Visual Servoing of a Sensor Arm for Mine Detection Robot Marwa[1]

Talha Manzoor, Adnan Munawar, Abubakr Muhammad
Department of Electrical Engineering
LUMS School of Science and Engineering, Lahore, Pakistan
{talha.manzoor, adnan.munawar, abubakr}@lums.edu.pk

Abstract

This paper presents a visual servoing system developed at LUMS for the landmine detecting robot named *Marwa*. The system is coupled with a robotic arm whose payload is a metal detector. Visual information is provided by a custom built stereo vision system to maintain a fixed orientation and elevation of the metal detecting sensor with respect to the ground directly in front of it. This is essential for successful working of the sensor. First, the problem is defined and the motivation for the development of the robot is discussed. Then, the mechanical design of the arm is explained, followed by an explanation of the strategy we employ for total coverage of a mine infected area. The terrain representation and its reconstruction using stereo vision is described. The integration of the vision component with the arm joint feedback is then presented, along with an overview of the real time controller and programming environment.

1 Introduction

The detection and removal of buried landmines is a global problem being faced by many countries. Due to the geographic extent of the problem as well as the danger associated with deploying a human into a mine-field, it makes sense to think about an automated, possibly robotic solution to the landmine clearing problem. A completely robotic solution still has a long way to go from current state-of-the-art in robotics technology [1]. However, significant levels of automation can still be achieved if subtasks such as detection of mines with occasional intervention of a human operator is made possible.

Following this philosophy, work has been done by several researchers on several aspects of the problem such as use of multi-agent architecture and search strategies for demining by using multiple robots [2, 3]; achieving efficient terrain traversability and overall mobility of the robot [4, 5]; and a high focus on terrain profiling and efficient manipulation for adjusting the sensor orientation [6]. In this paper, we describe a visual servoing technique to maintain the required sensor orientation and elevation with respect to the ground, as an important aspect of the complete mine-detection and clearing problem.

Our motivation for developing a landmine detecting robot came from an international robotics competition by National Instruments (NI). In 2009, NI Arabia in collaboration with LebMAC (Lebanese Mine Action Center) invited proposals for the NI Mine Detection Robot Design (NI MDRD) contest in Lebanon [7]. LUMS design named as *Marwa* went on to win the competition in 2011 with an award in the *Vision & Sweeping Algorithms* category [8]. A critical component of the system was its sensor arm for landmine detection. Following the taxonomy used in standard literature, we have implemented a *binocular, stand-alone, position-based, dynamic look-and-move* visual servoing system. A block diagram for the control of the wrist angle can be seen in Figure 2.

Figure 1: Sensor arm in lab (Left). Arm, controllers and stereo vision setup mounted on Marwa (Right).

From literature reported at the time of writing of this paper, our system can be best compared with the Gryphon landmine detecting robot [6], which places a high focus on terrain profiling and efficient manipulation for adjusting the sensor orientation using visual processing. Gryphon uses a visual servoing system with a stereo vision subsystem coupled with a pantographic manipulator. Our system differs from that of Gryphon in the design of the manipulator and methodology of scanning the area. Our design focuses on simplicity, cost effectiveness and ease of deployment & maintenance in a developing world environment. Mechanical design complexity has been deliberately kept to the bare minimum and a high emphasis has been placed on computational resources and control algorithms. In this paper, we hope to present a simple and easily replicable visual servoing system to benefit researchers with limited

[1]This work has been done at the Laboratory for Cyber Physical Networks and Systems (CYPHYNETS) at LUMS. It was made possible by a LUMS Faculty Initiative Fund (FIF) grant, a German Academic Exchange (DAAD) grant (ALVeDA) and equipment donated by National Instruments, Arabia Division.

resources to work in this area.
The paper is organized as follows. Section 2 explains the mechanical design of the arm. Then Section 3 describes the coverage strategy for scanning of large minefields. Section 4 discusses the internal representation of the terrain profile, followed by an explanation of the stereo vision system in Section 5. Section 6 introduces the hardware and the programming environment along with a breakdown of the real-time tasks. We conclude in Section 7.

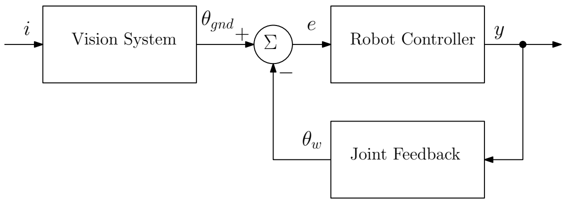

Figure 2: Block diagram for visual servoing of wrist pitch angle. Here i represents the stereo image pair, θ_{gnd} is the angle of the ground normal and θ_w is the wrist pitch angle. Similar controllers have been implemented for the rest of the arm joints.

2 Mechanical Design Overview

2.1 Construction

The metal detecting sensor has been mounted on the end effector of the robotic arm and the main purpose of the arm is to keep its end effector aligned with respect to the successive ground plains. We choose a non-conventional kinematic solution to the problem by designing a PRPRR robotic manipulator (P: Prismatic, R: Revolute). This mechanism is able to keep the end effector carrying the metal detecting sensors aligned with ground profile at all times as well as protecting the manipulator from steep or suddenly encountered obstacles in highly uneven terrains.
The manipulator is also capable of scanning on surfaces comparatively higher than the robot height due to its design. This saves the robot from the need to ascend unnecessarily for scanning.
The base of the arm is a prismatic joint. The metal detecting sensor is much smaller in size as compared to Marwa's front face thus this degree of freedom comes in handy for scanning by panning the front of the robotic vehicle, along its face. Thus, together with robot motion, the joint helps create an XY-raster movement for the mechanical arm, thus ensuring that no area is left undetected following the passage of the vehicle. The scanning motion of the robot is shown in Figure 3.
The second joint is revolute and is designed to align the arm according to the required angle for surface tracking as well as aligning itself to raised surfaces that can be better scanned without the need of the robot itself to climb upwards (See Figure 3).

The third joint is prismatic which is essential in the design of the manipulator. This prismatic joint is constructed using a screw rod. The screw rod moves with the manipulators arm length and thus is used to retract or extend the robotic arm in order to reach for the ground length. This is shown in Figure 3.

Figure 3: Demonstration of the working of the arm joints. In order from top to bottom: **1.** The panning joint. **2.** The elbow revolute joint. **3.** The retracting joint. **4.** The wrist pitch joint. **5.** The wrist roll joint

The fourth joint is for tilting the sensor upwards or downwards as shown in Figure 3. This freedom is very important for ground tracking. For the design process, a speed breaker like ground profile of acceptable minimum curvature was considered for tracking. This joint, in combination with the previously discussed prismatic joint are necessary to keep the sensors aligned with a speed breaker like ground profile.
The last joint is important for tracking surfaces or obstacles that are slanted by a roll angle from perspective of the robotic vehicle. This freedom is also shown in Figure 3.

2.2 Simplicity of Control

The greatest advantage in the given robotic manipulator is its simplicity of control. The joints are designed such that they are independently controlled and capable of remaining dormant without affecting the motion of the remaining joints till they are needed. For relatively easy terrains the manipulator can track the ground using just the third (prismatic) joint and the fourth (revolute) joint. The third prismatic joint can be slanted at an angle of $60°$ for normal terrain ground tracking. Since this joint is slanted and protrudes outwards from the robotic vehicle, it has an advantage while negotiating obstacles. It can either be retracted using the third joint or it can be elevated using the second joint or moved by a combination of both.

2.3 Sensor Payload

The critical joint in our system is the elbow revolute joint. This joint then determines the sensor payload which is given as

$$W_{\text{Payload}} = \left(\eta \frac{60P}{2\pi r \omega g}\right) - W_{\text{Arm}}$$

where η is the Gearing Efficiency, P is the Mechanical Power of the motor in Watts, r is the maximum moment arm, ω is the required motor RPM and g is the gravitational constant. For our mechanical setup ($P = 45, \omega = 5$, $\eta = 0.7$) we find that our sensor payload is 7 kg.

3 Coverage and Scanning Strategy

3.1 Area Covered in a Single Sweep

While standing stationary, the arm's DOF's enable the robot to scan a certain rectangular area immediately in front of it. This means that it is not required to move the robot forward for every small horizontal patch it has to scan, which would reduce the overall speed of operation. Instead, this movement can be provided by the arm, scanning every horizontal patch through a sideways movement of the sensor, then moving the end-effector forward and repeating the sideways movement. This continues until the limits of the arm joints are reached, at which point the arm is retracted and Marwa moves forward to scan the next rectangular patch. Scanning motion along the sideways direction is achieved through the panning joint, while scanning motion in the forward direction is achieved by a combined movement of the elbow revolute and prismatic joints (Figure 3). The sense of both directions can be seen in Figure 4. Another possible manipulator design such as that of Gryphon will enable the robot to scan semi-circular regions on each of it's sides.

3.2 Scanning Larger Areas

To start scanning a real-world mine infected area, it is necessary that Marwa enters the field in a head-on direction. Until any mines are detected, it would keep moving straight forward, stopping at constant intervals to scan the rectangular patch immediately in front of it. Marwa continues in this fashion until the opposite boundary of the area is reached, at which point (if no mines have been detected) it can declare a vertical slice of the area safe. It then turns 180° to cover the adjacent vertical slice, and continues this behavior until the whole area has been scanned. This can be visualized in Figure 4. Although we have discussed a scanning strategy for a rectangular area, a similar procedure can be adopted for circular areas by starting at the circumference and moving in towards the center. Also note that while Marwa's manipulator is attached to its front, it would also be possible to attach it to the side, in which case the robot would not enter the minefield immediately but would start scanning in a sideways manner. Gryphon is an example of such a system.

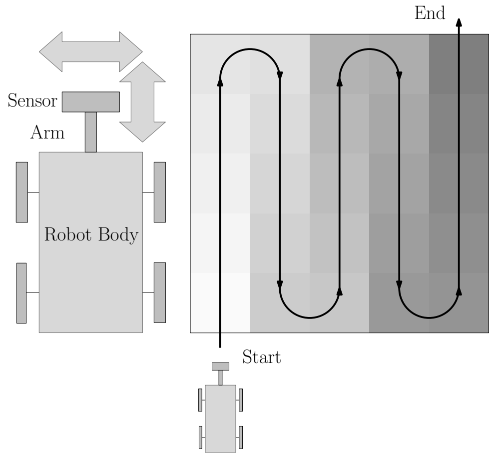

Figure 4: left: Possible scanning directions while Marwa is stationary. The sensor may be moved in forward or sideways direction. **right:** Coverage strategy of minefield adopted for Marwa. The path followed is indicated by the directed lines. A region is represented by a singly shaded area. Marwa scans a single region while remaining stationary and then moves to the next.

3.3 Sensor Orientation

Consideration of sensor orientation while scanning is extremely essential for landmine detection. For any sensor such as a metal detector or ground penetrating radar, we can only be confident of the authenticity of the scanning results if the following two conditions are met. Firstly, the sensor should remain as close to the ground as possible, and secondly, it should always be pointing directly towards the scanning area. These requirements can be visualized as shown in Figure 5. It can be seen that at each point, the sensor must maintain a constant distance from the ground and remain tangent to the patch directly beneath it.

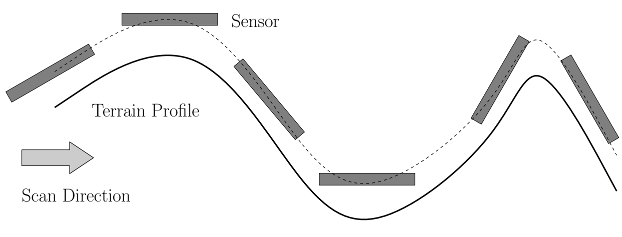

Figure 5: Cross sectional view of a single horizontal sweep of the metal detector.

4 Terrain Representation

4.1 Piecewise Planar Mapping

To cater for the sensor pose requirements discussed in Section 3.3, we employ a piecewise planar representation of the underlying terrain. The resolution of the representation dictates the accuracy with which the terrain is represented and so with high enough resolution, we are able to represent the extremely rugged terrains encountered in real-world minefields. Such an example can be seen in Figure 6. Currently, the resolution is initialized manually at system startup, but a clustering technique can be applied on the point cloud to determine the required resolution at run-time. This can be an interesting avenue for future work.

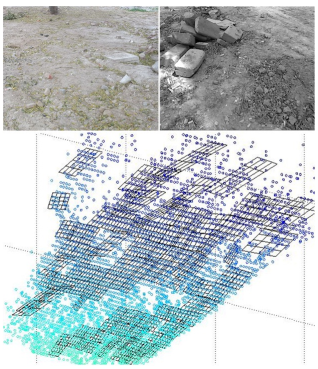

Figure 6: top: Example of an outdoor field used for Marwa's experiments and a patch of uneven ground encountered during one of the scans. **bottom:** The corresponding point cloud. The point cloud has been overlaid by its piecewise planar representation,and rendered at a different angle for visual clarity.

4.2 Calculating References for the Arm Controller

Each plane in our representation has a corresponding normal vector, and the normal of the metal detector is aligned with this vector, to mantain the desired orientation while it passes over that plane. Moreover, the point with the least depth in that patch, is used as a reference to mantain the desired elevation of the sensor. This information is used to calculate the desired end effector pose as an input to the arm controller. Once the terrain is represented in this manner, we may also use knowledge of the vehicle and arm kinematics, along with an interpolation of the normal vector angles for each plane, to generate whole trajectories for the each of the wrist joints.

4.3 Encountering Holes in the Point Cloud

The point clouds we obtain are generated through stereo reconstruction which often yields incomplete data. Currently, missing patches are covered up with interpolating planes, which is a risk in context of the sensitive application of humanitarian demining. However, our terrain representation can only be as good as our stereo correspondence algorithm which is an inherent bottleneck in any stereo vision technique. So until an economical and accurate depth sensor (such as the increasingly popular RGB-D cameras) is developed for outdoor applications, our only alternatives are sensors which are much too expensive for commercial deployment in a developing world country.

5 Ground Profiling Through Stereo Vision

5.1 The Stereo Vision Setup

We have implemented our vision system with two Logitech c500 webcams attached to a custom built stereo rig (Figure 7). The stereo system is mounted on the robot body with a downward looking stance. It can be seen that the setup is extremely simple and easily replicable. We have used the open-source OpenCV library to program our vision apparatus. Moreover, all vision algorithms we use, are well documented and easily accessible [9].

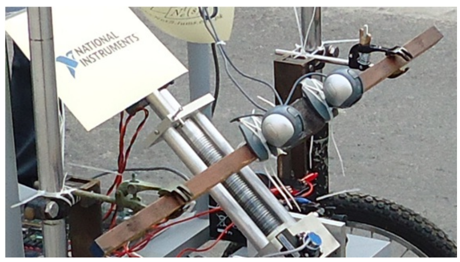

Figure 7: The stereo system mounted on Marwa.

5.2 Procedure for 3D Reconstruction

To generate the terrain map discussed in Section 4 we go through the following steps.

5.2.1 Calibration

Individual and stereo camera calibration is only required once for a fixed stereo rig. As a calibration object, we used a regular chessboard pattern. A few snapshots from a calibration procedure can be seen in Figure 8

Figure 8: A few images from the calibration process.

5.2.2 ROI and Image Rectification

After startup, we specify our Region Of Interest from the live feed of the cameras, which is directed to the ground immediately in front of the sensor.
In order to calculate a disparity map, the images must be rectified in order to make them row-aligned. For this we used Bouget's calibrated method, which is faster than other available methods [9] and made convenient by the fact that we perform stereo calibration beforehand.

5.2.3 Finding Correspondences

The quality of the correspondence algorithm is a major bottleneck in any multi view reconstruction problem. For matching corresponding image points, we have used a simple correlation based Block Matching technique, where the correlation function is a simple Sum of Squared Differences (SSD) window. Despite the low robustness of Block Matching, its real time performance makes it a more suitable candidate for robotics applications than other methods like the Graph Cuts or Semi-Global Block Matching algorithms.

5.2.4 Disparity Map and 3D Reconstruction

After finding correspondences, we construct the disparity map, which is then used to reconstruct the scene in 3D. Some results can be seen in Figure 9

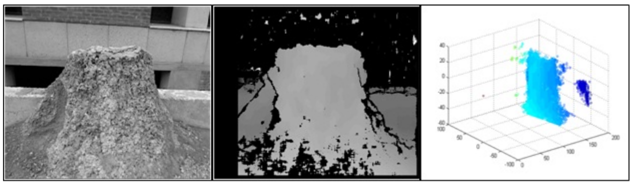

Figure 9: Snapshots of a scene, its disparity map and 3D point cloud as generated by our stereo system.

5.2.5 Plane Fitting Via PCA

We use simple PCA (Principle Component Analysis) to fit a plane through the point cloud and compute the normal vector. A sample of the results from this process can be seen in Figure 10. Once we have this vector, we can project it onto the plane of the desired wirst joint to generate a reference for it's controller.

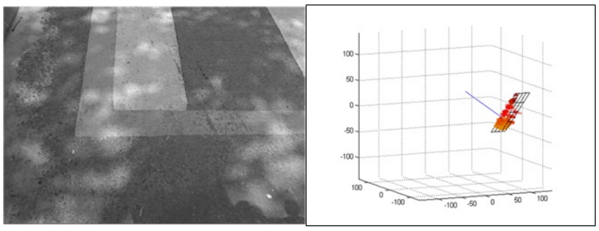

Figure 10: Sample image and corresponding point cloud with best fit plane and surface normal found by PCA

5.2.6 Arm Retraction Computation

Next, we find the distance of the point nearest to the camera in the point cloud. We then subtract the required elevation of the sensor from the ground from this distance. This gives us the reference for the arm retraction joint.
We repeat all mentioned steps in a loop so that the reference variables for the controller are continuously updated. However, calibration and specification of the ROI are a one-time process.

5.3 The Speed Breaker Experiment

Here we present the results of a very simple outdoor experiment, a speed breaker on a road. The scene is shown in Figure 11. Such terrain is unlikely to be encountered in real demining scenarios. However, it provides us a nice insight on whether or not the internal representation is capable of following changes in the underlying surface as Marwa moves forward. Also note that the terrain under consideration is smooth enough to be approximated by just a single plane. The evolution of the surface normal as Marwa moves over the bump can be seen in Figure 12.

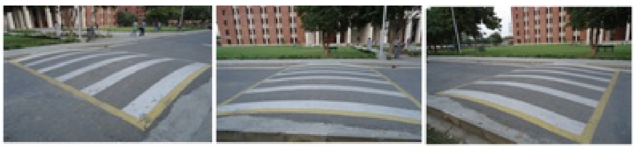

Figure 11: The speed breaker scene used for one of Marwa's experiments.

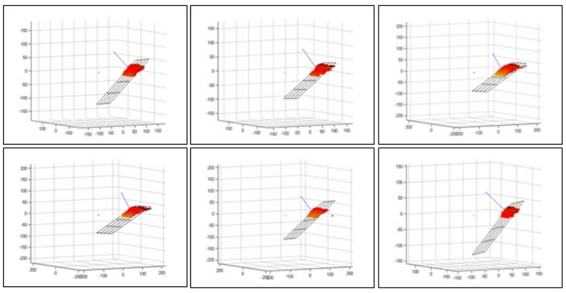

Figure 12: Time change in computed surface normal of the speed breaker as viewed by Marwa.

6 Real Time Implementation and Hardware

Our system uses National Instruments' (NI) Single Board RIO (SbRIO) and relevant modules for low level hardware control. The board includes a real time processor and FPGA, programmable in LabVIEW's Real Time Module. Extensive documentation both for hardware and software issues is available on the web.
The overall flow of data for the system can be seen in Figure 13.

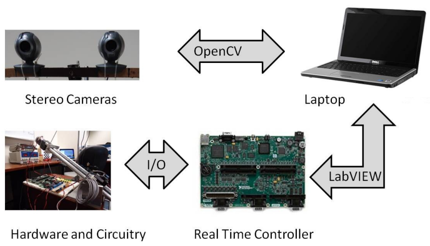

Figure 13: Overall data flow of the system.

At runtime, the main control loop runs in LabVIEW on the laptop. Communication with the cameras is carried out by executing the dll files containing wrapper functions of our OpenCV code. The main control loop contains two real time tasks
1. Vision: Compute the surface depth and normal vector to update the controller references.
2. Control: Use the reference values to set the input for the joint actuators.
Communication between the controller and PC has been carried out using LabVIEW's shared variable feature.

7 Conclusion

There is still some time before robots completely replace humans in humanitarian demining. Nevertheless, significant progress can be made in solving individual tasks such as automated mine-detection. In this paper, we have presented the development of a visual servoing system for a landmine detection robot. Although no change has been made to the state of the art, we hope that the development procedure described over here may be of use to researchers who are new in this field and aspire to solve the problem at a global level.

8 Acknowledgements

The authors acknowledge their colleagues Bilal Talat, Sufian Azhari and Tahir Javed for help with conducting experiments and setting up electronics. Also acknowledged is the mechanical arm's machinist, Mr. Zahid from Hamza Engineering, Lahore.

References

[1] J. Trevelyan. Robots and landmines. *Industrial Robot: An International Journal*, 24(2):114–125, 1997.

[2] P. Santana, J. Barata, H. Cruz, A. Mestre, J. Lisboa, and L. Flores. A multi-robot system for landmine detection. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, volume 1, pages 8–pp. IEEE, 2005.

[3] R. Cassinis, G. Bianco, A. Cavagnini, and P. Ransenigo. Strategies for navigation of robot swarms to be used in landmines detection. In *Advanced Mobile Robots, 1999.(Eurobot'99) 1999 Third European Workshop on*, pages 211–218. IEEE, 1999.

[4] P. González de Santos, J.A. Cobano, E. Garcia, J. Estremera, and MA Armada. A six-legged robot-based system for humanitarian demining missions. *Mechatronics*, 17(8):417–430, 2007.

[5] E. Cepolina and M. Zoppi. Cost-effective robots for mine detection in thick vegetation. In *Proceedings of the 6th International Conference on Climbing and Walking Robots, Catania, Italy*, 2003.

[6] E. Fukushima, P. Debenest, M. Freese, K. Takita, Y. Oishi, and S. Hirose. Development of teleoperated landmine detection buggy gryphon for practical humanitarian demining tasks. In *Experimental robotics: the 10th International Symposium on Experimental Robotics*, page 225. Springer Verlag, 2008.

[7] National Instruments Arabia Division. Ni mine detection robot design contest. `http://digital.ni.com/worldwide/arabia.nsf/web/all/7D2F5DA6CFCD0A8D8625766C0036B351`, March 2010.

[8] Muhammad A. et al. Marwa: A rough terrain landmine detection robot for low budgets. In *Proceedings of the 43rd International Symposium on Robotics, Taipei, Taiwan*, 2012.

[9] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.