Distributed Robotic Mapping of Extreme Environments

S. M. Thayer*, M. Bernardine Dias, B. Nabbe, B. Digney, M. Hebert, and A. Stentz

Robotics Institute, Carnegie Mellon University

5000 Forbes Avenue, Pittsburgh, PA 15213

ABSTRACT

In the extreme environments posed by war fighting, fire fighting, and nuclear accident response, the cost of direct human exposure is levied in terms of injury and death. Robotic alternatives must address effective operations while removing humans from danger. This is profoundly challenging, as extreme environments inflict cumulative performance damage on exposed robotic agents. Sensing and perception are among the most vulnerable components. We present a distributed robotic system that enables autonomous reconnaissance and mapping in urban structures using teams of robots. Robot teams scout remote sites, maintain operational tempos, and successfully execute tasks, principally the construction of 3-D Maps, despite multiple agent failures. Using an economic model of agent interaction based on a free market architecture, a virtual platform (a robot colony) is synthesized where task execution does not directly depend on individual agents within the colony.

Keywords: Distributed Robotics, Free Market Architecture, Cooperative Stereo, 3-D Mapping, Robot Colony

1. MOTIVATION

Military Operations in Urban Terrain (MOUT) pose fierce constraints such as limited visibility, complex and expansive fortifications, limited intelligence, and the presence of native populations and other non-combatants that prohibit deployment of large forces [1,2]. Further, the use of asymmetric threats, e.g. biological and chemical agents, against both land forces and indigenous populations in urban settings is an increasing likelihood [3]. These conditions place land forces and non-combatants in a highly non-deterministic, dangerous, confrontational, and volatile environment. An effort to identify and improve the ability of ground forces to project sufficient force and safeguard non-combatants is underway. This program, called the MOUT Advanced Concept Technology Development (ACTD), focuses on improving operational effectiveness in urban areas [4, 5].

The development of robotics technology will enable minimally invasive and precise MOUT operations that reduce risk to both ground forces and non-combatants by removing soldiers from dangerous and sometimes confrontational tasks [6]. Potential tasks for robotic systems include mine sweeping, reconnaissance, security and monitoring presence, and communications infrastructure [7]. DARPA has undertaken the task of enabling distributed robotics technology to execute urban reconnaissance missions. As a part of this effort, DARPA's Software for Distributed Robotics (SDR) program is looking at revolutionary approaches to the development of multi-agent robotic systems within this task domain. Under the auspices of the SDR Program, our team is producing software technology designed to construct interior maps of urban structures using groups of homogenous mobile robots. This concept software will be demonstrated on surrogate, simplistic mobile robotic platforms and eventually ported to mission capable mobile robotic systems.

Our approach to the problem focuses initially on the production of reliable, accurate, and fault-tolerant software by exploiting the redundancy inherent in a group of homogenous robots, or colony. Our methodology is to model multi-agent dynamics in terms of economic activity; thus, the motivation for robot effort is directly coupled to the rewards of task-execution [8]. We will incorporate learning and perception to build systems that degrade gracefully when faced with component or agent failures, that learn to execute tasks more efficiently than is possible as individual agents, and that deliver timely and accurate intelligence with respect to urban reconnaissance operations.

2. APPROACH

Some tasks, such as robotic exploration or reconnaissance, can be effective only if carried out by a team of robots. A robot team can accomplish a given task more quickly than a single agent can by dividing the task into sub-tasks and executing them

* Correspondence: Email sthayer@ri.cmu.edu : Phone (412)-268-7816, Fax (412) 268-5895

concurrently. A team can also make effective use of specialists (e.g., an agent that scouts an area, picks up objects, hauls payload) rather than require that each robot be a generalist, capable of performing all tasks but expert at none.
Difficulty arises, however, in coordinating all of these robots to perform a single, global task. One approach is to consider the robot team as a single robot "entity" with many degrees of freedom. A central computer coordinates the group optimally to perform the specified task. The problem is that optimal coordination is computationally difficult; the best known algorithms are exponential in complexity, so this approach is intractable for teams larger than a few robots. Further, the approach assumes that all information about the robots and their environment can be transmitted to a single location for processing and that this information does not change during the construction of an optimal plan. These assumptions are unrealistic for applications such as exploration in an unknown environment where communication is limited and robots behave in unpredictable ways. Another weakness with this approach is that it produces a highly vulnerable system. If the leader (the central planning unit) malfunctions, a new leader must be available or the entire team is disabled. A system that surmounts a single point of failure is crucial in applications such as urban reconnaissance.

Local and distributed approaches address the problems that arise with centralized, globally coordinated approaches. Each robot operates largely independently, acting on information that is locally available through its sensors. The robots are better able to respond to unknown or changing environments, since they sense and respond to the environment locally. A robot may coordinate with other robots in its vicinity, perhaps to divide a problem into sub-problems or to work together on a sub-task that cannot be accomplished by a single robot. Typically, little computation is required, since each robot need only plan and execute its own activities. Also, little communication is required, since the robots communicate only with others in their vicinity. The disadvantage of distributed approaches is that the solutions can be grossly suboptimal.

Consider an economic system for coordinating robots. In market economies, individuals are free, within given bounds, to exchange goods and services and enter into contracts at will. Individuals are in the best position to understand their needs and the means to satisfy them. At times they cooperate with other members of the society to achieve an outcome greater than that possible by each member alone. At times they compete with other members to provide goods or services at the lowest possible cost, thus eliminating waste and inefficiency. Combining the best of both approaches, market economies are robust and nimble like distributed approaches, yet allow for some centralization for optimization when time and computational resources permit.

In the past decade there has been growing interest in multi-agent systems. Mataric [18] presents a comprehensive summary of some of the principal efforts in this area of research. Jensen and Veloso [15], Švestka and Overmars [28], and Brumitt and Stentz [13] are examples of the centralized approach to control a multi-robot system organized hierarchically. A number of researchers have developed biologically inspired, locally reactive, behavior-based systems to carry out simple tasks [9, 10, 11, 17]. These distributed systems have found applications in many different domains. Additional review of recent approaches to multi-robot teaming can be found in [8]. To the best of our knowledge, we are the first to use a free market architecture for controlling a team of self-interested agents that perform tasks and solve problems.

3. FREE MARKET ARCHITECTURE[1]

3. 1 Revenues and Costs in the Economy

When a team of robots is modeled as an economy to perform a task, the goal of the team is to execute the task well while minimizing costs. A function, ***trev***, is needed that maps possible task outcomes onto revenue values. Another function, ***tcost***, is needed that maps possible schemes for performing the task onto cost values. As a team, the goal is to execute some plan ***P*** such that profit, ***trev(P) – tcost(P)***, is maximized. Individual cost and revenue functions must also be designed to provide a means for distributing the revenue and assessing costs to individual robots.

The sum of individual revenues and costs will determine the team's revenues and costs. However, the distribution is not even: individuals are compensated in accordance with their contribution to the overall task, based on factors that are within the control of each. An individual that maximizes its personal production and minimizes its personal cost receives a larger share of the overall profit. Therefore, by acting strictly in their own self-interests, individuals maximize not only their own profit but also the overall profit of the team.

3.2 Determining Price via the Bidding Process

[1] See [8] for a more detailed description of the free market architecture.

The team's revenue function is not the only source of income for the robots. A robot can also receive revenue from another robot in exchange for goods or services. In general, two robots have incentive to deal with each other if they can produce more aggregate profit together than apart. Such outcomes are win-win rather than zero-sum. The *price* dictates the payment amount for the good or service. Since many factors that could affect the price may be hidden or complex, a common approach to determining price is to *bid* for a good or service until a mutually acceptable price is found. A plausible bidding strategy is to start by bidding a price that is personally most favorable, and then successively retreat from this position until a price is mutually agreed upon.

Note that a given robot can negotiate several potential deals at the same time. It bids the most favorable price for itself for all of the deals, successively retreats from this position with counter bids, and closes the first deal that is mutually acceptable. Note also that a deal can be multi-party, requiring that all parties agree before any part of the deal is binding. The negotiated price will tend toward the intersection of the supply and demand curves for a given service. If a service is in high demand or short supply, the price will be high. This information will prompt other suppliers to enter the fray, driving the price down. Likewise, if demand is low or supply high, the low price will drive suppliers into another line of business. Thus, price serves to optimize the matching of supply to demand. Finally, it is important to note that price and bidding are low bandwidth mechanisms for communicating aggregate information about costs.

3.3 Opportunistic Optimization, Learning, and Adaptation

Conspicuously absent from the free market system is a rigid, top-down hierarchy. Instead, the robots organize themselves in a way that is mutually beneficial. Thus, robots will cooperate if they have complementary roles, i.e., if all robots involved can make more profit by working together than by working individually. Conversely, robots will compete if they have the same role, i.e., if the amount of profit that some can make is negatively affected by the services provided by the others. The flexibility of the market model allows the robots to cooperate and compete as necessary to accomplish the assigned task. Since the aggregate profit amassed by the individuals is directly tied to the success of the task, this self-organization yields the best results. But this does not preclude centralized optimization; instead, it is opportunistic as circumstances permit.

One of the greatest strengths of the market economy is its ability to deal successfully with changing conditions. Since the economy does not rely on a hierarchical structure for coordination and task assignment, the system is highly robust to changes in the environment, including destroyed or injured robots. Disabling any single robot will not jeopardize the system's performance. By adding escape clauses for "broken deals," any tasks undertaken by a robot that malfunctions can be re-bid to other robots, and the entire task can be accomplished. It is also important to realize that a robot does not need to remain idle if only a portion of its functionality is disabled. The market architecture allows a partially disabled robot to bid out the use of its available resources opportunistically. For example, if a robot gets stuck while executing a task, it can still use its computing resources as needed while attempting to recruit another robot to mobilize itself. Another key to optimality is that a robot can always sub-contract a task it undertook if the conditions change and it becomes more profitable to outsource (i.e., get a different robot to carry out the task). Thus, the market model allows the robots to deal with dynamic environments in an opportunistic and adaptive manner.

Within the market economy, a robot can learn new behaviors and strategies as it executes missions. This learning applies to both individual behaviors and negotiations as well as to the entire team. Individual robots may learn that certain strategies are not profitable, or that certain robots are apt to break a contract by failing to deliver the goods or proper payment. Individuals may also learn to identify areas where the risk of injury is high and share this knowledge with members of the colony. The robot team may learn that certain types of robots are in over-supply, indicated by widespread bankruptcy or an inability to make much money. Conversely, the robot team may learn that certain types of robots are in under-supply, evidenced by excessive profits captured by members of the type. Thus, the population can learn to exit members of one type and recruit members of another.

3.4 Constructing the Free Market Architecture

Consider a team of robots assembled to perform a task. A software agent, representing the user, negotiates with the team to perform the task. The user desires the team to perform the task well while minimizing costs. To accomplish this, the user defines a revenue function that maps possible task outcomes onto revenue values, and a cost function that maps possible schemes for performing the task onto cost values. With the aim of maximizing their individual profits, the robots bid on parts of the task. They receive revenue for performing their subtasks, and they are charged for costs incurred during execution. Once they have received their assignments, the robots can continue to negotiate among themselves, buying and selling subtasks to minimize costs. If the task changes during execution, perhaps due to new information, new subtasks are created that are bid out to the robots. If a robot malfunctions or dies during execution, its subtasks are bid out to the surviving robots. Robot resources, such as sensing, communication, and computing, can be bought and sold in the marketplace to concentrate resources where they are most productive.

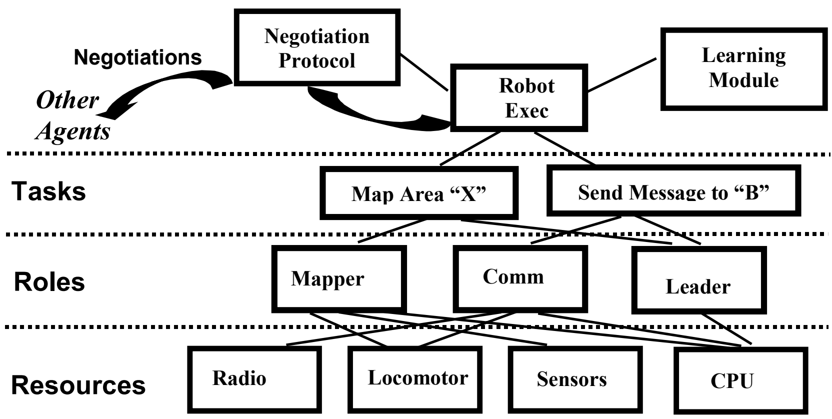

Figure 1: Robot software architecture for each individual in a robot colony

As shown in Figure 1, the architectural for each robot is layered. In the bottom layer are the resources under the robot's control, such as sensors, effectors, locomotors, computers, and communication devices. These resources are available to the robot to perform its tasks, and any unused resources can be sold to other robots in the colony. The next layer consists of the robot's roles for accomplishing tasks. Roles are application-specific software modules that implement particular robot capabilities or skills, such as collecting sensor data or hauling materials. The roles utilize resources (in the layer below) to execute their tasks. Roles execute tasks that match their specific capabilities. They receive assignments by bidding on tasks offered by other robots. As they execute their tasks, they may generate other tasks or subtasks to be bid out to the other robots.

At the top layer in the architecture, the robot executive coordinates the activities of the robot and its interactions with other robots. The executive bids on tasks for the robot to perform and offers tasks for sale. It matches tasks to roles, schedules the roles to run, resolves contention for resources, and offers unused resources for sale to other robots. The executive is equipped with a learning module that enables it to perform better over time by learning which negotiation strategies are the most effective.

4. COOPERATIVE STEREO MAPPING ROLE

One important role for individuals within robot colonies is distributed mapping. This role is illustrated in Figure 2, where robots 1 and 2, each equipped with one camera, are controlled by the planner so that the fields of view of their cameras provide maximum coverage of the environment. Assuming that the planner has also controlled the robots so that the fields of view overlap (at positions A and B for example), it is possible to reconstruct the 3-D geometry of the environment at those positions by processing the images from the two robots. Reconstructing 3-D maps from two robots is the cooperative stereo approach to map reconstruction. Collecting 3-D maps reconstructed from different positions provides a complete representation of the environment built cooperatively (the mapped area is shown in yellow or lightly shaded for b/w prints in Figure 2). This representation is suitable for:

- **Viewing:** The user can see the environment from different viewpoints, measure distances to or sizes of objects of interest, and generate floor plans for future use.
- **Route planning:** The location of potential obstacles can be derived from the 3-D maps. The obstacles can be detected at far greater range than can be achieved by conventional safeguarding sensors such as sonar. This information can be used directly by a route planner.

- **Coverage planning:** A difficult problem in exploration by multiple robots is to decide where to look next in order to ensure coverage of the environment. Using the maps, it is possible to reason about the relative positions of surfaces in order to plan paths for coverage. In particular, it is possible to detect parts of the environment that are occluded by surfaces visible from the recorded robot positions, and which should be explored.

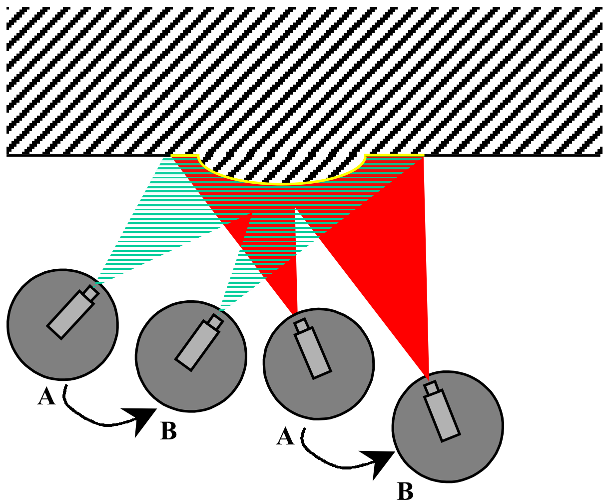

Figure 2: Distributed mapping scenario with two robots cooperatively constructing a stereo map.

It is important to note that while in principle it is possible to recover 3-D data from a single robot with stereo cameras, using cooperating robots has important advantages. One obvious advantage is the savings in hardware and computation time since a single camera is needed on each robot and the full load of the stereo computation does not reside on any given robot. A more important advantage, however, is that stereo reconstruction accurate enough for rendering can be achieved from a long range by taking advantage of the fact that the robots may be separated by a wide baseline. To be useful, the system must generate at least qualitatively correct reconstruction up to 10-20m, which is not possible using the short baseline that is typical on small robots. To illustrate this point, the graph in Figure 3 shows the error in depth Z as a function of Z for the type of camera currently used with a field of view of 60 degrees and for different baseline values B=0.1, 0.2,0.5,1,2m. The error is plotted using the standard stereo error formula $\Delta Z = Z^2/Bf$. The graphs clearly show that reconstruction past a few meters from the robot is not practical for short baselines and that baselines on the order of one meter or more become necessary. Such large baselines can be achieved only through the use of multiple robots.

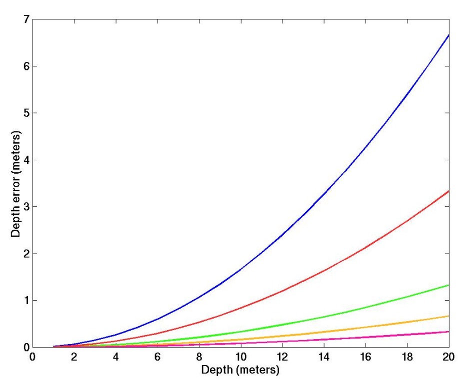

Figure 3: Error in reconstructed depth as a function of distance from the camera for different baselines.

Robust recovery of the scene geometry from multiple robots and extraction of information from the maps involves:

Cooperative Calibration: Although the relative positions and orientations of robots 1 and 2 are known to some degree from dead-reckoning and other means of internal pose estimation, the accuracy of the relative pose between them is typically not sufficient for stereo reconstruction. Cooperative calibration recovers relative poses of robot cameras automatically from image features.

Spare Feature Estimation: Once the robots are mutually calibrated, the next task is to compute the 3-D positions of sparse features in the scene.

Dense Reconstruction: The final step in reconstructing scene geometry from robots 1 and 2 is to convert the sparse structure to a dense reconstruction of the environment suitable for rendering and texture mapping.

4.1 Cooperative Calibration

Given images of a scene from arbitrary robot positions, the first task is to recover the relative poses of the cameras. These poses must be computed precisely since an error of a few image pixels in mapping points from one image to the next may lead to large errors in the 3-D reconstruction. In particular, the pose estimates from dead reckoning are not precise enough for image matching. The general approach is to extract point features from the images, find correspondences between the two sets of features and recover the relative poses of the cameras from the feature matches. Typical features used for calibration are shown in Figure 4. These features were extracted using an interest operator derived from the Harris operator [43].

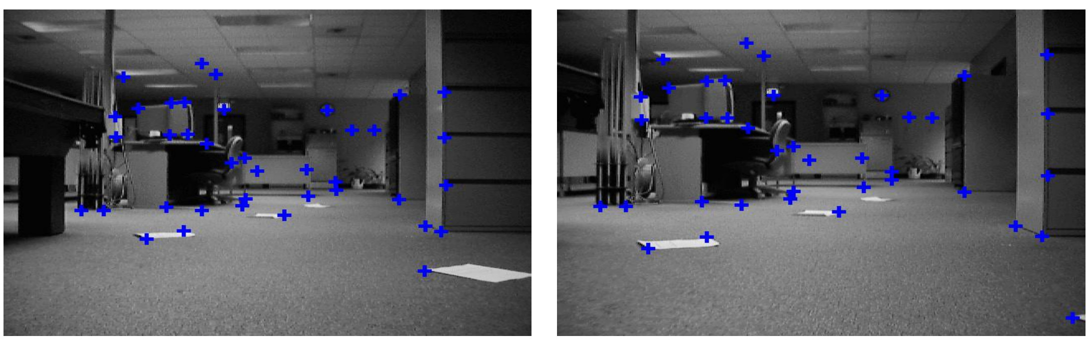

Figure 4: Typical sparse features extracted from indoor images using the Harris operator.

4.1.1 Feature matching using photometric invariants

Instead of using local correlation to match the features, we use a technique based on photometric invariants that was originally introduced in [37] and used for image matching in [42]. Given a feature location **p** = (x,y), the general idea is to compute the derivatives up to order 3 (I_x, I_{xx}, I_{xy}, etc.) of the image in a neighborhood around **p.** Then, combine the derivatives into quantities that are invariant to rotation and translation in the image. For example, the quantity $I_x^2 + I_y^2$ is the magnitude of the gradient, which is clearly invariant by rotation. For derivatives up to order 3, it can be shown that 8 such invariants exist so that each feature $\mathbf{m}^i_j$ can be described by a 8-vector $\mathbf{v}^i_j$ which characterizes the local distribution of intensity and is geometrically invariant.

Invariance to scale is also important since the scale of the features may vary substantially across images in general poses. Although direct scale invariance is not possible, the derivatives can be easily computed at different scales since they are computed by convoluting derivatives of Gaussian filters of variance, for example, $I_x = G_x(\sigma) \otimes I$, where σ is the variance of the Gaussian filters. The vectors $\mathbf{v}^i_j$ are computed over a range of values of σ corresponding to a range of image scale factors. Features that have similar invariant vectors are matched. The similarity between vectors is defined as the weighted sum of squared differences between their coordinates. A similar approach is used in [39] for recovering epipolar geometry.

4.1.2 Using Prior Pose Estimates

The second enhancement to the standard image-based calibration algorithm is to take advantage of the fact that we do have information about the relative poses from the robots' own internal pose estimation systems. In fact, we not only have an approximate estimate of the robot pose but also an estimate of the uncertainty of the pose. The situation is illustrated in

Figure 5. Using camera 1 as the reference camera, the true pose of camera 2 with respect to camera 1 is (**R**,**T**); the relative pose reported by the positioning systems is (**R'**,**T'**); and the error between the true and estimated poses is (dθ,d**T**) where dθ is the error in orientation in the plane of travel of the robot. Although (dθ,d**T**) is not known, the positioning system can provide a bound on $d\theta_{max}$ – the maximum angular deviation from the true orientation – and $d\mathbf{T}_{max}$ – the maximum position error which places all the possible positions of camera 2 in a disk centered at **T** (Figure 5(a)). Those bounds define a region in which the true pose (**R**,**T**) may lie: (**R**,**T**) is in the region defined by all the possible (**T'**+d**T**) and **R**(θ'+dθ) with $\|d\mathbf{T}\| < d\mathbf{T}_{max}$. Given a feature position in image 1, $\mathbf{m}_1$, the set of corresponding epipolar lines in image 2 is defined by the parameter vectors $(\mathbf{T}'+d\mathbf{T}) \times \mathbf{R}(\theta'+d\theta)\ \mathbf{m}_1$ for all possible values of d**T** and dθ. This set of lines defines a search region for the correspondences with $\mathbf{m}_1$ – shown in green (gray shaded area for b/w prints) in Figure 5(b) around the initial estimate of the epipolar line shown in red (darker line for b/w prints). If the initial position error is small, the search problem is reduced to searching in the vicinity of the epipolar line and the calibration amounts to a small adjustment of **R** and **T**. For larger position errors, the region covers a larger area of the image but still eliminates most of the outliers in practice.

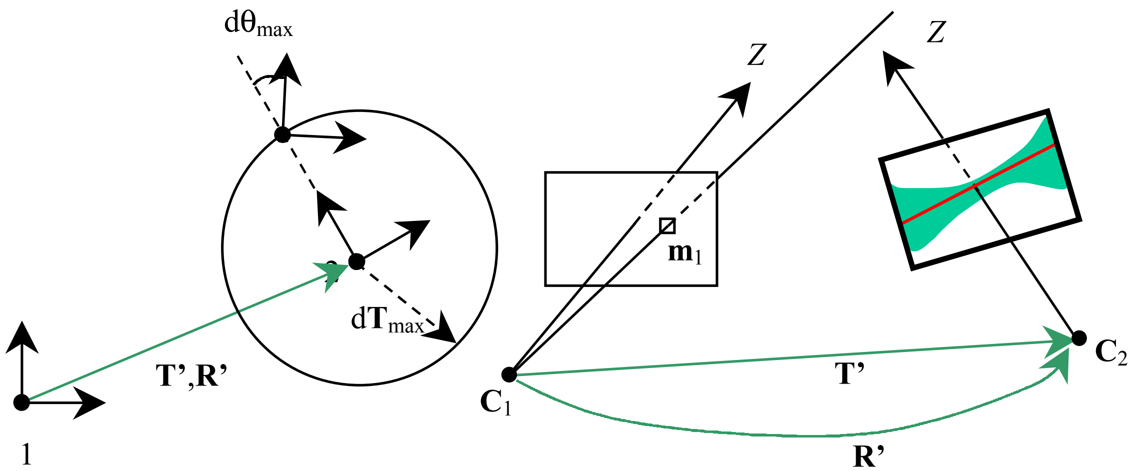

Figure 5: Using prior pose estimations to limit search

4.2 Sparse Feature Estimation

Given the calibration information from the two robots, the 3-D locations of each feature can be recovered by triangulation. In this case, the coordinates of each feature are computed independently. We have also applied global estimation techniques, which attempt to refine robot positions and feature positions by minimizing the distance between the image features and the image projection of the corresponding points – bundle adjustment technique [33]. Such global techniques become beneficial only when a substantial number of images are used. In our case, the experiments show that those techniques do not substantially affect the result and that straight triangulation is sufficient.

For example, Figure 6 (left) shows the depth reconstructed of matched features of Figure 4, as well as the same result (right) (right) in which the 3-D points corresponding to the features are shown in an oblique view. The lines are drawn from the point features to the ground plane. A few objects are indicated to facilitate visual registration of the data. This first reconstruction step provides a representation of the sparse structure of the environment. This sparse information can be reasonably used for planning or partial occlusion reasoning.

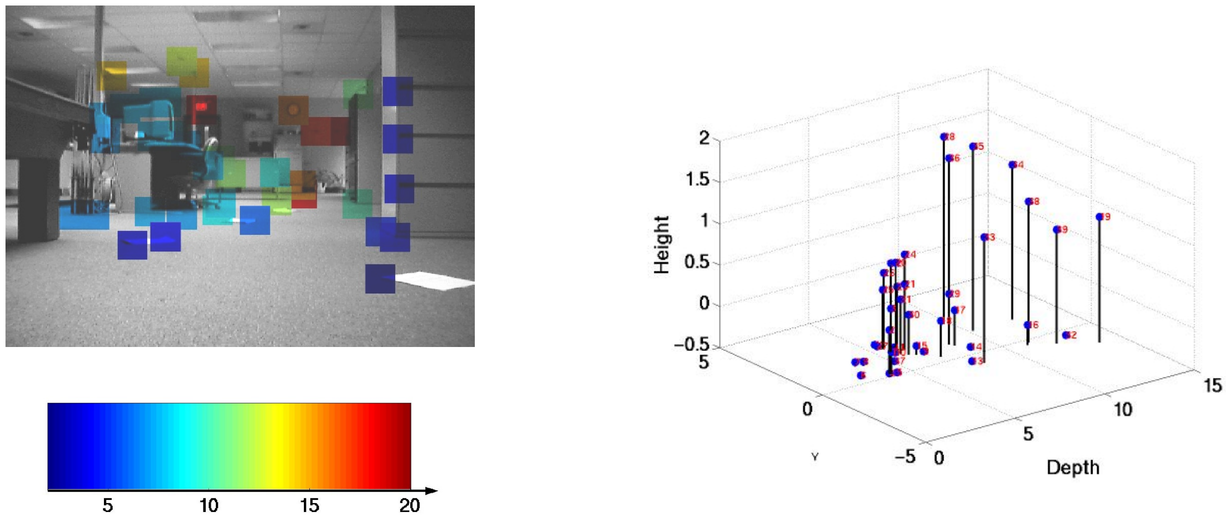

Figure 6: (Left) Sparse depth map computed from the features of Figure 4. The depth is indicated by a color-coded square centered at each feature. The color scale is indicated below the figure. (Right) Oblique view of the reconstructed 3-D points corresponding to features of Figure 4.

4.3 Dense Reconstruction

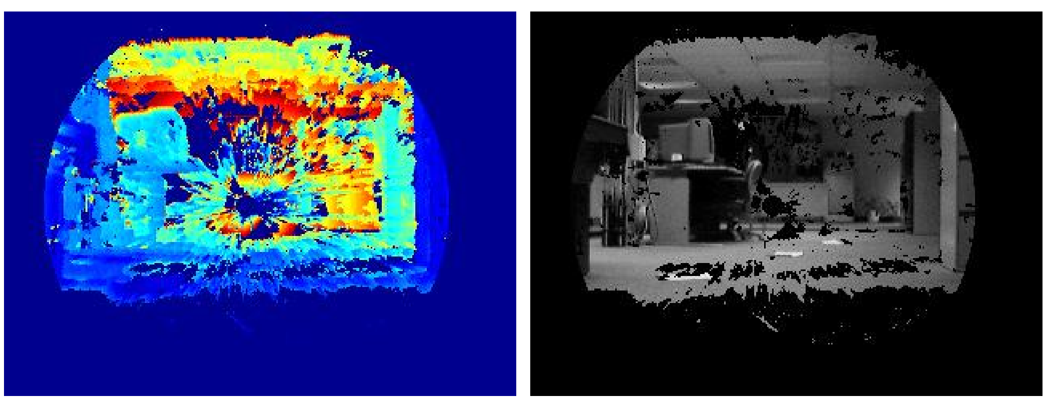

Figure 7: Color-coded depth map after filtering (left) and corresponding mask (right)

Sparse feature estimation provides useful information for visualization and planning but is insufficient for realistic rendering of the environment, which requires the recovery of the coordinates of a dense set of pixels in the image. Given two robot positions and assuming that the calibration problem has been solved using the techniques previously described, the problem is equivalent to the standard problem of stereo except that the images are taken from different robots instead of from a single camera rig, as is normally the case.

The general approach to stereo is to rectify the images so that the epipolar lines are the scanlines of the images and to search along the scanlines for matching locations based on a local comparison criterion such as correlation or sum of squared differences (SSD.) The rectification is critical because searching along arbitrary scanlines is not computationally practical. While in principle this approach is feasible, there are two key differences with standard stereo from fixed cameras, which are both related to the fact that the relative positions and orientations of the robots are arbitrary. First, as noted, the baseline between the robots may be large. Although necessary for accuracy reasons, this complicates the matching between images because of potentially large distortions between the two images. Second, the search for correspondences is complicated by the fact that rectification may lead to unacceptable warping of the images. Some preliminary results are shown in Figures 7 and 8(a) and 8(b).

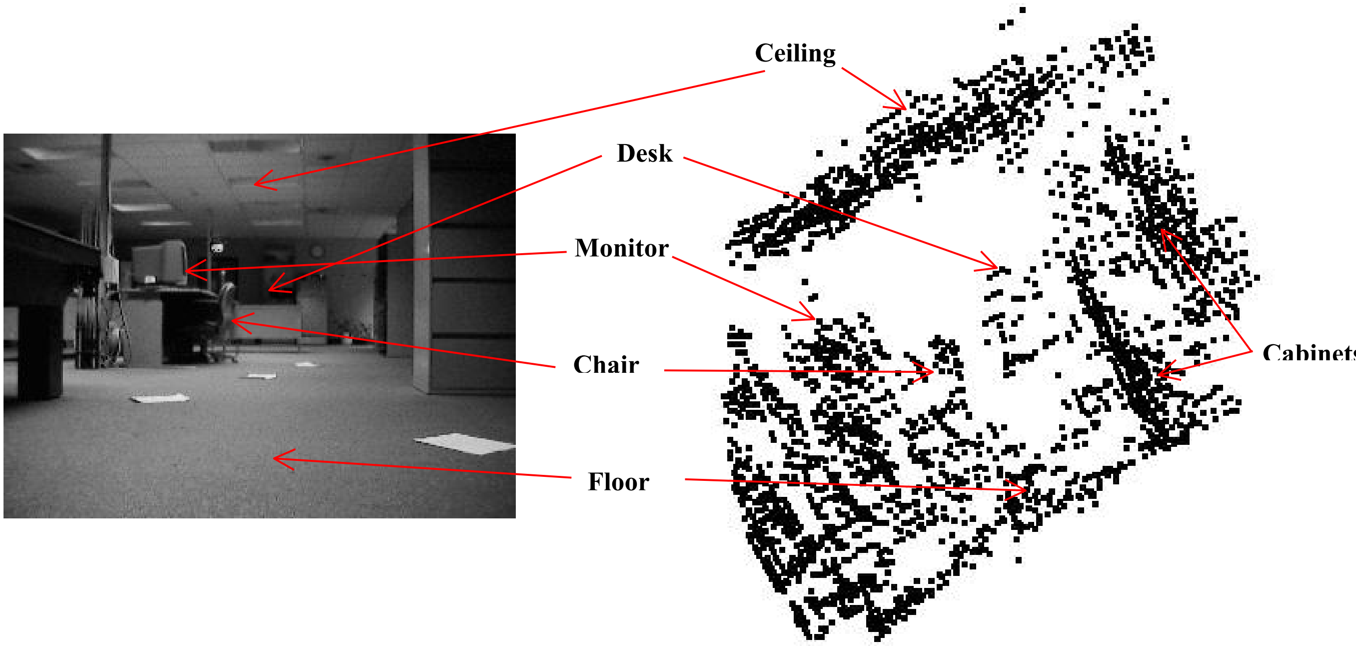

Figure 8: 3-D View of the point cloud constructed by our cooperative stereo mapping

5. RESULTS

This architecture was first verified by tasking a team of mobile robots with distributed mapping. [2] The mobile robots, located at different starting positions in a known simulated world, were assigned the task of visiting a set of pre-selected observation points. The robot colony was structured as illustrated below:

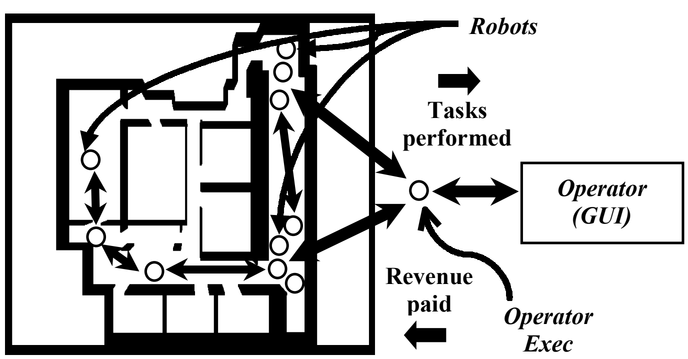

Figure 9: Organizational structure for a colony of robots engaged in distributed mapping

This problem is equivalent to the distributed traveling salesman problem, where the observation points are the cities to visit. Each agent was equipped with a map of the world, which enabled it to calculate the cost associated with visiting each of these cities. The costs were the lengths of the shortest paths between cities in an eight-connected grid, interpreted as money. The interface between the human operator and the team of robots was a software agent, the *operator executive* (*exec*). The *exec*

---

[2] Detailed descriptions of the implementation and results can be found in [8].

conveyed the operator's commands to the members of the team, managed the team revenue, monitored the team cost, and carried out the initial city assignments. Being a self-interested agent, the *exec* aimed to assign cities quickly while minimizing revenue flow to the team. In our initial implementation, the *exec* adopted a greedy algorithm for assigning tasks. Once the *exec* had completed the initial city assignments, the robots negotiated among themselves to subcontract city assignments. Only single-city deals were considered, and the robots continued to negotiate among themselves until no new, mutually profitable deals were possible. Thus, negotiations ceased once the system settled into a local minimum of the global cost. Dias and Stentz published preliminary results[8].

We also tested system response to dynamic conditions in a scenario similar to exploration of a partially known world. The operator designated a set of 14 observation points, or cities, to be visited in a simulated world. Four robots negotiated, won tours, and set off to explore. Whenever a robot approached a doorway, a new observation point was triggered inside the "newly observed" room. When a new goal was triggered, the robots ceased their current tours and began negotiations to determine new tours that were most profitable in light of the new information. This resulted in an additional six cites for exploration, for a total of 20 cites. Through this exploration scenario.[3], the robots evolved by adapting to dynamic conditions.

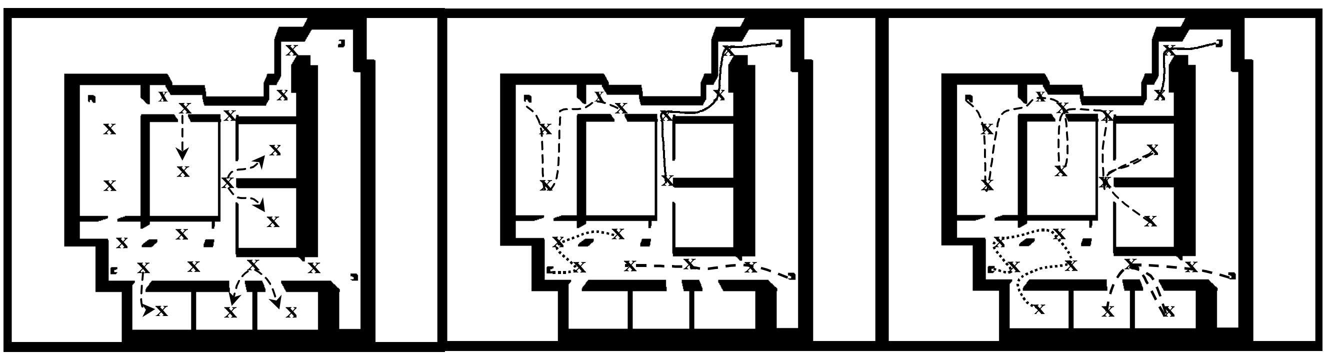

Figure 10: Results from dynamic re-negotiations.

6. SUMMARY

We have outlined the design of a colony of mobile robots capable of providing maps from the interior of urban structures. Traditionally, new paradigms are brittle and may not convey any advantage when deployed outside of the laboratory environment [9]. Our primary mission is to exploit multiple robots as a means to develop capability that is robust to individual failure much in the same way as traditional military units operate. These units exploit redundancy such that critical capability is preserved when losses are incurred; consequently, they can continue to operate when faced with multiple casualties.

Our architecture is unique in that task execution is only loosely coupled to the existence of individual robots. This is an important technological leap that must be developed in order to place robot colonies into extreme environments to perform dangerous tasks. Colony performance can be enhanced by the development of accelerated learning through "on-line" and instantaneous exchange of experiences. The ubiquitous connectivity within a colony provides the infrastructure to promote the exchange of successful skills and behaviors in a trivial fashion. Through our research, apprenticeship in established robot colonies will be measured in "seconds of download time."

Secondary goals of this effort are to equip this software architecture such that new applications can be readily programmed. As a first order enabler, the design of this software infrastructure has been abstracted from the task. In the worst case, new applications would require coding with provided application programmer interfaces (APIs). To eliminate this requirement, we are also pursuing the construction of task archives and behavioral seeding technologies that would enable robots to learn new tasks by downloading "seeds" from an archive. These seeds would grow into fully functional, perhaps even optimal, task level capabilities within our fertile distributed learning infrastructure. Once matured, this functionality would be stored in a

[3] A movie of this evolution is available at http://www.frc.ri.cmu.edu/projects/colony/frcworld.shtml

task archive for future mission capability. Thus, a process of iterative refinement for colonies of robots would be available. This archive of robot-derived experience, knowledge, and data is a first step in creating accelerated cultural learning and knowledge acquisition for robots, a distinctly human phenomenon.

7. FUTURE WORK

The results described above provide the building blocks for geometric environment reconstruction from multiple robots. Many issues must be addressed before they can be assembled into an effective mapping system. First, there are many choices of parameters for rectification and stereo – for example, size of matching window or interpolation mode – and extensive experiments are needed to determine the optimal choices in a typical environment. Secondly, there is the need to incorporate maps from multiple pairs of robots into composite maps. The multi-map approach will support addressing of maps from different locations, but registration of individual maps is necessary to ensure coherence of the representation throughout the environment. Map integration can be accomplished by one of two means. First, the individual 3-D maps from robot pairs can be iteratively modified to yield an optimal combined map. An alternative that we have used successfully is to find correspondences between the individual maps constructed from pairs of robots and to estimate the relative poses between the maps based on the correspondences. A global optimization approach is then used to find the optimal set of poses and correspondences between maps. This is the approach we proposed in [36]. We will apply both approaches to the multi-robot map building problem and will design the integration system based on the evaluation of those approaches. In addition, we are examining the following:

Interactive Queries: An operator-designated region, line, or point of interest from an approximate floor plan is used to guide retrieval and reconstruction from local stereo matching.

Generating Floor Plans: A cross section from an integrated map consisting of all the points from all the cooperative stereo maps taken in that location.

View-Based Rendering: generating realistically rendered views from arbitrary positions.

ACKNOWLEDGEMENTS

This research was sponsored in part by DARPA under contract "Cognitive Colonies" (contract number N66001-99-1-8921, monitored by SPAWAR). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the U.S. Government. The authors thank the additional members of the Cognitive Colonies group for their valuable contribution: Charles Smart, Brian Frederick, Vanessa De Gennero, and Dean Boustead.

REFERENCES

1. Brig. General J. R. Groves, Jr. "Operations in Urban Environments" *Military Review*, August 1998.
2. MOUT ACTD Program Overview. http://mout.actd.org
3. Lt. Gen. J. Rhodes, "Jane's Defense Weekly Interview" *Jane's Defense Weekly*, Feb 4, 1998.
4. MOUT ACTD Fact Sheet. http://mout.actd.org
5. MOUT ACTD Operational Requirements. http://mout.actd.org/req.html
6. SFC L. Lane, "Robots out Front" *Soldier's Online*, April 1995.
7. D. G. Knichel, "Robotics Insertion Technology" May 1997.
8. M. B. Dias and A. X. Stentz, "A Free Market Architecture for Coordinating Multiple Robots" CMU-RI-TR-99-42, Carnegie Mellon University, December 1999
9. L. W. Grau, "Bashing the Laser Range Finder with a Rock" *Military Review*, May 1997.
10. Arkin, R. C., "Cooperation without Communication: Multiagent Schema-Based Robot Navigation" *Journal of Robotic Systems*, Vol. 9, No.3, pp. 351-364, 1992.
11. Arkin, R. C., Balch, T., "AuRA: Principles and Practice in Review" *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 9, No. 2/3, pp.175-188, 1997.
12. Brooks, R. A., "Elephants Don't Play Chess" *Robotics and Autonomous Systems*, Vol. 6, pp.3-15, 1990.
13. Brumitt, B. L., Stentz, A., "Dynamic Mission Planning for Multiple Mobile Robots" *Proceedings of the IEEE International Conference on Robotics and Automation*, No. 3, pp. 2396-2401, 1996.
14. Golfarelli, M., Maio, D., Rizzi, S., "A Task-Swap Negotiation Protocol Based on the Contract Net Paradigm" Technical Report CSITE, No. 005-97, 1997.
15. Jensen, R. M., Veloso, M. M., "OBDD-based Universal Planning: Specifying and Solving Planning Problems for Synchronized Agents in Non-Deterministic Domains" *Lecture Notes in Computer Science*, No. 1600, pp. 213-248, 1999.
16. Johnson, N. F., Jarvis, S., Jonson, R., Cheung, P., Kwong, Y. R., Hui, P. M., "Volatility and Agent Adaptability in a Self-Organizing Market" *Physica A*, Vol. 258, No. 1-2, pp. 230-236, 1998.

17. Lux, T., Marchesi, M., "Scaling and Criticality in a Stochastic Multi-Agent Model of a Financial Market" *Nature*, Vol. 397, No. 6719, pp. 498-500, 1999.
18. Mataric, M. J., "Issues and Approaches in the Design of Collective Autonomous Agents" *Robotics and Autonomous Systems*, Vol. 16, pp. 321-331, 1995.
19. Pagello, E., D'Angelo, A., Montsello, F., Garelli, F., Ferrari, C., "Cooperative Behaviors in Multi-Robot Systems through Implicit Communication" *Robotics and Autonomous Systems*, Vol. 29, No. 1, pp. 65-77, 1999.
20. Parker, L. E., "ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation" *IEEE Transactions on Robotics and Automation*, Vol. 14, No.2, pp. 220-240, 1998.
21. Sandholm, T., "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations" *Proceedings, Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pp. 256-262, 1993.
22. Sandholm, T., Lesser, V., "Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework" *Proceedings, First International Conference on Multiagent Systems (ICMAS-95)*, pp. 328-335, 1995.
23. Schneider-Fontán, M., Mataric, M. J., "Territorial Multi-Robot Task Division" *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 5, 1998.
24. Schneider-Fontán, M., Mataric, M. J., "A Study of Territoriality: The Role of Critical Mass in Adaptive Task Division" *Proceedings, From Animals to Animats 4, Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*, MIT Press/Bradford Books, pp. 553-561, 1996.
25. Schwartz, R., Kraus, S., "Negotiation On Data Allocation in Multi-Agent Environments" *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp.29-35, 1997.
26. Shehory, O., Kraus, S., "Methods for Task Allocation via Agent Coalition Formation" *Artificial Intelligence Journal*, Vol.101, No:1-2, pp.165-200, May, 1998.
27. Smith, R., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver" *IEEE Transactions on Computers*, Vol. C-29, No. 12, December, 1980.
28. Švestka, P., Overmars, M. H., "Coordinated Path Planning for Multiple Robots" *Robotics and Autonomous Systems*, Vol. 23, No. 4, pp. 125-133, 1998.
29. Tambe, M., "Towards Flexible Teamwork" *Journal of Artificial Intelligence Research*, Vol. 7, pp. 83-124, 1997.
30. Veloso, M., Stone, P., Bowling, M., "Anticipation: A Key for Collaboration in a Team of Agents" Submitted to the *3rd International Conference on Autonomous Agents*, pp. 1-16, 1998.
31. Wellman, M., Wurman, P., "Market-Aware Agents for a Multiagent World" *Robotics and Autonomous Systems*, Vol. 24, 1998.
32. Zeng, D., Sycara, K., "Benefits of Learning in Negotiation" *Proceedings of the AAAI National Conference on Artificial Intelligence*, pp.36-41, 1997.
33 P. Beardsley, A. Zisserman, D. Murray, "Sequential Updating of Projective and Affine Structure from Motion" *International Journal of Computer Vision*, Vol. 23, No. 3, 1997.
34 P. Chang, M. Hebert, "Omnidirectional Structure from Motion" *Proc. Workshop on Omnidirectional Vision*, IEEE Press, 2000.
35 O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint* MIT press, 1993.
36 D. Huber, O. Carmichael, M. Hebert, "3-D Map Reconstruction from Range Data" *Proc. IEEE International Conference on Robotics and Automation*, 2000.
37 J.J. Koenderink, A.J. van Doorn, "Representation of Local Geometry in the Visual System" *Biological Cybernetics*, 55:367-375, 1987.
38 S. Laveau, O.D. Faugeras "3-D scene representation as a collection of images" *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1994.
39 P. Montesinos, V. Gouet, R. Deriche, D. Pele, "Differential Invariants for Color Images" *Proc. 14th International Conference on Pattern Recognition*, 1998.
40 M. Pollefeys, R. Koch, L. Van Gool, "A Simple and Efficient Rectification Method for General Motion" *Proc. IEEE International Conference on Computer Vision*, Corfu, 1999.
41 S. Roy, J. Meunier, I. Cox, "Cylindrical Rectification to Minimize Epipolar Distortion" *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
42 C. Schmid, R. Mohr, "Local Grayvalue Invariants for Image Retrieval" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530-534, 1997.
43 C. Schmid, R. Mohr, "Comparing and Evaluating Interest Points" *Proc. International Conference on Computer Vision*, 1997.
44 P. Sturm and R. Hartley, "Triangulation" *Computer Vision and Image Understanding*, 68(2), 1997.
45 Z. Zhang, R. Deriche, O. Faugeras, Q.T. Luong, "A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry" *Artificial Intelligence Journal*, Vol. 78, 1995.