Autonomous Stair-Hopping with Scout Robots

Sascha A. Stoeter, Paul E. Rybski, Maria Gini, Nikolaos Papanikolopoulos

Center for Distributed Robotics
Department of Computer Science and Engineering
University of Minnesota, Minneapolis, U.S.A.
{*stoeter, rybski, gini, npapas*}*@cs.umn.edu*

Abstract

Search and rescue operations in large disaster sites require quick gathering of relevant information. Both the knowledge of the location of victims and the environmental/structural conditions must be available to safely and efficiently guide rescue personnel. A major hurdle for robots in such scenarios is stairs.

A system for autonomous surmounting of stairs is proposed in which a Scout robot jumps from step to step. The robot's height is only about a quarter step in size. Control of the Scout is accomplished using visual servoing. An external observer such as another robot is brought into the control loop to provide the Scout with an estimation of its pose with respect to the stairs. This cooperation is necessary as the Scout must refrain from ill-fated motions that may lead it back down to where it started its ascend. Initial experimental results are presented along with a discussion of the issues involved.

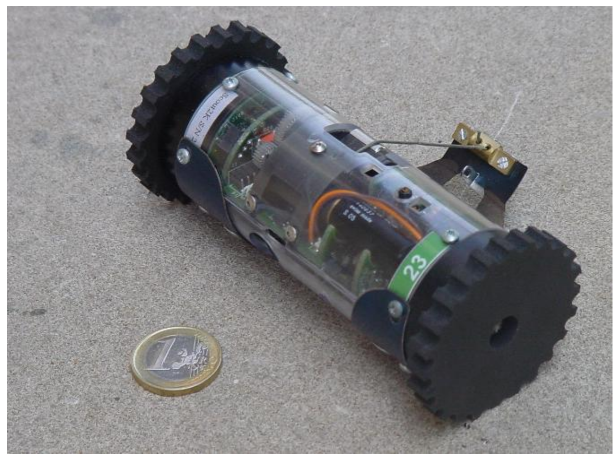


Figure 1: *Closeup of the salient Scout robot shown in comparison to a 1 Euro coin. The spring foot is located on the miniature robot's far side.*

1 Introduction

Stairs are ubiquitous in urban environments. They provide means to overcome height differences that occur naturally (e.g., the slope of a hill) or artificially (e.g., different levels in a building). Humans can easily make use of these installations.

Robots in urban settings often encounter stairs during their passage through the environment. Search and rescue operations in urban environments, as well as exploration and surveillance tasks, necessitate stair surmounting: help must reach a point quickly or no other paths are available due to, for instance, collapsed walls. The vast majority of robots, however, is ill-equipped to handle this situation. Robots typically lack the necessary shape, power, intelligence, or weight. The insufficient traditional approach is to flag stairs as a dead-end and continue searching for an alternative route.

The Scout robot depicted in Fig. 1 is well-suited for these kinds of tasks. Its salient feature, the jump, allows it to overcome obstacles other robots cannot. While its reduced size permits access to tight areas, it also limits the integration of powerful processors required for autarky. The present approach of teleoperating does not scale well and is inappropriate for large numbers of deployed robots.

This work proposes a support framework for urban rescue missions in which a human operator is assisted by autonomous behaviors. While a Scout robot autonomously approaches and surmounts flights of steps by jumping one step at a time, the human operator is freed to focus attention somewhere else. The control loop is closed through an assisting agent that provides sensor information for pose approximation.

The following section discusses relevant work. Section 3 gives an overview of the Scout robot. The software architecture to make the system work along with a prototype of a wearable user interface are presented in Section 4. Section 5 presents the proposed solution and Section 6 describes initial results. Future work is outlined in Section 7.

2 Related Work

Stair surmounting has received some attention for applications in hazardous urban environments. The most prominent scenario is sending a large robot into a nuclear reactor core for cleanup and maintenance tasks. The robot described by Arai *et al.* surmounts stairs using specialized wheels that act similar to gear wheels on the steps [2]. Littman *et al.* present a system consisting of three connected units that can lift each other from step to step [9]. Small robot designs that can surmount stairs include different combinations of legs and wheels [1, 4, 5] or self-reconfiguration [13, 17]. All of these systems are similar in that the robots are teleoperated by human operators for the entire mission.

The authors are aware of only a single piece of work that attempts surmounting stairs autonomously using closed-loop control [18]. Xiong *et al.* utilize an Urban II from iRobot equipped with tracks and an onboard vision system that even worked when shadows were cast on the steps. The system cannot handle stairs with steep slopes.

Cao *et al.* provide a nice overview of research in the cooperative, multi-robot domain [6]. Architectures for robot cooperation and task-decomposition are plentiful [3, 10]. In ACTRESS [8], for instance, cooperation is created by explicit requests for help from a robot that lacks a function required for executing its current task.

3 Hardware Description

The Scout was developed by the Center for Distributed Robotics at the University of Minnesota [14]. Scouts have a cylindrical shape, 40 mm in diameter and 115 mm in length, and weigh about 200 g. Each Scout moves with a unique combination of locomotion types including rolling using the wheels mounted on both ends of its body and jumping using a spring mechanism often simply referred to as a foot. Rolling allows for efficient traversal of smooth surfaces, while jumping allows Scouts to operate in uneven terrain and pass over obstacles as high as 35 cm, depending on the strength of the spring foot. The trajectory of the jump is determined solely by selecting an angle; the force of the jump cannot be controlled for a given mounted spring.

The electronics of Scouts include microcontrollers, transmitters, magnetometers, tiltometers, and shaft encoders. In addition, they have a modular sensor payload that includes a miniature video camera (with an optional pan-tilt unit) and video transmitter or a microphone. A Scout can roll for several hours while transmitting video and can execute about 100 jumps before exhausting a set of batteries.

Scouts are accompanied by a much larger Ranger robot. A Ranger is capable of traveling longer distances and over rough terrain. Equipped with a magazine and a spring-based delivery mechanism, it can deliver up to ten Scouts into a target area. The Ranger's powerful onboard computer can be used to coordinate Scouts and relay status information.

4 Control Architecture

As a Scout's own processor is extremely limited because of size and power constraints, intelligence for control decisions must be provided externally. Stair-hopping with Scouts involves visual servoing [7] that necessitates auxiliary hardware for image analysis. A computer equipped with a framegrabber is required to run image-processing algorithms. This could be either a Ranger or another machine within reception range of the Scout's analog video transmission.

To overcome the Scouts' limitations and to tie multiple computers together for complex missions, a distributed software architecture has been developed that supports the transparent integration of remote resources [15]. It takes a functional view of missions. Instead of fitting missions to whole robots that are seen as individuals with control over their own bodies, all hardware resources, including the robots, are partitioned into finely grained resources that can be requested by functional components. The architecture consists of the four subsystems shown in Fig. 2:

- The *User Interface* subsystem provides run-time control and feedback. A possible wearable user interface is shown in Fig. 3. A monitor to view video from a Scout or Ranger is attached to the operator's sleeve with velcro. The personal digital assistant (PDA) located to its left can be used to teleoperate a Scout.
- *Mission Control* hosts prioritized functional components that, together, make up a solution for a mission. Components are partially-ordered and can execute in parallel.

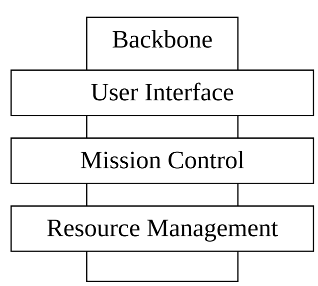


Figure 2: *The core of the distributed software architecture is comprised of four subsystems.*

Figure 3: *Operator wearing prototype PDA user interface. The monitor attached alongside the PDA on the sleeve is used to view the video from a Scout.*

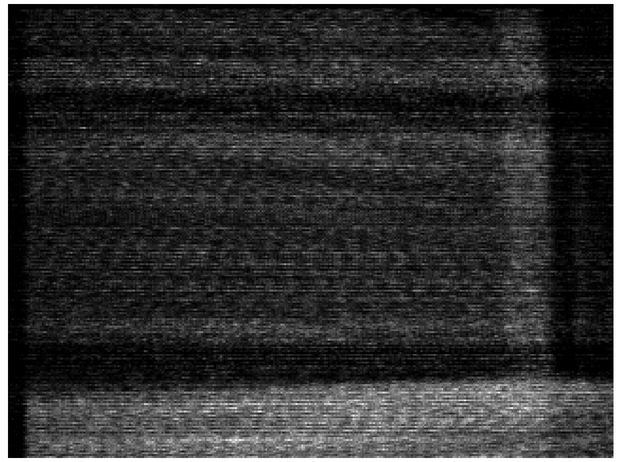


Figure 4: *A typical frame as broadcast by a Scout. Here, the robot is facing a step.*

- The *Resource Management* contains resources that can be requested by functional components. Examples of such resources include cameras and transmission frequencies.
- The *Backbone* provides basic services that connect the other three subsystems transparently over all computers available for a mission.

When a mission is started through a user interface, the top-level functional component requests all resources for which it has an immediate need. For the target application, this means requesting a Scout and any resource capable of externally observing the Scout jump up the stairs. It is important to request resources by naming required capabilities rather than a distinct piece of hardware. For instance, one should request a resource that features a camera and locomotion instead of a specific robot. This way, the resource pool can be shared economically by concurring missions.

5 Autonomous Stair-Hopping

With its unique mode of locomotion, the jump, a Scout can go places other robots cannot. However, a Scout by itself is incapable of surmounting stairs because of limited computational resources, lack of appropriate sensors, and bad position for sensing; it lacks the computational abilities to analyze a video stream, its camera is located too close to the next step to obtain a focused image, and the Scout's own shadow could create an unusable image. For instance, a Scout's video signal is rather poor as demonstrated in Fig. 4 due to noise in the analog video link. Noise is primarily introduced from the Scout's own motors and reflections from obstacles around it, including the floor. In its most devastating incarnation, noise prevents the framegrabber's tuner from synchronizing the video signal.

Additional challenges are introduced from occasionally dropped packets of the shared, low-bandwidth control data link and from the Scout's small size that makes it hard to identify the robot in an image.

Hence, for autonomous stair-hopping, a Scout requires assistance from one or more entities that add to its capabilities and help it overcome the steps.

5.1 Overview of Proposed Approach

This work proposes solving the stair-climbing problem using a Scout. While it attempts to jump from step to step, it requires external input about its performance. An observer assists the Scout by estimating its location based on visual information.

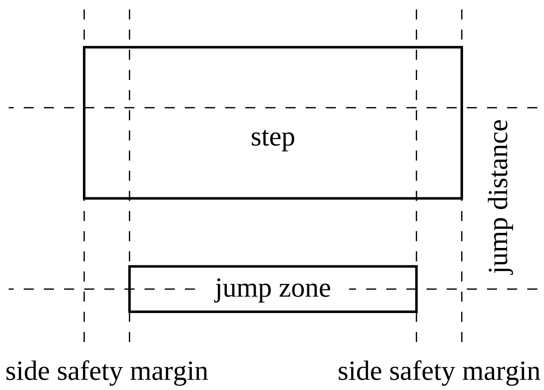
The observer must be positioned at a good location for adequate image quality [16]. First, it must have a good field-of-view that shows the steps and the Scout. Second, the observer must be located in proximity because of the limited transmission range of the video signal. The observer has to determine two properties of the Scout's location:

Scout location with respect to the step. After touchdown, the Scout typically bounces and rolls until it settles. The final location can only be predicted within large bounds. Simply assuming a likely orientation and moving in that direction to align for the next jump could result in the Scout tumbling down the stairs.

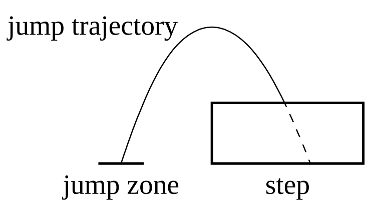
Goal reached. As soon as the Scout has reached the top of the stairs, it can continue on its own. The observer is released.

For both properties, the observer has to determine the geometric attributes of the stairs. Additionally, it has to locate the Scout and compute its pose with respect to the supporting and the next steps.

During the initial stage of autonomous stair-hopping, the Scout maneuvers toward the first step. Once there, it is guided into the jump zone. As defined in Fig. 5, the jump zone is a rectangular region from which the Scout can safely leap onto the step before it. Two safety margins of equal size on the sides of the jump zone prevent the Scout from falling off the step's sides. The edge of the step is directly under the highest point of the trajectory to maximize the available clearance during the flight.



(a) bird's eye view



(b) profile view

Figure 5: *A Scout can safely leap onto the next step when it is fully contained in the jump zone and its main axis is aligned with the step.*

Using the shaft encoders, minuscule corrections are accomplished to bring the Scout in alignment with the step. At last, a jump command is issued. The process is then repeated until all steps are taken.

5.2 Functional Decomposition

Fig. 6 depicts a solution in terms of the presented software architecture. A box stands for a functional component as defined for this architecture. If a box contains other boxes, the corresponding functional component is a composite functional component made up of simpler functional components. A labeled transition names a condition under which it is taken. Not shown are the necessary actions to secure the required resources.

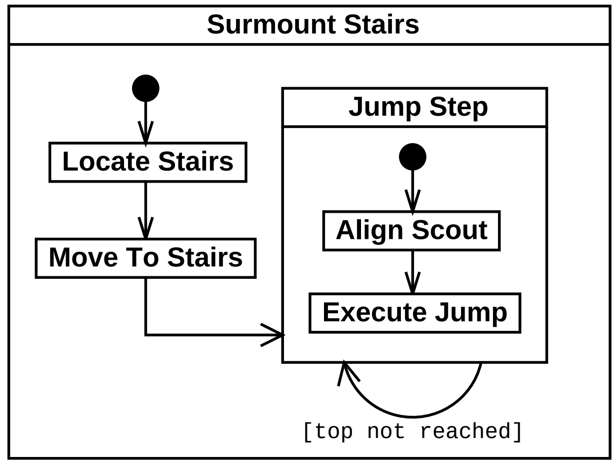
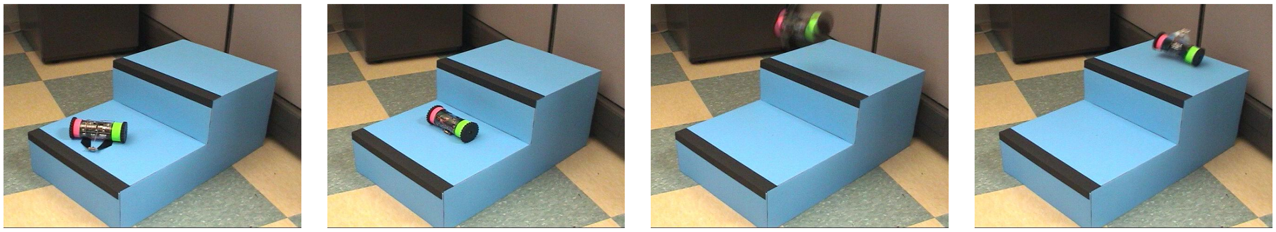


Figure 6: *Mission diagram for surmounting stairs.*

The two functional components Locate Stairs and Move To Stairs are responsible for positioning the Scout in front of the first step. A description of the composite Jump Step component follows. First, the Align Scout component is executed. Initially, the Scout is located in the observer's image and then tracked. Finally, after the Align Scout component has terminated successfully, the Execute Jump component is started to elevate the Scout to the next step. Once the top is reached, Jump Step terminates and with it Surmount Stairs. Control shifts back to the operator who can continue exploring the environment.

5.3 Implementation Details

Previous experiments in tracking Scouts have utilized pattern matching [12] and active contour models [11]. Neither approach proved adequate for the given task. In this work, the ends of the Scout's body have been marked with different colors to aide pose estimation. To keep with tradition, the left end has been given a red ring, while the one on the right end is green. The rings are approximately 1 cm wide. Each marker is tracked individually with its own region of interest. A captured image is smoothed with a 5×5 median filter and then converted from the RGB to the HLS color model. Thresholding of the image takes place with the known ranges of hue, lightness, and saturation from both markers. A circle is fit to the largest blob in the thresholded image for each region of interest.



(1) intitial position (2) before jump (3) mid-flight (4) impact

Figure 7: *Frames from a successful autonomous hop to the next step.*

Given the radii of the circles and the distance between their centers, inconsistencies as a result of tracking the wrong object can be detected. For instance, consider another Scout traveling closely past the one of interest. It then becomes possible that the tracker locks onto the wrong wheel. This, however, is discovered quickly because the circles' distances contradict with their calculated sizes. In this case, or when one marker is lost, the tracking is reseeded with an enlarged region of interest based on the location and size of the remaining marker.

The staircase is extracted using edge detection followed by a Hough transform. Given the fixed dimensions of the Scout, its appearance in the image, and its location with respect to the stairs, the steps' dimensions can be estimated. The Scout's orientation can be calculated from the position of the colored markers. While the estimation is quite accurate, improved results can be achieved with a Kalman filter.

6 Initial Experimental Results

For ensuring the practicality of experimentation, a portable staircase consisting of two steps was built. Each step is 12 cm high, 30 cm wide, and 25 cm deep. These known parameters are not used in the experiments; rather the stairs are localized by the vision algorithm exclusively. The slightly reduced height of the steps over those of regular stairs makes the problem harder, because a lower impact point increases the Scout's kinetic energy during its descent, accentuating the bouncing effect upon landing. Likewise, the greater depths of the steps provide more room for the uncontrolled setting-down process and will make it more difficult for observers placed on the ground in future versions of the system to see the Scout.

The presented system runs on a 1.6 GHz Pentium 4 under Linux. An uncalibrated camera mounted under the ceiling serves as the external observer. Images with a resolution of 640×480 pixels are captured with a framegrabber and can be analyzed in real-time (i.e., with the full NTSC frame rate of almost 30 Hz). This is about three times the used control command rate of the Scout. The Scout and the staircase always remained in the observer's field-of-view.

To approach the stairs, a path from the Scout's initial location to the first step is computed and expressed as a parametric cubic Bézier curve. The pose vector consisting of a 2D image coordinate and the orientation is compared with the path. Control commands are computed with a standard PID controller. Preliminary results from experiments show that the Scout follows the path to the first step closely. As both wheels do not start spinning at the same time in the general case, the Scout diverts from its path initially. This is corrected over time.

The sequence of frames from Fig. 7 demonstrates a successful autonomous hop from one step to another. While the alignment of the Scout with the next step is always successful, the jump does not necessarily always succeed. A jump is executed as a series of three steps: pulling the foot around the body, sensing the gravitational field and tilting the body to the set jump angle, and releasing the winch cable with subsequent unlocking of the spring foot. As control of the jump angle ceases before the Scout becomes airborne, the body can loose its tilt while the winch cable is unwound internally. Consequently, the Scout does not follow the intended trajectory. At other times, the robot bounces off the step in an unfortunate way on impact.

7 Summary and Future Work

A support framework for urban rescue missions was introduced. It consists of a PDA-based user interface and a software architecture that allows for assisting a human operator with high-level behaviors. A functional component for autonomously surmounting a flight of stairs was presented. This is achieved by controlling the robot using visual servoing based on external video input.

Future efforts will concentrate on finishing the implementation of the autonomous stair-hopping capabilities (i.e., utilizing the Scout's own camera in addition to the external one).

The color rings will be replaced by colored wheels. This will yield an increase in tracking accuracy because the markers will be farther apart. It will also allow observers on the ground to track Scouts from the side.

An interesting extension would be to analyze the jumps. The trajectories of the jumps and especially the landing position seem random, but there could be a pattern for a particular Scout and jumping angle. This would help to tackle stairs outside the observer's field-of-view. The Scout could analyze its first assisted jumps. If there actually is a pattern contained in how it lands, then it could predict future landings and free the external observer to go about other tasks.

Acknowledgments

This material is based upon work supported by Microsoft Corporation and by the Defense Advanced Research Projects Agency, Microsystems Technology Office (Distributed Robotics), ARPA Order No. G155, Program Code No. 8H20, issued by DARPA/CMD under Contract #MDA972-98-C-0008.

References

[1] H. Adachi, N. Koyachi, T. Arai, A. Shimiza, and Y. Nogami. Mechanism and control of a leg-wheel hybrid mobile robot. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, volume 3, pages 1792–1797, Piscataway, NJ, U.S.A., Oct. 1999.

[2] T. Arai, H. Yoshikawa, M. Takano, S. Ozono, G. Odawara, T. Miyoshi, K. Shimo, and T. Mikami. A stair-climbing robot for maintenance: "A MOOTY". In *Proc. of the Seminar on Remote Handling Equipment for Nuclear Fuel Cycle Facilities*, pages 444–456, 1985.

[3] S. Blum. OSCAR – Eine Systemarchitektur für den autonomen, mobilen Roboter MARVIN. In R. Dillmann, H. Wörn, and M. von Ehr, editors, *Proc. of Autonome Mobile Systeme*, Informatik aktuell, pages 218–230, Karlsruhe, Germany, Nov. 2000. Gesellschaft für Informatik, Springer.

[4] N. G. Bourbakis. Kydonas – an autonomous hybrid robot: Walking and climbing. *IEEE Robotics and Automation Magazine*, 5(2):52–59, June 1998.

[5] M. Buehler, R. Battaglia, A. Cocosco, G. Hawker, J. Sarkis, and K. Yamazaki. SCOUT: A simple quadruped that walks, climbs, and runs. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, volume 2, pages 1707–1712, Leuven, Belgium, May 1998.

[6] Y. U. Cao, A. S. Fukunaga, and A. B. Khang. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, Mar. 1997.

[7] S. Hutchinson, G. D. Hager, and P. I. Corke. "A Tutorial on Visual Servo Control". *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.

[8] Y. Ishida, H. Asama, S. Tomita, K. Ozaki, A. Matsumoto, and I. Endo. Functional complement by cooperation of multiple autonomous robots. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 2476–2481, 1994.

[9] F. Littman, E. Villedieu, and H. Chameaud. A train-like vehicle for intervention missions inside nuclear plants. *Transactions of the American Nuclear Society*, 66:551, 1992.

[10] F. R. Noreils. Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment. *International Journal of Robotics Research*, 12(1):79–98, Feb. 1993.

[11] D. P. Perrin, E. Kadioglu, S. A. Stoeter, and N. Papanikolopoulos. Localization of miniature mobile robots using constant curvature dynamic contours. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, Washington, DC, U.S.A., May 2002.

[12] P. Rößler, S. A. Stoeter, P. E. Rybski, M. Gini, and N. Papanikolopoulos. Visual servoing of a miniature robot toward a marked target. In *Proc. of IEEE International Conference on Digital Signal Processing*, Santorini, Greece, July 2002.

[13] D. Rus. Self-reconfiguring robots. *Intelligent Systems*, 13(4):2–4, 1998.

[14] P. E. Rybski, N. P. Papanikolopoulos, S. A. Stoeter, D. G. Krantz, K. B. Yesin, M. Gini, R. Voyles, D. F. Hougen, B. Nelson, and M. D. Erickson. Enlisting rangers and scouts for reconnaissance and surveillance. *IEEE Robotics and Automation Magazine*, 7(4):14–24, Dec. 2000.

[15] S. A. Stoeter, P. E. Rybski, M. D. Erickson, M. Gini, D. F. Hougen, D. G. Krantz, N. Papanikolopoulos, and M. Wyman. A robot team for exploration and surveillance: Design and architecture. In *Proc. of the Int'l Conf. on Intelligent Autonomous Systems*, pages 767–774, Venice, Italy, July 2000.

[16] K. Tarabanis, R. Y. Tsai, and A. Kaul. Computing occlusion-free viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(3):279–292, Mar. 1996.

[17] C. Ünsal and P. K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 1742–1747, San Francisco, CA, U.S.A., Apr. 2000.

[18] Y. Xiong and L. Matthies. Vision-guided autonomous stair climbing. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 1842–1847, San Francisco, CA, U.S.A., Apr. 2000.