Image Segmentation Techniques: Comparison and Improvement

Charbel Fares[1], Alain Bou Malham[2]
[1]Holy Spirit University of Kaslik, Faculty of Sciences, Lebanon.
[2]Digipen Middle East
charbelfares@usek.edu.lb, aboumalh@digipen.edu

Abstract

Region Growing is an image segmentation approach particularly used in Artificial Intelligence in which neighboring pixels are examined and added to a region class if no edges are detected. This process is iterated for each boundary pixel in the region. If adjacent regions are found, a region-merging algorithm is used, in which weak edges are dissolved and strong edges are left intact. Region Growing offers several advantages over conventional segmentation techniques. The algorithm is also very stable with respect to noise. Regions will never contain too much of the background, as long as the parameters are defined correctly. Other techniques that produce connected edges, like boundary tracking, are very unstable. We can take advantage of several image properties, such as low gradient or gray level intensity value, at once. There are, however, several disadvantages to region growing. It is very expensive computationally. It takes both serious computing power (processing power and memory usage) and a decent amount of time to implement the algorithms efficiently. In this paper, we presented a list of segmentation techniques and we proposed a new segmentation algorithm. The results of an objective evaluation of these segmentation techniques are shown in a comparative study. Moreover, we proposed a hybrid variant that combines two techniques. For each of these two techniques, we will examine three characteristics: Correctness, Stability with respect to parameter choice, and Stability with respect to image choice.

Keywords: *Image Segmentation, Hybrid Approach, Region Growing, Region Merging, Mean Graph.*

1. INTRODUCTION

In computer vision, segmentation refers to the process of partitioning a digital image into multiple regions. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as color, intensity, or texture.

Segmented images are now used routinely in a multitude of different applications, such as, diagnosis, treatment planning, in the robotics, localization of pathology, geology, study of anatomical structure, meteorology, computer-integrated surgery, among others. However, image segmentation remains a difficult task due to both the variability of object shapes and the variation in image quality. In spite of the most complex algorithms developed nowadays, segmentation continues being very dependent on the application. With the aim of obtaining segmentation methods more exact and more effective, several techniques have been proposed in the literature. Unfortunately, segmentation is a complex problem with no exact solution. Noise and other image artifacts can cause incorrect regions or boundary discontinuities in segmented objects.

The main challenge in object detection is the amount of variation in visual appearance. For example, cars vary in shape, size, coloring, and in small details such as the headlights, grill, and tires. Visual appearance also depends on the surrounding environment. Light sources will vary in their location with respect to the object, their intensity, and their color. Nearby objects may cast shadows on the object or reflect additional light on the object. The appearance of the object also depends on its pose; that is, its position and orientation with respect to the camera. For example, a human face will look much different when viewed from the side than viewed frontally. An object detector much accommodates all these variations and still distinguishes the object from any other patterns that may occur in the visual world.

The rest of the paper is structured as follows. In section 2 the proposed algorithm is shown. The evaluation and comparison are shown in section 3. Finally, conclusions are sketched in Section 4.

2. PROPOSED ALGORITHM: MEAN GRAPH

Unsupervised image segmentation algorithms have matured to the point that they provide segmentations which agree to a large extent with human intuition. The time has arrived for these segmentations to play a larger role in object recognition. It is clear that unsupervised segmentation can be used to help cue and refine various recognition algorithms. However, one of the stumbling blocks that remain is that it is unknown exactly how well these segmentation algorithms perform from an objective standpoint.

Most presentations of segmentation algorithms contain superficial evaluations which merely display images of the segmentation results and appeal to the reader's intuition for evaluation. There is a consistent lack of numerical results, thus it is difficult to know which segmentation algorithms present useful results and in which situations they do so. Appealing to human intuition is convenient, however if the algorithm is going to be used in an automated system then objective results on large datasets are to be desired.

We present the results of an objective evaluation of two popular segmentation techniques: the mean shift [1] and the graph-based segmentation algorithms [2]. As well, we look at a hybrid variant that combines these algorithms. For each of these algorithms, we examine three characteristics:

1. *Correctness*: the ability to produce segmentations which agree with human intuition. That is segmentations which correctly identify structures in the image at neither too fine nor too coarse level of detail.

2. *Stability with respect to parameter choice*: the ability to produce segmentations of consistent correctness for a range of parameter choices.

3. *Stability with respect to image choice*: the ability to produce segmentations of consistent correctness using the same parameter choice on a wide range of different images.

If a segmentation scheme satisfies these three characteristics, then it will give useful and predictable results which can be reliably incorporated into a larger system.

2.1 Segmentation Algorithms

We have chosen to look at mean-graph segmentation as it is generally effective. The efficient graph-based segmentation algorithm was chosen as an interesting comparison to the mean shift in that its general approach is similar; however it excludes the mean shift filtering step itself, thus partially addressing the question of whether the filtering step is useful. The combination of the two algorithms is shown as an attempt to improve the performance and stability of either one alone. Then we describe each algorithm and further discuss how they differ from one another.

2.1.1 Mean Shift Segmentation

The mean shift segmentation technique is one of many techniques under the heading of ``feature space analysis". The mean shift technique is comprised of two basic steps: a mean shift filtering of the original image data (in feature space), and a subsequent clustering of the filtered data points. Below we will briefly describe each of these steps and then discuss some of the strengths and weaknesses of this method.

- **Filtering:** The filtering step of the mean shift segmentation algorithm consists of analyzing the probability density function underlying the image data in feature space. Consider the feature space consisting of the original image data represented as the (x, y) location of each pixel, plus its color in L*u*v* space (L*, u*, v*). The modes of the probability density function (pdf) underlying the data in this space will correspond to the locations with highest data density. In terms of segmentation, it is intuitive that the data points close to these high density points (modes) should be clustered together. Note that these modes are also far less sensitive to outliers than the means of, say, a mixture of Gaussians would be.

The mean shift filtering step consists of finding the modes of the underlying pdf and associating with them any points in their basin of attraction. Unlike earlier techniques, the mean shift is a non-parametric technique and hence, we will need to estimate the gradient of the pdf, f(x), in an iterative manner using kernel density estimation to find the modes. For a data point x in feature space, the density gradient is estimated as being proportional to the mean shift vector:

$$\widehat{\nabla} f(\mathbf{x}) \propto \frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|\right)} - \mathbf{x} \qquad (1)$$

where x_i are the data points, x is a point in the feature space, n is the number of data points (pixels in the image), and g is the profile of the symmetric kernel G. We use the simple case where G is the uniform kernel with radius vector h. Thus, the above equation simplifies to:

$$\widehat{\nabla} f(\mathbf{x}) \propto \left[\frac{1}{|S_{\mathbf{x},h_s,h_r}|} \sum_{\mathbf{x}_i \in S_{\mathbf{x},h_s,h_r}} \mathbf{x}_i\right] - \mathbf{x} \qquad (2)$$

where S_x,h_s,h_r represents the sphere in feature space centered at x and having spatial radius h_s and color (range) radius h_r, and the x_i represent the data points within that sphere. For every data point (pixel in the original image) x we can iteratively compute the gradient estimate in Eqn. 2 and move x in that direction, until the gradient is below a threshold. Thus we have found the points where $\widehat{\nabla} f(\mathbf{x}') = 0$, the modes of the density estimate. We can then replace the point x with x0, the mode with which it is associated. Finding the mode associated with each data point helps to smooth the image while preserving discontinuities. Intuitively, if two points xi and xj are far from each other in feature space, then xi $\notin$ $S_{x,hS,hr}$ and hence xj doesn't contribute to the mean shift vector gradient estimate and the trajectory of xi will move it away from xj . Hence, pixels on either side of a strong discontinuity will not attract each other. However, filtering alone does not provide segmentation as the modes found are noisy. This "noise" stems from two sources. First, the mode estimation is an iterative process; hence it only converges to within the threshold provided (and with some numerical error). Second, consider an area in feature space larger than $S_{xj,hs,hr}$ and where the color features are uniform or have a gradient of 1. Since the pixel coordinates are uniform by design, the mean shift vector will be 0 in this region, and the data points will not move and hence not converge to a single mode. Intuitively, however, we would like all of these data points to belong to the same cluster in the final segmentation. For these reasons, mean shift filtering is only a preprocessing step, and a second step is required in the segmentation process: clustering of the filtered data points {x'}.

- **Clustering:** After mean shift filtering, each data point in the feature space has been replaced by its corresponding mode. As described above, some points may have collapsed to the same mode, but many have not despite the fact that they may be less than one kernel radius apart. Clustering is described as a simple post-processing step in which any modes that are less than one kernel radius apart are grouped together and their basins of attraction are merged. This suggests using single linkage clustering, which effectively converts the filtered points into segmentation [3].

- **Discussion:** Mean shift filtering using either single linkage clustering or edge-directed clustering produces segmentations that correspond well to human perception. This algorithm is quite sensitive to its parameters. The mean shift filtering stage has two parameters corresponding to the bandwidths (radii of the kernel) for the spatial (hs) and color (hr) features. Slight variations in hr can cause large changes in the granularity of the segmentation, as shown in Figure 1. By adjusting the color bandwidth we can produce over-segmentations as in Figure 1-b which shows every minute detail, to reasonably intuitive segmentations as in Figure 1-f which delineate objects or large patches, to under-segmentations as in Figure 1-g which obscure the important elements completely. This issue is a major stumbling block with respect to using mean shift segmentation as a reliable preprocessing step for other algorithms, such as object recognition. For an object recognition system to actually use a segmentation algorithm, it requires that the segmentations produced be fairly stable under parameter changes and that the same parameters produce stable results for different images, thus easing the burden of parameter tuning.

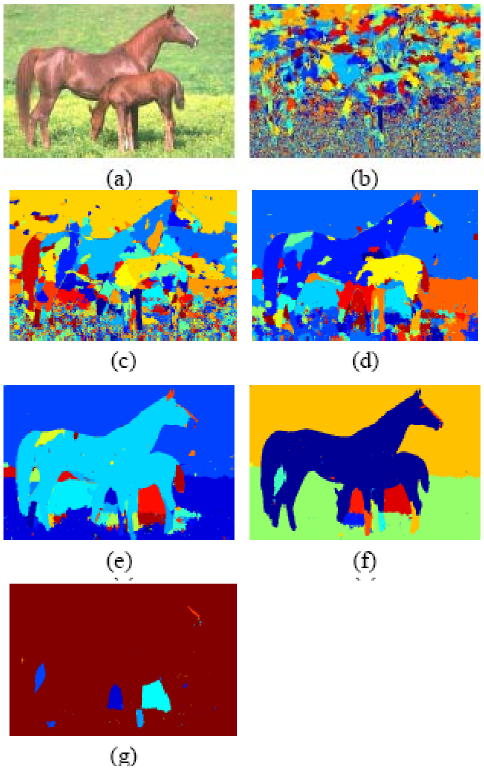

Figure 1: *Changing scores for different segmentation granularities: (a) Original image, (b)-(g) mean shift segmentations using scale bandwidth (hs) 7 and color bandwidths (hr) 3, 7, 11, 15, 19 and 23 respectively.*

2.1.2 Efficient Graph-based Segmentation

Efficient graph-based image segmentation is another method of performing clustering in feature space. This method works directly on the data points in feature space, without first performing a filtering step, and uses a variation on single linkage clustering. The key to the success of this method is adaptive threshold. To perform traditional single linkage clustering, a minimum spanning tree of the data points is first generated (using Kruskal's algorithm [4]), from which any edges with length greater than a given hard threshold are removed. The connected components become the clusters in the segmentation. This method eliminates the need for a hard threshold, instead of replacing it with a data-dependent term. More specifically, let G = (V,E) be a (fully connected) graph, with m edges and n vertices. Each vertex is a pixel, x, represented in the feature space. The final segmentation will be S = (C1, ...,Cr) where Ci is a cluster of data points. The algorithm is shown in table 1.

We can make the algorithm more efficient by considering only the 100 shortest edges from any vertex instead of the fully connected graph. This does not result in any perceptible quality loss. In contrast to single linkage clustering which uses a constant K to set the threshold on edge length for merging two components in Eqn. 3, efficient graph-based segmentation uses a variable threshold. This threshold effectively allows two components to be merged if the minimum edge connecting them does not have length greater than the maximum edge in either of the components' minimum spanning trees, plus a term. $\tau = \frac{k}{|C^{t-1}_{\mathbf{x}_i}|}$. as defined here, τ is dependent on a constant k and the size of the component. Note that on the first iteration, li = 0 & lj = 0, and $|C^0_{\mathbf{x}_i}| = 1$ & $|C^0_{\mathbf{x}_j}| = 1$. So k represents the longest edge which will be added to any cluster at any time, k = lmax. Also, as the number of points in a component increases, the tolerance on added edge length for new edges becomes tighter and fewer mergers are performed, thus indirectly controlling region size. However, it is possible to use any non-negative function for τ which reflects the goals of the segmentation system. The merging criteria in Eqn. 3 allows efficient graph-based clustering to be sensitive to edges in areas of low variability, and less sensitive to them in areas of high variability. This is intuitively the property we would like to see in a clustering algorithm. However, the results it gives do not have the same degree of correctness as mean shift-based segmentation, as demonstrated in Fig. 2. This algorithm also suffers somewhat from sensitivity to its parameter, k.

Table 1: *Algorithm of Graph-Based Segmentation*

1. Sort $E = (e_1, ..., e_m)$ such that $|e_t| \leq |e_{t'}|\ \forall t < t'$
2. Let $S^0 = (\{\mathbf{x}_1\}, ..., \{\mathbf{x}_n\})$, in other words each initial cluster contains exactly one vertex.
3. For $t = 1, ..., m$
   (a) Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the vertices connected by e_t.
   (b) Let $C^{t-1}_{\mathbf{x}_i}$ be the connected component containing point $\mathbf{x}_i$ on iteration $t-1$, and $l_i = \max_{\text{mst}} C^{t-1}_{\mathbf{x}_i}$ be the longest edge in the minimum spanning tree of $C^{t-1}_{\mathbf{x}_i}$. Likewise for l_j.
   (c) Merge $C^{t-1}_{\mathbf{x}_i}$ and $C^{t-1}_{\mathbf{x}_j}$ if
   $$|e_t| < \min\{l_i + \frac{k}{|C^{t-1}_{\mathbf{x}_i}|}, l_j + \frac{k}{|C^{t-1}_{\mathbf{x}_j}|}\} \tag{3}$$
   where k is a constant.
4. $S = S^m$

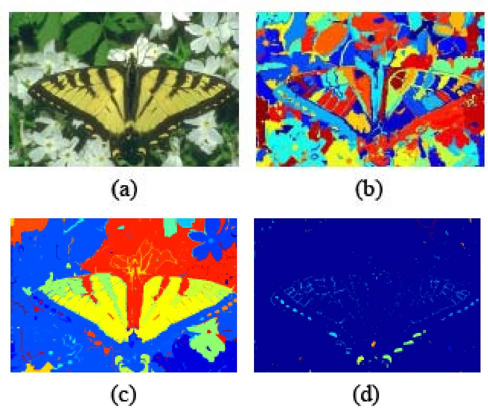

Figure 2: *Changing scores for different parameters using efficient graph-based segmentation: (a) Original image, (b)-(d) efficient graph-based segmentations using scale bandwidth (hs) 7, color bandwidth (hr) 7 and k values 5, 25, and 125 respectively.*

2.1.3 Hybrid Segmentation Algorithm

An obvious question emerges when describing the mean shift based segmentation method and the efficient graph based clustering method: can we combine the two methods to give better results than either method alone? More specifically, can we combine the two methods to create more stable segmentations that are less sensitive to parameter changes and for which the same parameters give reasonable segmentations across multiple images? In an attempt to answer these questions, the third algorithm we consider is a combination of the previous two algorithms: first we apply mean shift filtering, and then we use efficient graph-based to give the final segmentation. The result of applying this algorithm with different parameters can be seen in Fig 3. Notice that for hr = 15 the quality of the segmentation is high. Also notice that the rate of granularity change is slower than either of the previous two algorithms, even.

3. EVALUATION AND COMPARISON

The first comparison we performed considered the correctness of the fourth algorithms. All three algorithms had the potential to perform equally well on the dataset given the correct parameter choice. On average over the parameter set, however, the hybrid algorithm performed slightly better than the mean shift algorithm, and both performed significantly better than the graph-based segmentation. We can conclude that the mean shift filtering step is indeed useful, and that the most promising algorithms are the mean shift segmentation and the hybrid algorithm.

The second comparison we performed considered stability with respect to parameters. In this comparison, the hybrid algorithm showed less variability when its parameters were changed than the mean shift segmentation algorithm. Although the amount of improvement did decline with increasing values of k, the rate of decline was very slow and any choice of k within our parameter set gave reasonable results. Although the graph-based segmentation did show very low variability with k = 5, changing the value of k decreased its stability drastically.

Finally, we compared the stability of a particular parameter choice over the set of images. Once again, we see that the graph-based algorithm has low variability when k = 5, however its performance and stability decrease rapidly with changing values of k. The comparison between the mean shift segmentation and the hybrid method is much closer here, with neither performing significantly better. For the three characteristics measured, we have demonstrated that both the mean shift segmentation and hybrid segmentation algorithms can create realistic segmentations with a wide variety of parameters; however the hybrid algorithm has slightly improved stability.

The complexity of our proposed algorithm (Mean-Graph) is O(n*m).

4. CONCLUSIONS

In this article we compared and evaluated image segmentation algorithms. Our work consists of comparing the performance of segmentation algorithms based on three important characteristics: correctness, stability with respect to parameter choice, and stability with respect to image choice. If an algorithm performs well with respect to all of these characteristics, it has the potential to be useful as part of a larger vision system. For our comparison task, we chose to compare two popular segmentation algorithms: mean shift-based segmentation and graph-based segmentation scheme. We also proposed a hybrid algorithm which first performs the first stage of mean shift-based segmentation, mean shift filtering, and then applies the graph-based segmentation scheme, as an attempt to create an algorithm which preserves the correctness of the mean shift-based segmentation but is more robust with respect to parameter and image choice. Thus, we would choose to incorporate the hybrid method into a larger system.

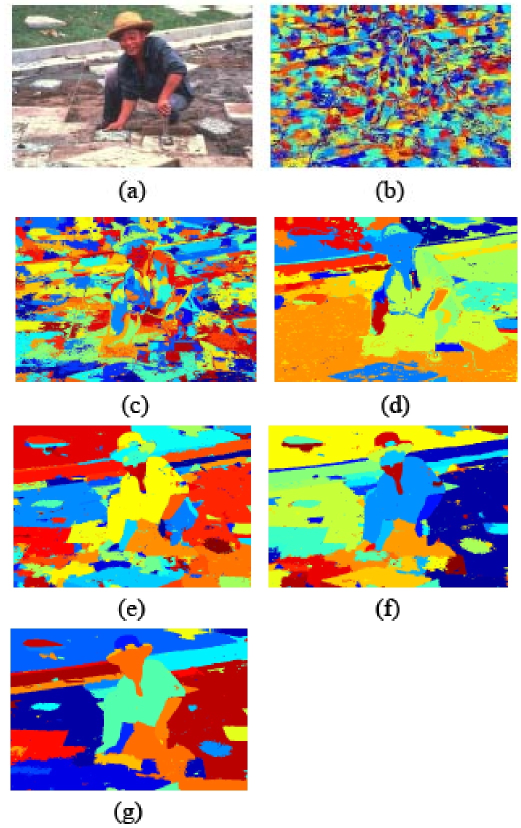

Figure 3: *Changing scores for different parameters using a hybrid segmentation algorithm which first performs mean shift filtering and then efficient graph-based segmentation: (a) Original image, (b)-(g) segmentations using scale bandwidth (hs) 7, and color bandwidth (hr) and k value combinations (3, 5), (3, 25), (3,125), (15, 5), (15, 25), (15,125) respectively.*

5. REFERENCES

[1] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis", IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24, pp. 603-619

[2] P. Felzenszwalb, D. Huttenlocher, "Efficient Graph-Based Image Segmentation", Intl Journal of Computer Vision, 2004, 59 (2).

[3] C. Christoudias, B. Georgescu, P. Meer, "Synergism in Low Level Vision", Intl Conf on Pattern Recognition, 2002, 4, pp. 40190.

[4] Leo Grady and Eric L. Schwartz; "Isoperimetric Graph Partitioning for Data Clustering and Image Segmentation" IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. XX, 2004.