A Region-Based Approach to Stereo Matching for USAR

Brian McKinnon and 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, but methods must also be in place for dealing with the large volume of data that stereo vision necessitates, in order to maintain a practical frame rate. 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.

Our own interests in this area 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 AI technology becomes 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.

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. 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 as a robot becomes entangled with debris is unreasonable. 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.

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 [5], Carson et al., [3] and Ishikawa and Jermyn[4].

Lowe's work [5] 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. 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 [7], but nothing as unstructured as USAR.

In Carson et al.'s [3] Blobworld representation, pixels in an image are assigned to a vector containing their color, texture, and position. 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.

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. 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. We divide the process of performing region matching in stereo vision into five stages: Image Blurring, 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.

3.1 Preprocessing

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 affected by the quality of the image capture hardware itself. 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 [8] - that is, lines and edges in different directions are treated in a similar fashion.

This 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 two convolution masks across the image. 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.

3.2 Region Growing

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 then mark a single pixel as examined and place it on the stack. We 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. If any of these pixels is an edge in the edge map, it is ignored (edges thus form a strong boundary for regions). Each new pixel is 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 is placed on the stack to further extend the region. 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.

Our initial approach to growing a set of regions using this algorithm 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 was 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 by comparing each to others that it abuts or overlaps, and merging if one of two thresholds are exceeded. The first of these is the percentage of pixels that overlap, requiring 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 second pair in Figure 1). Each region is represented by a map between the

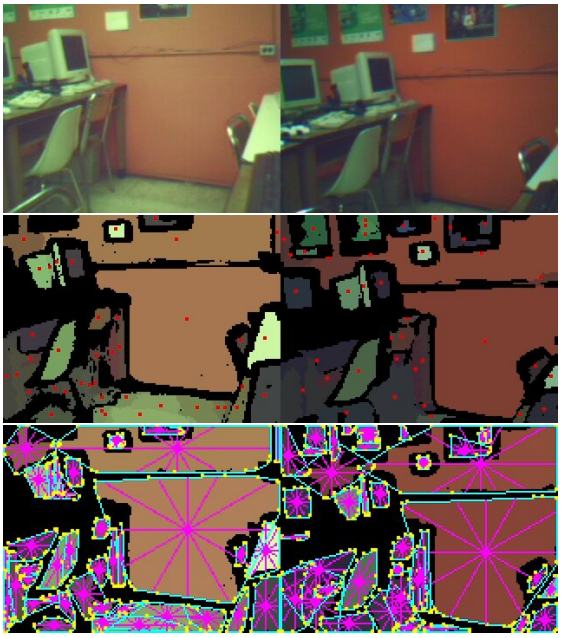

Fig. 1. Segmented regions (middle) from the raw image pair (top) after blurring and edge detection, with convex hulls plotted (bottom).

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.

3.3 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 is a complex process that can easily consume a great deal of the limited computational resources available. Our initial stereo matching process involved examining 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. 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 final stereo pair in Figure 1 illustrates the result of this process.

3.4 Stereo Matching

With convex hull simplification, 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 makes the time required to compute 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 top stereo pair in Figure 2 illustrates the matching of three regions between the raw stereo sample pair in Figure 1. The lines plotted in each region are used as an indication to a teleoperator of the angle that one would region have to be oriented to match the orientation of the other, and can also be used as estimation of confidence in an autonomous situation.

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 (Figure 2, top). 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 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.

Figure 2 also illustrates 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 (Figure 2, middle), 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 (Figure 2, bottom) illustrates a mismatch between two similarly shaped hulls.

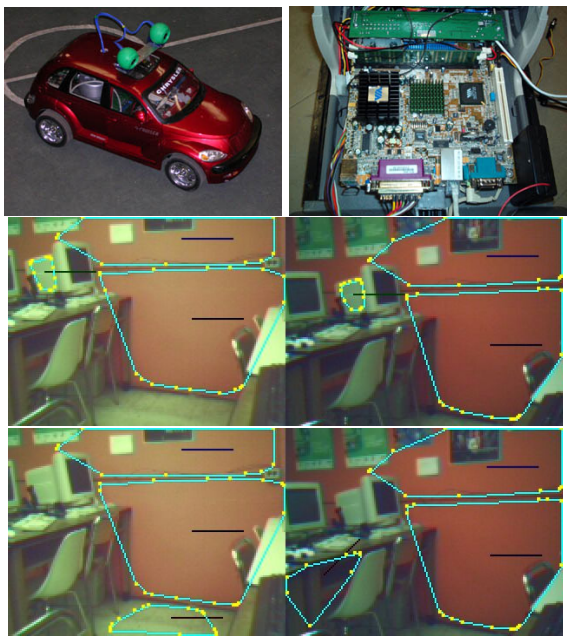

Fig. 2. Implementation and Demonstration. Top: Spike, the mobile robot used for this work. Middle and Bottom: sample matches.

We have observed very good improvement through the use of region simplifications with no decrease in match accuracy. As a baseline for comparison,

the hardware described above delivers a capture rate of 2.5-2.9 frames per second when performing no vision processing. Using region matching (including all phases described above) with convex hulls results in a frame rate of 2.3-2.5 fps, while not employing convex hulls results in a frame rate of 1.5-1.8 fps.

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. This lays the ground work for the use of higher level facilities employing stereo vision, including 3D scene interpretation, mapping, localization, and autonomous control, some of which have already been employed in our systems with single-camera vision [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 automatically produce the Fundamental Matrix [6], which allows an 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. We also intend to replace the elements of vision based sensing in our autonomous systems with stereo vision using this approach.

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, John Anderson, Shawn Schaerer, and Ryan Wegner. The keystone fire brigade 2004. In *Proceedings of RoboCup-2004*, Lisbon, Portugal, 2004.
3. 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.
4. Hiroshi Ishikawa and Ian H. Jermyn. Region extraction from multiple images. In *Eigth IEEE International Conference on Computer Vision*, July 2001.
5. David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the Int. Conf. on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.
6. 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.
7. 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.
8. 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.