A Region-Based Approach to Stereo Matching for USAR

Brian McKinnon, Jacky Baltes, and John Anderson

Autonomous Agents Laboratory
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, R3T 2N2, Canada
{ummcki01, jacky, andersj}@cs.umanitoba.ca

Abstract. Stereo vision for mobile robots is challenging, particularly when employing embedded systems with limited processing power. Objects in the field of vision must be extracted and represented in a fashion useful to the observer, while at the same time, methods must be in place for dealing with the large volume of data that stereo vision necessitates, in order that a practical frame rate may be obtained. We are working with stereo vision as the sole form of perception for Urban Search and Rescue (USAR) vehicles. This paper describes our procedure for extracting and matching object data using a stereo vision system. Initial results are provided to demonstrate the potential of this system for USAR and other challenging domains.

1 Introduction

This paper describes our current research into practical stereo vision for autonomous and teleoperated robots. Stereo vision as a form of perception has many benefits for autonomous intelligent systems: in ego motion detection and simultaneous localization and mapping (SLAM) for example. For a teleoperated vehicle, stereo vision can be used to assist a human operator in judging distances, marking landmarks for localization purposes, and identifying desired objects in the environment.

The domain with which we employ stereo vision is that of Urban Search and Rescue (USAR). The goal of USAR is to allow robotic vehicles to explore urban areas following a disaster, locating human victims as well as dangerous situations (e.g. gas leaks). Robotic vehicles have the advantage of being able to traverse narrow voids that would be difficult for humans to reach, and also alleviate the need to place human rescue workers in dangerous situations. While this is a extremely useful application for robotic, computer vision, and artificial intelligence technologies, it is also a important challenge problem for these areas. NIST, for example, provides a standard challenge domain at a number of locations around the world each year, allowing a practical testbed for researchers to compare approaches to various aspects of this problem.

A. Bredenfeld et al. (Eds.): RoboCup 2005, LNAI 4020, pp. 452–463, 2006.

Our own primary interests lie in artificial intelligence and computer vision. We design autonomous robots that use vision as the sole sensor to support ego-motion detection, localization, and mapping. Recognizing that it will be some time before embedded systems become powerful enough and AI technology sophisticated enough for an autonomous system to perform well in such a challenging domain, we also work to provide vision-based solutions to enhance human teleoperation.

In order to deal with the unpredictability of the USAR domain, we follow two main principles in our work. First, all solutions must make as few assumptions regarding the nature of the domain as possible. For the purposes of vision, this means that we cannot assume that any camera calibration will remain perfect throughout a run, or that we can make assumptions about the nature of lighting. Secondly, our vehicles must be considered disposable, since damage and loss can occur. We thus attempt to provide solutions using commonly available equipment. This in turn supports the principle of generality: cheaper vehicles with less specialized hardware force us to deal with problems using intelligent software. For example, using visual ego-motion detection as opposed to relying heavily on shaft encoders for localization [2].

This paper describes the process by which we provide stereo vision for autonomous and teleoperated robotic vehicles under the conditions typical of USAR. We outline a novel algorithm for matching stereo images based on regions extracted from a stereo pair, and detail the steps taken at various points in the overall vision process to adhere to the twin goals of basic hardware and general solutions. We begin by reviewing other recent efforts to use vision as a primary perceptual mechanism, and follow by describing the phases involved in visual interpretation using our approach. Initial results of employing this approach in practice are then provided.

2 Related Work

Stereo vision is attractive because it generates a depth map of the environment. There are two basic approaches to the matching of stereo images: pixel-based and region-based approaches. Pixel-based approaches use feature points to match between images. Matching pixels between images is typically made more efficient through the use of epipolar constraints: if the cameras are perfectly calibrated, only one row in the image needs to be examined to find the same pixel in both images. While this is a reasonable approach from a computational standpoint, this method suffers from the problem of mismatching feature points. A single pixel, on the whole, provides very little evidence to support a match. Calibrating the cameras in order to support this is also non-trivial, and in domains such as USAR, the expectation that a fine calibration will remain accurate over time is an unreasonable one. Matching regions rather than pixels is an alternative intended to decrease mismatching, because much larger areas are matched to one another. However, these larger regions require correspondingly greater computational resources for a match to be performed. The approach we detail in

Section 3 improves on standard region-based matching through the simplification of regions, requiring fewer computational resources for matching while still maintaining robust matching.

The two most important steps in region-based matching are the identification and representation of features in the image. Research is active in this area, since current approaches often encounter environments that cause failure rates to become unmanageable. Examples of approaches currently being advocated include those of Lowe [9], Carson et al., [5] and Ishikawa and Jermyn[7].

Lowe's work [9] introduces an object recognition system known as Scale Invariant Feature Extraction (SIFT). This approach uses a feature representation that is invariant to scaling, translation, and rotation, as well as partially invariant to changes in illumination. Scale invariance is achieved through the use of the Gaussian kernel as described in [8]. For rotational invariance and efficiency, key locations are selected at the maxima and minima from the difference of the Gaussian function applied in scale space. Once a set of keys are defined for a given object, live images are scanned and objects are selected using a best-bin-first search method. Bins containing at least three entries for an object are matched to known objects using a least square regression. Experimental results show that the system is effective at detecting known objects, even in the presence of occlusion, since only three keys are necessary for a match to occur. Lowe and others have employed this method to implement a localization for reasonably structured environments [11], but nothing as unstructured as USAR.

In Carson et al.'s [5] Blobworld representation, pixels in an image are assigned to a vector containing their color, texture, and position. Colors are smoothed to prevent incorrect segmentation due to textures, and are stored using the L*a*b* color format. Texture features employed for categorization include contrast, anisotropy (direction of texture), and polarity (uniformity of texture orientation). Regions are grouped spatially if they belong to the same color and texture cluster. A gradient is generated in the x and y directions, containing the histogram value of pixels in that region. For matching, the user must begin by selecting blobs from the image that will be used for comparison against a database. Regions are matched to the database by the quadratic distance between their histograms' x and y values, in addition to the Euclidean distance for the contrast and anisotropy texture. This method was used as a basis for an optimal region match in [4]. It is unclear, however, how robustly the method handles translation of the blobs. In addition, this system is not directly usable for an autonomous system, since the user must select the candidate blobs.

Wavelet-based Indexing of Images using Region Fragmentation (WINDSURF) [1] is another recent approach to region extraction. In this approach the wavelet transform is employed to extract color and texture information from an image. A clustering algorithm is used on the output coefficients from the wavelet transform, producing regions that contain similar color and texture waves. By using only the coefficients, regions are clustered without considering spatial information. This means that images cannot be compared based on the location of the regions. However, it allows matching that is invariant to position and

orientation differences. One limitation in this approach is that the region count must be defined, so clustering of dissimilar regions can occur in the presence of images that contain more features than expected.

3 Pragmatic Stereo Vision for USAR

The aim of our overall approach is to identify useful regions and match them between stereo images, with limited computational resources and under conditions typical of the USAR domain. In fully autonomous systems, stereo-matched regions are intended as input to routines for ego-motion detection, localization, and map-building (as employed originally in [3, 2] using a single camera). In teleoperated systems, this is intended to enhance the remote perception of the teleoperator by providing information about the distance and movement of objects.

We divide the process of performing region matching in stereo vision into six stages: Color Correction, Image Blur, Edge Detection, Region Extraction, Region Simplification, and Stereo Matching. Each of these stages performs a specific function in terms of allowing a visual scene to be matched using stereo vision. We examine the role and implementation of each of these phases in turn.

3.1 Color Correction

Under the varied conditions of Urban Search and Rescue, we cannot assume uniform lighting. Thus, there will be situations where imbalances exist in the color response of the two cameras capturing the stereo image.

We have found that in general extracting stereo matches can proceed with little inaccuracy due to color differences between the two stereo images (which can easily be seen in many stereo image pairs, such as the raw image pair shown at the top of Figure 1). We have also found, however, that normalization is useful in supporting human teleoperation in situations where ambient lighting is low. For those purposes, we perform color correction by normalizing the color channels.

Our method for normalization involves using the mean and standard deviation of the color channels. While the naive computation methods for these would require two passes through the data with a third to normalize, we do this in a single pass. This is done by relying on the assumption that the mean and standard deviation over a sequence of images are relatively stable, which will generally be the case in a reasonably slow-moving vehicle in a USAR scenario. Thus, at time t we use the mean of the image from time $t-2$ and the standard deviation of the image at time $t-1$ as approximations to the current values, and normalization can ensue at the same time future standard deviation and mean values are calculated.

Normalization is performed by defining a range of two standard deviations on either side of the mean as the entire range for the color channel. This allows outliers to be discarded, resulting in a more accurate representation of the color range in the image.

3.2 Image Blurring

Raw images obtained under the conditions typical of a USAR domain are extremely prone to noise. In addition to texture and lighting variations in the domain itself, noise is exacerbated by the approach we take to the USAR problem. Given current technology, any equipment that can be viewed as cheap enough to minimize the loss of a robot will have to include inexpensive video capture devices, which are highly prone to noise. This noise makes the detection of edges, and ultimately regions and objects, extremely difficult, and thus inconsistencies introduced by noise must be minimized through some form of image smoothing, such as blurring.

We employ Gaussian blurring in this process for a number of reasons. First, Gaussian blurring provides circular symmetry [12] - that is, lines and edges in different directions are treated in a similar fashion.

More importantly, a Gaussian blur can deliver this and remain efficient. In order to perform a Gaussian blur, a bell curve is approximated with integer values, typically binomial coefficients selected from Pascal's Triangle. These particular numbers have a useful principle that allows for an efficient implementation: to apply a blur of $N = k$, the coefficients of i and j, such that $i + j = k$ can be convoluted. For example, to apply a 3x4 blur, k =5 is selected, but rather than having to use the coefficients of 5 (the set {1 5 10 10 5 1}), the coefficients of 2 (the set {1 2 1}) and 3 (the set {1 3 3 1}) can be used - the first horizontally, and the second vertically to the result of the first. The result for each pixel is then normalized by dividing by the sum of the two coefficients. The practical result of this is that a small area of blur is repeated to generate large areas, rather than requiring the additional computational expense of blurring with a larger set of coefficients. The middle element of Figure 1 illustrates the result of a such a blur on a sample stereo image pair.

3.3 Edge Detection

Preprocessing via color correction and smoothing leaves the image in a state where we can begin to identify objects in the image with the expectation of some degree of accuracy. The first step in this process is to determine rough boundaries through edge detection.

We employ Sobel edge detection in our implementation, because it is computationally simple and proves robust under a variety of conditions. Sobel edge detection involves the application of convolution masks across the image. We employ two masks, for the horizontal and vertical dimensions respectively:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

After applying each mask, normalization is performed by dividing each pixel value by four. The resulting pixels are examined against a threshold value, where values larger than the threshold indicate an edge, as shown in the bottom element of Figure 1.

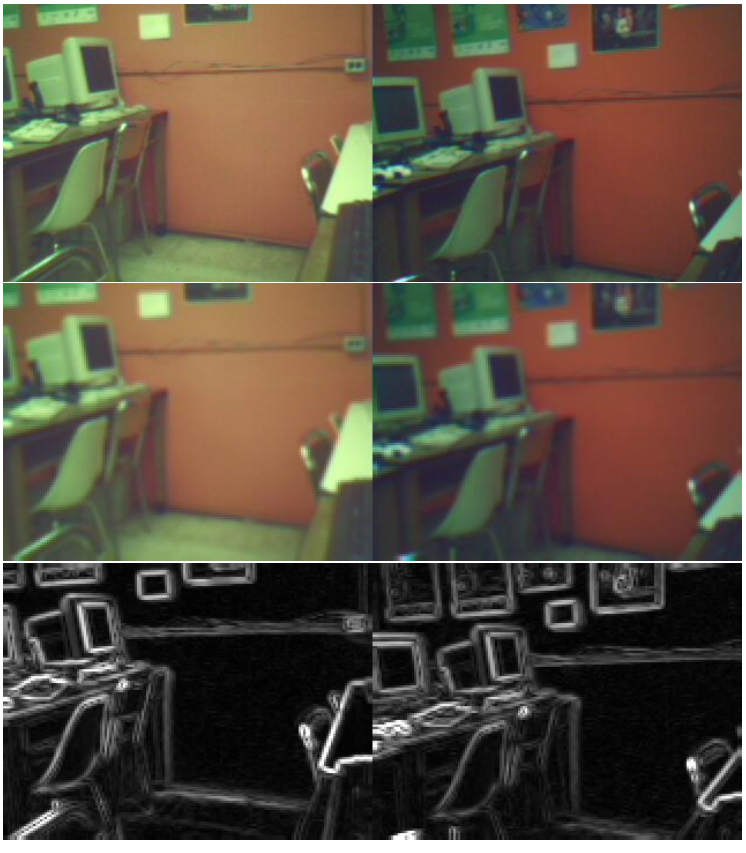

Fig. 1. The product of applying a Gaussian blur (middle) on the raw image pair (top). The blurred image is then subjected to Sobel edge detection (bottom).

3.4 Region Growing

Having obtained a set of strong edges from a smoothed image, we employ this along with the original smoothed image to determine a set of regions in each individual image.

Our approach to region segmentation involves growing regions from individual pixels using a stack-based approach. At any point, the pixels on the stack represent those from which future expansion of the region will take place. We begin by marking each pixel in the smoothed image as unexamined, and marking a single unexamined pixel as examined and placing it on the stack. We then repeatedly pop the topmost entry off the stack, and attempt to grow the region around it by examining the pixels immediately above and below, and to the left and right of that pixel. Each of these is tested to see if it is an edge pixel in the edge map, in which case it is ignored (allowing edges to form a strong boundary for regions). Each is also tested to see if it is a color match to the region being built, by summing the squares of the differences across all color channels. If this

value falls below a defined threshold for error, the pixel is considered to be a part of the current region, and that pixel is placed on the stack to further extend the region; if not, the pixel is ignored. The threshold for color error is the mean color value of all pixels currently in the region, allowing the threshold to adapt as the region is grown. The algorithm terminates with a completed region once the stack is empty. A threshold is set on the acceptable size of a grown region, and if the region size falls below this level, the region is discarded.

To extend this algorithm to grow a set of regions, we must recognize two things: first, it should be possible for an area to be part of more than one region in initial stages, since an image will generally be factorable into regions in a number of different ways. Thus, the algorithm must allow any pixel to be potentially claimed by more than one grown region. Second, once we throw a region away as being too small, we do not wish to start growing other regions within this same area, as this has already proved unfruitful. Similarly, once we have defined a region, it will be more useful to start new regions outside that defined area.

Our initial approach was to begin searching the image for a non-visited pixel, growing a region using the algorithm described above (while marking each pixel as examined when it is placed on the stack), and then starting the next region by searching for an unexamined pixel. This approach is functional, but in practice, linear scanning wastes resources because many unsuccessful regions are attempted. We have found it more fruitful to begin with randomly selected points (20 for a 320 x 240 image), selecting the location of each after regions have been grown from all previous points.

We also attempt to merge regions based on degree of pixel overlap. Each region is examined with others that it abuts or overlaps, and regions are merged if one of two thresholds are exceeded. The first of these is the percentage of pixels that overlap - this value requires a significant overall similarity, and is generally most useful in merging small regions. For merging larger regions, the likelihood of a large percentage overlap is small, and so the threshold used is a total pixel overlap. By using overlap rather than color separation as a basis for merging, shadows can be properly joined to the objects that cast them, for example, or glare to the objects the glare is placed upon, without having to set an excessively high color threshold.

At this point, we have a collection of strong regions in each of the two images (the top stereo pair in Figure 2). Each region is represented by a map between the original image and the region (a set of boolean pixels where each 1 indicates a pixel present in the image), as well as a set of region attributes: its size, mean colour value, centroid, and a bounding box.

3.5 Region Simplification

The next step in providing useful visual information to a robotic rescue agent is the matching of regions between a pair of stereo images. This, however, is a complex process that can easily consume a great deal of the limited computational resources available. Our initial stereo matching process involved examining

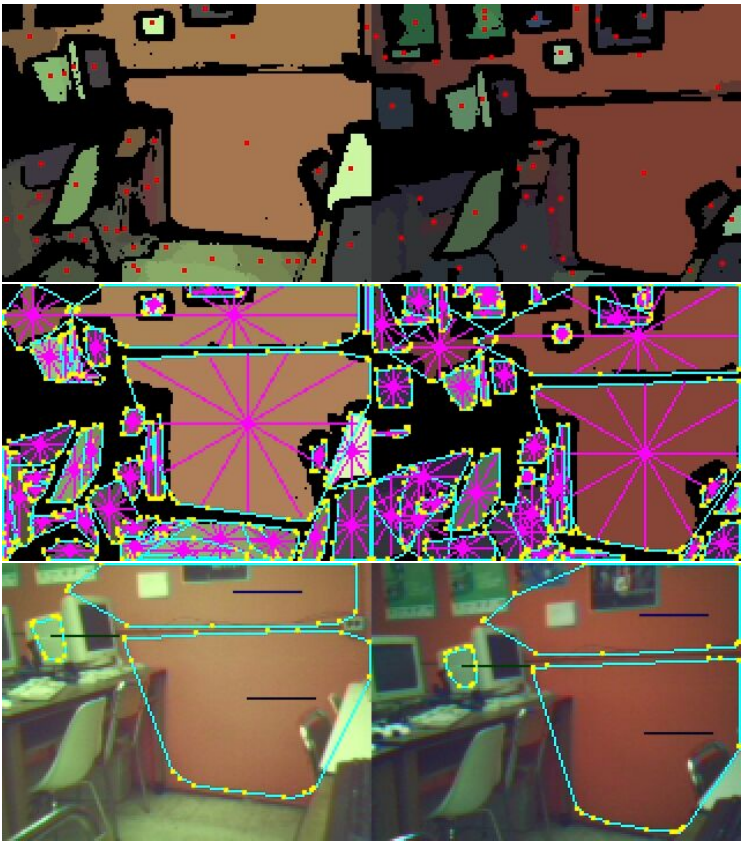

Fig. 2. Segmented regions (top), with convex hulls plotted and distance lines from the centroid added (middle). Stereo-matched regions (bottom) are bounded by a colored box, with a line emanating from the centroid of the image.

all pixels that could possibly represent the same region across the stereo pair, requiring checking for a match between hundreds of pixels for each potential match. We have considerably simplified this process by simplifying the structure of the regions themselves, allowing us to match a much smaller set of data. This process is analogous to smoothing noise out of an image before looking for edges.

We simplify regions by generating a convex hull for each, allowing us to replace the set of points outlining the region with a simpler set describing a polygon P, where every point in the original point set is either on the boundary of P or inside it. We begin with the boolean grid depicting each image. The exterior points along the vertical edges (the start and end points of each row) are used to generate a convex hull approximating the region using Graham's Scan [6]. We form a representation for the convex hull by drawing radial lines at 5 degree intervals, with each line originating at the centroid of the region and extending to the hull boundary. The length of each such line is stored, allowing an array of

72 integers to describe each region. The middle stereo pair in Figure 2 illustrates the result of this simplification process.

3.6 Stereo Matching

Once an image has been segmented into regions and simplified, regions must be matched across stereo images. Before simplifying regions, our original approach was limited in that it required superimposing region centroids and matching pixels. This was particularly troublesome for large regions. With convex hull simplification, however, the efficiency of matching can be greatly improved. With each convex hull, the very first stored value represents the distance from the centroid to the hull boundary at the 0-degree mark. A comparison of the similarity of two regions can then be easily performed by summing the squares of the differences of the values in the 72 corresponding positions in the two arrays (implicitly superimposing the centroids). Beyond greatly decreasing the number of individual points to match, this representation allows time required to make a comparison independent of region size. There is no particular threshold to a match - each region is matched to its strongest partner in the corresponding stereo image. We do, however, constrain matches for the purposes of maintaining accuracy by forcing a match to be considered only after its appearance in three successive video frames. This is particularly useful for noisy and poorly lit environments such as USAR. The bottom stereo pair in Figure 2 illustrates the matching of three regions between the raw stereo sample pair. The lines in each region are used as an indication to a teleoperator the angle that one would region have to be oriented to match the orientation of the other. That is, straight horizontal lines require no reorientation. The use of these lines will be explained momentarily.

Since we are matching regions without regard to the location in the visual frame, similar regions can be matched despite unreasonable spatial displacement. This is equally true without employing convex hulls, and is part of the nature of this domain. Because the robot is moving over very uneven terrain, cameras are likely poorly calibrated, and as the domain is unpredictable, we cannot make strong assumptions about the position of a region in each of a pair of stereo images. If this were employed in a domain where such assumptions could be made, the process could be made more accurate by strongly constraining the distance between potential matches in regions in a stereo pair, thereby lowering the number of potential matches that would have to be considered.

4 Performance

This system has been implemented and tested using an unmodified area in the Computer Science department at the University of Manitoba. *Spike*, the robot used in this project, is a one-sixth scale model radio-controlled toy car with rear wheel drive (See Figure 3). The radio controller has been modified to allow the vehicle to be controlled through the parallel port of any PC. The PC used, a 533MHz C3 Eden VIA Mini-ITX, with a 256Mb flash card, is carried in the interior of the vehicle. For this hardware configuration, we developed our own

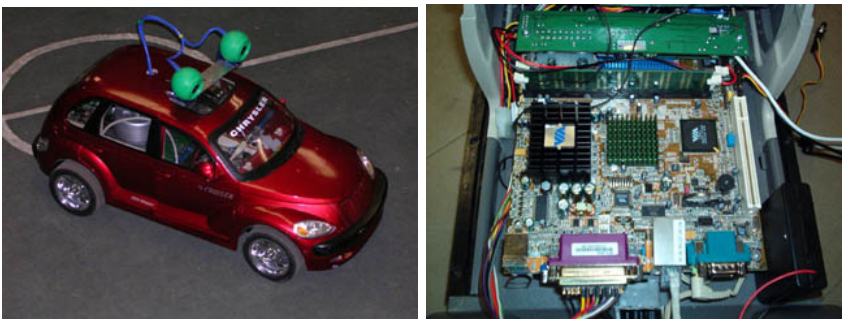

Fig. 3. Spike, the mobile robot used for this experiment

miniaturized version of the Debian Linux distribution, refined for use on systems with reduced hard drive space. The vision hardware consists of two USB web cameras capable of capturing 320 by 240 pixel images. The cameras are mounted on a servo that allows the stereo rig to pan in a range of plus or minus 45 degrees.

Figures 2 and 4 illustrate the matching abilities of this system. The raw image shown previously results in 40 regions for the right image and 42 regions for the left (including regions carried forward from previous frames). For applications in teleautonomous vehicle control, we currently display to the operator only the three strongest stereo matches present, in order to provide useful information without cluttering the camera view. In the first sample match (the bottom image pair in Figure 2), the three strongest image matches (as indicated by the colored boxes surrounding the matched pairs) are all correct, despite the offset in images due to the camera angle. Lighting in the area varies over time, resulting in changes in the matched pairs. The second sample match (the top image pair in Figure 4) illustrates a mismatch between two similarly shaped hulls. The line emanating from the image on the right hand side indicates the angle by which the right camera would have to be rotated for these shapes to match with a similar orientation, allowing a teleoperator or autonomous system to judge the likelihood of such an occurrence. The third image set (the bottom image pair in Figure 4) illustrates a similar correct match, but with one different stereo pair forming one of the three strongest matches.

We have observed very good improvement through the use of region simplifications with no decrease in match accuracy. With all vision processing turned off using the computational resources described above (that is, running only the other control functions), we achieve a capture rate of 2.5-2.9 frames per second. Activating region matching (including all phases described above) using convex hulls as the basis for a match results in a frame rate of 2.3-2.5 fps, while not employing convex hulls results in a frame rate of only 1.5-1.8 fps. We are currently investigating methods for speeding up this process further. In particular, the matching of regions is computationally expensive because the stereo system is entirely uncalibrated at the moment. Therefore, each region must be compared

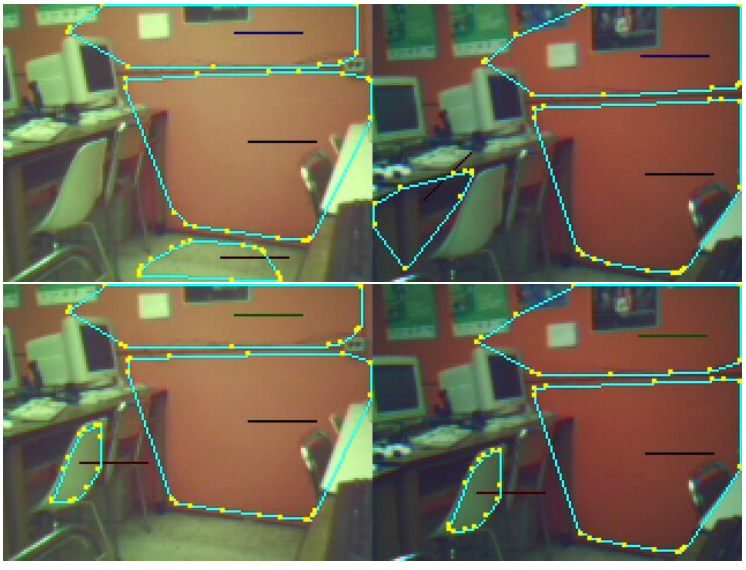

Fig. 4. Demonstration of matching in an unknown environment. The top pair shows a spatially incorrect match, while the bottom pair shows a different match than Figure 2.

against all other regions. By adding at least a rough calibration, some of these comparisons can be avoided: for example, matching regions with extremely large of spatial disparities are unlikely even in a USAR environment.

5 Conclusion

This paper has described our approach to stereo vision matching, which forms the basis of visual perception for both autonomous and teleoperated robots. With the region-based object extraction and stereo matching implemented, the ground work is laid for the use of higher level facilities employing stereo vision. These include 3D scene interpretation, mapping, localization, and autonomous control, some of which we have already employed in systems that use single-camera vision [3, 2].

The next step in this ongoing development is to include elements of camera calibration suitable for the USAR domain. The goal is to design a self-calibrating system that can produce the Fundamental Matrix without human interaction [10]. The Fundamental Matrix allows the object matching search to be constrained to a single line, rather than the entire image. This will improve the running time and accuracy of the stereo pair matching process. Parallel to this, we intend to replace the elements of vision based sensing in our autonomous systems with stereo vision using this approach. This will involve taking the matched stereo pairs and calculating the distance to each object found in the images by measuring the disparity observed in each image. Once the set of objects have a

distance associated with them, these will serve as the basis for map generation (which is currently performed by using ego-motion detection across the entire image), which in turn will support better localization and path planning.

The research presented in this paper represents a core component in the development of vision to support autonomous and teleoperated processing in complex domains such as USAR. It is also applicable to many other domains, and indeed, will be even more efficient in domains where assumptions about lighting, color calibration, and predictability in the environment can be made.

References

1. Stefania Ardizzoni, Ilaria Bartolini, and Marco Patella. Windsurf: Region-based image retrieval using wavelets. In *DEXA Workshop*, pages 167–173, 1999.
2. Jacky Baltes and John Anderson. A pragmatic approach to robot rescue: The keystone fire brigade. In *Proceedings of the AAAI Workshop on Rescue Robots*, 2002.
3. Jacky Baltes, John Anderson, Shawn Schaerer, and Ryan Wegner. The keystone fire brigade 2004. In *Proceedings of RoboCup-2004*, Lisbon, Portugal, 2004.
4. Ilaria Bartolini, Paolo Ciaccia, and Marco Patella. A sound algorithm for region-based image retrieval using an index. In *DEXA Workshop*, pages 930–934, 2000.
5. Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.
6. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, and Cliffort Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2nd edition, 2001.
7. Hiroshi Ishikawa and Ian H. Jermyn. Region extraction from multiple images. In *Eigth IEEE International Conference on Computer Vision*, July 2001.
8. Tony Lindeberg. Scale-space: A framework for handling image structures at multiple scales. In Egmond aan Zee, editor, *Proc. CERN School of Computing,*, September 1996.
9. David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
10. Quang-Tuan Luong and Olivier Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *The International Journal of Computer Vision*, 17(1):43–76, 1996.
11. Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *I. J. Robotic Res*, 21:735–760, 2002.
12. F. Waltz and J. Miller. An effecient algorithm for gaussian blur using finite-state machines. In *SPIE Conf. on Machine Vision Systems for Inspection and Metrology VII*, 1998.