An Outdoor Stereo Vision Brick Recognition System for Construction Robots

Lynne A. Slivovsky*, Kris Rahardja*,
Jeff Edwards*, Avi Kak*, and Yasuo Tanaka**

*Robot Vision Lab, 1285 Electrical Engineering Building, Purdue University, West Lafayette, Indiana 47907-1285, USA.
**Technical Research Laboratory, Hitachi Construction Machinery Co., Ltd., 650 Kandatsu-machi, Tsuchiura, Ibaraki, 300 JAPAN

ABSTRACT

Our work addresses the problem of using robot vision to recognize and localize objects in outdoor environments. Successful applications of robot vision are mostly found in indoor environments where illumination and backgrounds can be carefully controlled. For outdoor applications, a vision system must be robust with respect to the changing ambient illumination, not to mention the difficulties caused by complex and varied backgrounds against which an object must be recognized. In this paper we describe an outdoor stereo-vision brick recognition system for a construction robot. Our algorithms can accommodate large variations in the reflectivity properties of the background and shadowing effects caused by adjoining buildings and other features of the landscape. The performance of the algorithms remains unaffected when the background is changed from, say, dirt to grass. We also have achieved tolerance with respect to the changing sun angle and some shadowing caused by clouds.

1. INTRODUCTION

The construction industry can benefit greatly from the automation of some of its labor intensive and expensive components, as stated by Pritschow, et. al.[1]. There they describe a mobile robot capable of performing masonry construction tasks such as laying bricks. A vision system like ours teamed up with their robot would be quite a team.

The goal of our system is to recognize and localize bricks in a natural and complex environment. The bricks used in our experiments vary in color from orange to red to brown, where each brick is almost homogeneous in color. The size of the bricks is approximately

20.0 cm x 10.0 cm x 5.5 cm. We used these measurements to construct a wire-frame model for the bricks although these numbers are accurate only to within 0.25 cm due to the irregularity of the brick manufacturing process. And so from the beginning, the size of any particular brick that we attempt to recognize is not known as precisely as one would like. Although not as hard as finding a needle in a haystack, we recognize a brick from a pile of a variable number of randomly placed bricks. (Currently we stop after the successful recognition of a single brick. It would be a minor adjustment to continue locating all bricks that are sufficiently visible to the sensors.)

We do the recognition and localization in an outdoor environment. The lighting is dependent on the position of the sun and the type of cloud cover. Shadowing is caused by these two elements as well as the surrounding landscape; e.g. trees, shrubs, and buildings. In most vision systems, experiments are conducted in indoor environments where one can control lighting conditions, backgrounds and shadowing effects. Outdoor vision experiments are few and far between. Most concern themselves with mobile robot navigation. We have developed an outdoor object recognition system that uses stereo vision.

The robustness we claim for our system was achieved by using a model-based approach to the grouping of features extracted by low-level image processing routines. (By model-based we mean using as much knowledge as possible about the objects and the backgrounds for figure-ground separation.) We first perform a region-based segmentation of both the left and right images using a quadtree data structure. To the extent that some of the model knowledge consists of knowing that smooth man-made objects cannot result in regions with chaotic shapes, small complex-looking regions output by the segmenter are merged with adjoining regions with statistically similar properties. Next, Hough transformation[2] is used to extract line segments from the region boundaries. We then search through these segments looking for all lines that could serve as candidates for the boundary of a brick, and group together those that could form a brick face; this is done on the basis of criteria such as parallelity, the gray levels adjoining the lines, etc. We are then finally ready to perform stereo matching. Stereo correspondence is set up for parallel lines that were selected as candidates for brick faces in each of the two images. This correspondence results in the system hypothesizing a 3-dimensional face of a brick in the scene. The proposed brick face is verified by applying tests based on the geometrical model of the brick to the size of the face edges. Calculation of the position and orientation of the recognized brick is then simple. Figure 1 shows a flowchart outlining the steps to our algorithm. Figure 2 shows a sequence of intermediate images of a sample brick scene.

2. REGION AND SEGMENTATION ANALYSIS

Segmentation is a very important step in a vision system. Poor segmentation leads to poor performance in later steps of an algorithm. We use a region based segmentation algorithm similar to the split-and-merge algorithm[2] to segment the left and right images. The goal of the segmentation is to group together pixels in the image that

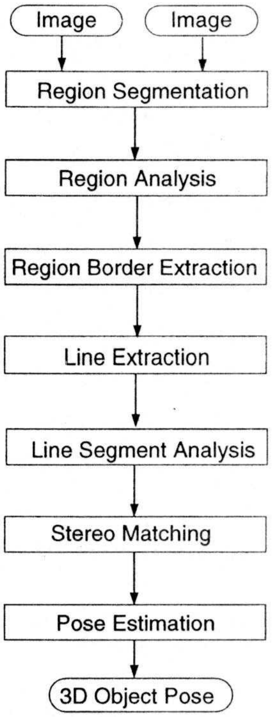
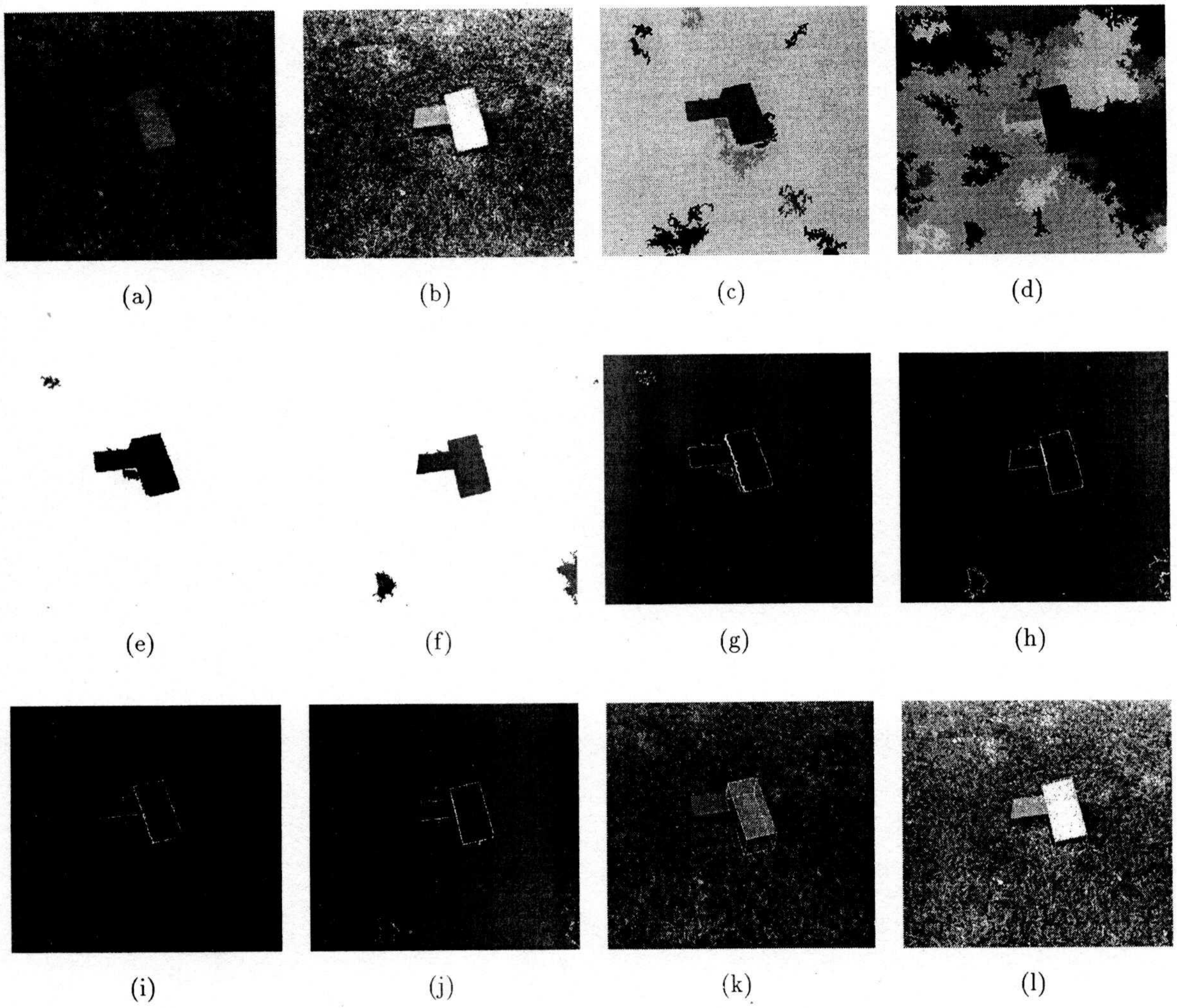


Figure 1: Flowchart of our algorithm.

are statistically similar. A quadtree data structure of the image is first built. Merging of regions is done to improve the segmentation. This is done in three stages, with each stage testing a different merging criteria. First, we merge neighboring quad structures that have a maximum minus minimum pixel intensity difference within a threshold value: $Max_{PixelIntensity} - Min_{PixelIntensity} < \epsilon$. Second, we merge neighboring regions that have an average pixel intensity difference within a threshold value: $Region1_{AvePixelIntensity} - Region2_{AvePixelIntensity} < \epsilon$. And third, we merge small regions to a larger neighboring region with the closest average pixel intensity: $Area_{Region} < Area_{Threshold} < \epsilon$.

Each of these criteria requires a threshold to be given. In order to set these thresholds to make our recognition system robust to changes in the outdoor environment, many test images were examined to obtain threshold values that would perform in different lighting situations. Since the quadtree data structure maintains square regions when first constructed, criteria 1 is implemented to obtain non-square regions. Criteria 2 insures that a small gradient of pixel intensity across a region does not hinder the entire region from being obtained. Criteria 3 helps to eliminate very small regions that are most likely due to noise.

Noting that the objects we are trying to recognize are polygonal, we prune the region space in order to decrease the running time of the rest of our algorithm. We do this by computing the shape complexity of each region as $perimeter^2/area$ and merge those



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 2: (a) and (b) The original left and right, respectively, image pair; (c) and (d) Results after segmentation and the three merge passes; (e) and (f) Images after removal of complex regions; (g) and (h) Region borders; (i) and (j) Extracted line segments overlaid on region borders; (k) and (l) Original images with the wire frame model overlaid on the recognized brick.

regions with a high complexity with a neighboring region. Images showing the segmentation both before and after the merging of complex regions are shown in Figures 2(c), 2(d), 2(e), and 2(f). After eliminating complex regions complete figure-ground separation is almost attained. An example of the improved segmentation can be seen in Figures 2(c) and 2(d). The complexity threshold was also set by looking at the same test images as above. A precise threshold for this test is not as important as the three above because any linear sections of a complex region would be easily determined as outliers in the stereo matching process. Of course, it is desirable to not have to deal with an outlier in the first place, but the presence of an outlier would not incapacitate the stereo matching process.

3. LINE EXTRACTION AND ANALYSIS

Since most regions in an image have jagged borders, the boundary pixels are used as input to a Hough transform to generate a Hough map. The Hough space is defined by θ, the orientation of a line, and by d, the perpendicular distance from the origin to the line. We generate a histogram of edge pixels of the region boundaries to create the Hough map. The peaks in the Hough space are located and are hypothesized to be line segments, ideally all and only brick edges, in the image. This again is done for both the left and right region maps. The endpoints of the line segments are obtained by tracking the edge pixels along the lines corresponding to Hough peaks in the region map.

At this time we do line segment analysis to acquire additional information to pass on to the stereo matcher. The average pixel intensities for each side of a line segment are calculated and those line segments that have similar intensities for at least one side and are at approximately the same orientation are paired together. Figures 2(g), 2(h), 2(i), and 2(j) show the borders of the final regions from the segmentation and the line segments extracted from the left and right images of our sample scene. These line segments are overlaid on the region borders.

4. STEREO MATCHING

Finding the correct stereo correspondences between individual points in a pair of stereo images is extremely difficult and computationally expensive. In order to make our system robust and efficient we perform the stereo matching process at a higher level; meaning we match line segments, not individual pixels. This greatly reduces the number of elements in each image to be matched and hence its cost. And by having more information about the possibly corresponding left and right objects a more reliable match can be made. An indoor stereo vision system using this approach of matching higher level components with the aid of their features was demonstrated by Huynh and Owens[3]. The more efficient idea of matching higher level components can be related to putting together a jigsaw puzzle. A puzzle in which every piece is the same color is much more difficult than putting together those same pieces now colored so that the entire puzzle has a pattern to it.

Being in an outdoor environment our scenes have inherent difficulties that must be addressed by the stereo matcher. We recognize a brick in a pile of randomly placed bricks. The setup for the experiment itself causes many undesirable problems that need to be dealt with. Bricks occlude each other. To some extent the uneveness of the ground that the bricks are set upon occludes their bottom edges. In the case where the bricks are placed on grass it is much worse. Blades of grass occlude much of a brick face and at times are as tall as the side of a brick. The effects of these problems first show up during the segmentation and then propagate through the rest of the system.

The majority of line segments extracted from our test images are either whole or partial brick edges from the top faces of the bricks in the scene. Making note of that, our system attempts to reconstruct the top face of a brick during the stereo matching process by matching the four line segments of the face in the stereo image pair. Of course, all four line segments of the top brick face are not always found. Three sides of a brick may be used to hypothesize the location of the fourth. With two or more sides missing predictions of the remaining ones become unreliable.

In the case where four sides, including even a partial side, of a brick face have been extracted during segmentation and line segment analysis our algorithm proceeds in the following manner (these steps are performed for both the left and right images):

1. For both the left and right images:
   (a) Extract line segments forming a parallelogram for the image.
   (b) Compute line intersections to obtain four corner points of the parallelogram.
   (c) Check that a minimum number of line segment endpoints are within some ϵ of the computed intersections.
   (d) Extract an intact brick face.
2. Perform stereo matching of line segments based on the positions and orientations of the segments in the left and right images.
3. Compute the 3-D world coordinates of the corners of the face.
4. Verify face hypothesis.

First, one pair of parallel lines, with a common average neighboring pixel intensity between them, is selected. We then search for another pair of parallel line segments that would form a trapezoid with the first pair. The average neighboring pixel intensity need not be the same as the first pair. The intersection points of the four line segments are then calculated. This is done by using the line parameters from the Hough space corresponding to each line segment.

We then determine how many line segment endpoints are within a certain (projected) distance from the calculated intersection points; requires checking a total of 8 endpoints. By setting the minimum required number of endpoints that fall within this specified distance we compensate for situations in which the line segment extracted from above was either too long or too short. In instances where the line segment extracted was too long it usually was not long by a great amount. A more visually dramatic case is when a line segment extracted is too short. In these instances half of a brick edge could be missing, yet by calculating the intersections of the extended line segment (a line) this is no problem.

Once the brick faces edges are known in both images the correspondence between them is determined. The positions of the line segments in the 2-dimensional images as well as the fact that the angle of a line segment in the left image is approximately the angle of its corresponding line segment in the right image guide us in doing this. Li[4] states that, depending on camera positions and orientations relative to scene objects, one can assume these angles to be equal. We do not require such a tight constraint in our system.

With the stereo correspondence of the brick face edges in place finding the correspondence of the face corners is trivial. We do this for the calculated intersection points. We then calculate the 3-dimensional coordinates of the brick face corners. To verify this is in fact a brick face we compute the 3-dimensional distances between corners and compare these measurements to our brick model. If these measurements are within an acceptable distance to the model measurements we have successfully recognized a brick.

5. EXPERIMENTS

The vision system was designed for a mobile platform that allows experiments to be conducted in an outdoor environment. The platform is a portable rig, see Figure 3, that holds all necessary equipment. The system components are:

- Operating system: VxWorks 5.1
- 1 - MVME167 Board (Motorola 68040)
- 1 - Image Technologies FG150 frame grabber
- 1 - 500 MByte SCSI hard disk
- 2 - Monochrome cameras
- 2 - Black and white monitors

The setup of the mobile platform allows cameras to be placed in any of numerous positions in a two dimensional vertical grid. The capability to expanding the number of cameras in our system, say to a trinocular vision system, is already physically present.

Calibration of the cameras was performed in an indoor environment by utilizing our standard calibration procedure[5]. We could have done this in an outdoor environment where the experiments were being conducted. However, a nice controlled environment is preferable since we need sufficiently precise calibration.

We tested our system with a wide variety brick scenes on both grass and dirt. These test pairs contained differences in image brightness, glare, occlusion, and different degrees

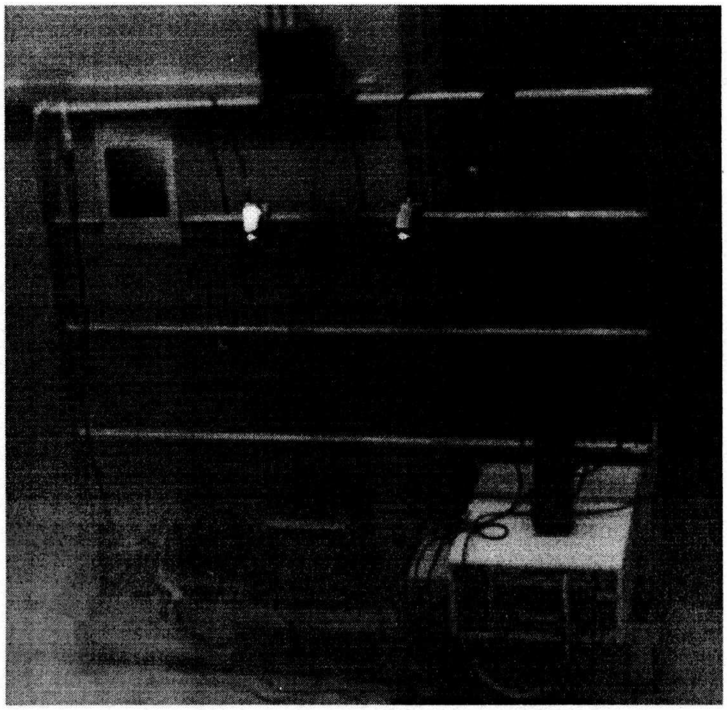


Figure 3: System Setup.

of focus in the camera lens. Figures 2(k) and 2(l) shows the correct recognition of a brick from the sample grass scene. A test image pair of a brick scene on dirt overlaid with the wire frame model on the located brick is shown in Figures 4. Note the abundance of debris, wood chips and leaves, throughout the scene. Through experimentation we verified the robustness of our system. The false recognition rate was negligible. The average processing time for the entire algorithm to be executed is two and a half minutes. Much of this time is during the execution of the segmentation and line extraction phases with the stereo matching taking less than one second.

6. CONCLUSION

A system such as ours can easily be extended to accommodate various other types of building materials such as different size bricks and concrete blocks. It can also be coupled with a brick laying robot to create a skilled and efficient labor saving machine. The vision system and robot can work in parallel. As the robot is laying a brick, the vision system can be locating the next brick to pick up.

In this paper we presented an outdoor stereo vision brick recognition system for a construction robot. We have described the problems one encounters when taking a vision system out of its indoor bubble and into the real world. These are uncontrollable light sources, shadowing, and occlusion. Our solution at overcoming these problems

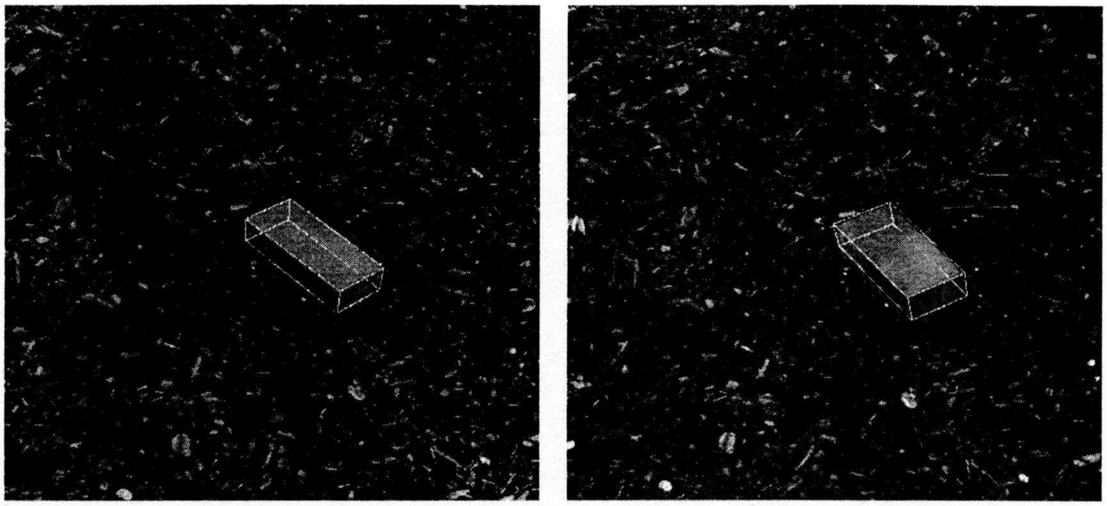

(a) (b)

Figure 4: Original images with the wire frame model overlaid on the recognized brick.

was then laid out. Our region segmentation module performs figure-ground separation by iteratively merging neighboring regions that satisfy certain criteria. Choosing the thresholds for these criteria is an important step. They must be flexible enough to cover the wide range of possible lighting yet well defined to prevent a computational explosion. Line segment analysis was done to extract region borders, which are ideally brick edges as well. The stereo matching process proved to be a unique algorithm, and when an ideal set of brick edges were extracted it performed perfectly.

ACKNOWLEDGEMENTS

We would like to thank the Hitachi Construction Co. Ltd. for their support in funding this project.

REFERENCES

1. "A Mobile Robot for On-Site Construction of Masonry," Proc. of the IEEE/RSJ/GI Int. Conf. on Intelligent Robots and Systems, vol. 3, pp. 1701-1707, 1994.

2. A. Rosenfeld and A. C. Kak, *Digital Picture Processing.* Second Edition, Vol. 2, Academic Press, New York, 1982.

3. D. Q. Huynh and R. A. Owens, "Line labelling and region segmentation in stereo image pairs," Image and Vision Computing, vol. 12, no. 4, pp. 213-225, 1994.

4. Z. Li, "Stereo Correspondence Based on Line Matching in Hough Space Using

Dynamic Programming," IEEE Trans. on System, Man, and Cybernetics, vol. 24, no. 1, pp. 144-152, 1994.

5. K. Rahardja and A. Kosaka, "Automatic Camera Calibration," *Robot Vision Laboratory Memo, School of Electrical and Computer Engineering, Purdue University,* RVL Memo #37, March 14, 1995.