# Synergism in Low Level Vision

Christopher M. Christoudias[1], Bogdan Georgescu[2] and Peter Meer[1,2]
(1) Electrical and Computer Engineering; (2) Computer Science Department
Rutgers University, Piscataway, NJ, 08854-8058, USA
cmch, georgesc, meer@caip.rutgers.edu

**Abstract**

*Guiding image segmentation with edge information is an often employed strategy in low level computer vision. To improve the trade-off between the sensitivity of homogeneous region delineation and the oversegmentation of the image, we have incorporated a recently proposed edge magnitude/confidence map into a color image segmenter based on the mean shift procedure. The new method can recover regions with weak but sharp boundaries and thus can provide a more accurate input for high level interpretation modules. The* Edge Detection and Image SegmentatiON *(EDISON) system, available for download, implements the proposed technique and provides a complete toolbox for discontinuity preserving filtering, segmentation and edge detection.*

## 1 Introduction

Two of the most important low level vision operations are *image segmentation* and *edge detection*. In this paper both are considered to be based on the same, piecewise constant image structure model. The two operations are complementary in nature. Image segmentation focuses on *global* information and labels the input into homogeneous regions. Edge detection, on the other hand, focuses on *local* information and labels the pixels which are assumed to be located at discontinuities. In principle, both operations should give the same result, the edges corresponding to the boundaries of the homogeneous regions. In practice, however, the results differ significantly since local and global evidence may lead to different conclusions.

Combining the outputs of image segmentation and edge detection to improve the quality of the segmented image, is an old idea. In a recent survey paper [5] seven different strategies were distinguished for combining similarity (region) and discontinuity (edge) information. They were divided into two classes: embedded integration and post-processing integration. In the former case the discontinuity information is used during the delineation process, while in the latter case it is employed only to control the fusion and/or refinement of already delineated patches. Both approaches are present in the technique we are proposing, since the information provided by the edge detector is integrated differently into the two modules of the image segmenter: filtering and fusion.

The discontinuity and homogeneity informations were associated in many different ways in the lierature, here we only mention a few of the approaches. Region growing can be guided by the edge map, e.g., [8] [12] for gray level, and [4] for color images. (Note that the extensive anisotropic diffusion literature usually targets the issue of image smoothing and not segmentation.) Boundary information can be used in the labeling process by incorporating it into a vector field derived from a color image, e.g., [9], [14]. The edge map can be employed to refine the delineated region boundaries, e.g., [1]. Information about homogeneity and discontinuity can be also captured by statistical measures and fused under Bayesian rules, e.g., [6], [7].

We use two recently proposed techniques which facilitate a more versatile combination of edge detection with image segmentation. The mean shift based image segmentation [2] is reviewed in Section 2. A generalization of the traditional Canny edge detection procedure which also employs the confidence in the presence of an edge [11] is reviewed in Section 3. The new, combined image segmentation procedure is described in Section 4 and experimental results are shown in Section 5. A short description of the *Edge Detection and Image SegmentatiON* (EDISON) system implementing the proposed technique is given in Section 6.

## 2 Image Segmentation with Mean Shift

A large class of image segmentation algorithms are based on feature space analysis. In this paradigm the pixels are mapped into a color space and clustered, with each cluster delineating a homogeneous region in the image. In the traditional clustering techniques the feature space is modeled as a mixture of multivariate normal distributions,

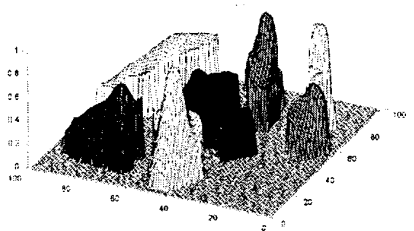


Figure 1: Mode and basin of attraction based classification of a complex feature space. The clusters were shifted apart for better visibility.

1051-4651/02 $17.00 © 2002 IEEE

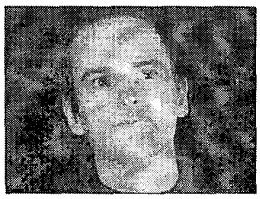


Figure 2: The gray level *face* image used in the examples.

which can introduce severe artifacts due to the elliptical shape imposed over the clusters or due to an error in determining their number.

The *mean shift* based nonparametric feature space analysis eliminates these artifacts. Let $f(\mathbf{x})$ be the (unknown) probability density function underlying a $p$-dimensional feature space, and $\mathbf{x}_i$ the available data points in this space. Under its simplest formulation, the mean shift property can be written as

$$\widehat{\nabla f(\mathbf{x})} \sim \left( \underset{\mathbf{x}_i \in S_{h,\mathbf{x}}}{\text{ave}} [\mathbf{x}_i] - \mathbf{x} \right) \tag{1}$$

where $S_{h,\mathbf{x}}$ is the $p$-dimensional hypersphere with radius $h$ centered on $\mathbf{x}$. Relation (1) states that the estimate of the density gradient at location $\mathbf{x}$ is proportional to the offset of the mean vector computed in a window, from the center of that window. The mean shift property was introduced in pattern recognition in 1975, and was recently applied to several computer vision problems. See [3] for a detailed presentation.

Recursive application of the mean shift property yields a simple *mode detection* procedure. The modes are the local maxima of the density, i.e., $\nabla f(\mathbf{x}) = 0$. They can be found by moving at each iteration the window $S_{h,\mathbf{x}}$ by the mean shift vector, until the magnitude of the shifts becomes less than a threshold. The procedure is guaranteed to converge [3].

When the mean shift procedure is applied to every point in the feature space, the points of convergence aggregate in groups which can be merged. These are the detected modes, and the associated data points define their *basin of attraction*. The clusters are delineated by the boundaries of the basins, and thus can have arbitrary shapes. The number of significant clusters present in the feature space is automatically determined by the number of significant modes detected. See the example in Figure 1 for the decomposition of a complex 2D feature space which was represented through its underlying density function.

In the color image segmentation algorithm proposed in [2] a five-dimensional feature space was used. The $L^*u^*v^*$ color space was employed since its metric is a satisfactory approximation to Euclidean, thus allowing the use of spherical windows. The remaining two dimensions were the lattice coordinates. A cluster in this 5D feature space thus

(a) (b)

Figure 3: Mean shift segmentation of the *face* image. (a) The input with the region boundaries overlayed. (b) Segmented image.

(a) (b)

Figure 4: Segmentation at a higher resolution than in Figure 3.

contains pixels which are not only similar in color but also contiguous in the image.

The quality of segmentation is controlled by the spatial $h_s$, and the color $h_r$, resolution parameters defining the radii of the (3D/2D) windows in the respective domains. The segmentation algorithm has two major steps. First, the image is *filtered* using mean shift in 5D, replacing the value of each pixel with the 3D (color) component of the 5D mode it is associated to. Note that the filtering is discontinuity preserving. In the second step, the basins of attraction of the modes, located within $h_r/2$ in the color space are recursively *fused* until convergence. The resulting large basins of attraction are the delineated regions, and the value of all the pixels within are set to their average. See [2] and [3] for a complete description and numerous examples of the segmentation algorithm. It is important to emphasize that the segmenter processes gray level and color images in the same way. The only difference is that in the former case the feature space has three dimensions, the gray value and the lattice coordinates.

The mean shift based color image segmentation is already popular in the computer vision community and several implementations exist. To optimize performance we have reimplemented the technique. In the filtering step, a speed-up of about five times relative to the original segmenter was obtained by not applying the mean shift procedure to the pixels which are on the mean shift trajectory of another (already processed) pixel. These pixels were directly associated with the mode to which the path converged. The approximation does not yield a visible change in the filtered image.

In the fusion step, extensive use was made of region ad-

jacency graphs (RAG) and graph contraction with a *union-find* algorithm [13, pp.441–449]. The initial RAG was built from the filtered image, the modes being the vertices of the graph and the edges were defined based on four-connectivity on the lattice. The fusion was performed as a transitive closure operation on the graph, under the condition that the color difference between two adjacent nodes should not exceed $h_r/2$. At convergence, the color of the regions was recomputed and the transitive closure was again performed. After at most three iterations the final labeling of the image (segmentation) was obtained. Small regions (the minimum size, $M$ is defined by the user) were then allocated to the nearest neighbor in the color space. Note that this postprocessing step can be refined by employing a look-up table which captures the relation between the smallest significant color difference and the minimum region size.

The new implementation of the color image segmenter was also tested for equivariance under 90° rotations on the lattice. That is, when the input image is rotated the segmented image rotates accordingly. This property assures that the output of the processing does not depend on the order in which the pixels in the image are processed.

The $320 \times 240$ gray level *face* image (Figure 2) is typical for the class of images used in face recognition/tracking applications. The relative small dynamic range, the presence of highlights on the face, the shadows around the chin, etc., make feature extraction from such gray level images challenging.

Indeed, when the image is segmented at a lower resolution ($h_s = 7, h_r = 4.5, M = 20$) only very few facial features are recovered (Figure 3), which may not be satisfactory in a tracking application. On the other hand, when the resolution is slightly increased ($h_s = 7, h_r = 4.0, M = 20$) a significant clutter (nonsalient regions) appears, but important features such as the chin or the full contour of the mouth are still missed (Figure 4). Note the sensitivity of the gray level image segmentation to the value of $h_r$.

## 3 Edge Detection with Embedded Confidence

Edge detection is maybe the most investigated low level vision operation. While a large number of techniques were proposed, today the gradient based edge detectors are the ones most frequently used. They have three processing steps applied in sequence: gradient estimation, nonmaxima suppression and hysteresis thresholding. The edge map is derived from the input based on two gradient magnitude thresholds. However, using the gradient magnitude for decisions causes a well known deficiency, sharp edges with small magnitudes can be detected only at the expense of allowing a large amount of edge clutter. A recently proposed generalization of the gradient based edge detection procedure eliminates this trade-off [11].

The idea behind the new method is illustrated in Figure 5. Assume that the two differentiation masks employed by the gradient operator are defined in an $n \times n$ window. These masks, together with the data in the window, can be represented as three vectors in $\mathcal{R}^{n^2}$. The two vectors corresponding to the masks define the gradient subspace (a hyperplane), while the data is an arbitrary vector in $\mathcal{R}^{n^2}$.

Computation of the gradient vector is equivalent to projecting the data into the subspace, its orientation $\theta$ being the angle between the projected data and one of the mask vectors. Note that only the part of the data in the gradient subspace is employed when computing the gradient vector.

The parameter $\theta$ can be used to generate an ideal edge prototype, i.e., a unit step-edge passing through the center of the $n \times n$ window and oriented at $\theta$. The value of a pixel of the prototype is obtained by integrating across its cross-section. The edge prototype is also a vector in $\mathcal{R}^{n^2}$ and in general will not be located in the gradient subspace, though, by definition must be in the plane of the projection and the data. The prototype is the template of the normalized pattern which would be present in the optimal case. Thus, $\eta$ the cosine of the angle between the data and the template, measures the confidence in the presence of an edge obeying the assumed model. The critical observation is that $\eta$ is computed in $\mathcal{R}^{n^2}$, thus including *new* information from the $(n^2-2)$-dimensional orthogonal complement of the gradient subspace. Therefore, $\eta$ is a measure *independent* of the gradient magnitude.

Let $0 < \rho \leq 1$ be the normalized ranks of the gradient magnitude values, i.e., the percentiles of their cumulative distribution. For each pixel two values are now available: $\rho$ and the confidence $\eta$. Since they are independent, a $\rho\eta$–diagram can defined. In Figure 6a the $\rho\eta$–diagram of the *face* image is shown. Note the presence of many weak (small $\rho$) but accurate step-edges (large $\eta$).

In [11] the nonlinear processing steps of the the gradient based edge detection, nonmaxima suppression and hysteresis thresholding, were generalized to exploit all the information available in the $\rho\eta$–diagram. Instead of gradient magnitude thresholds two *thresholding curves* were used, and the decisions were taken based on the sign of the algebraic distances of a point from these curves. The traditional

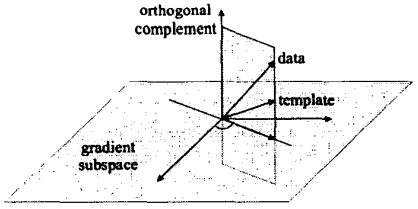
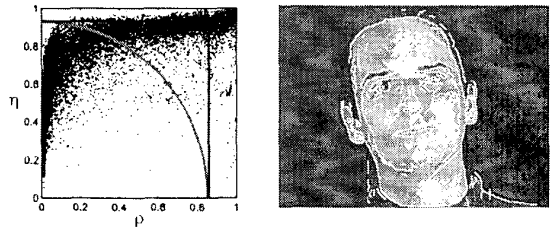



Figure 5: The principle of embedded confidence generation.




(a) (b)

Figure 6: The edge detection with embedded confidence of the *face* image. (a) The $\rho\eta$–diagram, with the employed hysteresis thresholding curves. (b) The input with the detected edges overlayed.

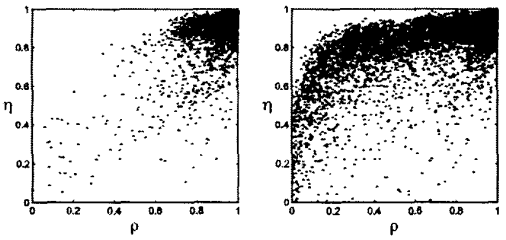
procedure which is based exclusively on gradient magnitude, i.e. $\rho$, then is equivalent to using two vertical lines as thresholding curves.

The new edge detection technique can retain pixels on sharp but weak edges which will be located in the upper-left corner of the $\rho\eta$–diagram (Figure 6a). In resulting edge map of the *face* image (Figure 6b) the contours of the chin and the mouth are detected. See [11] for a more detailed treatment of edge detection with embedded confidence and numerous examples.

## 4 Synergetic Image Segmentation

From the previous two sections we can conclude that using only the gradient magnitude information to guide the image segmentation is not the optimal strategy. Regions with sharp but weak boundaries may still remain undetected since their boundaries may not be adequately supported by the gradient.

Analyzing the $\rho\eta$–diagrams of the pixels located on the boundaries of the delineated regions further strengthens this conclusion. When the *face* image is undersegmented (Figure 3) all the boundary pixels are located in the upper-right corner (Figure 7a), i.e., have high gradient magnitudes. However, once the image is oversegmented (Figure 4) the boundary pixels are nonselectively dispersed everywhere in the $\rho\eta$–diagram (Figure 7b).




(a) (b)

Figure 7: The $\rho\eta$–diagrams of the boundary pixels in the segmented *face* images. (a) Undersegmentation, Figure 3a. (b) Oversegmentation, Figure 4a.

To incorporate all the information available in the $\rho\eta$–diagram, the pixel $(i, j)$ is associated with the weight $w_{ij}$,

(a) (b)

Figure 8: Synergetic segmentation of the *face* image. The same mean shift segmentation parameters were used as in Figure 3. (a) The input with the region boundaries overlayed. (b) Segmented image.

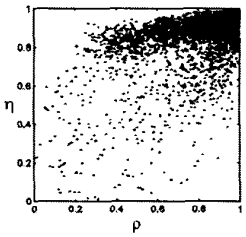



Figure 9: The $\rho\eta$–diagram of the boundary pixels in the synergetic segmentation, Figure 8a.

computed as

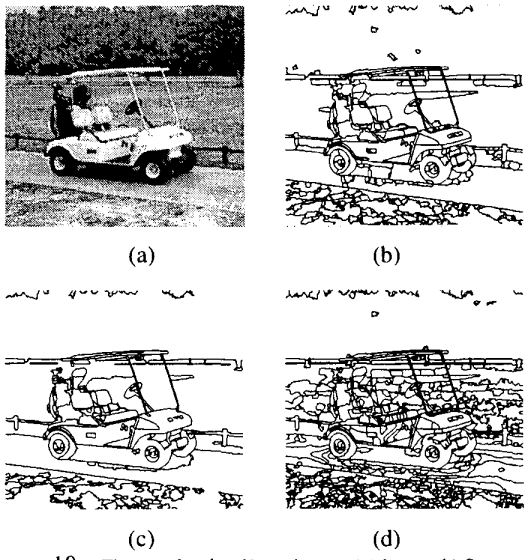
$$w_{ij} = a_{ij} * \rho_{ij} + (1 - a_{ij}) * \eta_{ij} \qquad (2)$$

where $0 \leq a_{ij} \leq 1$ is an attribute which controls the blending of gradient magnitude $\rho$ and local pattern $\eta$ information. For $\rho < 0.02$ the weights are taken 0. In general $a_{ij}$ should be derived from top-down processes focused on enhancing the features of an object sought in the image, or should capture a priori global information such as ecological statistics of segmentation [10]. In our examples the same $a_{ij}$ value was used for all the pixels in the image. Note that the definition (2) can be replaced with any other expression more suitable for a specific class of applications.

The synergetic image segmentation integrates the weights into the mean shift based image segmenter. In the *filtering* step a weighted average is then used in (1), with the weights being $1 - w_{ij}$. Since for pixels close to an edge these weights are small, the discontinuity preserving property of the mean shift based filtering is further enhanced.

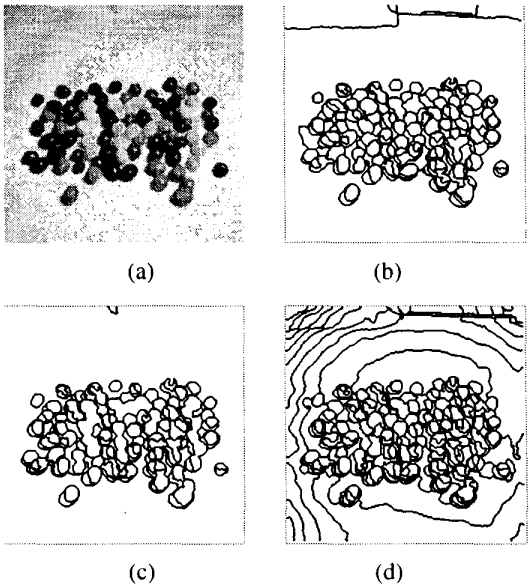
To integrate the discontinuity information into the *fusion* step, for each edge in the region adjacency graph (RAG) of the filtered image a *boundary strength* measure, $e$ is computed by averaging the values $w_{ij}$ for the pixels on the boundary shared by two regions. The transitive closure operations are then performed on this weighted graph, with the additional condition of $e < t_e$.

The synergetic segmentation of the *face* image (Figure 8) employed the same parameters as the undersegmentation in Figure 3. The size of the gradient window was $n = 5$, $a_{ij} = 0.3$ and $t_e = 0.9$. Most of the important features



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

Figure 10: The gray level *golf-cart* image. (a) Input. (b) Synergetic segmentation. (c) Mean shift segmentation with the same parameters. (d) Mean shift oversegmentation.



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

Figure 11: The color *jelly-beans* image. (a) Input. (b) Synergetic segmentation. (c) Mean shift segmentation with the same parameters. (d) Mean shift oversegmentation.

are now recovered, and the segmented image is a satisfactory rendition of the input. The $\rho\eta$–diagram of the boundary pixels (Figure 9) is extended toward lower gradient values but almost exclusively for the pixels which have high confidence. Compare with Figure 7.

## 5 Experimental Results

The performance of the synergetic segmentation was also assessed with three images of different natures. The 548×509 gray level *golf-cart* image has many fine details and a textured background (Figure 10a). Using the parameters $h_s = 7, h_r = 4.5, M = 50$ the mean shift segmentation yields the boundaries in Figure 10c. The synergetic segmentation, with $n = 7$, $a(i,j) = 0.5$ and $t_e = 0.6$ recovers much more details (Figure 10b), like the contiguous contour of the road, or the hood and front wheel of the cart. These features cannot be extracted even when mean shift oversegmentation ($h_s = 7, h_r = 3.5, M = 50$) is used, which introduces a large amount of clutter (Figure 10d).

The 256×256 color *jelly-beans* image (Figure 11a) is very challenging since the objects are small and many of them have highlights. The segmentation parameters were $h_s = 7, h_r = 6, M = 50$, $n = 5$, $a(i,j) = 0.3$ and $t_e = 0.55$. The synergetic segmentation correctly separates most of the objects, and also delineates the weak borders in the background (Figure 11b). Undersegmentation (Figure 11c) joins several objects, while oversegmentation ($h_s = 4, h_r = 4, M = 50$) introduces significant clutter (Figure 11d).

In the 575×437 color *museum* image (Figure 12a) note the strong illumination gradient along the side wall and the fine details on the back wall. The employed segmentation parameters were $h_s = 7, h_r = 6, M = 50$, $n = 2$, $a(i,j) = 0.3$ and $t_e = 0.5$. The fusion module of the mean shift segmenter successfully handles the presence of illumination gradient and the wall is delineated as a single structure (Figure 12c). To recover the details bounded by weak edges synergetic segmentation is needed (Figure 12b), where these edges are well represented in the the employed weight map (Figure 12d). Note also the improved delineation of the cubes in the foreground. Using strong oversegmentations also failed to detect any of these features before the clutter becomes dominant.

## 6 The EDISON System

The processing modules described in this paper were integrated into the *Edge Detection and Image SegmentatiON* (EDISON) system. The system is implemented in C++, and its source code is available on the web at

`www.caip.rutgers.edu/riul/`

EDISON provides a versatile graphic interface (Figure 13) to perform any of the three basic low-level vision operations: discontinuity preserving filtering, segmentation, edge detection; both separately and synergistically. The user has control over any of the parameters, can display the original, the filtered or the segmented image with or without overlaying the boundaries of the delineated regions. The gradient, confidence and weight maps are shown in separate windows. All these images and maps can be saved for future processing. The edge detection module can be run separately. Employing vertical thresholding curves it

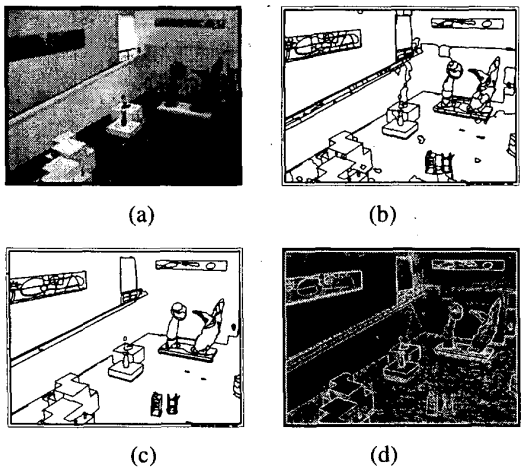


Figure 12: The color *museum* image. (a) Input. (b) Synergetic segmentation. (c) Mean shift segmentation with the same parameters. (d) The employed weight map.

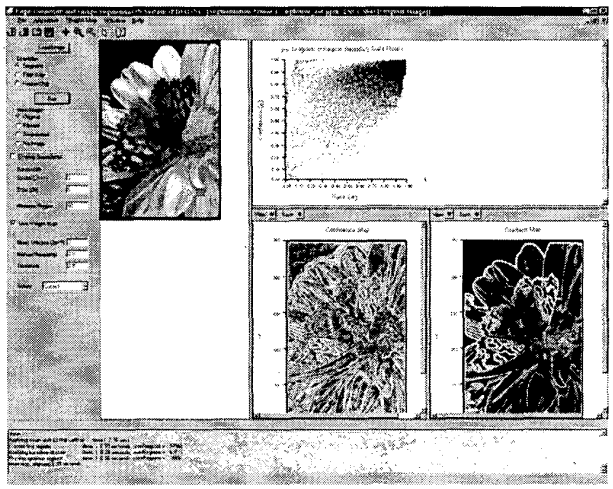


Figure 13: Screen capture of the EDISON system performing synergistic image segmentation. The $\rho\eta$–diagram of the boundary pixels is at top right, the gradient and confidence images are at bottom right.

defaults to the traditional "Canny" technique.

The EDISON system can also be run in command line mode. Both the GUI and the command line implementation is available for the Windows environment, and the command line implementation for the UNIX/Linux environment.

We have succinctly described two recently developed computational modules performing fundamental low-level computer vision operations: segmentation and edge detection. Their strengths can be combined into a synergetic technique which extracts weak but significant features from images.

## Acknowledgement

The National Science Foundation grant IRI 99-87695, and the Rutgers Undergraduate Research Fellow for C.M. Christoudias are gratefully acknowledged. Thanks Doug DeCarlo (Rutgers) for discussions on implementing EDISON, and Christian Vogler (UPenn) for the face image.

## References

[1] C. C. Chu and J. K. Aggarwal. The integration of image segmentation maps using region and edge information. *IEEE Trans. Pattern Anal. Machine Intell.*, 15:1241–1252, 1993.

[2] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *7th International Conference on Computer Vision*, pages 1197–1203, Kerkyra, Greece, September 1999.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24, May 2002.

[4] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Trans. Image Process.*, 10:1454–1466, 2001.

[5] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cufi. Yet another survey on image segmentation. In *7th European Conference on Computer Vision*, Copenhagen, Denmark, May 2002.

[6] D. Geiger and A. Yuille. A common framework for image segmentation. *International J. of Computer Vision*, 6:227–243, 1991.

[7] J. F. Haddon and J. F. Boyce. Image segmentation by unifying region and boundary information. *IEEE Trans. Pattern Anal. Machine Intell.*, 12:929–948, 1990.

[8] J. Le Moigne and J. C. Tilton. Refining image segmentation by integration of edge and region data. *IEEE Trans. Geoscience and Remote Sensing*, 33:605–615, 1995.

[9] W. Y. Ma and B. S. Manjunath. Edge flow: A framework of boundary detection and image segmentation. *IEEE Trans. Image Processing*, 9:1375–1388, 2000.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *8th International Conference on Computer Vision*, volume II, pages 416–423, Vancouver, Canada, July 2001.

[11] P. Meer and B. Georgescu. Edge detection with embedded confidence. *IEEE Trans. Pattern Anal. Machine Intell.*, 23:1351–1365, 2001.

[12] T. Pavlidis and Y. T. Liow. Integrating region growing and edge detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 12:225–233, 1990.

[13] R. Sedgewick. *Algorithms in C.* Addison-Wesley, 1990.

[14] M. Tabb and N. Ahuja. Multiscale image segmentation by integrated edge and region detection. *IEEE Trans. Image Process.*, 6:642–655, 1997.