E90 PROJECT PROPOSAL: STEREO VISUAL ODOMETRY

By: Yavor Georgiev
Advisor: Bruce Maxwell
Date: Tuesday, November 15, 2005

I. TABLE OF CONTENTS

II. ABSTRACT

I propose the development of a stereo-based visual odometry system for mobile robot localization. This project would integrate with the existing robot control system developed at the department, and would add functionality for accurate mapping and robust localization. The video input to the system is provided by a Videre STH-DCSG/-C stereo head, and the target platform is the Pentium III-based RWI Magellan Pro. No additional equipment will be necessary aside from what is already available at the department, so the project will be completed at a negligible cost.

III. INTRODUCTION

Accurate localization and mapping is a problem of great significance in the mobile robotics community, and especially in the context of Urban Search and Rescue (USR) applications, where a teleoperated (or sometimes autonomous) agent is dispatched inside an unknown hazardous environment to collect topological information on a time-critical task. Almost always the environments surveyed are unstructured and previously unknown, so creating accurate maps is essential should a human rescue effort be necessary.

There is a number of notable localization methods being used today in the robotics community, and visual odometry is the only approach that fit our performance needs and cost constraints. We considered sonar-based methods, but the results[1] obtained by other researchers were not as general as we needed. Laser range-finder solutions are definitely very well-suited to our task, but the cost of the equipment necessary was prohibitive. Visual odometry, as presented in the work of David Nistér emerged as a good tradeoff between reliability, cost, and implementation complexity. The necessary stereo equipment has already been purchased by Prof. Maxwell using grant funds, and this implicit expense has been excluded from the scope of this project.

This proposal presents a brief technical description of the visual odometry process as outlined in this work[2], as well as a section on sustainability and design constraints. Then it highlights major implementation steps in the context of this project, considering the equipment available, and previous work done at the department that can be reused. An implementation timeline and critical path analysis follow, and a section on the author's qualifications.

IV. TECHNICAL DISCUSSION

4.1 OVERVIEW

The stereo head selected for this project is Videre Design's STH-DCSG/-C. It consists of two color 1/3" format CMOS imagers, with a maximum resolution of 640x480 pixels, sufficient for the needs of this project. The stereo baseline of the rig is 9cm.[3]

Visual odometry is the process of estimating the location and trajectory of a moving object in real time, based on a video feed coming from a camera (or a set of cameras), mounted on top of that object. This can be accomplished using one camera (monocular), or two cameras (stereo), with the latter method being preferred for this project due to increased robustness and decreased complexity of implementation.

The basic flow of the stereo visual odometry process is detailed in the following section. Video footage coming from the stereo rig is analyzed to find features, reliable landmarks that can be tracked continuously in the video sequence. These feature points in the left and right video streams are then triangulated to 3D points, using prior knowledge of the distance between the two imagers. Once 3D points are obtained, a 3-point geometric solution is used to infer the location of the stereo rig. Through a random sampling process, this solution is refined until a robust camera pose estimate has been obtained. Tracking the camera position over time gives is the trajectory of its movement, which contains all necessary odometric information.

4.2 PROCESS STEPS

4.2.01 Harris Corners

Figure 1: Videre's STH-DCSG/-C stereo head.

The first step in the process is the detection of features in every frame of the video that comes in to either of the two video feeds. Harris corner detection is an algorithm that processes an image and discovers the corners of the objects shown: it finds the locally averaged moment matrix computed from the image gradients, and then combines the eigenvalues of the moment matrix to compute a corner "strength," of which maximum values indicate the corner positions. The movement of a Harris corner in the video feed as the stereo head moves is the basic piece of information that allows us to make position estimates.

4.2.02 Feature matching and Tracking

Once the Harris detector has estimated the positions of the corners (features) in the image, we use the Sum Squared Difference (SSD) method to match each feature to its position in the subsequent image in the video stream. This will result in the creation of "feature tracks," which represent the movement of each feature with time.

The next step of the process is to match feature tracks between the left and right video feeds, which allows us to see how a certain feature's movement is reflected in each video feed.

4.2.03 Camera Calibration

The next step is the determination of the camera calibration parameters, which gives us the relationship between pixel distances in the video image, and distances in the real world. The calibration process is not part of the visual odometry algorithm, and is executed only once. Many tools exist that will perform automatic calibration, including the MATLAB Calibration Toolbox[4] and also the Small Vision System calibration tool[5]. This step determines a variety of intrinsic camera parameters, which will be used subsequently.

4.2.04 Stereo Triangulation

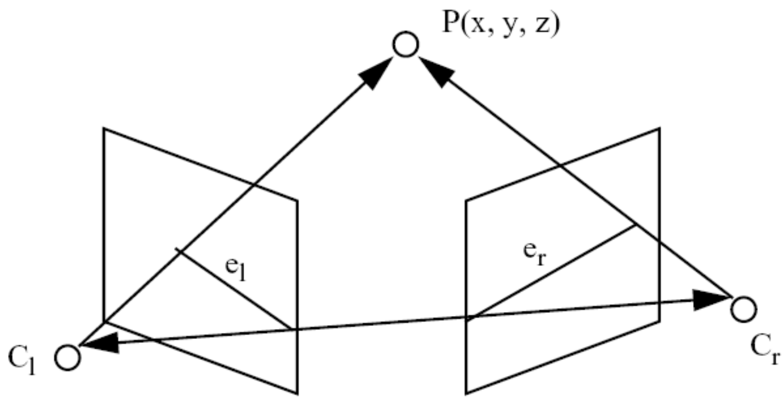


Figure 2: Epipolar geometry for triangulation

Once we have obtained the set of feature tracks as viewed by the left and the right image, we are ready to convert those to trajectories in world coordinates, with using epipolar geometry principles and the calibration parameters determined during the calibration step. Knowing the image coordinates of a point P in the world in both images, and the distance between the centers of projection of the two cameras (determined using the stereo baseline and calibration information), we can reconstruct the relative Cartesian coordinate position of the point relative to the stereo rig.

The result of this step is the resolution of features into 3D points in world coordinates.

4.2.05 Camera Pose Estimation

With each video frame now decomposed into 3D points, we use the preemptive Random Sample Consensus (RANSAC) algorithm[6] with 3 feature points[7],[8] to obtain a pose estimate for the camera. Running this on every frame will provide us with a continuously updated camera movement trajectory, in relation to the point where the process was started.

4.2.06 Firewall

After a certain set of features has been tracked over a certain number of frames, we drop some of the features tracked, and add some new ones to the set we are tracking. Also, at a given time interval, we drop all feature points tracked, and start afresh, to deal with error accumulation and propagation.

4.3 PERFORMANCE SPECIFICATION

Successful implementations of this approach yield real-time solutions of satisfying accuracy, after the necessary optimization steps. As demonstrated[9], this system can maintain tracking over distances as large as 0.5km in sparse environments, with minimal propagation error. For the needs of this project, the environments observed will be relatively dense, and the distances considered will not exceed 200-300m. The system needs to perform in real time, which is a concern, since the platforms used by other researchers exceed the specifications of the mobile systems available at the department. Therefore, development will proceed in parallel on a high-performance desktop system (the Quickbeam server, equipped with a 2GHz CPU and 1GB of memory), and our mobile platforms (running 900MHz Pentium III CPUs with 512MB of memory). Thus the performance of the system will be verifiable on the intended hardware, in case the mobile bases prove too slow. Parallel development overhead will not be a concern, since both systems run similar versions of Debian Linux, and use the same compiler.

4.4 REALISTIC DESIGN CONSTRAINTS AND SUSTAINABILITY

No new materials will be purchased for the needs of the project. The equipment that will be used includes: Videre stereo camera system, a computer workstation, and Magellan mobile robot. The manufacture of all of these sophisticated electronics incurred significant environmental cost, but the length of their life cycle makes them a practical and sustainable investment in the context of our society. All of these pieces have been in use for at least four years, and they will continue to be useful (with some upgrades) for a number of years to follow. The odometry method developed for this project will become an integral part of Swarthmore's Urban Search and Rescue system, which is an ongoing project, with direct positive ethical consequences.

There is no easy way to dispose of the system's components after the end of their useful life cycle. Computer systems can generally be sold back to the manufacturer, ensuring an environmentally responsible way of reworking, recycling, or disposing of their components. By law, battery elements have to be disposed of by an accredited company to ensure proper treatment of heavy metals and acids. At this point, decreasing the environmental cost of producing electronics is in the hands of the semiconductor and battery industry. There is no other practical way of realizing robotics projects.

All software and hardware developed and used for this system will adhere to appropriate IEEE (hardware) and ANSI (coding) standards. Any software and documentation developed will be available to the public under the General Public License in the public domain, for the advancement of science, and the free and timely dissemination of scientific knowledge.

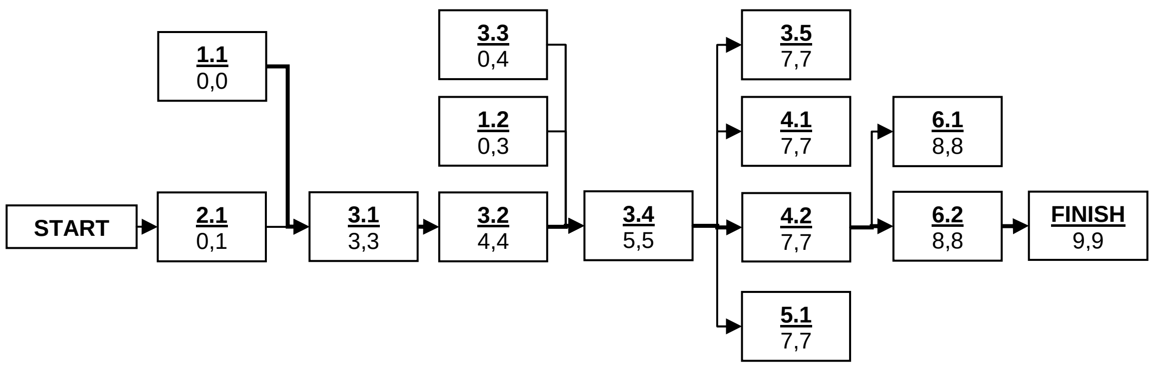
V. PROJECT PLAN

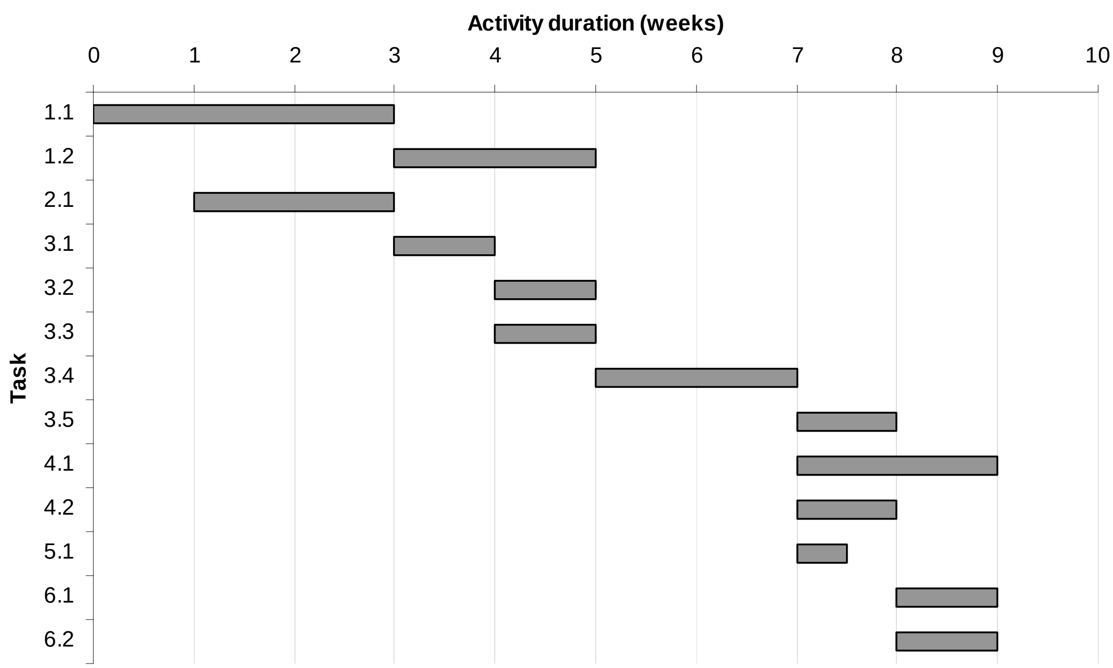
5.1 TASK ANALYSIS

Task	Description
5.1.01 Literature Review	
General literature review	Review all literature covering implementation specifics
RANSAC and 3-point method literature review	Understand details of the RANSAC and the 3-point pose estimation methods
5.1.02 Hardware Setup	
Connect stereo head and set up necessary drivers	Connect stereo rig and install all necessary software and drivers. Implement necessary interface to fetch video frames in real time and process them.
5.1.03 Algorithm Implementation	
Harris corner detector	Modify Harris existing features implementation, and implement single-camera feature tracks over a few frames. Match features between frames using simple SSD. **Deliverable:** Images with marked Harris corners
Feature matching	Match feature tracks between left and right cameras, track same feature in both cameras. **Deliverable:** Show feature track for same feature in both images
Stereo head calibration	Use SVS (or MATLAB) to calibrate the stereo head. Implement triangulation.
Implement RANSAC with 3-point algorithm	Implement RANSAC and iterative refinement. **Deliverable:** Show 3D trajectory of camera motion
Integrate with existing robot architecture	Set up the system as part of the robot's existing software architecture **Deliverable:** Show robot moving with accurate odometry
5.1.04 Testing	
Debugging and optimization	Ensure reliability and real-time performance.
Accuracy testing	Use linear positioner to estimate accuracy of trajectory estimation.
5.1.05 Documentation	
Review documentation	Make sure the code and development process are properly documented so future modification and extension is easy. **Deliverable:** Detailed documentation in HTML format
5.1.06 Reporting	
Develop presentation	Develop E90 presentation and necessary demonstrations. **Deliverable:** PowerPoint presentation
Write report	Write E90 report **Deliverable:** Final report

5.2 CRITICAL PATH METHOD (CPM) ANALYSIS AND GANNT DIAGRAM

Task ID	Name	Duration	Needs	Feeds
1.1	General literature review	3 weeks	-	3.1
1.2	RANSAC and 3-point method literature review	2 weeks	-	3.4
2.1	Connect stereo head and set up necessary drivers	2 weeks	-	3.1
3.1	Harris corner detector	1 week	2.1, 1.1	3.2
3.2	Feature matching	1 week	3.1	3.4
3.3	Stereo head calibration	1 week	-	3.4
3.4	Implement RANSAC with 3-point algorithm	2 weeks	3.2, 3.3, 1.2	3.5
3.5	Integrate with existing robot architecture	1 week	3.4	-
4.1	Debugging and optimization	2 weeks	3.4	-
4.2	Accuracy testing	1 week	3.4	6.1, 6.2
5.1	Review documentation	0.5 week	3.4	-
6.1	Develop presentation	1 week	4.2	-
6.2	Write report	1 week	4.2	-





CPM analysis on this project shows an overall project duration of 9 weeks, which is well within the bounds of a 13-week semester, leaving plenty of time to take care of any unanticipated setbacks. Since the project does not depend on the delivery of any materials, any problems should be solvable by the allocation of more time to work on the appropriate task.

VI. PROJECT COST

As stated already, no significant upfront monetary costs will be incurred during this project. The only exception is the need for the assembly of a power supply cable, so that the stereo head can be powered from the robot motherboard, which can be met with parts available at the electronics shop in Hicks at a negligible cost. To this end, some assistance may be required from Ed Jaoudi, the department's Electronics Specialist. In terms of assistance, significant help will be required from the project advisor for the clarification of concepts, and implementation advice. Prof. Maxwell is on leave this year, so availability may be an issue, but e-mail, voice chat, and appropriately scheduled (albeit short) meetings should provide sufficient access.

VII. PROJECT QUALIFICATIONS

The author of this project would like to believe his experience as a robotics research assistant and inside the classroom have prepared him for the successful completion of the project. Other resources – Prof. Maxwell's help, and the appropriate literature – will also be accessed in case of 7need. Although the author has little prior experience in the field of computer vision, exposure to relevant computer science, robotics, and mathematical concepts (supposedly) constitute sufficient preparation.

VIII. REFERENCES

[1] H. Choset, K. Nagatani, and N. Lazar, "The Arc-Transversal Median Algorithm: an Approach to Increasing Ultrasonic Sensor Accuracy," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, 1999.

[2] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2004.

[3] "STH-DCSG/-C Stereo Head User's Manual," http://www.videredesign.com/docs/sthdcsg.pdf

[4] "Camera Calibration Toolbox for Matlab," http://www.vision.caltech.edu/bouguetj/calib_doc

[5] "Small Vision System Calibration," http://www.videredesign.com/docs/calibrate-3.2.pdf

[6] D. Nistér, "Preemptive RANSAC for live structure and motion estimation," in *Proc. of the 9th International Conference on Computer Vision*, Nice, 2003.

[7] D. Nistér, "A minimal solution to the generalised 3-point pose problem," in *CVPR 2004*, June 2004.

[8] R. Haralick, C. Lee, K. Ottenberg and M. Nölle, "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *International Journal of Computer Vision*, 13(3):331-356, 1994.

[9] "Visual odometry"