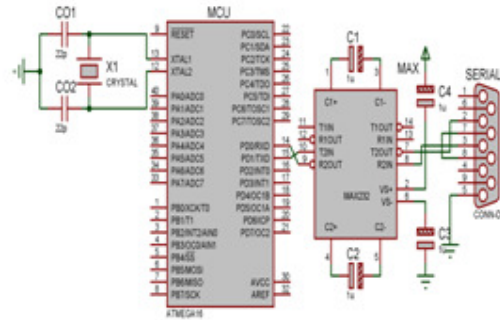


Figure 28. Circuit Diagram for ATmega16 serial interface with PC

port. DB9 connectors are used for serial communication. The circuit diagram for serial communication is shown in Fig.28. UART communication is used with 57600 BAUD rate, 8 bit frame size, 1 stop bit and parity mode disabled.

### 5.3. Servo Motor Control

An (unnumbered) acknowledgements section may be inserted if required. Servo-motors have been used in head and hands. For providing shoulder motions, high-torque (10 kg-cm), coreless, metal-gear, dual-bearing TowerPro MG995 – Standard Servo motors [30] have been used. For grip rotation and motions in the head, relatively low-torque (8 kg-cm), 3-pole, plastic-gear, dual-bearing TowerPro SG-5010 - Standard Servo motors [31] have been used. For light-weight applications like finger movements and small camera-movements low-torque (1.8 kg-cm) TowerPro SG91R - Micro Servo motors [32] have been used. They are shown in Fig. 29.

Servo-motors require approximately 5V and 500mA supply each. So, to power-up all the 14 servo-motors used in this robot linear voltage regulator IC 7805 [33] has been used for constant 5V supply. In normal operation of 7805, it provides only 500mA current. But due to the high current requirement of the system for driving servo-motors, a modified circuit is used incorporating TIP127 [34] which is a pnp Epitaxial Darlington Transistor with large current gain (1000) as show in Fig.30. This circuit can support up to 8A output current.

Servo-motors are controlled by pulse from microcontroller. A signal of 20ms period is sent continuously. For 0 degree position the duty period of the pulse is 1.5ms. For -45 degree and +45 degree the duty periods are 1ms and 2ms respectively.

### 5.4. Operation of Ultrasonic Rangefinder

Detection and avoidance of obstacles is a must for autonomous mobile robots. For this purpose, TS601 ultrasonic electric telemeter module (Fig.31) has been used which is capable of measuring distance within 0.03-3M [35].

It sends the data into pulses having different pulse-widths depending on the measured distance. The pulses from SIG pin are read in ATmega16 using 16bit Timer-counter.

### 5.5. LDR operation for light-sensing

Insufficient lighting may result in poor 3-D map. So, luminosity has been sensed and necessity of lighting has been detected and the robot is equipped with self-lighting scheme (Fig.32).





Figure 29. Servo-motors (a) MG-995 (b) SG-5010 (c) SG 91R

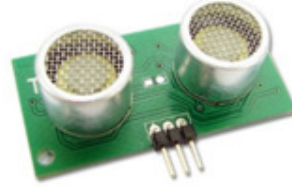


Figure 31. TS601 ultrasonic electric telemeter module

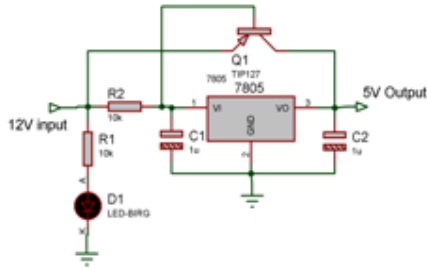


Figure 30. Voltage regulation circuit with high output current

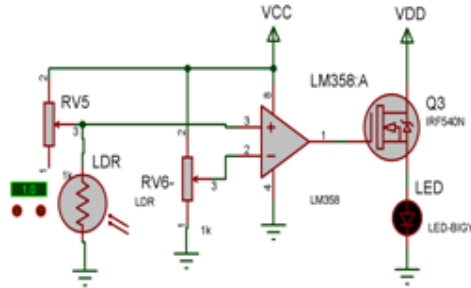


Figure 32. Light sensing and self-illumination circuitry

Locally available LDR has been used which shows good variation of resistance with the luminosity. To differentiate the insufficient lighting, comparator IC LM358 [36] has been used. IRF540N [37] is used for high current switching of LED.

## 6. SYSTEM PROTOTYPE

Fig.33 shows the block diagram of the overall setup. The completed robot in one piece is shown in Fig.34.

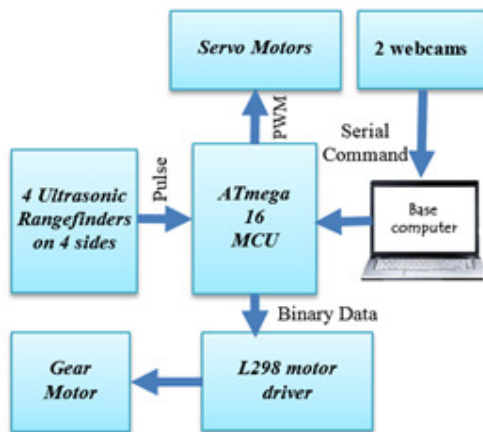


Figure 33. Overall block diagram of the electrical system design



Figure 34. 16 Degrees-of-Freedom Mapping Robot



## 7. CONCLUSIONS

This work manifests the innovative architectural design and mechanical construction of a 16 Degrees-of-Freedom Mapping Robot in detail. To power the mechanical framework, the designed electrical circuitry are also shown. This construction is perfectly suitable for recognizing, localizing and mapping of objects in an unknown area thereby providing a complete picture of an area. This robot is built from low-cost but lightweight and durable materials suited for its purpose. Stereo vision is implemented for accurate localization and mapping. For optimization, dynamic programming with image pyramiding without sub-pixel estimation is used for 3D view generation.

## REFERENCES

- [1] MATLAB. [Online]. Available: <http://www.mathworks.com>
- [2] Microsoft Visual Studio 2010. [Online]. Available: <http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express>
- [3] OpenCV [Online]. Available: [opencv.org/](http://opencv.org/)
- [4] S. Thrun et al., "Minerva: A second-generation museum tour-guide robot," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA'99), Detroit, MI, May 1999, pp. 1999–2005.
- [5] S. Thrun et al., "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), San Francisco, CA, Apr. 2000, pp. 321–328.
- [6] F. Dellaert et al., "Using the condensation algorithm for robust, vision based mobile robot localization," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'99), Fort Collins, CO, Jun. 1999.
- [7] J. B. Hayet et al., "Visual landmarks detection and recognition for mobile robot navigation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'03), Madison, WI, Jun. 2003, pp. 313–318.
- [8] J. Kosecka et al., "Qualitative image based localization in indoors environments," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'03), Madison, WI, Jun. 2003, pp. 3–10.
- [9] K. Chong and L. Kieeman, "Large scale sonarray mapping using multiple connected local maps," in Proc. Int. Conf. Field and Service Robotics, Canberra, Australia, Dec. 1997, pp. 538–545.
- [10] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
- [11] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in Proc. IEEE Int. Symp. Computational Intelligence in Robot. Autom. (CIRA), Monterey, CA, Nov. 1999, pp. 318–325.
- [12] M. Bosse et al., "An atlas framework for scalable mapping," in Proc. IEEE Int. Conf. Robotics and Automation (ICRA'03), Taipei, Taiwan, Sep. 2003, pp. 1899–1906.
- [13] B. Kuipers et al., "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), New Orleans, LA, Apr. 2004, pp. 4845–4851.
- [14] Kriegman, D. J., Triendl, E., & Binford, T. O. (1989). Stereo vision and navigation in buildings for mobile robots. *Robotics and Automation, IEEE Transactions on*, 5(6), 792-803.
- [15] Se, Stephen, David G. Lowe, and James J. Little. "Vision-based global localization and mapping for mobile robots." *Robotics, IEEE Transactions on* 21.3 (2005): 364-375.
- [16] Bradski, G. and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly: Sebastopol, CA, 2008.
- [17] Bay, Herbert, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: "Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008
- [18] Veksler, O. "Stereo Correspondence by Dynamic Programming on a Tree." University of Western Ontario.
- [19] Park, CS; Park, HW. "A robust stereo disparity estimation using adaptive window search and dynamic programming search." *Pattern Recognition*, 2000.
- [20] Thevenaz, P; Ruttimann, UE; Unser, M. "A Pyramid Approach to Subpixel Registration Based on Intensity." *IEEE Transactions on Image Processing* (1998) Vol. 7, No. 1.

- [21] Koschan, A; Rodehorst, V; Spiller, K. "Color Stereo Vision Using Hierarchical Block Matching and Active Color Illumination." Pattern Recognition, 1996.
- [22] Trucco, E; Verri, A. "Introductory Techniques for 3-D Computer Vision." Prentice Hall, 1998.
- [23] Hartley, R; Zisserman, A. "Multiple View Geometry in Computer Vision." Cambridge University Press, 2003.
- [24] Bouguet, JY. "Camera Calibration Toolbox for Matlab." Computational Vision at the California Institute of Technology®. [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- [25] Logitech (2012), "Logitech HD Webcam C310" [On-line]. Available: <http://www.logitech.com/en-us/product/hd-webcam-c310?crd=34#section=specs> [Nov. 1, 2012].
- [26] Atmel. (2010, Oct. 20). "ATMEGA 16 datasheet." [On-line]. Pp. 1-356. Available: [www.atmel.com/Images/doc2466.pdf](http://www.atmel.com/Images/doc2466.pdf) [Sept. 1, 2012].
- [27] STMicroelectronics. (2000, January). "L298 Dual full-bridge driver". [On-line],pp.1-12. Available: [noel.feld.cvut.cz/hw/st/1918.pdf](http://noel.feld.cvut.cz/hw/st/1918.pdf). [September 2, 2012].
- [28] Fairchild Semiconductor. (2009, May). "1N4001-4007 General Purpose Rectifiers". [On-line]. pp. 1-3. Available: [www.fairchildsemi.com/ds/1N/1N4001.pdf](http://www.fairchildsemi.com/ds/1N/1N4001.pdf) [September 2, 2012].
- [29] Maxim. (2004, March). "MAX232 Datasheet." [On-line]. pp. 1-7. Available: [www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf](http://www.datasheetcatalog.org/datasheet/texasinstruments/max232.pdf) [September 4, 2012].
- [30] TowerPro "TowerPro MG995 Servo Specifications and Reviews" [On-line] Available: [www.servodatabase.com/servo/towerpro/mg995](http://www.servodatabase.com/servo/towerpro/mg995) [October 3, 2012].
- [31] TowerPro "TowerPro SG-5010 Servo Specifications and Reviews" [On-line] Available: [www.servodatabase.com/servo/towerpro/sg-5010](http://www.servodatabase.com/servo/towerpro/sg-5010) [October 3, 2012].
- [32] TowerPro "TowerPro SG91R Servo Specifications and Reviews" [On-line] Available: [www.servodatabase.com/servo/towerpro/sg91r](http://www.servodatabase.com/servo/towerpro/sg91r) [October 3, 2012].
- [33] National Semiconductor (2000, May 2). "7805 datasheet". [On-line]. pp. 1-3. Available: [pira.cz/pdf/78xx.pdf](http://pira.cz/pdf/78xx.pdf). [September 1, 2011].
- [34] Fairchild Semiconductor. (2008, October). "Tip127 datasheet". [On-line]. pp.1-4. available: [www.redrok.com/PNP\\_Darlington\\_TIP127\\_-100V\\_-5A\\_Hfe1000\\_TIP127.pdf](http://www.redrok.com/PNP_Darlington_TIP127_-100V_-5A_Hfe1000_TIP127.pdf) [September 2, 2012].
- [35] Vegarobokit.com (2012). "Ultrasonic Electronic Eye Telemeter Module". [On-line]. Available: [www.vegarobokit.com/index.php?route=product/product&product\\_id=410](http://www.vegarobokit.com/index.php?route=product/product&product_id=410) [September 10, 2012].
- [36] Texas Instruments (2005, October), "LM158/LM258/ LM358/ LM2904 Low Power Dual Operational Amplifiers". [On-line]. pp. 1-33. Available: [www.ti.com/lit/ds/symlink/lm158-n.pdf](http://www.ti.com/lit/ds/symlink/lm158-n.pdf) [September 12, 2012].
- [37] International IR Rectifier (2001, March), "IRF540NHEXFET® Power MOSFET" ". [On-line]. pp. 1-8. Available: [www.irf.com/product-info/datasheets/data/irf540n.pdf](http://www.irf.com/product-info/datasheets/data/irf540n.pdf). [September 12, 2012].

## Authors

Md. Shafayat Hossain was born in Dhaka, Bangladesh. He is currently pursuing his B.Sc. degree in electrical and electronic engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh. His research interests include analytical modeling & simulation of Nano-scale electron device, Robotics, computer vision and embedded system design.



Ahmedullah Aziz was born in Dhaka, Bangladesh. He is currently an undergraduate student of Bangladesh University of Engineering and Technology (BUET) in department of Electrical and Electronic Engineering. His current research interests include Robotics, Novel semiconductor based thin film characterization, embedded system design etc.



Mohammad Wahidur Rahman was born in Chandpur, Bangladesh, in 1991. He is currently an undergraduate student of Bangladesh University of Engineering and Technology (BUET) in department of Electrical and Electronic Engineering. His current research interests include Novel semiconductor based thin film characterization, Computer Vision technology, Video processing etc.

