Rule-Based Segmentation for Intensity-Adaptive Fiducial Detection

Jong-Weon Lee[a] and Ulrich Neumann[b]

Computer Science Department
Integrated Media Systems Center
University of Southern California, Los Angeles, CA 90089-2561

ABSTRACT

This paper describes a new fiducial detection method for use under varying lighting conditions without manual control of any parameters. We developed the algorithm especially for vision-based Augmented Reality (AR) systems. The major problem in Augmented Reality is the registration between the virtual world and the real world. The user's pose in both worlds should be exactly the same. Vision-based AR is an attractive approach to the registration problem, however the fiducial detection methods used in many systems operate only under restricted lighting conditions. We developed a rule-based algorithm to segment regions of an image to detect known fiducials under varying lighting conditions. The algorithm is based on simple spatial and intensity relations among fiducials and their backgrounds. Rules and membership functions are defined from those relations. Rules are applied to find transition regions, and membership functions locate an edge position within a transition region. Edges are clustered to segment regions in an image. A vision-based AR system using our method operates under varying lighting conditions, including uneven lighting. This detection method extends the operating conditions of vision-based AR systems.

Keywords: Augmented Reality, edge detection, region segmentation, rule-based detection

1. INTRODUCTION

Augmented reality (AR) is a technology that merges virtual objects with the real world by using real and virtual cameras. A real video camera generates images of the real environment, and a virtual camera generates images of 3D graphics models to enhance a user's perception of and interaction with the real world[1]. Accurate registration between real and virtual scene elements has to be achieved to make AR systems useful in real applications. Techniques for achieving accurate registration can be divided into two areas, non-vision technology and vision-based technology. Vision-based AR systems detect fiducials using image processing or computer vision techniques to solve the registration problem. Boundary detection, color segmentation, watershed, and template matching techniques are used for current systems. These techniques often fail under varying lighting conditions or require manual parameter changes. Since AR systems are sensitive to changes in the environment, their use is restricted. Even with controlled lighting conditions objects can shadow fiducials. Color and intensity values in shadowed areas are quite different from those without shadows. These differences require different threshold values for detecting fiducials. Therefore we need a detection method that works under any lighting conditions without manually controlling any parameters, including uneven lighting condition in one image.

Several AR systems use vision-based techniques. Some authors mention their approaches to different lighting conditions while others give little information relating the environmental restrictions under which their systems operate. Madritsch, Leberl, and Gervautz[8] developed a camera-based beacon tracking system. They use red LEDs as beacons and accept lighting restrictions for their system. They suggest combining infrared LEDs with infrared filters to reduce dependence on lighting conditions. State, Hirota, Chen, Garrett, and Livingston[11] developed a hybrid system, which combines a vision-based tracking system and a magnetic tracking system. The algorithm to detect fiducials is based on the ratios of RGB component values. They restrict lighting conditions and mention diminishing fiducial detection performance with changing lighting conditions despite the use of adaptive brightness evaluation for each landmark. Uenohara and Kanade[12] discuss how the appearance of objects varies depending on pose and illumination change. They solve this problem in two ways. They require users to locate objects at the initial recognition step and precapture reference images around feature points in varied illumination with

[a] Email: jonlee@graphics.usc.edu
[b] Email: uneumann@graphics.usc.edu

predefined positions of objects. Normalized correlation is computed with all reference images, and they select the point with the highest score over a threshold value. The authors claim that the values of normalized correlation are insensitive to the intensity changes. The complexity of the detection algorithm increases as objects become more complex and more reference images are required.

As illustrated by the above three systems, vision-based AR systems require a fiducial detection algorithm, and the performance of detection algorithms diminishes when lighting conditions vary. We developed a rule-based algorithm to detect fiducials under varying lighting conditions. The algorithm uses relations among homogeneous regions that remain invariant under illumination. We define rules and membership functions from these relations and use them to detect fiducials. We also define rules and a membership function for the shape of our fiducials, an ellipse, and used them for a shape verification procedure that is fast and robust under rotations of a camera. The results shows that our method is robust under varying lighting conditions. After a description of the algorithm in Section 2 we present results in Section 3 and a discussion in Section 4.

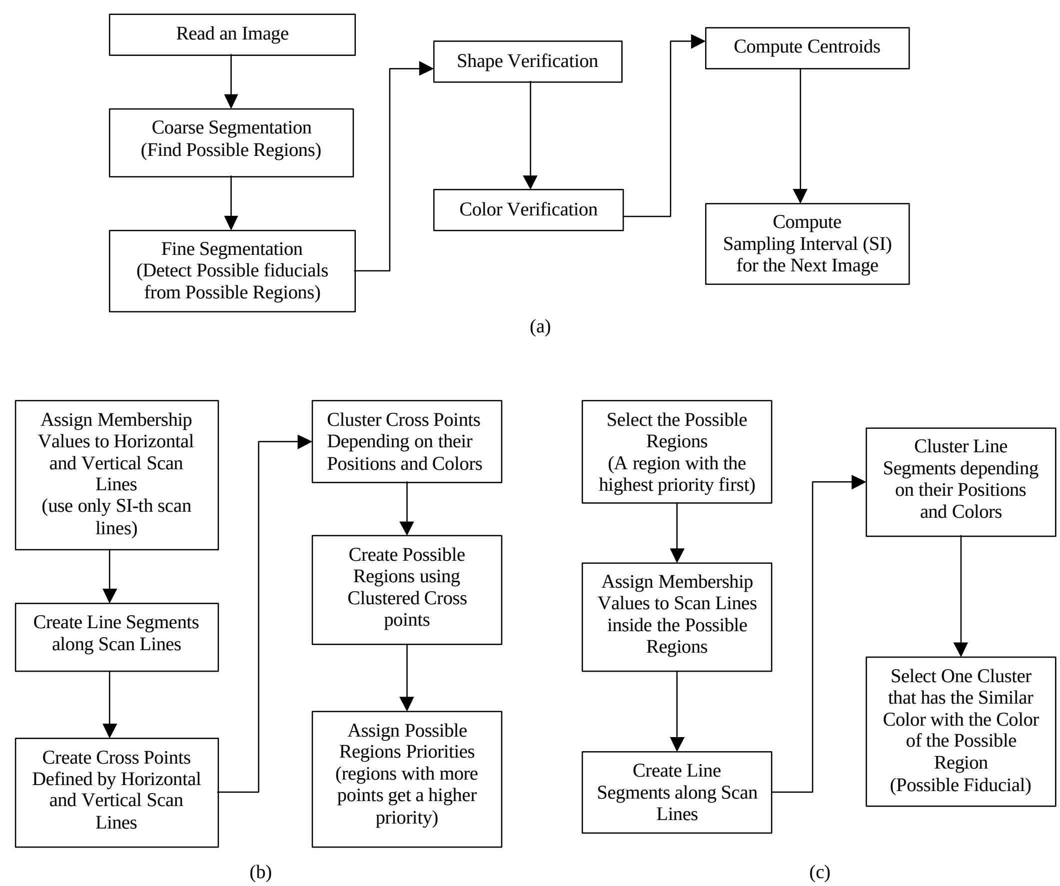


Figure 1. (a) Overview of the Algorithm (b) Coarse Segmentation (c) Fine Segmentation

2. DESCRIPTION OF THE ALGORITHM

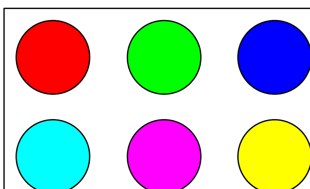


Figure 2. Color Fiducials

We are only interesting in fiducials (Figure 2) that occupy small areas of an image, so a multi-resolution approach is applied for efficiency in our detection algorithm (Figure 1(a)). The segmentation algorithm is divided into two stages, coarse segmentation and fine segmentation. Coarse segmentation (Figure 1(b)) quickly skims through an image and finds potential fiducial regions, and a more expensive fine segmentation method (Figure 1(c)) detects possible fiducials. Shape and color verifications are applied to eliminate false fiducials from possible fiducials. Coarse and fine segmentation methods share two steps, assigning membership values and creating line segments. They differ in two ways. The coarse segmentation finds potential regions using only sampled horizontal and vertical scan lines, but the fine segmentation uses all scan lines in small selected regions to detect possible fiducials. The other difference is that coarse segmentation clusters cross points created with horizontal and vertical line segments, but the fine segmentation clusters line segments along horizontal and vertical scan lines. Clustering in both segmentations is based on locations and colors of cross points or line segments. Rules and fuzzy algorithms are used for detecting transition regions between a fiducial and its background, and membership functions are applied to localize the position of edges. Details of the algorithm are presented in following sections.

2.1. Rules and membership functions

Rules and membership functions are defined from relations between fiducials and their backgrounds. We assume fiducials have solid colors and they are encompassed by homogeneous regions. This means interesting edges exist between two homogeneous regions (Figure 3). Rules are applied to detect transition regions that contain fiducial edges. Transition regions do not always contain fiducial edges, but false transition regions will be eliminated. Membership functions locate the positions of edges inside the transition regions. We assume that the best location of an edge inside a transition region is a pixel whose intensity is close to the average value of left and right regions of a transition region. Rules and membership functions are defined as follows.

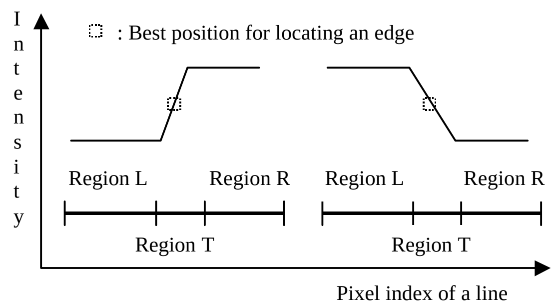


Figure 3 Step Edge Model

1. Monotonic average rule
   - *Avg(R) > Avg(T) > Avg(L)* or *Avg(L) > Avg(T) > Avg(R)*, where *Avg(J)* is an average intensity value of a region *J*.
2. Distribution rule
   - (*Max(T) – Min(T)*) > (*Max(R) – Min(R)*) & (*Max(T) – Min(T)*) > (*Max(L) – Min(L)*), where *Max(J)* and *Min(J)* indicate the maximum and minimum intensity values of a region *J*.
3. Overlapping rule
   - *Max(R) < Min(L)* when *Avg(R) < Avg(L)*
   - *Max(L) < Min(R)* when *Avg(L) < Avg(R)*
4. Membership functions

$$m = m_1 \times m_2 \qquad (1)$$

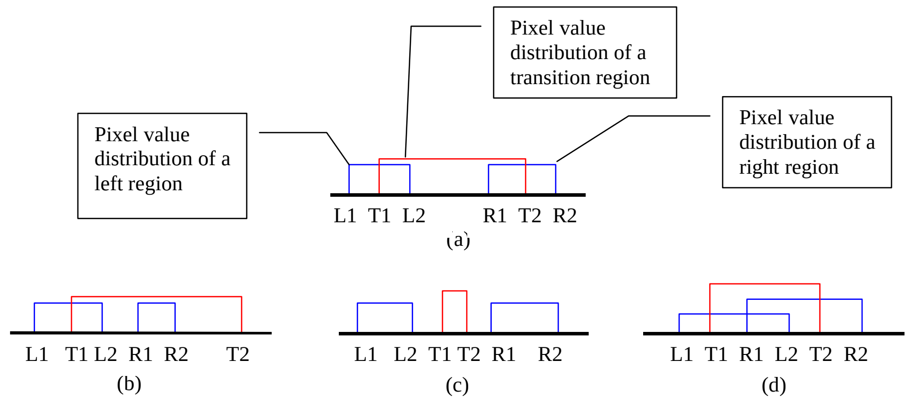
$$m_1 = \frac{2 \times MIN\left(\left|Avg(R) - Avg(T)\right|, \left|Avg(L) - Avg(T)\right|\right)}{\left|Avg(R) - Avg(L)\right|} \qquad (2)$$

$$m_2 = \frac{Min(R) - Max(L)}{Max(R) - Min(L)}, Avg(R) > Avg(L)$$

$$or \qquad (3)$$

$$m_2 = \frac{Min(L) - Max(R)}{Max(L) - Min(R)}, Avg(L) > Avg(R)$$

The membership function contains two different parts, m_1 and m_2. m_1 indicates a grade of closeness to the median intensity value between two neighbor regions. $m_1 = 1$, if $Avg(T) = |Avg(R) - Avg(L)| / 2$, and m_1 is less than 1 for other cases. m_2 indicates the grade of closeness to the ideal edge. If an edge is the ideal edge, *Min(R) – Max(L) = Max(R) – Min(L) or Min(L) – Max(R) = Max(L) – Min(R)*, then $m_2 = 1$. m_2 is less than 1 for other cases. Therefore the membership function is used to find a position of an edge that is close to the ideal edge. Some examples are shown at Figure 4.



Note. T is a transition region
L & R are left and right homogeneous regions
1 & 2 indicate minimum and maximum pixel values of a region

Figure 4. Examples of relationships among transition, left, & right regions (a) accepting case (b) unaccepting case, violating rule 1 (c) unaccepting case, violating rule 2 (d) unaccepting case, violating rule 3

2.2. Line segmentation

Each scan line is segmented into line segments according to their colors. Line segments separate regions that have different colors from neighbor regions and are created by connecting two edges that are detected by rules and membership functions described in the previous section. These line segments are clustered to form regions that have a solid color in an image or create cross points in an image.

2.3. Clustering

We expect fiducial with a solid color. For the ideal case, regions are easily segmented by clustering pixels with the same color values. However, in real images, a solid color region produces a distribution of pixel color values. This introduces a similarity measure to cluster pixels with similar color values. Often, similarity is measured with a distance metric and a threshold. Possible distance metrics used for a color similarity include absolute distance (e.g., Manhattan or Euclidean distance), 1-norm distance, 2-norm distance, ∞-norm distance, an angle between colors in the RGB color cube, and a square of cosine of a color angle. These metrics require thresholds to decide whether two color values are similar. It is difficult to automatically determine thresholds for different lighting conditions. Color values change nonlinearly depending on lighting conditions and camera characteristics, and they can not be easily predicted for different lighting conditions. Therefore we developed a similarity measure which uses local information existing in line segments. A uniform probability density function is created for each line segment, and two line segments are considered as having the same color when their uniform probability density functions overlap. Color similarity is checked whenever two line segments are next to each other, and we expect fiducials and their backgrounds to be distinct colors. The distribution of a line segment is defined by its minimum, average, and maximum color values. Find a deviation D, which is defined by $MIN(|Avg(J) - Min(J)|, |Avg(J) - Max(J)|)$, and create a uniform distribution by $Avg(J) - D$ and $Avg(J) + D$ (Figure 5). The choice of D reduces the effects of noise in an image. This

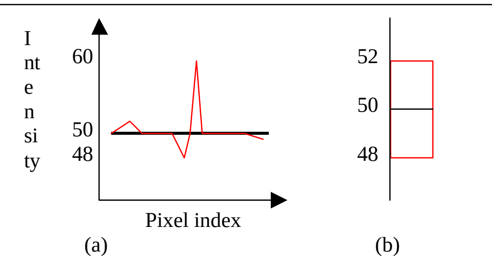


Figure 5 (a) Intensity values of a sample region, max = 60, avg = 50, & min = 48 (b) Uniform density function created with (a) simple density function works well for our detection algorithm, and it reduces the computation time of the clustering procedure.

2.4. Coarse segmentation

Fiducials occupy only small regions of an image and typically, only three or four fiducials are required for a vision-based AR system. We exploit these properties to reduce computation time. The coarse segmentation procedure is applied to find potential regions that are used in the fine segmentation procedure. Potential regions are defined by clusters of cross points that are created with sampled horizontal and vertical line segments in an image.

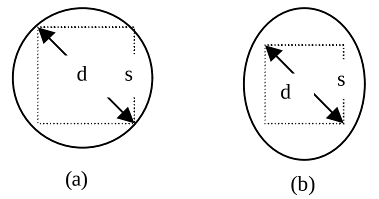


Figure 6. Maximum Sampling Interval (a) circle shape (b) ellipse shape

The sampling interval is the most important parameter in the coarse segmentation procedure. A small sampling interval increases computation time, and a large sampling interval may miss fiducials. Intersections among sampled horizontal and vertical lines form square grids containing four cross points. We create a potential region with a cluster that contains at least one cross point, but we use a sampling interval that creates at least four cross points for each fiducial. This improves robustness at a slight increase of computation time. A square grid must fit within the inner square of the expected minimum fiducial (Figure 6). We derive the size of the inner square of the smallest fiducial from the minor axis of a projected fiducial with equation (4). The optimal sampling interval is computed using equation (5). To adapt to the current viewing conditions, we use the minor axis size of the smallest fiducial of the previous image to compute the sampling interval for the current image.

$$s = \frac{d}{\sqrt{2}}, \text{ where } d \text{ is the size of minor axis of the smallest fiducial} \quad (4)$$

$$Sampling\ Interval(SI) = \frac{s}{2} \quad (5)$$

We apply the line segmentation procedure to every sampled horizontal and vertical line to detect line segments. Horizontal and vertical line segments create cross points if they intersect and have the same color. We cluster cross points depending on their positions and colors, and each cluster defines one potential region for the fine segmentation procedure. Potential regions larger than the maximum fiducial size are eliminated. A potential region priority is set in proportion to the number of cross points in its cluster.

2.5. Fine segmentation

Select potential regions in descending priority. Apply rules and membership functions along all horizontal and vertical scan lines inside the selected region. These line segments are clustered by their locations and color to create possible fiducials. Shape and color verification procedures select real fiducials from the possible fiducials.

2.6. Shape verification

Shape verification selects among possible fiducials for varied camera rotations and any viewing directions. Rules and a membership function are defined from the properties of fiducials that are invariant under the accepted range of camera poses. In practice, the rules eliminate nearly all false detections. The procedure is as follows.

1. Divide a boundary of a clustered region into four sub-boundaries with minor and major axes (Figure 7).
2. Measure deviations of sub-boundaries from straight lines connecting two end-points of axes using the following membership function.
   - W_I indicates weight of ***i***-th component of a sub-boundary, and D_i indicates a difference between horizontal positions of ***i***-th components of a sub-boundary and its corresponding straight line. The weight function is a triangle function so D_i has a larger weighting value when "*i*" is closer to the center of a sub-boundary.

$$m_{cir} = \sum_i W_i \times D_i \quad (6)$$

3. Apply following rules.
   - $|m_{cir}(\text{top-left})| \cong |m_{cir}(\text{bottom-right})|$ and $|m_{cir}(\text{top-right})| \cong |m_{cir}(\text{bottom-left})|$
   - $m_{cir}(\text{top-left}) < 0$, $m_{cir}(\text{bottom-right}) > 0$, $m_{cir}(\text{top-right}) > 0$, and $m_{cir}(\text{bottom-left}) < 0$
   - lower threshold < $|m_{cir}()|$ < upper threshold

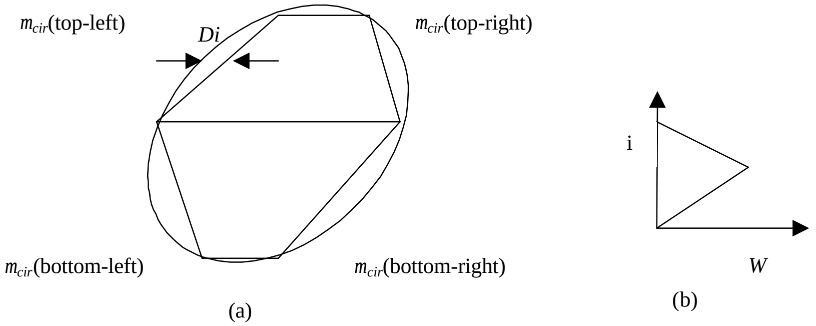


Figure 7. (a) Components of the shape test (b) A weight function used for computing m_{cir}(j)

2.7 Color verification

We use the same color verification method developed by Youngkwan Cho and detailed in his paper[10].

3. RESULTS

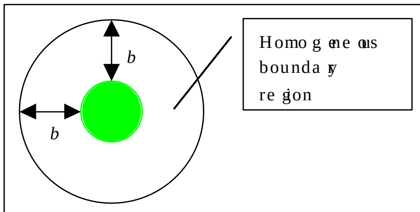
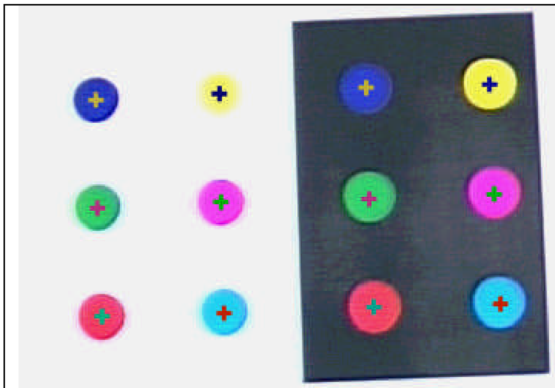


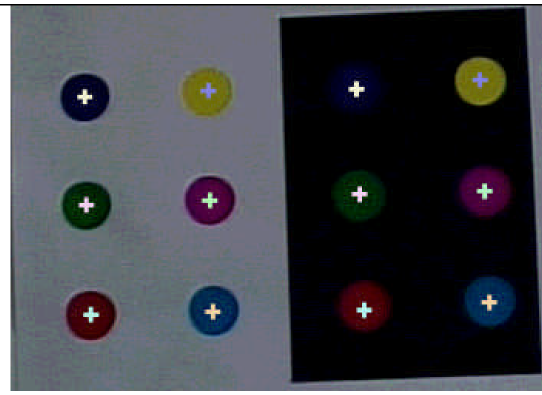
Figure 8 A required homogeneous region as a background of a fiducial with the size of a homogeneous boundary region.

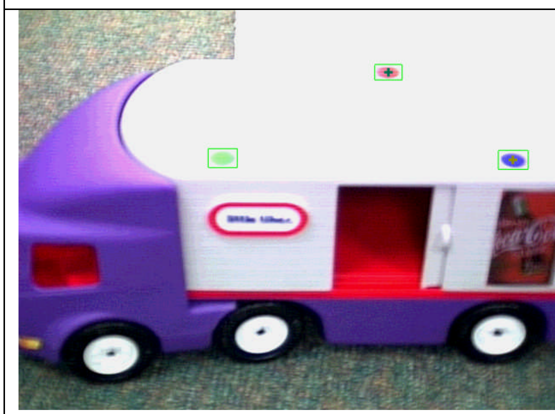
Our fiducial test set contains solid circles with six different colors (red, green, blue, yellow, cyan, and magenta). Test images are captured with a Sony DXC-151A color video camera with 640×480 resolution. We use two different lighting sources, daylight and fluorescent light, and different apertures of a camera to simulate different lighting conditions. The apertures range from 1.8 to 8.0. The algorithm is also tested on images with different backgrounds (Figure 8). Our method is compared with a gradient-based method used in our current AR system. We are exhaustively tested with testing conditions shown at Table 1. The gradient-based method detects most fiducials correctly with f = 2.0 ~ 4.0, but the presented method detects all fiducials under every aperture settings except a yellow fiducial on white background at f = 1.8 and a green fiducial on black background at f = 8.0 and 5.6. We note that screen images of the undetected fiducials are not easily perceived by the human eye either. Figure 9 shows sample images generated by the AR system with our algorithm. Figure 9 also shows that our algorithm detects fiducials under uneven lighting conditions with partial shadows over fiducials. Green rectangles indicate possible fiducials, and detected fiducials with assigned color values are marked with a cross in their center. The green fiducial in Figure 9(c) and the blue fiducial at Figure 9(h) are marked with green rectangles indicating their classification as possible fiducials. The system could not match them to a valid fiducial color so they failed the color verification stage.

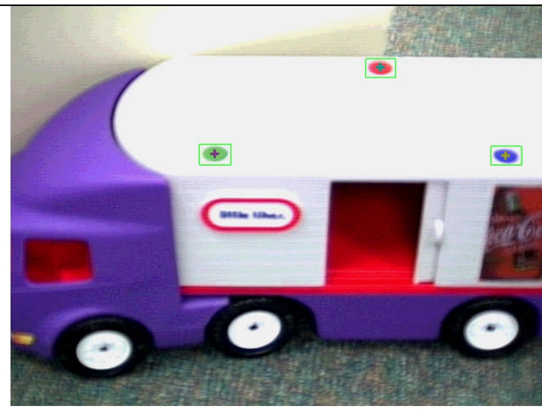
Color of fiducials	Red, Green, Blue, Yellow, Cyan, and Magenta
Number of fiducials	6 (one fiducial per each color)
Backgrounds	White and black backgrounds
Lighting sources	Day light and fluorescent light
Aperture (f)	1.8, 2.0, 2.8, 4.0, 5.6, and 8.0

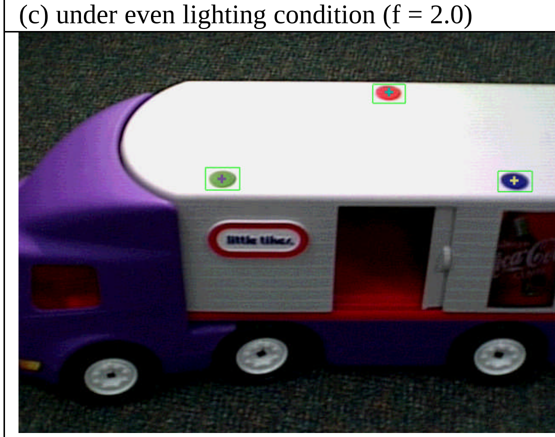
Table 1. Testing conditions


(a) all fiducials under brighter light condition (f = 2.0)

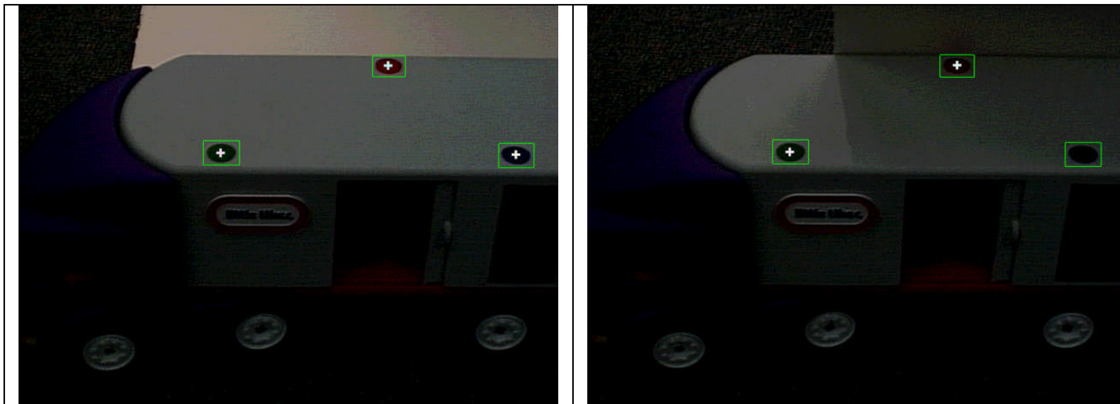

(b) all fiducials under darker lighting conditions (f = 5.6)


(c) under even lighting condition (f = 2.0)


(d) under uneven lighting condition (f = 2.0)


(e) under even lighting condition (f = 4.0)


(f) under uneven lighting condition (f = 4.0)

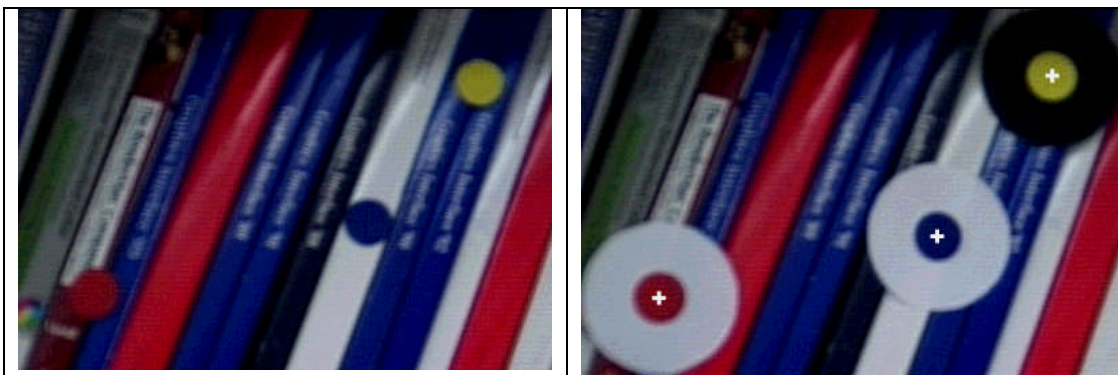


(g) under even lighting condition (f = 8.0) (h) under uneven lighting condition (f = 8.0)

Figure 9 Fiducial detection under varying light conditions

4. DISCUSSION

This paper describes a fiducial detection method that is robust for varied lighting conditions. The algorithm is unique in that it uses rules and membership functions extracted from relations among fiducials and homogeneous backgrounds. The algorithm detects fiducials without the need for any human intervention under varying lighting conditions, including uneven lighting conditions in one image. The algorithm extends the usability of vision-based AR systems.



(a) (b)

Figure 10 (a) Undetectable since there is no homogeneous region around a fiducial (b) One possible solution

The principle weakness of the algorithm is its need for a homogeneous background region around fiducials (Figure 8). If the homogeneous background region is small, the algorithm may fail to detect a fiducial (Figure 10 (a)). One approach to solve this limitation is to define fiducials that contain homogeneous areas around themselves (Figure 10 (b)). With these fiducials, the background is unrestricted.

Our AR system with the presented algorithm operates at four frames per second with images of 640x480 resolutions. The addition of tracking methods (e.g., optical flow) may reduce computation time. Estimates of camera motion or image motion could reduce the coarse searching regions and time, perhaps even allowing the direct application of the fine segmentation method to small regions. Another possible approach is to use our methods as part of a hybrid detection approach. Our algorithms could locate possible regions and adaptively estimate thresholds for use by simple and fast threshold-detection algorithms.

ACKNOWLEDGEMENTS

This research has been funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center with additional support from the Annenberg Center for Communication at the University of Southern California and the California Trade and Commerce Agency. We thank Dr. Suya You and Youngkwan Cho for their ideas and guidance in the implementation.

5. REFERENCES

1. Ronald Azuma, *A survey of augmented reality: course notes #9*, SIGGRAPH, 1995.
2. Michael Bajura and Ulrich Neumann, “Dynamic registration correction in augmented reality systems”, *Proceedings of IEEE Virtual Reality Annual International Symposium*, pp. 189-196, 1995.
3. Danna H. Ballard and Christopher M. Brown, *Computer Vision*, Printice Hall , 1981.
4. Christopher G. Healey and James T. Enns, “A perceptual colour segmentation algorithm”, Technical Report, University of British Columbia, 1997.
5. Athanasios Papoulis, *Probability, random variables, and stochastic processes Third Edition*, McGraw-Hill, New York, 1991.
6. J. Kender, *Saturation, hue, and normalized color: Calculation, digitization effects, and use*, CMU Computer Science Dept., Nov. 1976.
7. D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan, “Real-time vision-based camera tracking for augmented reality application”, Technical Report, 1997.
8. F. Madritsch, F. Leberl, and M. Gervautz, “Camera based beacon tracking: accuracy and applications”, *VRST ’96*, pp.101-108, 1996.
9. J. P. Mellor, “Enhanced reality visualization in a surgical environment”, AI Technical Report 1544, 1995.
10. Ulrich Neumann and Youngkwan Cho, “A self-tracking augmented reality system”, *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 109-115, July 1996.
11. Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark Livington, “Superior augmented reality registration by integrating landmark tracking and magnetic tracking”, *Proceeding of SIGGRAPH ’96*, pp. 439-446, 1996.
12. M. Uenohara and T. Kanade, “Vision-based object registration for real-time image overlay”, *Proceedings of Computer Vision, Virtual Reality, and Robotics in Medicine ’95*, pp. 13-22, 1995