

# Incremental Multiagent Robotic Mapping of Outdoor Terrains<sup>†</sup>

Kingsley Fregene<sup>‡</sup>, Raj Madhavan and Lynne E. Parker

Center for Engineering Science Advanced Research  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory (ORNL)  
Oak Ridge, TN 37831-6355, U.S.A.

E-mail: [kocfreg@ieee.org](mailto:kocfreg@ieee.org), [raj\\_madhavan@yahoo.com](mailto:raj_madhavan@yahoo.com), [parkerle@ornl.gov](mailto:parkerle@ornl.gov)

**Abstract**— We describe a scheme for building terrain maps of realistic outdoor environments by having multiple robotic agents navigate in them. The terrain map combines vision-based depth information of environmental features with an elevation gradient created by fusing vertical displacements obtained from inclinometer pitch angles with DGPS altitude data. Experimental results are presented to illustrate the practical application of this scheme.

**Keywords**— multirobot mapping, terrain mapping, depth-from-motion, robot vision .

## I. Introduction

THERE has always been interest in schemes for providing terrain maps to teams of robotic agents engaged in applications which require autonomous navigation. To be useful for applications like path-planning in realistic outdoor environments (which is our primary motivation), these maps should provide information about the location of objects/features in the environment and what the elevation gradient is across the area. Once the terrain map is known, paths may then be planned which are optimal in terms of the distance between origin and goal locations or the amount of energy expended, etc.

Our approach to outdoor multirobot terrain mapping is based on merging multiple local maps obtained by each member of the robot team during specific motion segments into a globally consistent metric map. The local maps combine the elevation gradient and vision-based depth (i.e., ranges) to environmental features. Each local map is obtained while the robots traverse the terrain of interest.

<sup>†</sup>The research reported in this article has been authored by contractors of the U.S. Government under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. The U.S. Government retains a nonexclusive royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes. Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, U.S. Department of Energy.

<sup>‡</sup> Corresponding author. Work done while visiting ORNL under the auspices of the HERE program. Currently with the Dept. of Electrical and Computer Engineering, University of Waterloo, Canada N2L 3G1 Fax: (519)746-3077.

Many researchers have studied the problem of mapping using robots. Thrun proposes a probabilistic algorithm for concurrent mapping and localization using a sample-based Monte Carlo Localization approach [1] and extends it to both multiple robots and 3D mapping of indoor environments [2]. The algorithm was implemented on a robot equipped with two laser rangefinders and a panoramic camera. A similar methodology for building metric maps through cooperation between multiple mobile robots has been adopted by Tercero *et al.* [3].

On the cooperative mapping front, Burgard *et al.* detail an explicit coordination mechanism that assigns appropriate target points to the robots in a team such that they effectively explore different regions [4] using an occupancy grid map that is built based on the data sensed by individual robots. Furthermore, a dead-reckoning-free robust positioning system for global mapping using multiple mobile robots is described by Dudek *et. al* in [5] by having the robots define a local coordinate system without reference to environmental features. In this robot-based representation, sensing errors remain localized to the individual robot. Inter-robot positions are determined by metric information but the resulting global map describes the neighbor relations between the robots in the form of a graph. Rekleitis *et. al* [6] report a graph-based exploration strategy for map construction in which two RWI B12 robots act in concert to reduce odometry errors. Each robot is equipped with a *robot tracker* sensor which tracks a geometric target installed on the other robot visually. In this scheme, the distance from one robot to the other is inferred from the height of the stripe pattern in the image.

Cooperative localization and occupancy-grid mapping of two homogeneous indoor robots each equipped with a stereo-vision head is described by Jennings *et al.* [7] and Little *et al.* [8]. Although grid-based maps have the advantage of allowing explicit modeling of free-space and ease of fusing data from different sen-

sors, they are often impractical for large unstructured environments due to the fixed grid-size and the accompanying computational burden. Additionally, false alarms and data association ambiguities are difficult to incorporate.

Howard *et al.* describe a cooperative localization and mapping (CLAM) [9] scheme in which two robots coordinate to reduce odometric uncertainty during unknown indoor environment exploration. Each robot is equipped with a color camera so that they ‘recognize’ each other using colored tags around their base. A disadvantage of the proposed approach is that at any given instant, only one robot of the team is allowed to move. In this way, the stationary robot estimates its own position with increased certainty than possible by odometry alone. This tends to limit both the speed and the accuracy of the maps that are constructed. Another disadvantage is that both the proposed cooperative localization and mapping approaches are centralized as opposed to the distributed approach we have adopted.

Huber and Hebert [10] consider three-dimensional mapping of large, unstructured terrain using terrestrial range sensors. Meshes are created from two range sensors - one ground-based and the other aerial. The aerial sensor is quite sophisticated in that it is capable of producing  $360^\circ \times 36^\circ$  field of view (FOV) range and reflectance images within a radius of 52 meters. Some issues associated with range mapping based on terrestrial sensors (for instance, lack of features, unknown transforms, etc.) are accounted for in the algorithm proposed therein.

In this work, a coordinate frame centered at the location of the DGPS base station is fixed to the environment of interest. This ensures that problems associated with combining locally generated maps are minimized. Furthermore, the scheme we present has no restriction on how many robots can move at any instant. Also, the sensors required are not unduly sophisticated. Mapping is decentralized and takes place in a realistic outdoor environment.

The organization of this paper is as follows: Section II gives an overview of the experimental setup and the robots used as mapping agents. The actual terrain mapping procedure is detailed in Section III followed by experimental results and concluding remarks in Sections IV and V respectively.

## II. Mapping agents and experimental setup

The robotic agents used in the mapping task are a team of Real World Interface (RWI) All TerRain Vehicles (ATRV-minis) depicted in Figure 1. Each robot possesses the full range of sensors utilized in the mapping task as follows:

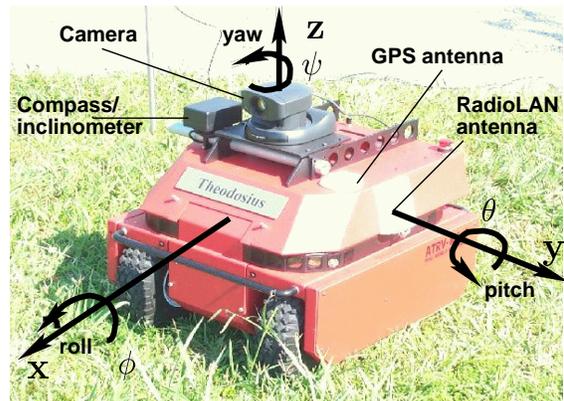


Fig. 1. A member of the ATRV-mini mapping team.

- (i) **Camera:** A pan-tilt-zoom (PTZ) capable CCD camera with focal length in the range  $5.4 - 64.8 \text{ mm}$  (i.e. a variable zoom system). Its pixel resolution is  $120 \times 160$  with a  $37^\circ$  FOV.
- (ii) **LADGPS:** An Ashtech-Magellan G12 GPS receiver/antenna which allows differential corrections to be received from a base station in the local area. Therefore, each vehicle in the team is part of a **Local Area DGPS (LADGPS)** system. Typical accuracies for this setup are in the  $10 - 15 \text{ cm}$  range with up to 10 satellites in view.
- (iii) **Magnetic compass:** The magnetic compass provides an external means of measuring the yaw angle  $\psi$ . It is calibrated and corrected for magnetic variation in the Northern hemisphere.
- (iv) **Inclinometers:** These provide measurements of the body pitch and roll angles,  $\theta$  and  $\phi$  respectively.
- (v) **Encoders:** The internal encoders provide measurements of the vehicle  $(x, y)$  positions and the respective translational velocities along those axes. Measurements of the heading (yaw) angle  $\psi$  and the angular rate are also provided.

Each robot in the team is essentially a node on a wireless LAN set up directly outdoors on the field (as shown in Figure 2) with a unique network identifier for each robot.

RWI’s *mobility* interface provides a transparent platform for querying/correcting the sensors on each robot, providing control/command signals and for inter-robot message passing. The essential capabilities of each sensor are encapsulated in a software server which is selectively loaded as desired.

While encoder readings are subject to drift during general motion, an **Extended Kalman Filter (EKF)** based decentralized localization scheme (described in a companion paper [11]) serves to correct these errors in an incremental manner. Some of the motion param-

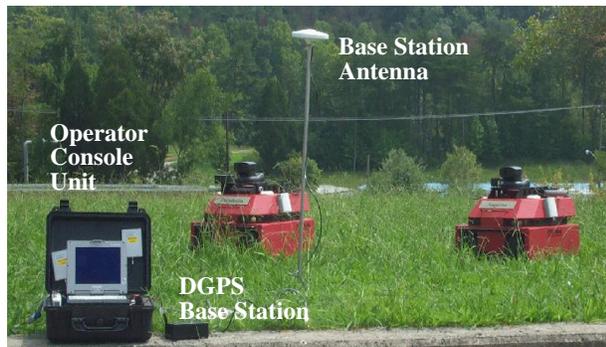


Fig. 2. The experimental setup.

eters to be used in depth range determination are also provided by the EKF-based scheme.

### III. The terrain mapping algorithm

Incremental terrain mapping takes place via four main processes. An incremental dense depth-from-camera-motion algorithm (which is an adaptation of the work reported in [12]) is used to obtain the depth to various features in the environment. The relative pose of the robots at these locations are associated with particular depth information. An elevation gradient of the terrain is determined by fusing GPS altitude information and vertical displacements obtained from inclinometer pitch angles. The depth and elevation information are then registered with their associated covariances. The terrain map is updated to incorporate the registered values at their proper coordinates. The covariances associated with each measurement provides the confidence the algorithm has in that measurement. In the case of overlapping areas, this confidence determines whether or not the map is updated. The overall schematic diagram of the algorithm is shown in Figure 3. Details about the implementation of each block are presented next.

#### A. Depth range determination

The depth of environmental features are determined from the optical flow between frames during small (known) camera motion. Assume a right handed coordinate system for the camera as shown in Figure 4. Two simple homogeneous transformations are required to go from the robot coordinate frame (as depicted in Figure 1) to the camera coordinates. This transformation from camera to robot can be represented as

$$H_c^r : [X_c \ Y_c \ Z_c \ 1]^T \longrightarrow [X \ Y \ Z \ 1]^T$$

where

$$H_c^r = \begin{bmatrix} \mathbf{R}_{z, \frac{\pi}{2}} \mathbf{R}_{x, \frac{\pi}{2}} & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix}$$

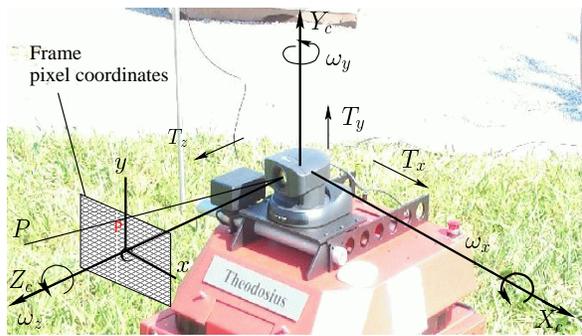


Fig. 4. Camera model and coordinates.

and  $r$  is the position vector of the camera center relative to the robot center of gravity while  $\mathbf{R}_{\cdot, \cdot} \in \mathbf{SO}(3)$  are rotation matrices about the indicated axes.

In the camera frame, each point in a scene has an associated position vector  $P = (X_c, Y_c, Z_c)^T$ . This is projected onto  $p = (x, y)^T$  in the image plane using the perspective projection:

$$x = \frac{fX_c}{Z_c}, \quad y = \frac{fY_c}{Z_c}$$

where  $f$  is the focal length of the projection and the point is located at a *depth*  $Z_c$  from the camera center. The image coordinate system is assumed centered at a center of projection corresponding to the frame center. On this basis, a simple camera model originally due to Sobel [13] is used to obtain the transformation from  $(x, y)$  in the image plane to actual pixel row and column. Define  $T = (T_x, T_y, T_z)^T$  and  $\Omega = (\omega_x, \omega_y, \omega_z)^T$  as the translational and angular velocities of the point due to camera motion. Then the image velocity  $V(x, y)$  is given by :

$$V(x, y) = d(x, y)\mathbf{F}(x, y)T + \mathbf{G}(x, y)\Omega \quad (1)$$

where  $d(x, y) = \frac{1}{Z_c}$  is the inverse depth (or *disparity*) and

$$\mathbf{F}(x, y) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix},$$

$$\mathbf{G}(x, y) = \begin{bmatrix} \frac{xy}{f} & -\frac{f+x^2}{f} & y \\ \frac{f+y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix}$$

The motion parameters  $T$  and  $\Omega$  in (1) are either estimated from the EKF-based scheme described in [11] or numerically computed from sensor data. In particular, the angular velocities are obtained from the angular rates about each axis by using the kinematic relation

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

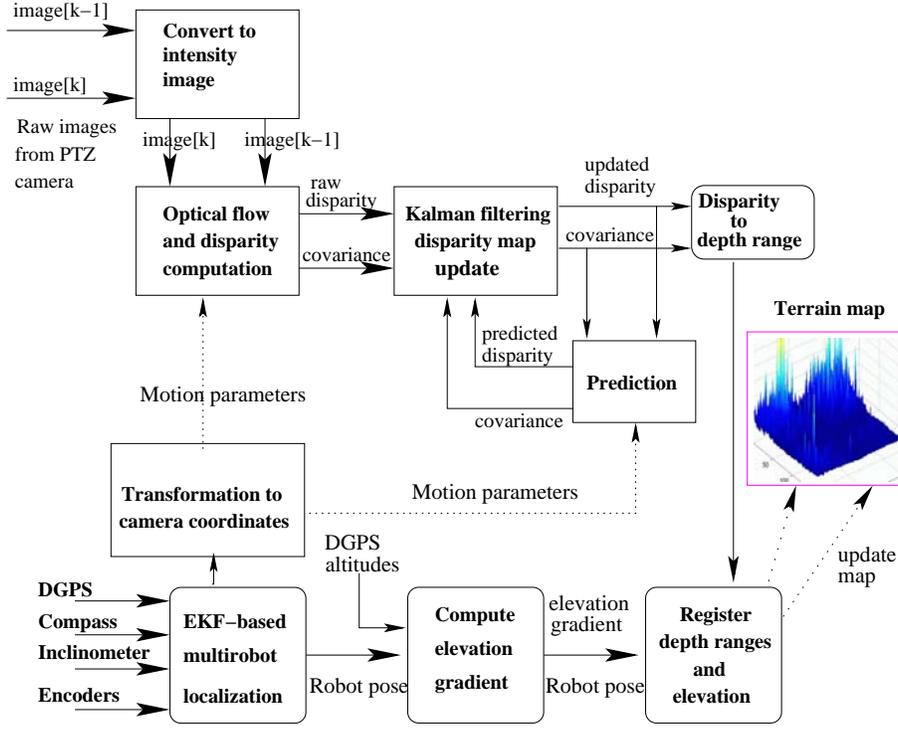


Fig. 3. The overall terrain mapping scheme.

Therefore, the only unknowns are the inverse depth  $d(x, y)$  and the image velocity  $V(x, y)$ . For small motions between frames,  $V(x, y)$  corresponds to the so-called *optical flow*. Thus, computing the optical flow for each pixel in the image frame provides a way to estimate dense depth from camera motion. The optical flow between two successive frames  $f_1$  and  $f_2$  (and associated variances) are obtained by a sum of squared difference (SSD) correlation-based algorithm as sketched in Figure 5:

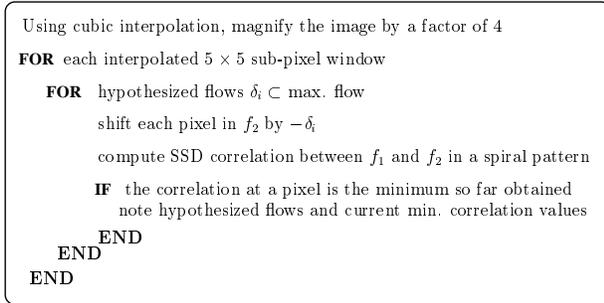


Fig. 5. Optical flow computation.

The reason for magnifying the image in step 1 is to obtain sub-pixel accuracy. This algorithm runs until the entire frame has been processed. The variance associated with the optical flow is determined by fitting the flow corresponding to the smallest SSD and its two

nearest neighbors to a parabola. Then the variance is  $\frac{2\sigma_n}{a}$  where  $a$  is the coefficient of the quadratic term in the parabola and  $\sigma_n$  is the uncertainty in pixel position [12]. Raw disparity is computed from (1) while the predicted disparity and variance are computed as

$$d(x, y) = (H^T P_m^{-1} H)^{-1} H^T P_m^{-1} \Delta \mathbf{x}$$

$$\sigma_d^2 = (H^T P_m^{-1} H)^{-1}$$

respectively.  $P_m$  is the covariance matrix obtained from the optical flow computation and

$$\Delta \mathbf{x} = V(x, y) - \mathbf{G}(x, y)\Omega, \quad H = \mathbf{F}(x, y)T$$

The predicted and raw disparities are fused using a Kalman filter, after which the pixel disparity is updated. To compose the update equations, let  $p_t^+$ ,  $p_t^-$  denote the updated and predicted covariances at the instant  $t$ , the covariance update equation and the Kalman filter gain  $K$  are respectively given by:

$$p_t^+ = \frac{p_t^- \sigma_d^2}{p_t^- + \sigma_d^2}$$

$$K = \frac{p_t^+}{\sigma_d^2}$$

The disparity at each pixel is then updated as

$$q_t^+ = q_t^- + K(d - q_t^-)$$

where  $q_t^+$  and  $q_t^-$  are the updated and predicted disparities, respectively.

## B. Obtaining an elevation gradient

Initially, a coordinate frame is fixed local to the area of interest (but global from the viewpoint of the members of the robot team). In this way, we ensure that all pose measurements are referenced to the same coordinate system. The 3D pose of the robot at the start location is registered, then the vision-based depth determination algorithm executes. The robot advances to a location near the closest object as inferred from the determined depth. GPS readings are used to determine vertical displacement of the robot between these two locations. This is fused with a vertical displacement measure  $h$  based on the inclinometer pitch angles and computed as:  $h = L \tan \theta$  where  $L$  is the offset between the robot center of gravity and the camera's optical center. The number of satellites acquired by the GPS is used as a measure to determine what confidence the algorithm should have in the vertical displacements provided by the GPS. The vertical displacements are monitored between two positions, say  $P_1$  and  $P_2$  as in Figure 6 to give an elevation profile for that motion segment corresponding to the depth map obtained. These depth and elevation information are registered to a local map which essentially encapsulates the terrain information within the robot's FOV between the two locations (see Sub-section III-C). At  $P_2$ , the robot turns away from objects inferred from the depth map, the object locations are marked, then new depth information is obtained. The robot advances to a point  $P_3$ , then creates a local map connecting  $P_2$  and  $P_3$  containing depth and elevation information. This process continues with each robot updating the global map as information significantly different from what is already registered on that map becomes available. The map making is incremental in the sense that the terrain is effectively partitioned into segments corresponding to location of inferred objects.

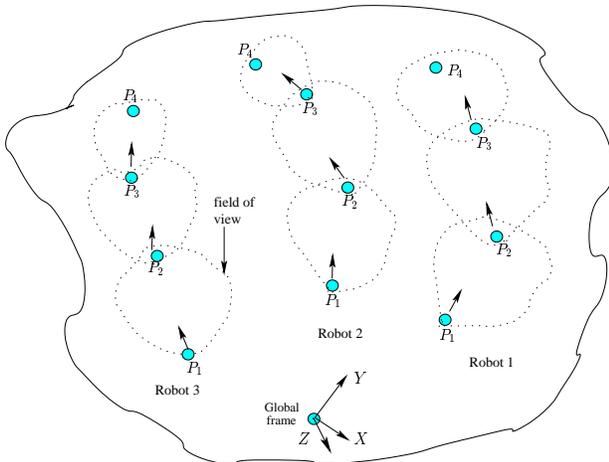


Fig. 6. Obtaining an elevation gradient.

## C. Elevation-depth registration

Registration of depth and elevation is simplified because a) there is a global coordinate frame in which all measurements are expressed; b) all the information to be fused are available in metric (rather than topological) form; c) the manner in which exploration is carried out implicitly takes depth information into account.

The registration is done after each robot traverses the region between two points, (say,  $P_1$  and  $P_2$  in Figure 6). On each robot, position coordinates  $((X, Y)$  in global frame) within the immediate FOV of the robot are matched with vertical displacements. Gaps are filled in by cubic interpolation so that when the robot arrives at  $P_2$ , a 3D map of a subset of the region it just departed is created. Since the robot advances up to points where the depth map indicates that 'interesting' features are present, the location of these features as marked on the map would correspond to the terminal point of the motion segment. So, depth registration simply consists of marking the  $(X, Y)$  coordinates of these points in the local map of that motion segment. The bearing information is calculated from the camera's angle of view and number of pixels per frame. (If we ignore the effects of lens distortion, 1 pixel of lateral displacement about the camera's  $Z_c$  axis approximately equals  $0.25^\circ$  in the present case).

## D. Updating the terrain map

The local maps generated by each of the mapping robots are combined into a global map based on the  $XYZ$  coordinate frame. For previously unmapped regions, the global map is updated as soon as the registration step is completed. In principle, the variances associated with the depth determination and the number of sensed GPS satellites may be utilized as confidence measures when the same region is traversed more than once. The global map update then depends on these confidence measures. However, this portion of the algorithm is still to be fully implemented. Therefore, the experimental results we report are based only on the *one-pass* update. The resulting map is itself a metric map available to path-planning algorithm in the form of a matrix from which a 3D surface is generated.

## IV. Experimental results and discussions

Our experiments show results obtained using two robots (**Augustus** and **Theodosius**) in an outdoor environment. The robots move slowly around the environment such that the *average* inter-frame displacement is no more than  $2cm$ .

The scenery within the robot **Augustus**' FOV during this particular motion segment is shown in Figure 7(a). The prominent object in this scenery is a tree behind which is a building. The filtered depth map

recovered for every pixel is shown in Figure 7(b). In this case, the lighter the area, the closer it is to the robot. Figure 7(c) shows the greyscale rendering of the state covariance after the last frame is processed. Here, the darker zones represent areas with lower covariances (i.e. more reliable depth estimates). Observe that high uncertainties (bright zones) are associated with the depth estimates for the building in the background and the skyline around the tree. Therefore, the 3D rendering of depth data depicted in Figure 7(d) should be interpreted in light of the aforementioned covariance image. Depth ranges to the tree's position and a slightly raised pavement behind it are more reliable than estimates to the building. We expect that the reliability of these estimates will improve as the robot gets closer to the features in question, but this has not been pursued further since our interest lies primarily in indicating areas where objects are present so that paths can be planned around them. Thus, using the mapping scheme we have described, the next motion segment for this robot starts on the other side of the tree - which is located about 10 m from the robot. Its orientation relative to the robot is obtained from the lateral pixel displacement relative to the center of the image frame.

Figure 8(a) shows the scenery within the FOV of *Theodosius*. The prominent features here are several culvert covers with a tree branch just over-hanging them. The recovered depth map is depicted in Figure 8(b) while the covariance rendering is shown in Figure 8(c). Observe that the depth estimates for the skyline and distant scenery have higher uncertainties associated with them. Also, Figure 8(c) suggests that there is more uncertainty in the estimate provided by *Theodosius* as compared with *Augustus*. This may be related to the level of scene complexity and actual ranges to the objects in the robot's FOV. Dense depth data are shown in Figure 8(d) for this case.

Both the elevation profile for the motion segments and the feature locations are mapped as shown in the partially updated terrain map (Figure 9). Although this update is still done offline for now, it shows the elevation profile across the area traversed by each robot (in the locally fixed coordinate frame centered at the LADGPS base station location) and prominent features within the robot's FOV during the motion segment are marked on the map. In reality, this terrain map is available to a path-planning algorithm as a *terrain matrix* in which the row and column indices correspond to  $X$  and  $Y$  positions while the actual matrix entry is the vertical displacement across the terrain ( $\Delta Z$ ). The path-planning routine utilizes the contents of this matrix to determine terrain properties like *traversability* and such like. Areas with prominent features would have entries several orders of magni-

tude higher than the neighboring displacement entries which would serve to flag them as areas to be avoided. Portions of the terrain still unexplored by the robots contain no elevation information at this instant. Once markers are placed at the approximate locations of the environmental features, exploration then continues on the other side of these markers (this is illustrated in the area explored by the robot *theodosius* in Figure 9).

## V. Conclusions and further research

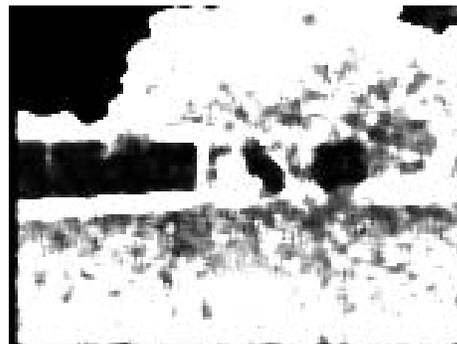
We have presented a scheme for multirobot outdoor terrain mapping which combines vision-based depth estimates with an elevation profile. The elevation profile is obtained by fusing vertical displacements from DGPS with those computed from inclinometer pitch angles while the robots traverse the terrain of interest. Experimental results have been presented to illustrate the application of this scheme to a terrain mapping problem which is partly motivated by path-planning requirements.

There are several ways to extend this work. The most obvious is to have explicit cooperation among the robots in the team, especially when identical or overlapping areas have been explored. Considering the fact that mapping based on depth-from-motion algorithms may be hard to apply in rough terrains, robots equipped with laser range-finders may be used in order to improve the bearing accuracy of features located in the environment. In this case, we would have a heterogeneous robot team and depth estimation can be done with greater accuracy by comparing vision-based and laser-based ranges. From a computational standpoint, it is probably sufficient to employ a feature-based (rather than iconic) depth estimation scheme since it is not usually necessary to mark the ranges to the entire FOV on the elevation profile. For this to be useful, it would be necessary to detect features of interest so that the algorithm tracks these rather than every pixel in the frame. Even for that kind of scenario, a vision-based scheme may not necessarily be the best approach for extremely rough and/or highly undulating terrains. It should be noted, however, that using a different scheme for depth range determination does not fundamentally change the mapping algorithm itself.

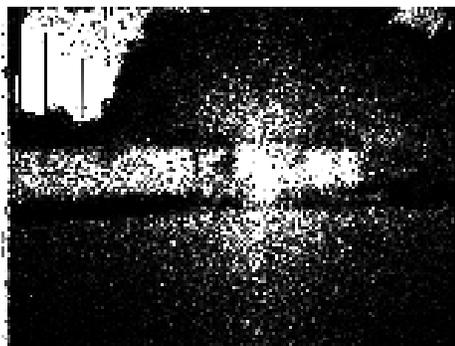
The kind of off-line map update shown in the experiments would typically suffice for path-planning in static or slowly-changing environments. Schemes to update the maps online in dynamic environments would be a significant extension to this work. Ultimately, we would like to combine the terrain mapping scheme presented here with the multirobot localization algorithms in the companion paper [11] so that robust cooperative localization and mapping (CLAM)



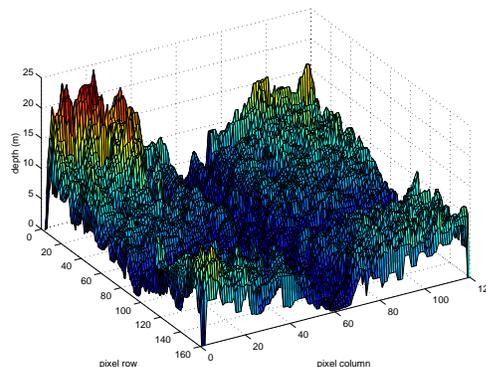
(a) Augustus: outdoor scene



(b) Augustus: recovered depth map



(c) Augustus: depth covariance



(d) Augustus: actual depth data

Fig. 7. Experimental results for the robot Augustus.

becomes possible. This work obviously represents a first effort - several of the issues raised above remain the subject of continuing research.

## References

- [1] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A Probabilistic Approach to Collaborative Multi-robot Localization," *Autonomous Robots*, vol. 8, no. 3, 2000.
- [2] S. Thrun, "A Probabilistic On-line Mapping Algorithm for Teams of Mobile Robots," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, May 2001.
- [3] J. Salido-Tercero, C. Paredis, and P. Khosla, "Continuous Probabilistic Mapping by Autonomous Robots," in *Proceedings of the International Symposium on Experimental Robotics*, Mar. 1999.
- [4] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative Multi-robot Exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, vol. 1, pp. 476–481.
- [5] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "Robust Positioning with a Multi-agent Robotic System," in *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Dynamically Interacting Robots*, 1993.
- [6] I.M. Rekleitis, G. Dudek, and E.E. Milios, "On Multi-agent Exploration," in *Proceedings of Vision Interface*, June 1998, pp. 455–461.
- [7] C. Jennings, D. Murray, and J. Little, "Cooperative Robot Localization with Vision-based Mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, vol. 4, pp. 2659–2665.
- [8] J. Little, C. Jennings, and D. Murray, "Vision-based Mapping with Cooperative Robots," in *Proceedings of SPIE - Sensor Fusion and Decentralized Control in Robotic Systems*, May 1998, vol. 3523, pp. 2–12.
- [9] A. Howard and L. Kitchen, "Cooperative Localisation and Mapping: Preliminary Report," Tech. Rep. TR1999/24, Dept. of Computer Science & Software Engineering, The University of Melbourne, July 1999.
- [10] D.F. Huber and M. Hebert, "A New Approach to 3-D Terrain Mapping," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, Oct. 1999, pp. 1121–1127.
- [11] R. Madhavan and K. Fregene and L. E. Parker, "Distributed Heterogeneous Outdoor Multi-robot Localization," *Submitted to the IEEE International Conference on Robotics and Automation*, Oct. 2001.
- [12] L. Matthies, T. Kanade and R. Szelinski, "Kalman Filter-based Algorithms for Estimating Depth from Image Sequences," *International Journal of Computer Vision*, vol. 3, pp. 209–236, 1989.
- [13] I. Sobel, "On Calibrating Computer-Controlled Cameras For Perceiving 3-D Scenes," *Artificial Intelligence*, vol. 5, pp. 185–198, 1974.

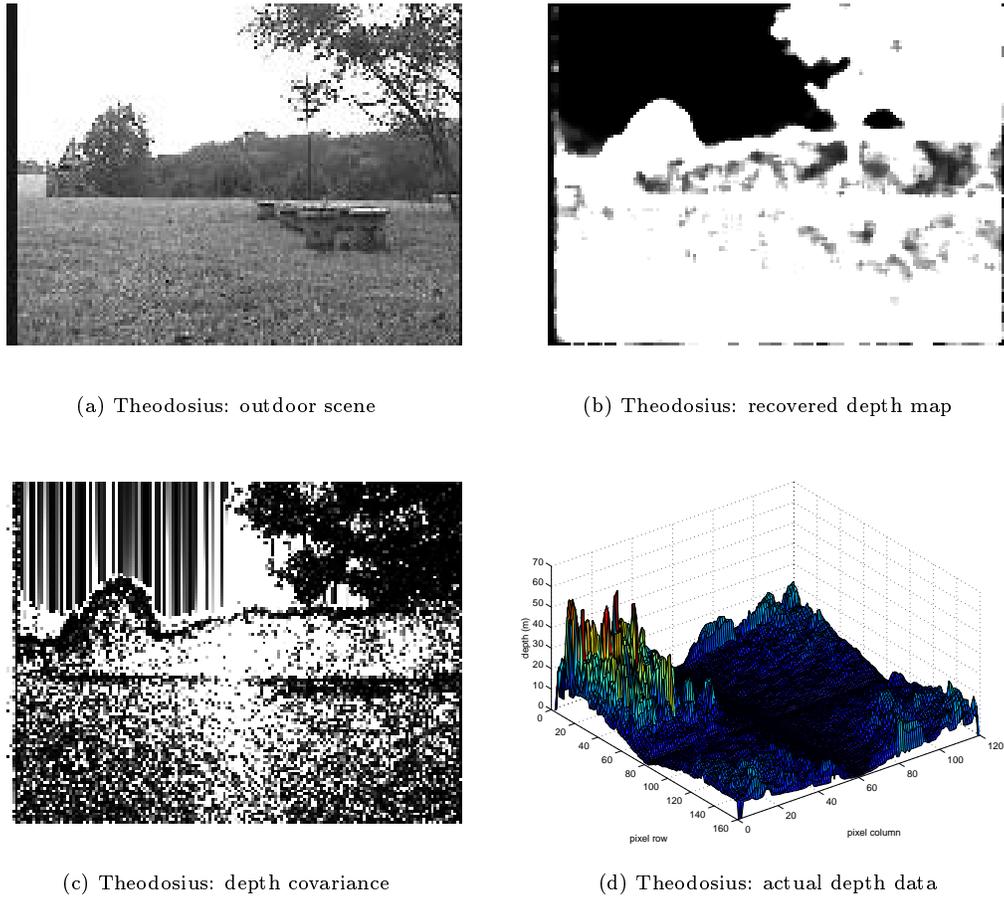


Fig. 8. Experimental results for the robot *Theodosius*.

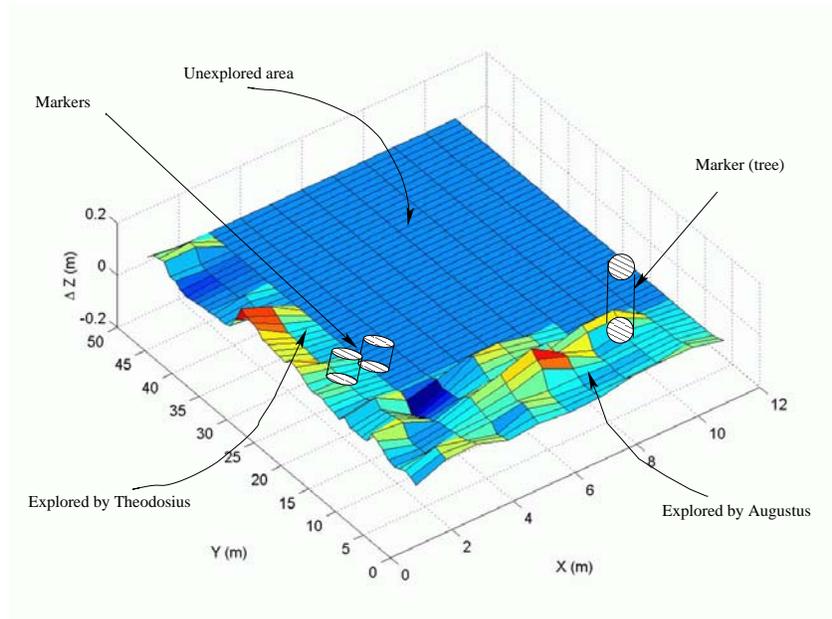


Fig. 9. Partially updated terrain map. The flat portion represents areas that the robots have yet to explore. For the area explored by *theodosius*, the markers represent the location of a pair of culvert covers in its FOV. The tree which features prominently in the area explored by *augustus* is so marked. For our purposes, these positions are available to path-planners as coordinates of a terrain matrix whose entries are much greater than surrounding entries.