



A Model-Based Interactive Object Segmentation Procedure

Jie Zou (zouj@rpi.edu)

Rensselaer Polytechnic Institute, ECSE Department
110 8th Street, Troy, New York, 12180, USA

Abstract

A global parametric shape model (boundary) of the object is optimized according to evidence accumulated from local features and the prior probability of the model parameters learned from already segmented training samples. The parametric boundary is then deformed to the most conspicuous edge pixels near it. We emphasize the effectiveness of human intervention. Any unsatisfactory automatic segmentation is corrected interactively with mouse clicks. We demonstrate the above segmentation procedure on a flower image database of 1078 samples from 113 species. 257 out of 1078 images are segmented without interactive correction. On average, 5.7 mouse clicks are needed for each image, and the segmentation process takes 15.2 seconds.

1. Introduction

Strong object segmentation could be widely used for visual pattern recognition, computer vision, content-based image retrieval, and model-based video coding. Although image segmentation has been intensively studied for decades, there still doesn't exist an "off-the-shelf" solution applicable to all types of images. Some recent approaches [3][8] consider image segmentation as a parameterized computational process, not a stand-alone vision task, i.e., an algorithm that generates different solutions dynamically according to requirements represented by a few adjustable parameters.

General image segmentation has been found to be extremely difficult. In [7], the authors argued that "... object segmentation for broad domains of general images is not likely to succeed, with a possible exception for sophisticated techniques in very narrow domains." A 1992 workshop organized by US National Science Foundation in Redwood, CA, stated that "computer vision researchers should identify features required for *interactive image understanding*, rather than their discipline's current emphasis on automatic techniques." Several interactive methods were subsequently proposed for image segmentation. "Magic Wand" and "Lasso" are the main selection tools in Photoshop. "Intelligent Scissors" [5] and "Intelligent Paint" [6] are two well-known general purpose, interactive object segmentation tools.

We believe that reliable general image segmentation is too difficult to achieve because of the fundamental complexity of modelling visual patterns that appear in generic images, and the intrinsic ambiguities of general image perception. Therefore, the procedure presented in this paper is designed for segmenting objects of a specific family that share some color and shape properties. By narrowing down to a specific domain, we can assume reasonable models, which are not too complex to be manageable. Reliable strong segmentation is still difficult to achieve even in specific domains if the pictures are taken under highly variable illumination and have complex background. In stead of pushing complicated automatic methods, we opt to seek the help from the user. We believe that, at the current stage, in some situations, efficient user intervention can be very effective. Allowing parsimonious user intervention guarantees reliable precise segmentation.

The segmentation method presented in this paper combines a model-based statistical approach with a morphological watershed algorithm. Interactive correction is used to refine the automatic segmentation when necessary.

2. Flower database

We collected a "stress-test" database of flowers with a digital camera for interactive visual pattern recognition research [9]. This interactive segmentation procedure is used to generate ground-truth segmentation of the flower image database. There are totally 1078 flower pictures of 113 species. Figure 4 shows four typical examples of flower pictures in our database. (The blue curves superimposed on the pictures are the automatic segmentation results.)

All pictures are 320 by 240 pixels, taken under highly variable illumination. Although we have so far photographed only "flat" flowers, their shape is arbitrary. The majority are yellow, white, blue, or red. Some of the flowers are quite out of focus, and several pictures contain multiple, tiny, overlapping flowers. The background is the real scene, which may include complex vegetation and sharp shadows. Humans have a remarkable ability to recognize such flowers, but it is difficult to conceive that a machine can segment them reliably.

3. Segmentation procedure

We present a model-based segmentation procedure based on color and shape distribution derived from a set of segmented training images. Each training image is segmented manually into a foreground and a background region. The foreground and background color distributions are estimated from the training images. A parametric model (here, a circle) is then fitted to each boundary and the distribution of its parameters is estimated from all the training images. Finally, the distribution of the deviation of the “exact” boundary from the parametric boundary is approximated by a one-parameter (Laplacian) density function.

After the above training process is completed, each new flower image is segmented as follows: 1) a circle (parametric boundary) is fitted by maximizing its posterior probability; 2) a likelihood map is generated by assigning to each pixel a value which indicates its likelihood of being a boundary pixel; 3) foreground and background “seed” pixels are located, and a seeded watershed algorithm is applied to the likelihood map to find a segmentation boundary; 4) if the boundary obtained by the watershed algorithm is unsatisfactory, it is corrected interactively by introducing additional seed pixels with mouse clicks, until an acceptable segmentation is reached.

The training and operational phase of the segmentation process are described in detail in the following sections.

3.1. Parameter estimation

For each training image, manual segmentation yields a *boundary* Γ_m , which *partitions* the image into *foreground* F_m and *background* F_m^c . The manual segmentation is considered as the ground truth. The interface used for manual segmentation is based on some of the concepts presented below, and will be explained in the corresponding sections.

Estimation of foreground and background color distribution. The 320 by 240 RGB image array: $\mathbf{I}(x, y) = (I^R(x, y), I^G(x, y), I^B(x, y))$ indicates the color at pixel (x, y) . It is difficult to estimate the complete color histogram of 256^3 values with a small number of training samples. We opt to assume that the probability densities of R, G, B are independent, therefore the foreground and background color distributions are:

$$P_{F_m}(\mathbf{I}(x, y)) = P_{F_m}^R(I^R(x, y))P_{F_m}^G(I^G(x, y))P_{F_m}^B(I^B(x, y)) \quad (1)$$

$$P_{F_m^c}(\mathbf{I}(x, y)) = P_{F_m^c}^R(I^R(x, y))P_{F_m^c}^G(I^G(x, y))P_{F_m^c}^B(I^B(x, y)) \quad (2)$$

These color distributions are estimated for each of the 256 values of I^R , I^G , and I^B from histograms of the RGB values of all the training flowers.

We explicitly write that the distributions depend on (x, y) in order to indicate that, given a partition Γ , the location (x, y) determines whether the pixel is a foreground

or background pixel. The position information (x, y) is not used for the histogram estimation except for determining the pixel’s category, i.e., foreground or background.

Parametric partition and associated parameter estimation. A parametric curve $\Gamma_a(\boldsymbol{\theta})$ is now used, as an approximation to Γ_m , to segment the image into foreground $F_a(\boldsymbol{\theta})$ and background $F_a^c(\boldsymbol{\theta})$. Our model of the flower boundary is a circle with parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_r)$, where the first two parameters define the center of the circle and the third its radius¹. For each flower image in the training set, the circle $\Gamma_a(\boldsymbol{\theta})$ is obtained by minimizing its mean square deviation from the exact boundary Γ_m .

The parameters of the multivariate density $P_{\boldsymbol{\theta}}(\theta_x, \theta_y, \theta_r)$ are estimated from the training set under the assumption of independent Gaussian distributions because there is not enough training samples to estimate the whole covariance matrix:

$$P_{\boldsymbol{\theta}}(\theta_x, \theta_y, \theta_r) = N(\mu_{\theta_x}, \sigma_{\theta_x}) N(\mu_{\theta_y}, \sigma_{\theta_y}) N(\mu_{\theta_r}, \sigma_{\theta_r}) \quad (3)$$

Estimation of the deviation of the circular shape model from the exact boundary. The radial deviation λ of the boundary pixels on the ground-truth boundary Γ_m from the circle Γ_a is now estimated from the pairs of boundary curves in the training set. Let \mathbf{r}_φ , a radius vector of circle Γ_a in direction φ , intersect Γ_a at r_a and Γ_m at r_m (Figure 1(a)). Set the radial deviation $\lambda = r_m/r_a$. The distribution of λ over all φ of all training samples is plotted in Figure 1(b), which can be closely approximated by a unity-mean Laplacian with parameter β ,

$$P(\lambda|\Gamma_m) = \frac{\beta}{2} e^{-\beta|\lambda-1|}. \quad (4)$$

We estimated β to be 5.52.

3.2. Automatic segmentation

These parameter statistics, accumulated from training samples, are used to segment new flower images.

Likelihood of a parametric partition. In order to compute the likelihood of a parametric partition $\Gamma_a(\boldsymbol{\theta})$ accurately, one needs a complex model of the correlation among pixels, which is unattainable in practice. We therefore attempt to model only the “average” correlation.

If the pixels were independent of each other, their joint distribution could be expressed as

$$Prob(F_a(\boldsymbol{\theta})) = \prod_{(x, y) \in F_a(\boldsymbol{\theta})} P_{F_m}(\mathbf{I}(x, y))$$

¹Although the circle model seems too simple for the flowers, we cannot afford very complicated models because: (1) We have too few training samples to estimate the parameters of a more elaborate model. (2) Even with the simple circle model, determining the optimal center and radius is so computation intensive that we must use gradient descent instead of exhaustive search. (3) With a more complicated model, the gradient descent algorithm is more likely to stop at a local extremum.

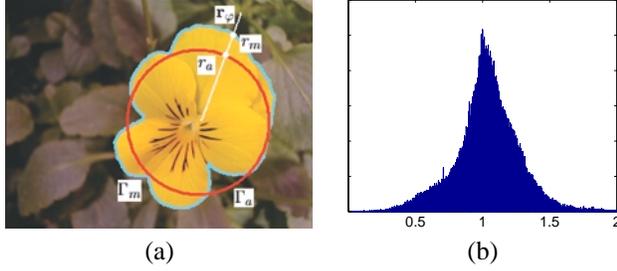


Figure 1: (a) r_φ intersects Γ_a at r_a and Γ_m at r_m . (b) The histogram of the ratio $\lambda = r_m/r_a$ over all φ of all training samples.

for the foreground, and

$$Prob(F_a^c(\theta)) = \prod_{(x,y) \in F_a^c(\theta)} P_{F_m^c}(\mathbf{I}(x,y))$$

for the background. Then

$$P[\mathbf{I}|\Gamma_a(\theta)] = Prob(F_a(\theta)) Prob(F_a^c(\theta)) \quad (5)$$

where $P_{F_m}(\mathbf{I}(x,y))$ and $P_{F_m^c}(\mathbf{I}(x,y))$ are the estimated foreground and background color distributions, which are evaluated with Equations (1) and (2).

However, image pixels in natural scenes are highly correlated. If we assume that (1) foreground pixels are independent from background pixels, (2) all the pixels in the foreground region depend completely on each other, i.e., the color of the other foreground pixels are completely determined by the color of any given foreground pixel, and (3) all the pixels in the background region depend completely on each other, the likelihood²:

$$P[\mathbf{I}|\Gamma_a(\theta)] \propto (Prob(F_a(\theta)))^{\frac{1}{\|F_a\|}} (Prob(F_a^c(\theta)))^{\frac{1}{\|F_a^c\|}} \quad (6)$$

where $\|F_a\|$ and $\|F_a^c\|$ indicate the number of foreground pixels and the number of background pixels, respectively.

As a compromise between total independence and total dependence, it is intuitive to use the following equation to approximate the likelihood.

$$P[\mathbf{I}|\Gamma_a(\theta)] \propto (Prob(F_a(\theta)))^{\frac{1}{\alpha\|F_a\|}} (Prob(F_a^c(\theta)))^{\frac{1}{\alpha\|F_a^c\|}} \quad (7)$$

Equation 7 is purely empirical. The constant α , which compensates the correlation of pixels, is set to 0.025. α is estimated to be the value that results, on the average, in fitting the best circles to the corresponding manual boundary curves of all training samples. As shown in Figure 2, the distribution of the quality of the fitted circles is quite flat over a broad range of α , which means that the likelihood is not sensitive to the precise value of α .

²In general, the RHS of Equation (6) doesn't add up to 1.

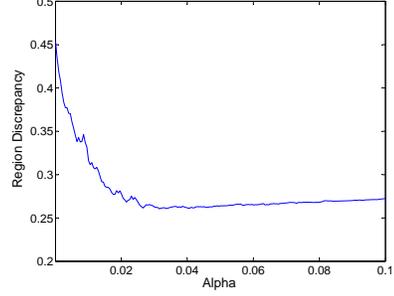


Figure 2: Average region discrepancy between manual and circle segmentation with respect to α .

The two terms in the likelihood are functions of θ . Whether the pixel (x,y) is in the foreground or the background region is completely determined by θ , and so is the likelihood.

Parametric segmentation. We fit a circle to each new image by maximizing the posterior probability of the parametric (circle) partition.

$$\theta^* = \operatorname{argmax}_{\theta} P(\Gamma_a(\theta)|\mathbf{I}) \quad (8)$$

From Bayes' Rule:

$$\theta^* = \operatorname{argmax}_{\theta} \{P(\mathbf{I}|\Gamma_a(\theta))P(\Gamma_a(\theta))\} \quad (9)$$

$P(\mathbf{I}|\Gamma_a(\theta))$ and $P(\Gamma_a(\theta))$ are computed with equation (7) and (3) respectively.

Generation of the boundary likelihood map. After parametric segmentation, the object is roughly segmented by a circle. However, the real object boundary is much more complicated than the model. The circle must be deformed to the actual object boundary.

Our approach combines Bayesian classification with morphological watershed segmentation to deform the circle to the most conspicuous boundary pixels near it. The deformation process guarantees that the final boundary is a closed curve.

If the pixel (x,y) is a boundary pixel, the random variable $\Gamma(x,y) = 1$; otherwise, $\Gamma(x,y) = 0$. We compute the probability, $P(\Gamma(x,y)|\lambda(x,y), \mathbf{f}(x,y))$, of each pixel to be a boundary pixel given its radial deviation ratio $\lambda(x,y)$ from the optimal parametric boundary, and the boundary feature $\mathbf{f}(x,y)$ chosen to differentiate boundary pixels from non-boundary pixels. To simplify the notation, we drop the argument (x,y) of Γ , λ , \mathbf{f} , and ∇ in the following discussion.

If we assume equal priors ($P(\Gamma)$), and λ is conditionally independent on \mathbf{f} , i.e., $P(\lambda, \mathbf{f}|\Gamma) = P(\lambda|\Gamma)P(\mathbf{f}|\Gamma)$,

$$P(\Gamma|\lambda, \mathbf{f}) = \frac{P(\lambda, \mathbf{f}|\Gamma)P(\Gamma)}{P(\lambda, \mathbf{f})} \propto P(\lambda|\Gamma)P(\mathbf{f}|\Gamma) \quad (10)$$

$P(\lambda|\Gamma)$ has been estimated from the training images (Equation 4). To model $P(\mathbf{f}|\Gamma)$, we begin with a very simple model, $P(\mathbf{f}|\Gamma) \propto |\nabla|$, where $|\nabla|$ is the magnitude of color gradients. The probability that a pixel is on the boundary then becomes

$$P(\Gamma|\lambda, \mathbf{f}) \propto |\nabla| \frac{\beta}{2} e^{-\beta|\lambda-1|} \quad (11)$$

Equation (11) reflects the intuition that we are looking for the pixels which are close to the circle, and have high color gradient. Applying Equation (11) to the entire image yields a 2D array with a boundary probability assigned to each pixel. We call it a *boundary likelihood map* (BLM). In the BLM, boundary pixels tend to have larger values than non-boundary pixels.

Deformation with seeded watershed algorithm. We assume that the center of the circle is inside the object³, and apply an algorithm similar to [1] to the BLM. Initially, the foreground and background regions contain only the seed pixels. The deformation process expands the boundaries around foreground seed pixels and around background seed pixels until the boundaries coalesce. In each iteration of the algorithm, the unlabelled pixels, which are connected to the current foreground or background region, are entered into an ascending ordered list according to their probability $P(\Gamma|\lambda, \mathbf{f})$. Then the pixel, which has the smallest probability, is retrieved from the list, and labelled either foreground or background according to the majority vote of its already labelled neighbors. When there is a tie, the pixel is randomly assigned a label. The algorithm ends when the sorted list is empty. It can be proved that the algorithm finds the watershed of the seed pixels on the BLM, so we call it *seeded watershed algorithm*.

This deformation method works even without the parametric segmentation step. Without that step, we have no optimal circle. Equation (11) becomes $P(\Gamma|\mathbf{f}) \propto |\nabla|$, and the BLM is simplified to the *gradient map*. We assume that the center of the image is inside the object, and the four corner pixels are outside the object, then find the watershed of the gradient map from these seed pixels. We segmented our initial set of training pictures with this method followed by interactive corrections (Section 3.3). However, the term $\frac{\beta}{2} e^{-\beta|\lambda-1|}$, which reflects the result of parametric segmentation, helps to prevent big, non-flower-like excursion. Figure 3 shows a good example.

3.3. Interactive correction

The automatic method described above cannot guarantee accurate segmentation of all the images. Interactive correction is necessary for robust strong segmentation.

³The assumptions about initial seed pixels are application dependent. The initial seed pixels will be overwritten by the user’s clicks.

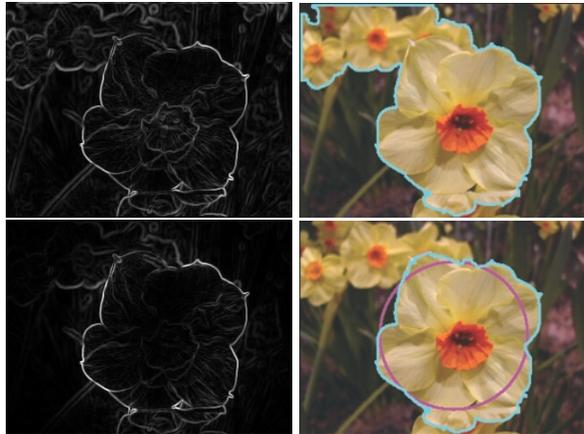


Figure 3: On the gradient map, the deformation process doesn’t stop at the “correct” boundary (the first row). However, on the BLM, the gradients that are near the fitted circle (red) are amplified, and the gradients that are far from the circle are attenuated. The segmentation result (blue curve) is therefore much better (the second row).

Our strategy for interactive segmentation is to let the system try its best to segment the object. If the user is satisfied, the segmentation is confirmed. Otherwise, the current segmentation is corrected by introducing new seed pixels with mouse clicks.

Automatic segmentation always precedes interactive correction. The user clicks on a misclassified pixel to shrink the foreground region if that pixel should be in the background, or to grow the object region if it should be in the foreground. The clicks are saved as seed pixels, and the seeded watershed algorithm is applied to the BLM with the updated seed pixels. The process is repeated until a satisfactory segmentation is reached (Figure 5).

The advantage of this strategy is that the machine always displays its segmentation results to the user. We observe that it is a consensus among all the users that the large mis-segmented regions are always chosen to be corrected first. Therefore, the segmentation quickly converges to the “exact” boundary. Depending on the application, the user can decide whether to correct the local small errors.

4. Experimental results and evaluation

The automatic segmentation process takes approximately 0.5 seconds on a 1GHz PC. The response time of the computer to each interactive correction is negligible (tens of milliseconds).

Figure 4 shows the automatic segmentation of four typical samples. We can observe that the parametric segmentations (circles) are fine for all four flowers. The segmentation



Figure 4: The results of automatic segmentation on four flower images.



Figure 5: Interactive correction. 7 foreground clicks (red circle dots) and 1 background click (red square dot) generate almost perfect segmentation.

for the bottom two and the top-left samples is perfect. The segmentation result for the top-right flower may also be acceptable depending on the application.

The flower in Figure 5 can be considered difficult, since its boundary is very complicated, and so are the color layouts of both the flower and the background. The automatic method cannot segment it correctly. However, with the help of the interactive correction from the user, a few clicks quickly yields near-perfect segmentation. Of course, the actual number of clicks depends on where the user clicks. Figure 5 shows one scenario, which took 7 foreground clicks and 1 background click. The correction process took less than 30 seconds. The intermediate results after 1, 2, 3, and after all 8 clicks are shown.

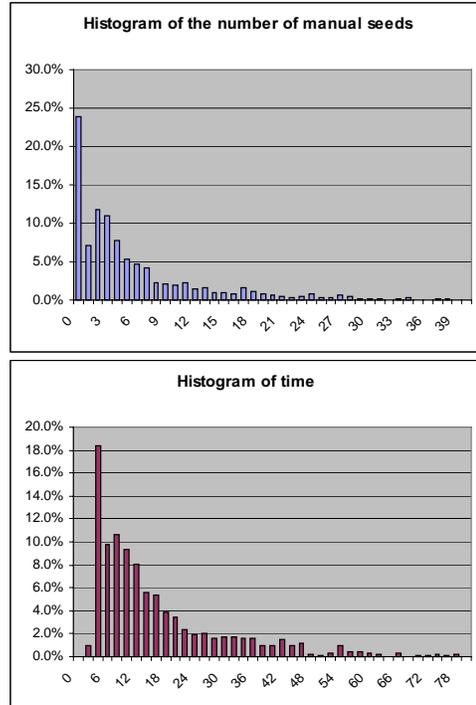


Figure 6: Histogram of the number of clicks (top) and histogram of interactive time in seconds (bottom).

We evaluated the automatic part of the segmentation procedure with 216 flower pictures from 29 species. The samples were partitioned in 6 different ways according to a leave-one-out-per-class experimental design, yielding an equivalent experiment with 174 (29 * 6) mutually-exclusive test samples and 187 (216 - 29) training samples.

The results of automatic segmentation were compared to the interactively produced “ground truth” segmentation. On the average, the region of automatic segmentation and “ground truth” overlapped by 75%, and the boundary was 8.7 pixels away from the “ground truth”. With “ground truth” segmentation, 141 out of 174 samples were correctly classified by a nearest-neighbor classifier based on eight color and shape features. With automatic segmentation, 84 out of 174 samples were correctly classified.

The interactive correction depends on the user’s expectations and patience. A computer engineering undergraduate student interactively segmented all the flower pictures in our database (1078 flowers from 113 species). The student was asked to segment the flower pictures as precisely as possible, since the results were used as the ground-truth strong segmentation for the development and evaluation of an interactive flower recognition system. A log subsystem embedded in the segmentation program recorded the number of seeds, the position of seeds, and the combined human

and machine time spent on each picture. Figure 6 shows the histograms of the number of manually-introduced seed pixels and the histogram of the time spent on each flower picture. On average, 5.7 seed pixels were needed for each picture, and the segmentation process took 15.2 seconds. 24% (257 out of 1078) pictures were segmented without interactive correction.

5. Summary and discussion

We proposed a fairly general procedure for model-based interactive object segmentation. The procedure was applied to segment flower pictures. Simpler models are currently preferable because they allow parameter estimation from a small set of samples. The automatic segmentation was quite good, considering the complexity of foreground and background color layout and the variability of the global shape of flowers. However, as expected, some user interaction was necessary for reliable strong segmentation. With the interactively-introduced seed pixels, the intermediate segmentation results approached the “exact” boundary with little human effort.

The effectiveness of human intervention for segmentation in practical applications should not be overlooked. Our automatic segmentation method is relatively simple comparing to the state-of-the-art segmentation algorithms. However, with parsimonious human intervention, the reliable precise segmentation is guaranteed with a cost of only a few seconds. This can be very useful for many practical problems, such as (1) generating ground-truth data for training and evaluating computer vision and pattern recognition systems, which is the purpose for us to develop this interactive segmentation procedure; (2) content-based image or video retrieval, where the goal of segmentation (the user interested region) can be ambiguous and human involvement is essential; (3) the first frame initialization for tracking and video compression; and (4) some medical image processing applications, where usually the image quality is bad and the accuracy is considered very important. In our opinion, some kinds of visible agents, which are understandable to both humans and computers, are essential for efficient human intervention. For example, “Intelligent Scissors” uses *live-wires* [5] and Blobworld uses *Blobworld representation* [2]. One advantage of our approach comparing to “Intelligent Scissors” is that our method approaches “exact” segmentation progressively and the user can stop at any acceptable segmentation. It may take long time for “Intelligent Scissors” to trace the boundary of the flower in Figure 5. Our method quickly converges to a satisfactory segmentation.

A large number of interactively segmented images are also very valuable for developing automatic segmentation algorithms. Recently, several sophisticated boundary cues have been carefully studied in [4] based on a large data set

of 12,000 manual segmentations. Incorporating more sophisticated models and their parameter statistics accumulated from interactively segmented samples will improve the performance of the automatic part of the segmentation procedure. Eventually, reliable fully automatic segmentation method may be developed.

Acknowledgments

The author would like to thank Prof. George Nagy for several valuable discussions, Borjan Gagoski and Greenie Cheng for collecting the flower database and interactively segmenting flower images. The author would also like to thank the anonymous reviewers for their critical and constructive comments.

References

- [1] R. Adams and L. Bischof, “Seeded Region Growing,” *IEEE T-PAMI*, vol. 16, no. 6, pp. 641-647, 1994.
- [2] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying,” *IEEE T-PAMI*, vol. 24, no. 8, pp. 1026-1038, 2002.
- [3] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE T-PAMI*, vol. 24, no. 5, pp. 603-619, 2002.
- [4] D.R. Martin, C.C. Fowlkes, and J. Malik, “Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues,” *IEEE T-PAMI*, vol. 26, no. 1, pp. 1-19, 2004.
- [5] E.N. Mortensen and W.A. Barrett, “Interactive Segmentation with Intelligent Scissors,” *Graphical Models and Image Processing*, vol. 60, No. 5, pp. 349-384, 1998.
- [6] L.J. Reese, *Intelligent Paint: Region-based Interactive Image Segmentation*, Masters Thesis. Department of Computer Science, Brigham Young University, Provo, UT, 1999.
- [7] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-Based Image Retrieval at the End of the Early Years,” *IEEE T-PAMI*, vol. 22, no. 12, pp.1349-1380, 2000.
- [8] Z. Tu and S.-C. Zhu, “Image Segmentation by Data-Driven Markov Chain Monte Carlo,” *IEEE T-PAMI*, vol. 24, no. 5, pp. 657-673, 2002.
- [9] J. Zou, *Computer Assisted Visual InterActive Recognition*, Ph.D. thesis, ECSE department, Rensselaer Polytechnic Institute, May, 2004.