A Pragmatic Approach to Robot Rescue: The Keystone Fire Brigade

Jacky Baltes and **John Anderson**
Autonomous Agents Laboratory
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
Canada R3T2N2
Email: jacky,andersj@cs.umanitoba.ca

Abstract

The Keystone Fire Brigade is a robotic rescue team that has competed in the 2002 competitions in both Fukuoka (RoboCup) and Edmonton (AAAI). The key elements of our approach are emphasis on autonomy, vision-based computing, and implementation on inexpensive robot bases (e.g., toy cars). This paper describes our motivations in developing the Keystone Fire Brigade, and describes the key elements of the design: the hardware employed and the visual processing algorithms used for localization and victim identification. We also describe our experiences in the test domains.

Introduction

Robotic rescue was conceived as a challenge problem that would provide a motivation for research in artificial intelligence, allow performance comparison between approaches, and encourage development in a useful application area (Kitano *et al.* 1999). In the time the competitions at RoboCup and AAAI have been operating, we have seen a varied array of approaches both shaped by and shaping the ongoing development of a set of standards for the competition environment (Jacoff, Messina, & Evans 2001). While this year's NIST testbed presented more than a thorough challenge to teleoperated systems (let alone fully autonomous agents) we have also seen ample evidence that real-world rescue environments can be significantly more challenging than our competition environment (Murphy, Blitch, & Casper 2002), providing even a greater challenge for the future.

This paper describes the Keystone Fire Brigade, the University of Manitoba's entry in the AAAI Robotic Rescue Competition. Our approach embodies several principles we believe will ultimately be important in successful robotic rescue problems: autonomy, a multiagent perspective, and parsimony.

While most teams in this year's competition relied on human control, we believe that ultimately, autonomous processing will be of greater importance in robotic rescue. In rescue settings, issues such as operator fatigue, lack of situational awareness of the operator, cognitive load on the operator, and the number of individuals an operator can control in real time (Casper 2002; Casper & Murphy 2002) all place limitations on human control.

While a number of presenters at this year's workshop have called for greater emphasis on combining autonomy and telecontrol, the current lack of success in this and other areas is largely due to the primitive state of autonomous control. We believe the best way to improve this is to explore the possibilities of dealing with the domain autonomously first. Teleoperation may be added to the system once the *hard* problems in autonomous operation have been tackled. This will also ultimately serve to uncover an appropriate balance between teleoperated and autonomous control.

Many problems are naturally amenable to solution through a collection of distributed agents, either through power of numbers or through a distribution of varying abilities. We believe that such an approach is a natural one in robotic rescue as well, first and foremost because of the success of similar human approaches. This view leads us to design agents that are meant to operate in groups of significant size. It also ties into our third aim, parsimony: one of the major goals of multiagent systems is the better use of resources, since it is often easier and cheaper to design a large number of much simpler agents than one agent expected to tackle the same task alone.

Parsimony itself is a worthwhile goal in any robotics application: any unproven addition can add cost without benefit. We believe, like others (Balch & Arkin 1994) that the addition of any component should be carefully considered. The cost (in terms of dollars, but also in terms of reliability, robustness, and versatility of the whole system) must be balanced with the efficacy of the equipment to the improvement of overall system performance.

In a domain where a multiagent approach is being considered, this is even more crucial: if the power of the solution is in the interaction between large numbers of agents, then the simpler we can make those agents, the greater the number that can be produced. In a domain such as robotic rescue, there is also an inherent danger to the units involved in the task. Parsimony is equally important from this perspective, in that the cheaper that robots can be made, the more expendable they may be considered in a dangerous domain.

The Keystone Fire Brigade is part of an ongoing attempt to embody these principles in the application of robotic rescue. The remainder of this paper details the hardware platforms used, the vision and control algorithms employed, and provides some commentary on the experiences with the team at the 2001 AAAI robotics competition.

Description of the Robot Hardware

The Keystone Fire Brigade robots are based on the 4 Stooges ((Baltes 2002a)), a small sized RoboCup team from the University of Auckland. The small sized (also called F180) league uses robots with a maximum diameter of 18cm. These robots play a game of five versus five soccer on a 2.80m by 2.30m playing field. Unlike most teams in the F180 league, the 4 Stooges do not make use of global vision. Instead, the design relies on small, inexpensive, and low power CMOS cameras, and enough local processing power for autonomous vision and robotic control. The original design of the 4 Stooges platform was intended to be both robust and versatile. In the Keystone Fire Brigade, we employ the same physical design on two different chassis to explore possibilities in the robotic rescue domain.

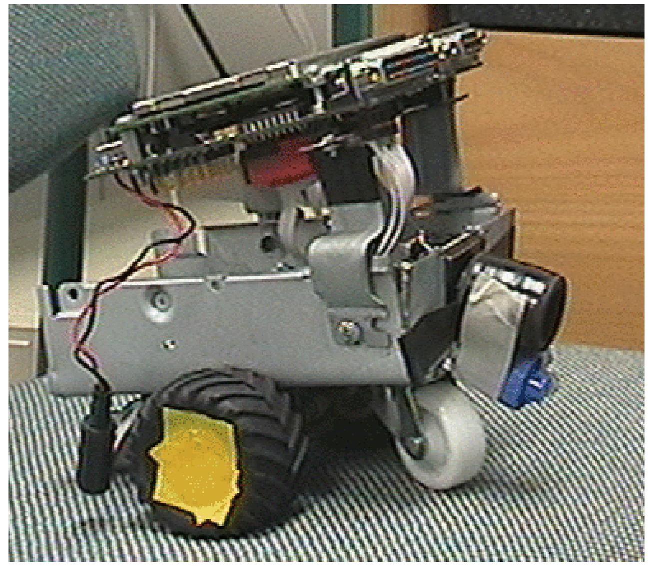
On-board processing for the robots in the Keystone Fire Brigade is provided by Thomas Bräunl's Eyebot controller (Bräunl 1999). The Eyebot controller consists of a 35 MHz 68332 processor with 2 MB of static RAM. The design is dated today, especially in terms of processing speed. However, the platform has the advantage of being comparatively cheap and also supports a direct connection between a CMOS camera and the processor itself. Furthermore, the Eyebot controller provides the necessary interface to connect motors, servos, gyroscopes, and many other sensors directly to the controller.

For the 2002 competition, we have employed this configuration on two different bases. The first is shown in Figure 1: this is the body of Curly, one of the original 4 Stooges. This base uses a commonly available Tamiya twin gearbox with off-road tires for locomotion, and has a swivelling front wheel. While being custom-assembled, the individual parts are cheap, affording the potential for significant numbers of these on a team.

The other base we have employed is a remote-controlled car model from Nikko, a Hummer (Figure 2). While not possessing a true four wheel-drive, it affords a higher ground clearance and significantly greater stability over rough terrain than the other base.

While neither of these bases would be suitable for real robotic rescue, they are sufficient for locomotion in the yellow zone of the NIST testbed (Jacoff, Messina, & Evans 2001), and can demonstrate some capability in the orange and red zones. Our intent in this year's competition was not to function perfectly in the orange and red zones, but to demonstrate the capabilities of autonomous performance and victim identification using vision for sensing. As such, we were more concerned with appropriate sensing and control algorithms than with a heavy-duty robotic base.

Figure 1: Curly: Robot platform using a simple twin gear box



Visual Processing in the Rescue Domain

The use of cheap platforms as those shown in Figures 1 and 2 has both advantages and disadvantages. While they make robotic agents reasonably cheap to construct and allow a greater number to ultimately be deployed, the mechanical designs themselves limit the utility of these agents in the more difficult sections of the RoboCup Rescue domain.

In future, we hope to overcome these limitations through collaboration with Dr. Nadir Ould-Kheddal's laboratory at Temasek Polytechnic, Singapore. Currently though, our focus is on overcoming the limitations of inexpensive platforms through the development of good autonomous vision algorithms suitable for use in embedded devices.

The greatest challenge on the current platform is to deploy these algorithms and deal with the low frame rates achievable by the Eyebot controller. We are looking in future at the Sharp Zaurus, with a 200 MHz processor, as an improved replacement for the Eyebot in this respect.

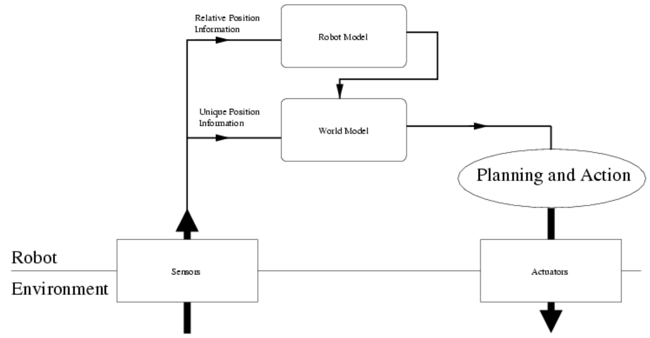
The following subsections describe the use of visual information in dealing with important elements of the robotic rescue problem.

Ego Motion Estimation

One of the biggest challenges in any robotic environment is ego-motion-estimation using visual information.

Figure 2: Hummer: Robot platform using a Toy Car base

Figure 3: System Design



This is true for robotic soccer, as the field size in the latter limits the amount of visual information useful for ego-motion estimation. Ego motion estimation through visual means is even more difficult, however, for robotic rescue: while there is a larger amount of visual information, much of this information is confusing or conflicting. In addition, there is much less structure that can be relied upon in the rescue environment. In soccer, for example, despite the smaller field size there is a great deal of structure in what can be seen. It is possible to employ ego motion estimation based on visual sighting of the boundaries surrounding the field, for example (Baltes 2002b). There are no such structures that can be relied upon in robotic rescue.

Because of difficulties of employing vision for ego motion estimation, most teams rely on odometry readings as a basis for robot localization. This solution works well in simple domains that use a hardwood floor or a flat carpet. The utility of odometry, however, is limited in the NIST robotic rescue domain: flooring is irregular and papers are scattered to promote slippage and the resulting compounding errors in odometric data. Like many elements of robotic control, this is another case where approaches successful in simple domains have proven far less successful in the NIST testbed.

To a robot using the cheap platforms that we have employed, the difficulties in the NIST domain are no worse than those encountered in any other domain: the gears and drive employed on a cheap platform are simply not accurate enough for use in gathering reliable odometric data.

For these reasons, the Keystone Fire Brigade does not use odometry. Rather, the robots use optical flow algorithms to substitute for odometry and to provide relative motion information. In addition, the optical flow information is used to update an internal robot model which is used as a last resort in case neither absolute localization, nor relative motion via optical flow can be detected (Figure 3).

Ego motion detection in our approach is done by looking for differences between received video frames. This is an exceptionally difficult task in robotic rescue for more reasons than the lack of structure mentioned above. In robotic rescue, the real world provides many different feature variations with shadow and light variations across these: far more than occurs across the expanse of playing field in robotic soccer. These variations are different enough between individual frame captures that the robot often assumes that it is in a different location when its actual location has not changed or has changed only slightly. In addition, this misclassification can happen as a result of external motion in some part of the image, and is compounded by the busy visual elements (small wallpaper patterns, floor tiles, etc.) used in the NIST testbed.

Since our robot does not include bump sensors or stall sensors, one of the tasks of the vision system is to determine if the robot is stuck and to suggest a different course of action in this case. The vision system must be able to take an image and detect a lack of motion, while attempting to take into account the likelihood of false motion detection described above.

To determine if the robot is blocked or otherwise stuck in one position, the image is broken up into 16 equal sized sub-images. Of these, only the bottom 8 sub-images need to be considered - everything else is further away and would not be likely to provide useful feedback regarding the motion of the robot. The system then computes the differences between the current and the previous image for each quadrant. The colour difference is defined as the sum of the absolute value of the differences in the red, green, and blue channels. If the difference in a sub-image is above a threshold, the quadrant is marked. If there are more than eight marked sub-images and the motors were turned on in the previous time step, than the system signals that the robot is stuck or blocked. We break the image into sub-images to allow for localized changes due to the motion of some other agent or other external motion in the image, to try to limit the number of false positives.

There are many possibilities for improving on this scheme. For example, by using prior knowledge about the scene (e.g., detecting likely candidates for straight lines), this simple method could be significantly improved and better motion estimates could be derived from the visual information.

However, this simple scheme has shown that it was sufficient during this year's competition. The Keystone Fire Brigade reliably detected when the robot was stuck and managed to free itself in all cases.

Obstacle Detection

Given that the difficulties of local vision-based scene interpretation, we felt that it would be useful to experiment with secondary obstacle avoidance mechanism to supplement the situations where the uncertainty of vision recognition was too high. Given the small size of the robots, as well as issues in an unstructured domain with proper reception of sonar pings, we felt that laser rangefinding would be of use. Following the pragmatic and inexpensive approach that we are advocating, rather than attempting to make use of expensive commercial laser rangefinders, we have adapted a simple off-the-shelf laser pointer for use as a rangefinder on one of these platforms. Figure 4 illustrates the 4 Stooges platform with the pointer mounted.

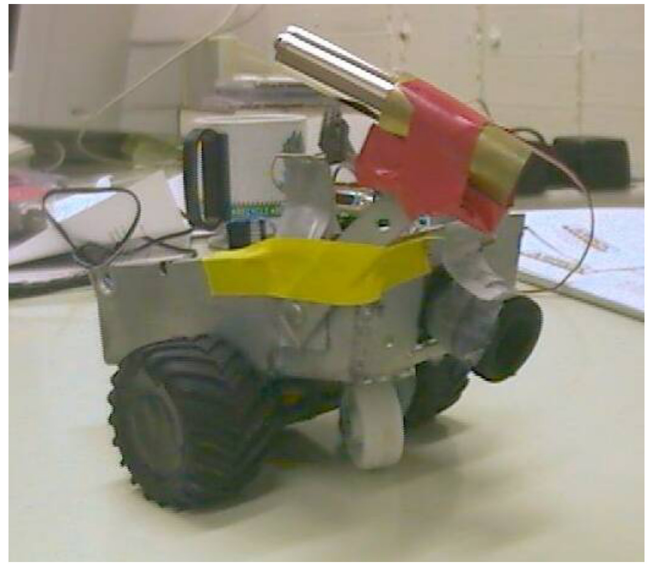
The laser pointer allows us to determine the distance between the robot and the closest obstacle in front of it. Image processing searches for the laser dot in the image: after calibrating the camera angle, the distance to an obstacle in front of the robot can be determined by the y-coordinate of the laser point in the captured image. We found the laser worked quite well as a supplement to the image processing described above.

We believe this mechanism may also be useful in supplementing the accuracy of our vision-based ego motion estimation, by providing a known, calibrated point within the image. Since there are issues of false movement recognition through lighting changes and other image differences, referencing the neighbourhood of the laser point image can be used as a further verification of movement (or lack of movement) by the robot, and help eliminate these false positives in recognizing blockage. We feel this can also be used for more accurate motion estimation using similar techniques: the point provides an absolute reference from which movement of lines (for example) in the image can be measured. Moreover, measuring distance movement of a calibrated laser point would also be possible in the dark, which would have been useful in the entrance to the yellow arena in this year's testbed. While not allowing detailed navigation, it would have allowed the robot to find and possibly follow walls in the dark to a better degree than our vision system did.

Map Generation

Since there is little a priori information about the map of the environment as well as no reliable odometry information available, it is difficult to generate a map.

Figure 4: Platform of Figure 1 with Mounted Laser Pointer



Errors in the map not only lead to misdirection to found victims, but may also prevent the robot from exploring parts of the arena entirely. That is, the map may incorrectly indicate that there is no feasible path between two areas in the arena, that an area has been previously searched, or that the area simply does not exist.

To construct a map, we first attempt to use a random walk algorithm to cover the arena. Our random walk algorithm simply causes the robot to drive straight for a random amount of time, then randomly turn into another direction and continue. This algorithm is very robust since the robot will not be confused by errors in the map.

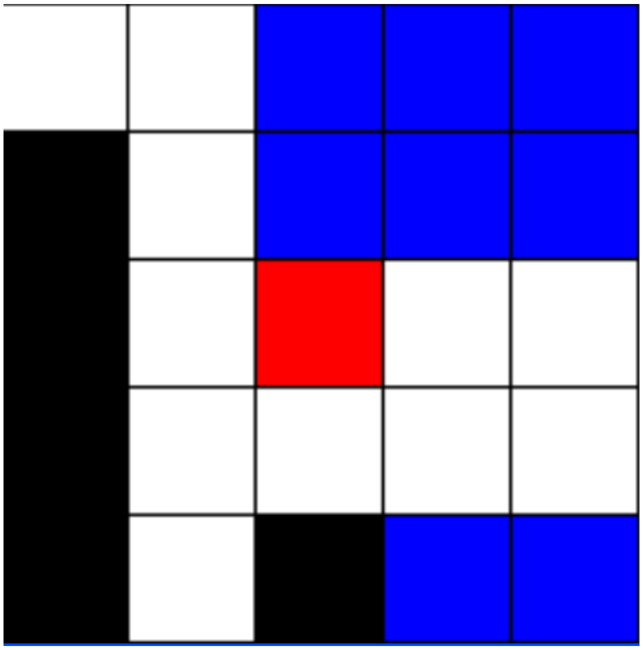
However, a purely random walk does not cover the area very efficiently. In our early experiments, the robot would make numerous random moves, but all limited to a small area.

We therefore extended the random walk algorithm to a hybrid algorithm that tried to combine the strength of both map generation with a random walk. This hybrid approach constructs a small local map, which allows the robot to efficiently search the surrounding area, and then uses random movements between areas.

The robot begins with a single small map (encompassing a 1.5m by 1.5m area around the robot) and starts to fill in any obstacles it detects. A sample of the local map is shown in Fig. 5. The robot is always in the center of the local map. In this example, the robot has detected an obstacle on the left side (in black), and marked the top and bottom right area as still unexplored. The white squares represent open areas.

We plan in the future to extend this work to include a case-based reasoning system that employs typical sensor readings to identify areas and to connect them via

Figure 5: Local Map



topologial paths.

Detection of Victims

The NIST domain includes a number of simulated victims, both whole mannequins and body segments. These victims provide a number of features which allow their detection, including their shape and colour, motion, sound, and heat.

In the 2001 competition, it was evident that visual information alone was sufficient to detect most victims since the human operators had no problem picking out the victims. One goal of this research is to develop methods that can perform similar visual identification in this environment without the help of the human operator.

This problem (scene interpretation) is an extremely difficult one. Many researchers have tried to develop scene interpretation systems but have failed so far. The difficulty is compounded for the 2002 competition through greater occlusion of skin on victims (through dirt, paint, etc.). This is a positive step in making the test environment more reflective of a real urban search and rescue environment. As was demonstrated in an invited talk at AAAI-02, however (Murphy 2002), this degree of skin occlusion is still not nearly what could be expected in a real disaster setting.

Our victim detection approach uses both colour as well as shape information. Flesh colored spots are marked as possible victim locations (these algorithms were trained beforehand on flesh patches of team members in the lighting used in the test arena).

We have developed a 12 parameter colour model which uses Red, Green, and Blue as well as three difference channels: Red - Green, Red - Blue, and Green - Blue. The differences are included in this parameter model because of their tendency to remain relatively constant in different views of objects of the same colour despite of lighting variations over a field. This approach is the same used in our Doraemon vision server (Anderson & Baltes 2002) and has proven itself in years of robotic soccer competition.

Currently, we use a simple blob detection scheme. The system signals that it has found a victim by calculating the apparent size and aspect ratio of a skin coloured blob. If these parameters are within limits, the system signals the detection of the victim by performing a series of 360 degree turns at the current location. It then continues to search the environment.

We plan to use a number of “typical” poses of human appendages hands, head, and feet to support pattern matching in the future.

Discussion

Since this was our first year in the competition, we selected the following goals for the Keystone Fire Brigade:

1. Show that the robots are able to determine correctly if they have visited a specific location previously or not. The main focus of our research has been this problem. The robots should avoid previously seen places and seek novel or unknown places in the environment.
2. Detect victims through interpretation of visual information alone. This detection uses colour hues as well as edge information to determine if the robot is sufficiently close to a victim or not. This part of the system will be evaluated by putting the robot close to a victim and measuring the recognition rate of the system.
3. Show that the mechanical design of the robots is capable of navigating the yellow and orange zones of the NIST playing field. This will be investigated by field trials. Some field trials will also be held in the red zone, but the terrain in the red zone may well require tracked or legged robots.
4. Move at systematically/randomly through the NIST playing field and cover an appropriate part of the playing field. The robots have not been programmed with a map of the playing field as this would be impossible to do in a real urban search and rescue scenario. Therefore, the robots will have to map the environment themselves and incrementally develop a map.

In the end we were more satisfied with the degree to which some of these goals were accomplished than others. Where there was less success than expected, it was mainly due to the positive steps taken to make this year’s NIST playing field more difficult than previous competitions. In particular, the placement of a tarpaulin over the entrance to the yellow zone proved a significant difficulty for our team. Working entirely with vision is a detriment in an area where steps have been taken to significantly darken it. The original robotic platforms carried no local lighting, but this was hastily added during breaks between field trials. Even then, dead reckoning was required to make an initial entrance, which itself did not go as well as desired. In future, we intend to supplement our visual algorithms with sensory elements not dependent on lighting level (such as the laser mentioned previously). In terms of the addition of the tarpaulin itself, we believe this is a very positive change to the nature of the arena: we do need to challenge all the sensing abilities of teams significantly, and vision is no exception. Anything that makes a testbed more like the real world it is intended to reflect can only ultimately be a positive step toward development of equipment to better function in the real world.

As far as the mechanical design of the robots was concerned, we were satisfied with the ability of the platforms to negotiate the yellow arena. Navigation in the orange arena would have been acceptable as well, but for the necessity of a climb into the orange arena that was beyond the height of our robots.

Overall, we were very pleased with the learning experience provided by this year's competition. Our intent was to aim for some basic goals and gather anecdotal evidence for how well the Keystone Fire Brigade is able to achieve these. It is non-trivial to develop a quantitative measure for how well a robot is able to perform the tasks mentioned above: while a score is available, that score must also be interpreted under the qualitative conditions that resulted in it. We are happy with the anecdotal evidence we came away with, and like others in the competition, we hope to put this evidence to good use in our plans for next year's entry.

References

Anderson, J., and Baltes, J. 2002. *The Doraemon User's Guide.* Department of Computer Science, University of Manitoba, Winnipeg, Canada.

Balch, T., and Arkin, R. C. 1994. Communication in reactive multiagent robotic systems. *Autonomous Robots* 1(1):27–52.

Baltes, J. 2002a. 4 stooges. In Birk, A.; Coradeschi, S.; and Tadokoro, S., eds., *RoboCup 2001: Robot Soccer World Cup V.* Berlin: Springer-Verlag. 559–562.

Baltes, J. 2002b. Localization for mobile robots using straight lines. In *Proceedings of Seventh International Conference on Control, Automation, Robotics and Vision.*

Bräunl, T. 1999. Eyebot: A family of autonomous mobile robots. In *Proceedings of the 6th International Conference on Neural Information Procoocessing (ICONIP99)*, 645649a.

Casper, J., and Murphy, R. 2002. Workflow study on human-robot interaction in usar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, 1997–2003. Washington, DC: IEEE.

Casper, J. 2002. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. Master's thesis, University of South Florida.

Jacoff, A.; Messina, E.; and Evans, J. 2001. Experiences in deploying test arenas for autonomous mobile robots. In Messina, E., and Meystel, A., eds., *Proceedings of the 2nd Performance Measures for Intelligent Systems Workshop (PerMIS)*, NIST Special Publication 982, 87–95. National Institute of Standards and Technology.

Kitano, H.; Tadokor, S.; Noda, H.; Matsubara, I.; Takhasi, T.; Shinjou, A.; and Shimada, S. 1999. Robocup-rescue: Search and rescue for large scale disasters as a domain for multi-agent research. In *Proceedings of the IEEE Conference on Systems, Men, and Cybernetics*, volume 6, 739–743. IEEE.

Murphy, R.; Blitch, J.; and Casper, J. 2002. Aaai/robocup-2001 urban search and rescue events: Reality and competition. *AI Magazine* 23(1):37–42.

Murphy, R. 2002. Robot-assisted urban search and rescue at the wtc: Where's the ai? Invited Talk at the 14th Innovative Applications of Artificial Intelligence Conference (IAAI-02), Edmonton, AB.