Object Segmentation Using Feature Based Conditional Morphology

M. Raffay Hamid
GVU Center, College of Computing
Georgia Institute of Technology
Atlanta GA 30318
raffay@cc.gatech.edu

Aijaz Baloch; Ahmed Bilal; Nauman Zaffar
Techlogix Inc.
800 West Cummings Park,
Suite 1200, Woburn, MA 01801
{aab;ahmed;nauman}@techlogix.com

Abstract

This paper presents a new technique to segment objects of interest from cluttered background with varying edge densities and illumination conditions from gray scale imagery. An optimal background model is generated and an index of disparity of the objects from this model is computed. This index estimates the disparity, both in terms of edge densities and edge orientation. We introduce Feature Based Conditional Morphology to process the representations that are most likely to belong to the object of interest and obtain a distilled edge map. These edges are linked using N^{th} order interpolation to get the final outline of the object. We compare our approach with 9 contemporary background subtraction algorithms as given in [8]. Our approach shows significant performance advantages and uses only the gray scale images, while the other approaches also need the color images for their algorithms. A comparison with the conventional morphological techniques is also made to highlight the advantages of our algorithms.

1. Introduction

In many cases object extraction is one of the initial steps for object and activity recognition systems [3, 6]. Most of the recent approaches for object extraction [7, 1, 2] use statistical models of the background that, in addition to other inputs, include the color information of the scene to segment the foreground from the background. Other techniques [4] are based on the background disparity model and assume the geometry of the background does not vary over time. This excludes many of the mobile platform applications, where the background is expected to change significantly over time. Our approach takes gray scale images as input with varying illumination conditions and changing background clutter. Moreover, no depth information about the background is a-priori assumed.

Feature Based Conditional Morphology (FBCM) begins by creating a model of the background from a set of training images over a range of varying illumination conditions and clutter taken over a period of time. The target image is compared with the background model and an index of disparity is computed for the entire distribution of edge density and orientation values. The difference image with the smallest index of disparity is registered as the optimal model and is selected for further processing. Conditional morphological filters are used to further clean the difference images. These filters foveate an N-connectivity window around every edgel and calculate the correlation between the edgel and its neighbors as well as between the optimal background. A comprehensive comparative study of several background subtraction techniques has been discussed in [8]. We tested the FBCM algorithm on the benchmark test images for seven canonical problem scenarios provided in [8]. The results were compared with the nine algorithms discussed in [8] to show the performance advantages of our approach. FBCM is also compared with the conventional morphological algorithms to highlight its advantages over the conventional techniques. In the following sections we discuss the details of FBCM algorithm and present results on some benchmark images.

2. Background Model and Edge Extraction

We begin by building a histogram model in which the score of two features, edge density and orientation is kept as they change over a period of time. This is done under the assumption that the background can only be a sub-set of a finite set of reference images. Each pixel acts like a bin and the variation of the different features of respective bins over time results in the background histogram model. The edge-extraction employs Kirsch Edge Operator, which models the gray scale change seen near an edge having various orientations [5]. In our algorithm we use 8-connectivity mask around every pixel. The resulting direction of the edge pixel is quantized into eight values, which are further quantized into four possible levels to establish equivalence between edges characterized by a high-to-low and a low-to-high intensity gradient.

3. Index of Disparity (IOD) Computation

The Index of Disparity (IOD) measures the difference between the input image and the background. We calculate the IOD of the input image with respect to the entire range of feature values of the background model. The set of values which gives the least IOD is considered the optimal background and is selected for further usage. A pixel is declared as an exclusive edge of the image after it passes either an edge-density or orientation sieve as discussed in sections 3.1 and 3.2 respectively. The IOD of the image is given by:

$$IOD = \sum P_{o(EdgeDensity)} \cup P_{o(Orientation)} \tag{1}$$

where $P_{o(EdgeDensity)}$ and $P_{o(Orientation)}$ are the outputs of Edge Density and Orientation Sieves respectively, such that

$$P_o(x,y)_{sieve} = \begin{cases} 1 & pixel(x,y)\ passes\ sieve \\ 0 & otherwise \end{cases} \tag{2}$$

3.1. Edge Density Sieve

Edge density represents the proportion of the edges in a window centered around pixel x, y:

$$E_d(x,y) = \frac{\sum_{y=Y-dY}^{y=Y+dY} \sum_{x=X-dX}^{x=X+dX} p(x,y)}{(2dX+1)(2dY+1)} \tag{3}$$

where dX and dY represent the horizontal and vertical displacements around the pixel in center. The denominator term represents the area of the window.

For a given window size, the edge density is calculated both for the image and the reference. The difference between the edge densities is given by:

$$\triangle E_d = E_d(x,y)|_{image} - E_d(x,y)|_{refImage} \tag{4}$$

If this difference is greater than a particular threshold, then the pixel passes the edge density sieve, i.e.

$$P_o(x,y)|_{EdgeDensitySieve} = \begin{cases} 1 & \triangle E_d > Thresh \\ 0 & otherwise \end{cases} \tag{5}$$

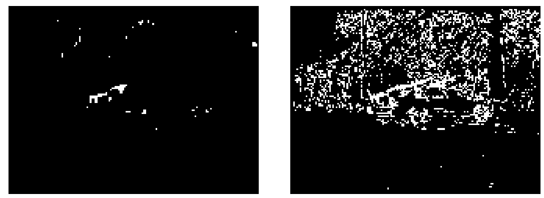
3.2. Orientation Sieve

If a pixel is blocked by the Edge Density Sieve, it can still be a valid edge of the object of interest if the slopes in corresponding windows of the image and the reference are different. The mode of the histogram of slopes of all the pixels in the window both for the image at hand and the reference is considered to represent the dominant slope prevalent in the window. The histogram of the slope distributions in the corresponding windows of the image and the reference can be uni-modal or multi-modal.

If the histogram of the slopes of edges in the corresponding windows of the image and the reference are uni-modal,

(a) (b)

Figure 1. (a) An input image (b) The best reference image selected from the background model.



(a) (b)

Figure 2. (a)The result of conventional background subtraction. (b) The result of Index of Disparity algorithm processing.

and these modes are the same, then the output should be zero, and one otherwise, i.e.

$$P_o(x,y)|_{OrientationSieve} = \begin{cases} 0 & m_{Img} = m_{Ref} \\ 1 & otherwise \end{cases} \tag{6}$$

Incase these histograms are multi-modal, then if the respective modes are the same for both image at hand and the reference image, then the output should be zero, and one otherwise, i.e.

$$P_o(x,y)|_{OrientSieve} = \begin{cases} 0 & m(k)_{Img} = m(k)_{Ref} \\ 1 & otherwise \end{cases} \tag{7}$$

Here k represents the mode of the histogram of slopes. Finally, if only one of these histograms is multi-modal while the other is uni-modal, then the output of the orientation sieve should be one. The underlying motivation behind the above sieve is to dispose of non-object edges on the basis of significant orientation parity between the image and the reference stream. The following example illustrates the IOD computation. The input and the selected optimal model are shown in Figure 1.

Figure 2 (a) shows the results of the conventional background subtraction on the input image with edge density threshold set at 0.5. Figure 2 (b) is the result of processing the same image with the IOD algorithm. The conventional method gets rid of pixels without considering the relative orientation information, whereas IOD algorithm regenerates a significant set of pixels that belong to the object of interest. Although there are some extraneous edges regenerated, the edges that belong to the object of interest have a continuous grade of slopes in their surrounding which is different than the non-object edges. These extraneous edges are removed in the orientation based erosion process as discussed in section 4.1. Moreover, the conventional method of background edge subtraction is much dependent on the threshold and hence requires fine tuning of the threshold for varying illumination conditions. If the same image is passed through the conventional background subtraction algorithm with the threshold set at 0.7, it produces no output edge pixels at all. Similarly, if the threshold is made small a large portion of background remains. On the other hand, IOD algorithm works over a large range of illumination conditions for a single threshold value. If the threshold in IOD is increased to 0.85 in the example shown in Figure 2, it still produces substantial volume of object pixels in the output.

4. Feature Based Conditional Morphology

In its most generic form Conditional Morphology can be characterized by morphological operations carried out upon the veracity of conditions parameterized by features computed in a volume around the pixel in focus. The Feature Based Conditional Iterative Morphology algorithm uses the slope distribution of the neighboring pixels as the feature. The volume exposed to the operation is foveated in an iterative fashion till it reaches a selected threshold. Each iteration is marked by the saturation of a certain volume to bring about any further morphological change. We now explain how these operations are carried out:

The a posteriori probability of finding neighboring edges with similar orientation around a pixel dictates if the edge/non-edge pixel in center needs to be transformed to a non-edge/edge pixel. Two probabilities are calculated in this regard, one is the probability of finding an edge in the neighborhood of the central pixel Ψ_a and the second is the probability of finding a pixel with same orientation as that of the central pixel Ψ_b. These probabilities are given as below:

$$\Psi_a(X,Y) = E_d(X,Y) \tag{8}$$

where $E_d(X,Y)$ represents the edge density as given in (3) around the point (X,Y) and

$$\Psi_b(X,Y) = \frac{\sum_{y=Y-dY}^{y=Y+dY} \sum_{x=X-dX}^{x=X+dX} p(x,y)}{(2dX+1)(2dY+1)} \tag{9}$$

Thus the combined probability of finding neighboring edges with same orientation as that of the pixel in focus can be expressed as:

$$\Psi(X,Y) = \Psi_a \Psi_b \tag{10}$$

(a) (b)

Figure 3. (a)The result of applying conventional erosion on the output of IOD algorithm. (b) The result of the proposed algorithm applied to the output of IOD algorithm.

4.1. Feature Based Conditional Erosion

The objective of this step is to discard (erode) randomly oriented edge pixels from the result obtained from IOD step to nullify the effects of varying illumination conditions. An edge pixel is declared non-edgel if it is randomly distributed noise pixel and this is determined by Ψ as follows:

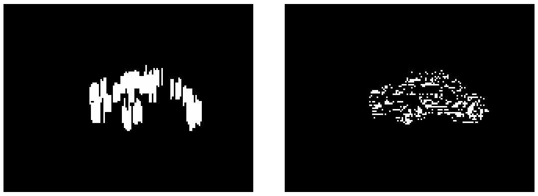
$$Edge(X,Y) = 0; \quad \Psi(x,y) < \Psi_{thresh}(WndSize) \tag{11}$$

where Ψ_{thresh} is a function of the window size for any particular iteration. If the edge sustains the above test, its orientation has no correlation with respect to the edge pixels in its neighborhood. This is determined by $\Psi = \Psi_a\Psi_b$ as follows:

$$Edge(x,y) = 0; \quad \Psi < \Psi_{thresh} \tag{12}$$

This process is carried out for a particular window size until it ceases to discard any further edges and the window size is increased iteratively till a selected threshold.

Figure 3 illustrates the orientation based erosion operation when applied to the result of IOD computation algorithm as shown in Figure 2. The conventional erosion procedure erodes the edge pixels without considering the orientation of the edges in the neighborhood. Moreover, as is the case of conventional background subtraction, where variable threshold is needed to account for varying intensities, a variable window size for conventional erosion is needed. FBCM works well for intensity variations across a range of windows sizes. The output of the conventional erosion operation becomes zero when the window size is increased to 15. On the other hand FBCM gives a substantial volume of the edges belonging to object while getting rid of more extraneous edges as the window size is increased.



(a) (b)

Figure 4. (a) The result of applying conventional dilation. (b) The result of the Feature Based Conditional Dilation.

(a) (b)

Figure 5. (a)The result of edge completion and filling when no prior knowledge about the geometry is used (b) The result when prior knowledge is included.

4.2. Feature Based Conditional Dilation

The objective of this step is to restore (dilate) non-edge pixels with similar orientation as that of their neighboring manifested edges to give allowance to weak edges suppressed during edge detection to be regenerated. A pixel will not be regenerated if the probability of finding edges around it is less than a particular threshold as defined in Equation 5. If so is not the case the pixel can still qualify to be regenerated if the probability of finding edges in the neighborhood with similar orientations as that of the pixel in focus is significant. This probability is governed by the orientation sieve of the corresponding pixel in the best selected reference as defined in Equation 6 and 7. To describe the combined role of all these conditions, we consider another attribute $\Psi^{'}$ as given below:

$$\Psi^{'} = P_o(x,y)\Psi_a\Psi_b \tag{13}$$

If $\Psi^{'} > \Psi^{'}_{thresh}$ then it is assumed that although the edge intensity of the pixel is not strong enough to be detected as an edge, since its orientation is similar to the neighboring pixels while at the same time different from the orientation of the corresponding pixel in the best reference frame, therefore the pixel is transformed into an edgel pixel. Figure 4 illustrate the processing advantages of Orientation Based Conditional Dilation algorithm.

5. Edge Completion and Filling

The final step is to complete the boundary of the object of interest and fill inwards to segment the object. The incorporation of the prior knowledge about the object of interest is useful in the solution of domain specific problems. For example, the prior knowledge about the likelihood of the existence of an edge belonging to the object of interest in a particular portion of the image helps in determining if a pixel belongs to the object of interest. We have developed an algorithm to link edges of a particular object of interest based on the prior information about the geometry of the object. This is done by dividing the image into quadrants, and the quadrants further into sub-quadrants. First of all the volume of edgels in each quadrant and its sub-quadrants is calculated and compared at their respective levels. Sub-quadrants having the least volume in their quadrants are ignored. This is done to suppress the noisy edges that may have sustained the previous tests. The edges at the various levels are parsed, in order to obtain certain control points, which dictate the outer contour. For instance, for a lower left sub-quadrant of the lower left quadrant, the lower left edge pixel is more likely to represent the silhouette than other edge pixels in this sub-quadrant. Similar mappings are carried out for different quadrants of the image, at different levels. This gives a set of C control points, which are joined to give the outer boundary of the object such that

$$C = \{p_1, p_2, ...p_m\} \tag{14}$$

The Cartesian product CXC represents the set of links that must be carried out for completing the boundary:

$$C = \{(p_1,p_1),(p_1,p_2),....(p_1,p_m),....(p_m,p_m)\} \tag{15}$$

All repetitive links are ignored. The linking between any two control-points is carried out using N^{th} order interpolation. The linear displacement between the points determines the number of intermediate control points required to arrive at the curve describing the contour, thus specifying the order of the interpolation. Once the silhouette of the object of interest is found it is filled to produce the final segmentation result.

Figure 5 (a) shows the output of both a naive row based edge completion algorithm when no prior information about the geometry of the object of interest is used. Figure 5 (b) shows the result when prior knowledge of object geometry is used to support the hypothesis that the pixel belongs to the object boundary.

6. Results

The FBCM algorithm was tested on the benchmark images provided in [8]. A comparison of these results was made with the 9 conventional techniques of object extraction on 7 canonical problem scenarios as described in [8]. Since [8] demonstrates that it performs better than 8 background subtraction methods, therefore by showing that our algorithm performs better than [8], we also demonstrate that it works better than the rest of the eight techniques mentioned in [8]. The total error, i.e., the sum of false positives and false negatives is used as the comparison metric. False positive are the number of pixels that do not belong to the object of interest but are extracted as the object. False negatives are the pixels that belong to the object of interest but are presnted as background by the algorithms. We use ideally segmented images for each of the 7 canonical problems to determine the number of false positives and negatives. The color images in [8] were first converted to grey scale images as input to the FBCM algorithm. No prior knowledge about the shape of the object was assumed. Figure 6 shows the output at various processing stages of FBCM algorithm. The results from [8] (see Column 6 for results of [8] in figure 6) and ideal segmentation (last column in figure 6) are also shown. Table 1 shows a summary of the false positives, false negatives and total errors for the seven problem scenarios and the grand total of errors by FBCM algorithm. The last row shows the grand total of errors by [8]. The FBCM algorithm has slightly more (23%) false positives and much fewer (75%) false negatives than [8]. The total error of FBCM is about 29% smaller than [8].

To show both the generizability and how prior information about the geometry of object of interest can be useful, we present various examples in Figure 7. The scenes include both the indoor and outdoor scenarios under varying lighting conditions and background clutter. The final results are presented next to the ideally segmented objects to highlight the accuracy and robustness of the FBCM algorithm.

7. Conclusions

An object segmentation algorithm, FBCM, is discussed that works well under varying illumination conditions and background clutter. The present algorithm uses only the gray scale images as input. The algorithm was applied to benchmark images with 9 canonical object extraction techniques in [8]. The results highlight the advantages of utilizing distilled edges of the object and utilizing both the edge density and orientation information to select the background model from a set of background images. Results are shown at various processing steps and compared with more conventional image processing techniques. An adaptive neighborhood size that responds to the amount of signal variation between the input and reference background images compnesates for object size and texture including the variations due to the lighting conditions. The incorporation of information such as color or a priori knowledge of expected objects is likely to improve accuracy and this is our future course of research.

Table 1. A comparison of the number of erros by FBCM algorithm and [8].

	False -ves	False +ves	Total
Moved Object	0	0	0
Time of Day	182	226	408
Light Switch	1687	223	1910
Weaving Trees	1109	439	1548
Camouflage	1287	263	1550
Foreground Aperture	851	99	950
Bootstrapped	1501	287	1788
FBCM Grand Total	6617	1537	8154
Wallflower Grand Total	5359	6119	11478

References

[1] N. Friedman and S. Russel. Image segmentation in video sequences: A probabilistic approach. *Thirteenth Conference on Uncertainty in Artificial Intelligence*, 1997.

[2] E. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. *Computer Vision and Pattern Recognition*, 1998.

[3] S. Intille, J. Davis, and A. Bobick. Real-time closed-world tracking. *Computer Vision and Pattern Recognition*, pages 697–703, 1997.

[4] Y. Ivanov, A. Bobick, and J. Liu. Fast lightening independent background subtraction. *International Journal of Computer Vision*, 2000.

[5] J.R.Parker. *Algorithms for Image Processing and Computer Vision*.

[6] D. Moore and I. Essa. Exploiting human actions and object context for recognition tasks. *IEEE International Conference on Computer Vision*, 1999.

[7] N. Ohta. A statistical approach to background subtraction for surveillance systems. *IEEE International Conference on Computer Vision*, 2001.

[8] K. Toyama, J. Krumm, B. Brummit, and B. Meyers. Wallflower: Principles and practice of background maintenance. *IEEE International Conference on Computer Vision*, 1999.

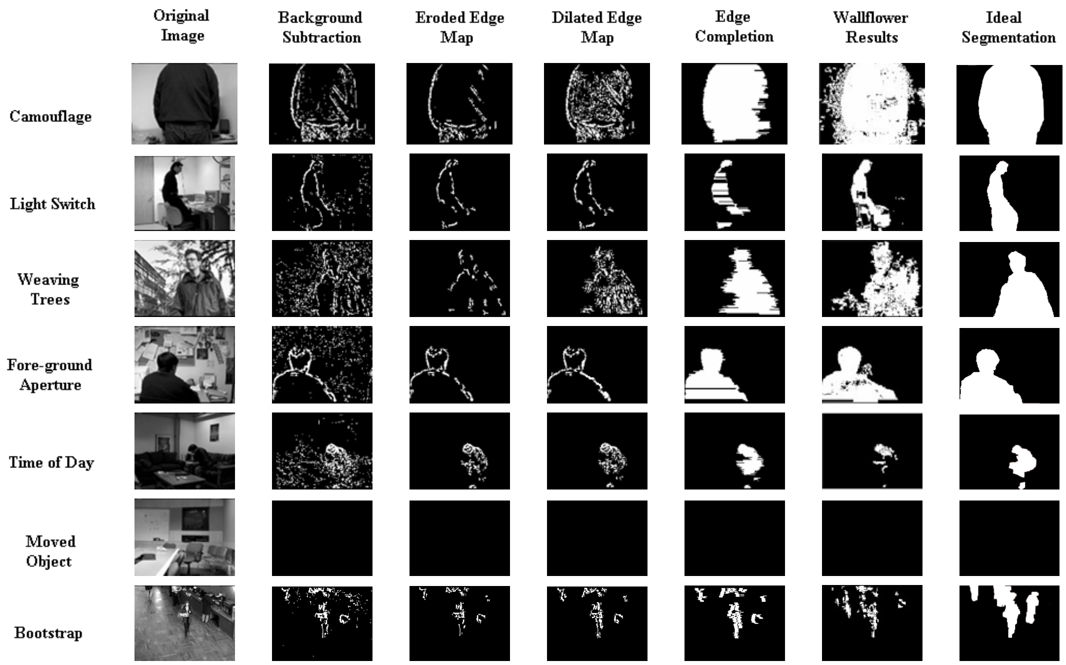


Figure 6. Examples of FBCM algorithm on seven canonical benchmark image sets in [8]. Rows show scenarios and the columns show outputs of various processing stages of FBCM. Wallflower results and ideal segmented results are also shown for comparison.

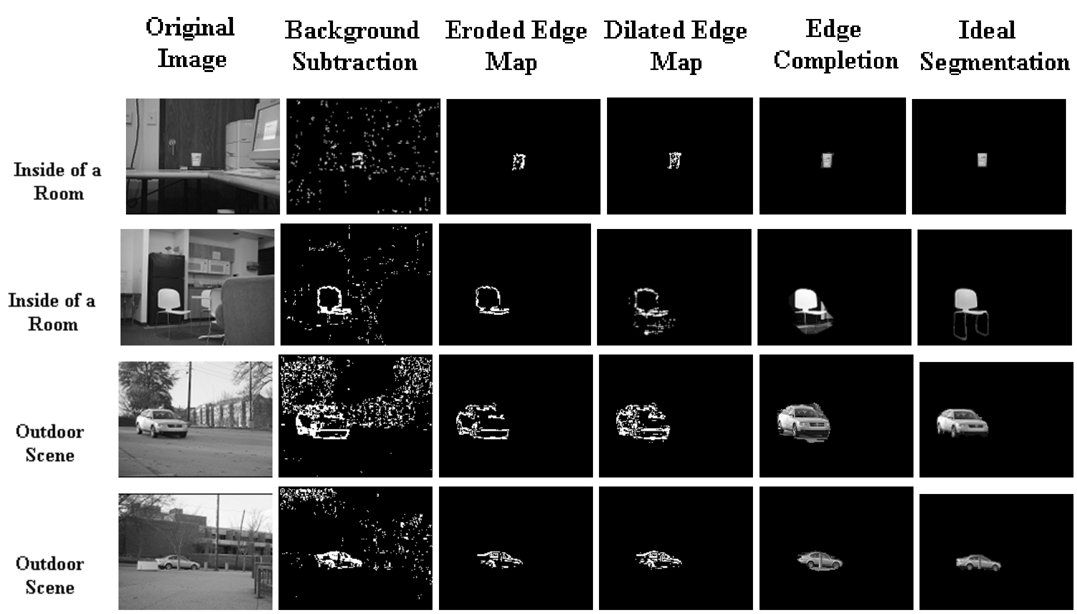


Figure 7. The outputs of various processing stages of the FBCM algorithm under different lighting conditions in indoor and outdoor scenes. The first two rows show indoor images with a glass and a chair as object of interest. The last two rows show show outdoor images with a car as the object of interst. The last column shows hand segmented objects.