2017 Asia Modelling Symposium

Implementation of Low Cost Stereo Humanoid Adaptive Vision for 3D positioning and Distance Measurement for Robotics Application with Self-calibration

Abul Al Arabi, Rayhan Sardar Tipu, Mohammad Raihanul Bashar,
Binoy Barman*, Shama Ali Monica†, Md Ashraful Amin
Computer Vision and Cybernetics Group, Department of Computer Science and Engineering
Independent University Bangladesh, Bashundhara R/A, Dhaka, Bangladesh
Mawlana Bhashani Science and Technology University*, Rangpur Medical College, Bangladesh†
email:(arabi,1330418,1320454)@iub.edu.bd (it.binoy,shamamonica)@gmail.com, aminmdashraful@iub.edu.bd

***Abstract*—Robots are getting smarter everyday with the implementation of computer vision system in it. It is now highly required for any robot to have a natural vision system or more likely humanoid vision system to interact with real life incidents. On the perspective of such imaging and vision, we propose an efficient method in order to determine the absolute view point of any desired image location. We used self calibration system and humanoid vision mechanism via stereo cameras to find the region of convergent of an object which with the help of a mathematical model can measure the distance of the object. With comparing different objects position it is also possible to determine the relative distance of the objects. Our system shows that, the real human eye tracking system used, can be possible for getting a realistic view of the image at the 3D point positioning.**

Keywords—Stereo Vision, Movable multi-cameras, distance measurement, robotic vision, image processing, humanoid vision, self calibration.

I. INTRODUCTION

In todays developed world, the stimulation of the digital manufacturing is a great concern topic. Specially for the three dimensional data, higher accuracy, precision and efficiency are a desired need now-a-days. Always its been a great challenge about getting the 3D object features not only accurately but also conveniently. In most of the cases, in order to perfect point position of any object, simple Dual view Stereo camera is used. But most times, its unable to give results according to the demand.

We already knew that, RGB-D cameras are able to capture RGB images with their sensing system attaching depth information over per pixel. These sensing system are one of the most attractive option not only for research but also for computer vision groups because for many years it has been used as a form of packaged factor.

In our work, we use a camera that developed by Logitech. It can able to capture 1280 x 720 pixels images and its captured 30 frames per second (fps).

In this work, we have proposed a system for 3D point positioning with dual Stereo Cameras but the working procedure of this task is based on the real Human Eye-tracking system. The camera is able to frequently side capturing. We used pan method for 3D vision using two cameras. First the object is placed at any position before the cameras, then one of the cameras finds the region of the detected object and sends signal to the other camera. And both of the cameras are movable via two separate servo motors at any direction where the object is placed. Then the cameras mutually find the convergent point of the object.

In the remainder of this paper, we first study the background of this works, then we describe the details about the approach including the mathematical approach. Then we showed our proposed model and to compute in the real time monitoring schema we experiment and simulate our task. Finally, we analyze the results to achieve the intended outcome. The paper ends with a general conclusion and future work.

II. BACKGROUND

Naturally, the point focused at the time of screen looking by human eyes is called the focus-point. Already the focus points working process of the human eye tracking system is being used in several domains. In [1], focus point is used by Sibert and Jacob as an alternation of mouse pointer of computer. This work based over the interaction of Human and computer. Focus point is also used at the field of neurology, as like, in [2] the movement of the focus point is analyzing for detecting the patients. And there proposed a result, of the stable and unstable movement for respectively healthy and diseased patients. Simplification of the sign in geometry can be possible with focus point, and this is shown in [3] with a relevant algorithm. For the visualization of volumetric data, a focus point based algorithm is used called ray-casting in [4]. To define the number of rays pixel distance of the focus point is used in this algorithm, which is able to adapted the level of volumetric data. On the other hand, there is a vast use of laparoscopic camera in the medical surgical treatment. But we can observed some difficulties at the time of positioning because of the unstable movement. The most used process of laparoscopic camera offered an automatic tracking system with a robotic holders. But still now the existing robotics holders are different from each other with DOF, joint types, link size and so on. But the laparoscope automated system is free from any robotic configuration [5-8]. Range scans [9, 10, 11, 12] and monocular camera [13] also used in many research to developed various computer vision communities.

2376-1172/17 $31.00 © 2017 IEEE
DOI 10.1109/AMS.2017.21

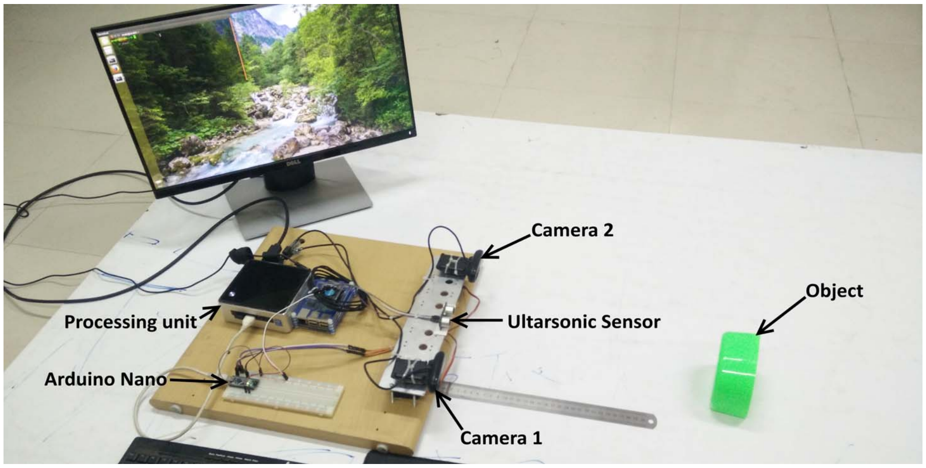

Fig. 1: Experiment Setup

Observing all these research task we make a relevant method over 3D point positioning. In our proposed system we used low cost cameras and single board low cost cpus. Our proposed system is efficient enough to run even in such a low cost and low powered cpus that paves the way of easy implementation in different robotic system without requiring high end processing power.

III. APPROACH

The main objective of our work is to find the common convergent point of any object that placed before the camera which is obtained by centering the object to the center of each frame.

The position of the object is arbitrary. The cameras will automatically find the object, because the cameras are placed on two separate servo motor that can move the cameras as it needs to locate the object. For the whole system to be mounted properly, we used a CNC machined aluminium frame. The frame has several spaces for placing the cameras. Our experiment setup is given as per figure 1. The angle of the cameras can be changed via a servo motor in which the cameras are placed. For the accurate measurement of the angles of the camera we used a meter scale attached to the frame. For finding the convergent point of any object we used the human - eye system. In this system the object is placed in any range in front of the camera. One of the camera first locate the object then send signal to the other camera. Then both of the cameras move according to the object location and find the common region of convergent of the object.

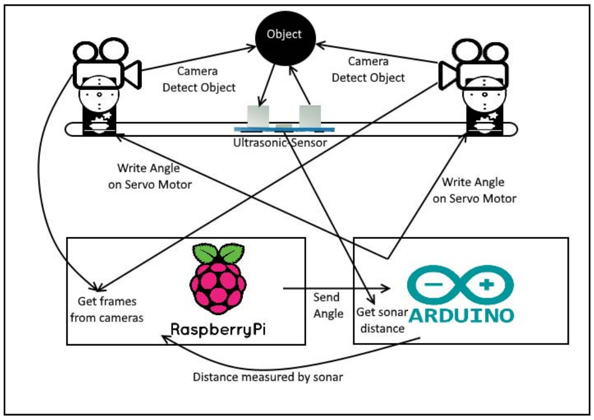

Fig. 2: System Overview

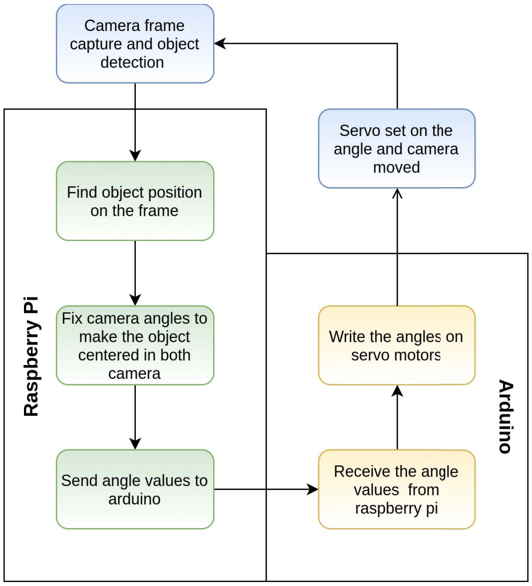

Fig. 3: System Flow Chart

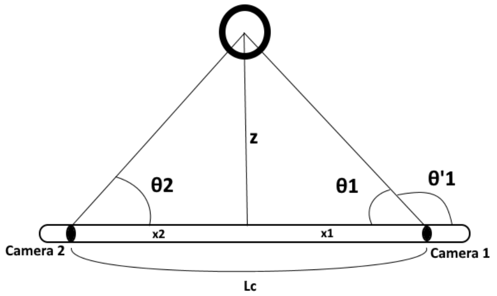

Fig. 4: Object position from stereo vision

A. Convergent Point Finding

First of all we considered some variables for finding the convergent point of the object. The distance between the two cameras was denoted by L_c and the distance between the object and camera frame is denoted by z (Figure 4) . The region of convergent depends on another option, we named it shadow effect. We considered the angel of the shadow zone of the cameras are θ_m, then the region of convergent without the shadow effect will be $180 - 2\theta_m$.

The system captures two frames from two different cameras. The CPU (raspberry pi and intel NUC in our case) then finds the object position and tried to converge those two object position found from two different frames by adjusting the angle of the servo motors which are driven by an arduino controller. Having centring the object in both frame the CPU then finds the distance of the object. The system data handling can be viewed from figure 2. A very brief and summarized system flow chart can be found from figure 3.

B. Mathematical Approach

We derived a generalized mathematical model for our algorithm. Let the distance between the two camera be L_c. The object distance be z and the angle of the two servo motors be $\theta'_1,\ \theta_2$ (right servo angle and left servo angle), figure 4.

Now if x_1 and x_2 be the horizontal distance of the object from right and left camera respectively then,

$$\begin{aligned} x_1 &= \frac{z}{tan\theta_1} \\ x_2 &= \frac{z}{tan\theta_2} \end{aligned} \tag{1}$$

Which can be expressed as per equation 2.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} tan\theta_1^{-1} \\ tan\theta_2^{-1} \end{bmatrix} * z \tag{2}$$

Hence,

$$z = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} tan\theta_1^{-1} \\ tan\theta_2^{-1} \end{bmatrix}^{-1} \tag{3}$$

Thus z can be found as,

$$z = (x_1 + x_2) * (tan\theta_1^{-1} + tan\theta_2^{-1})^{-1} \tag{4}$$

Since $x_1 + x_2 = L_c$ hence,

$$z = L_c * (tan\theta_1^{-1} + tan\theta_2^{-1})^{-1} \tag{5}$$

Here θ_1 is essentially $180 - \theta'_1$.

C. Self Calibration

The distance between the cameras is not a fixed one. Our system is adaptive and capable of modelling its own camera distance from the calibration procedure.

The ultrasonic sensor mounted on the frame (figure 2, 1) is capable of finding distance of an object that is positioned exactly on the middle line in between two cameras. The system on the very beginning senses the presence of object in front of the ultrasonic sensor. The distance reading from the ultrasonic sensor is then captured by the arduino. If the CPU needs that data then it fetches that data. With that in

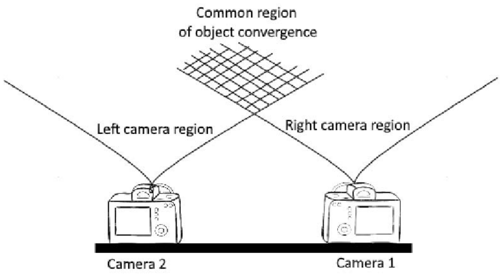

Fig. 5: Common region with static camera settings

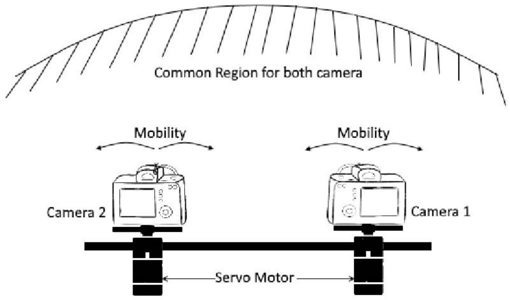

Fig. 6: Common region with movable camera settings

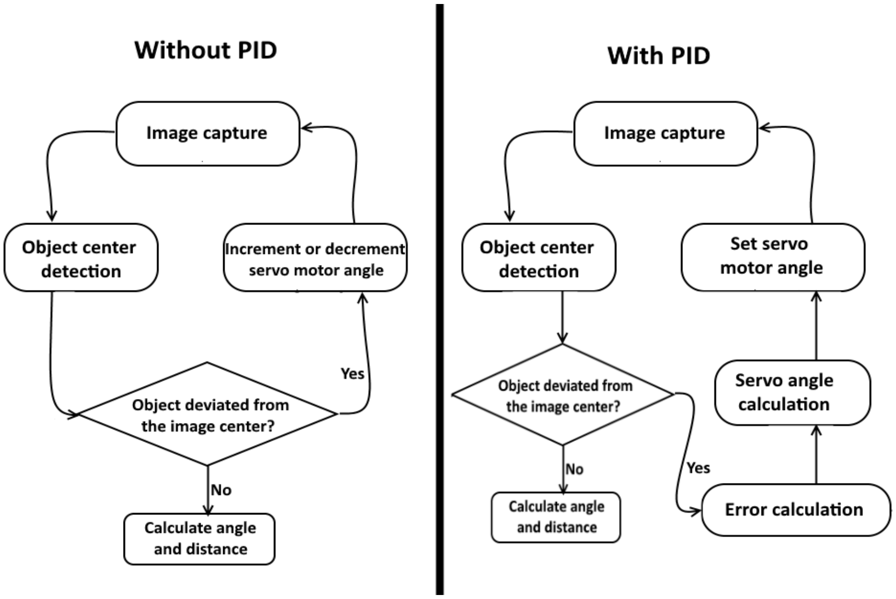

Fig. 7: Flow chart for angle fixing with and without PID system

hand in this scenario the values of $\theta_1,\ \theta_2,\ z$ are known. Thus in the calibration stage the value of L_c can be found from equation 6.

$$L_c = z * (tan\theta_1^{-1} + tan\theta_2^{-1}) \tag{6}$$

The system thus calibrates itself with different values of z and finds a reliable value for L_c that is used to compute further z values.

D. *Camera Angle Fixing by PID*

Our system uses a PID algorithm for controlling the servo hence camera angles. It uses the deviation of the object center from the absolute center of the image. It acts as the error value of the PID system. It can be modelled as per equation 7.

$$\begin{aligned} errorvalue &= objectcenter - imagecenter \\ p\ value &= kp\ *\ error \\ i\ value &= ki\ *\ \sum error \\ d\ value &= kd\ *\ error - error_{old} \\ angle_{new} &= angle_{old} - (p + i + d) \end{aligned} \tag{7}$$

Here $kp,\ kd,\ ki$ are constants which has been calibrated by iteration method.

IV. SIMULATION AND EXPERIMENT

For the implementation work we used Linux operating system platform. On this platform OpenCV was used along with python programming language. The cameras were connected in different USB bus as a single bus most of the case fails to handle data from multiple cameras. We have tested the whole system in a real life environment with fluorescence bulbs. As we know fluorescence bulbs causes banding or flickering in the cameras, hence it creates a challenge towards implementing the algorithm. Our system detects the object by background suspension method and also a filter avoids such flickering. As color space we used HSV system for its being efficient over RGB. The relevant Simulation and Implementation Tools are:

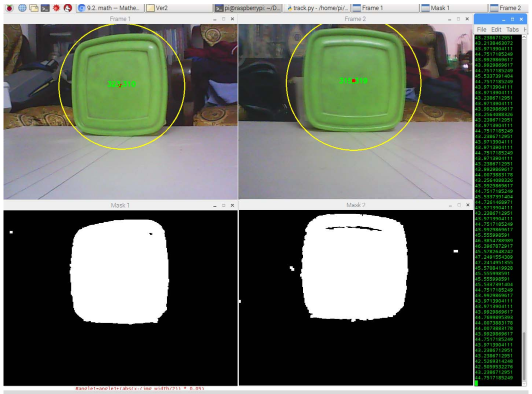

Fig. 8: Frames Capturing & Distance Measurement

A. *Camera*

Two webcams are used in this work. These cameras are made by Logitech and the resolution used from the cameras are 1280 x 720 pixels.

B. Camera Calibration

The camera calibration was performed via a close loop feedback system. Essentially, when the servo angle is vertical (90°) and the object is at perfect centre of the frame, then it represents a perfectly calibrated camera system. This procedure is repeated for both the camera.

C. Camera, Servo and Ultrasonic Sensor Platform

CNC machined parts were used to make the frame for the placement of the cameras. Two separate servo motors were used to place and move the cameras. There are different points on the frame so that the cameras can be placed at different distances.

D. SOLIDWORKS

Solidwork software is used for designing the CNC machined components. This is a solid modeling computer-aided design (CAD) and also a computer-aided engineering (CAE) software program. The whole mechanical system along with the servo motor has been simulated in this software and machining has been carried out with high precision to compensate error occurring from mechanical system.

E. Python IDE

Python is an open source language for managing hardware based programs. The Python Package Index(PyPI) hosts thousands of third-party modules for Python. Both Pythons standard library and the community-contributed modules allow for endless possibilities. This language is used in many fields such as Web and Internet Development, Database Access, Desktop GUIs, Scientific & Numeric, Education, Network Programming, Software & Game Development. In this system python is used for the GUI representation of the process.

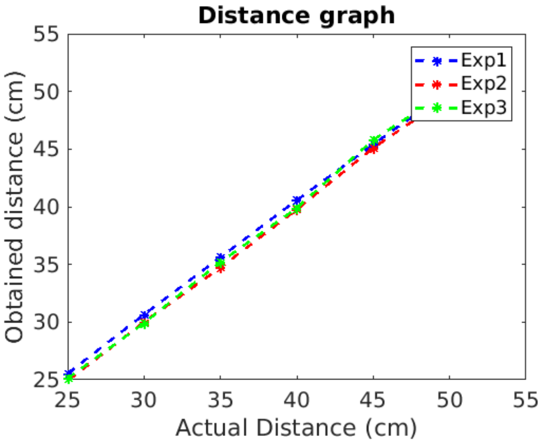

Fig. 9: Measured Distance vs Actual Distance

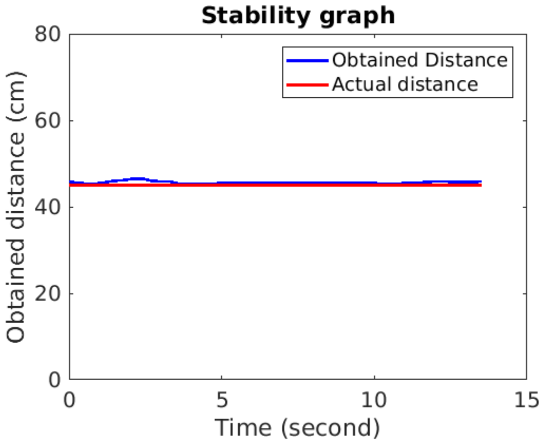

Fig. 10: Measured Distance Data stability

V. RESULT AND ANALYSIS

The system has been tested under intentional noisy environment figure 1. The system maximum frame rate is 30fps. With a delay of 50 millisecond limits the frame rate to 20fps (for testing case). It has been calibrated for a green object for the testing purpose. The image capturing involves a RGB to HSV color space for better immunity to noise and filtering. The contour detection uses the input probable size for noise removal.

The system showed an excellent performance while capturing and centring the object (figure 8). The two camera converges the object to the center of each frame and the mathematical model was able to measure the distance of the object. The small deviation in the measurement of the centroid of the object can lead towards errors. Hence, our system uses an moving averaging method for stability of the detected position and angel measurement. It minimizes the ripple of the detected centroid co-ordinates figure 10. The servo angels were set to a minimum step size of 0.5

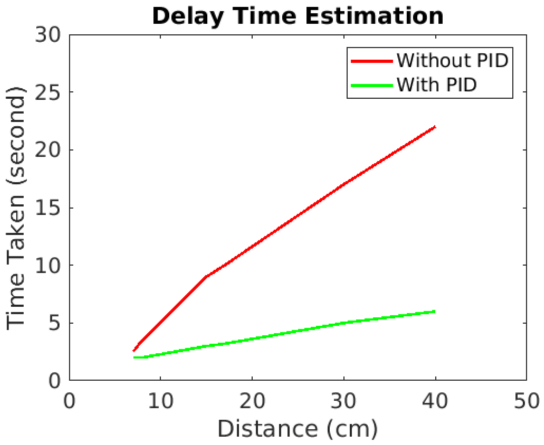

Fig. 11: Delay time to track object w/o PID system

degree. With a 50 pulses per second PWM, the vertical position (90°) pulse width was found to be 7.5 millisecond. Since cameras and servos was a low cost hence it had some noises that induced some ripples in the measured distance although those are negligible to some extent.

The result shows a very close approximation of the object distance, figure 9. For different positions the actual and system measured values are obtained. Although initially it was found to have some offset, it was diminished by further calibrations.

The object maximum distance is a subject to the minimum contour size. With lower size the maximum region of convergence increases but at the same time immunity to noise decreases leading the system to instability. With decrease in distance the contour size increases and hence limited to the maximum frame size of the camera. Again with angle values closing towards 90°, the tangent value closes towards much higher values. In such case with even a very small error the system showed instability and failed to determine the distance of the object, although it can approximate only. The distance and angle are non linear in relation. Thus with larger distance, the angle measurement being needed to be much more precise, the accuracy decreases. Hence it again limits the maximum measurable distance. Singular points (where only one or no camera can detect the object) are another main limitation of our work. When the object is placed in a position that no camera is able to find out the position of the object then this kind of situation happens. This is the position where no cameras can be set to focus on the object.

The servo angle measurement has been carried out both using PID and without PID. The flow chart for these two method is essentially different, figure 7. Without PID the system response is very low and there is a huge delay between moving the object and tracking it by adjusting servo angles. The contrast in delay time for these two system can be viewed from our experimental values, figure 11. The object was moved to different positions and the time taken to follow the object was measured. Although both system was closed loop, it is obvious that PID is much more superior one.

VI. CONCLUSION AND FUTURE WORKS

Our work incorporates very low cost devices but efficiently implements them to create a humanoid vision based distance measurement system that can effectively and easily be installed with existing robotics application. Also it is fast and devices used does not requires to be of high end which proves the robustness of the system. In future we plan to use zoom cameras to cover longer ranges. Also we aim at incorporating full 360°region with stepper motors for better angle and hence better distance measurement. Along with that a path planning algorithm is under development that will be fused with our current work for robotic path planning.

ACKNOWLEDGMENT

This work is fully supported by a grant from the Independent University Bangladesh.

REFERENCES

[1] Sibert, L. E., and Robert JK J. ”Evaluation of eye gaze interaction.” Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM, 2000.

[2] Iijima, A., Haida, M., Ishikawa, N., Minamitani, H., Shinohara, Y. ”Head mounted goggle system with liquid crystal display for evaluation of eye tracking functions on neurological disease patients.” Engineering in Medicine and Biology Society, 2003.Proceedings of the 25th Annual International Conference of the IEEE. Vol. 4. IEEE, 2003.

[3] Luebke, D., and Benjamin H. ”Perceptually driven simplification for interactive rendering.” Rendering Techniques 2001. Springer Vienna, 2001. 223-234.

[4] Levoy, M., and Ross W. ”Gaze-directed volume rendering.” ACM SIGGRAPH Computer Graphics 24.2 (1990): 217-223.

[5] Fahimi, F. Autonomous robots: modeling, path planning, and control. Vol. 107. Springer Science & Business Media, 2008.

[6] Zenkevich, S., and Yushchenko, A. ”Fundamentals of Control Robotic Manipulator.” MSTU Bauman, Moscow (2004).

[7] Sinyavskaya, E., Shestova, E., Medvedev, M., Kosenko, E. ”Positioning Method Basing on External Reference Points for Surgical Robots.” International Conference on Interactive Collaborative Robotics. Springer International Publishing, 2016.

[8] Pshikhopov, V.K., Medvedev, M.Y., Kostjukov, V.A., Gaiduk, A.R., Fedorenko, R.V., Gurenko, B.V., Kruhmalev, V. A., Medvedeva, T.N.: Design of robots and robotic system. A tutorial. SFU, Taganrog (2014).

[9] Sebastian, T., Burgard, W., and Fox, D. ”A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping.” Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on. Vol. 1. IEEE, 2000.

[10] Rudolph, T., and Burgard, W. ”Improving simultaneous mapping and localization in 3d using global constraints.” Proceedings of the national conference on artificial intelligence. Vol. 20. No. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

[11] May, S., Droeschel, D., Holz, D., Fuchs, S., Malis, E., Nchter, A., Hertzberg, J. ”Threedimensional mapping with timeofflight cameras.” Journal of Field Robotics 26.1112 (2009): 934-965.

[12] Robbie, S., and Newman, P. ”Discovering and mapping complete surfaces with stereo.” Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010.

[13] Clemente, L. A., Davison, A. J., Reid, I. D., Neira, J., Tardos, J. D. ”Mapping Large Loops with a Single Hand- Held Camera.” Robotics: Science and Systems. Vol. 2. 2007.