

Probabilistic Shape and Appearance Model for Scene Segmentation

S. S. Gleason
Oak Ridge National Laboratory
P.O. Box 2008
Oak Ridge, TN 37831-6011
gleasonss@ornl.gov

M. A. Abidi
The University of Tennessee
Electrical Engineering Department
Knoxville, Tennessee, 37996
abidi@utk.edu

H. Sari-Sarraf
Texas Tech University
Electrical Engineering Department
Lubbock, Texas, 79409
hamed.sari-sarraf@coe.ttu.edu

Abstract

Effective image segmentation of a digitized scene into a set of recognizable objects requires the development of sophisticated scene analysis algorithms. Progress in this area has been made through the development of a statistical-based deformable model that improves upon existing point distribution models (PDMs) for boundary-based object segmentation. Existing PDM boundary finding techniques often suffer from the shortcoming that global shape and gray-level information are treated independently during boundary optimization. A new deformable model algorithm is under development in which the objective function used during optimization of the boundary encompasses several important characteristics. Most importantly the objective function includes both shape and gray-level characteristics, so optimization occurs with respect to both pieces of information simultaneously. This new algorithm has been applied to geometric test images and a simple industrial-type scene for which results are presented.

1. Introduction

One of the most challenging tasks in computer vision is scene segmentation, that is, breaking up the image of some environment into its constituent parts. This is a task that is necessary in a variety of applications, including autonomous robot guidance. There are a variety of boundary- or surface-based scene segmentation approaches, including those that employ some type of deformable model such as simplex meshes [1] or deformable superquadrics [2]. These approaches are quite effective for object representation and object recognition in data-driven applications where object edges are fairly well-defined. By their nature data-driven approaches have more difficulty in segmentation applications where object edges are very noisy, faint, or obstructed. Hence, a new statistical-based deformable model algorithm for image analysis is being developed that is motivated by the need for an algorithm to perform automatic segmentation and recognition of objects with faint, missing, or obstructed edges within a complex background. An example of this type of application presented in this paper is the segmentation/recognition of objects within intensity images of an industrial-type scene.

The active shape model (ASM) developed by Cootes, et al. [3], was chosen as an appropriate starting point because of its ability to incorporate *a priori* information extracted from a training set to build a gray-level model (GLM) and a global shape model (GSM). These models are used during an iterative contour deformation process that adjusts the position and shape of the contour to match the boundary of the object within the image. Although ASM is an excellent starting point for the motivating application, it has a shortcoming in a key area: global shape and gray-level information are treated independently during optimization of the boundary position. This shortcoming limits ASM's robustness and accuracy in some applications. A new approach, the probabilistic shape and appearance (PSAM) algorithm has been developed to address this and other ASM shortcomings.

A few researchers have recently developed new statistical-based deformable models based on ASM. Wang, et al. [4] have developed a probabilistic based optimization scheme that uses Cootes' PDM and integrates Canny-edge information into the maximum a posteriori (MAP) objective function as the underlying image attraction force. Kervrann, et al. [5] have developed similar probabilistic techniques, but include Markov modeling on the local scale to promote boundary smoothness. Both of these ASM adaptations rely on edge information in the target image as the external attraction force, rather than the gray-level gradients proposed by Cootes. Duta, et al. [6] have also refined the ASM technique in terms of the image attraction force as well as the optimization approach to fit the boundary to the underlying image data. Consideration of boundary-point outliers is an important consideration in their work. Gleason, et al [7] have improved upon the original ASM by including more comprehensive gray-level information and have added constraints to the shape model to improve convergence. Even with the considerable research that has been performed, none of the resulting approaches address the shortcomings of existing PDMs outlined previously.

2. Shape and Gray-level Objective Function

First, we will define the boundary we are searching for as an (x, y) coordinate matrix, \mathcal{S}_v , of size $N \times 2$, where N is

the number of landmark points (LPs) needed to represent the boundary. Each row of \mathcal{S}_v corresponds to the (x, y) coordinate of an LP:

$$\mathcal{S}_v = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots & \dots \\ x_N & y_N \end{bmatrix}, \quad (1)$$

where the N LPs are defined by the coordinate pairs

$$\{x_j(i), y_j(i)\}; \forall j = 1, \dots, N. \quad (2)$$

Defining the boundary this way accommodates some flexibility in how the boundary is represented. The points in \mathcal{S}_v are typically LPs that lie directly on the boundary (as used in PSAM), but in addition, other (x, y) coordinate locations could be included that lie in image regions that are on the interior or exterior of the object boundary, for example. These points could be used to capture other internal object characteristics that are located at off-boundary positions.

Next, we will define a feature vector matrix, \mathbf{G} . In this formulation, the feature vectors (rows of \mathbf{G}) contain features extracted from the neighborhood of each LP on the boundary. If the length of each feature vector is m , then the size of \mathbf{G} will be $N \times m$. These feature vectors may contain any image information that is relevant to the given application, such as gray-level intensities in the neighborhood of each pixel, local texture measurements, or gray-level gradient information (as used in ASM and PSAM). We can write \mathbf{G} as

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \dots \\ \mathbf{g}_N \end{bmatrix}, \quad (3)$$

where

$$\mathbf{g}_j = h(\mathbf{i}_j), j = 1, \dots, N. \quad (4)$$

In Eq. 4, \mathbf{i}_j is a vector containing raw gray-level values sampled from the neighborhood of the (x_j, y_j) LP in the test image, and h is an operator (e.g. a gradient) that transforms these raw gray-level values into the feature vector, \mathbf{g}_j . Now that we have defined our parameters, we can formulate the probabilistic objective function.

In the PSAM boundary-finding application, the goal is to maximize the a posteriori probability of the boundary \mathcal{S}_v , given the measured image features in \mathbf{G} . Using the

compound version of Bayes rule [12], we can write the a posteriori probability expression for the boundary, \mathcal{S}_v , given a collection of feature vectors, \mathbf{G} , as

$$P(\mathcal{S}_v | \mathbf{G}) = \frac{p(\mathbf{G} | \mathcal{S}_v) P(\mathcal{S}_v)}{p(\mathbf{G})}, \quad (5)$$

where $P(\mathcal{S}_v)$ is the prior probability of a boundary instance, \mathcal{S}_v ; $p(\mathbf{G} | \mathcal{S}_v)$ is the conditional density of \mathbf{G} given the boundary instance, \mathcal{S}_v ; and $p(\mathbf{G})$ is the prior probability density for \mathbf{G} . The goal is then to optimize Eq. 5 by searching over all possible values of \mathcal{S}_v to find the one, \mathcal{S}_v^* , that corresponds to the MAP value as given by

$$P(\mathcal{S}_v^* | \mathbf{G}) = \max_{\mathcal{S}_v} \frac{p(\mathbf{G} | \mathcal{S}_v) P(\mathcal{S}_v)}{p(\mathbf{G})}. \quad (6)$$

Finding \mathcal{S}_v^* can be further simplified without losing the generality that we wish to maintain for a variety of boundary-finding applications. First, because the logarithm is a monotonically increasing function, optimizing the logarithm of a function yields the same result as optimizing the original function. Therefore, the objective function [we will now call it $J(\mathcal{S}_v)$] can be written as

$$J(\mathcal{S}_v) = \ln p(\mathbf{G} | \mathcal{S}_v) + \ln P(\mathcal{S}_v), \quad (7)$$

where we have dropped the term $-\ln p(\mathbf{G})$ since it is independent of \mathcal{S}_v and is therefore a constant.

Depending on the application at hand, the term $\ln p(\mathbf{G} | \mathcal{S}_v)$ may be difficult to calculate. The level of difficulty is contingent in large part on the level of independence that can be assumed for the current problem. If independence is assumed between the feature vectors in \mathbf{G} and if it is also assumed that each feature vector, \mathbf{g}_j , is dependent only on its corresponding location, (x_j, y_j) (i.e., j^{th} row of \mathcal{S}_v), then the conditional density of \mathbf{G} given \mathcal{S}_v can be rewritten as [12]

$$p(\mathbf{G} | \mathcal{S}_v) = \prod_{j=1}^N p(\mathbf{g}_j | S_{v,j,1}, S_{v,j,2}) = \prod_{j=1}^N p(\mathbf{g}_j | x_j, y_j). \quad (8)$$

Plugging this expression back into Eq. 7, we can write

$$J(\mathcal{S}_v) = \sum_{j=1}^N \ln p(\mathbf{g}_j | x_j, y_j) + \ln P(\mathcal{S}_v). \quad (9)$$

It is useful to note that the first term is the “data-driven” term of the objective function in that it depends on image characteristics (external energy term), while the second term is “model-driven” in that it is independent of the

image, depending only on prior distributions of boundary shape and location (internal energy term).

Note how this formulation can accommodate several important goals. First, this objective function allows simultaneous optimization with respect to image-derived gray-scale information (first term) and shape information (second term). Secondly, because this objective function is based on a probabilistic framework, we can interpret its value as a measure of how well the final boundary fits the distribution approximated by those contained in the training set. This measure can be broken down into two pieces: (1) the first term measures how well the final gray-level information matches that which was extracted from the training data, and (2) the second term measures how well the overall shape and location of the boundary matches that which was extracted from the training set.

Optimizing $J(\mathcal{S}_v)$ over all possible boundary vectors, \mathcal{S}_v , can be a daunting task for several reasons. First, depending on the number of LPs used to represent the boundary, \mathcal{S}_v could be a very long vector, and finding the maximum of J with respect to each of the $2N$ elements of \mathcal{S}_v can be computationally demanding. Applying principal component analysis (PCA) and then using only the significant modes of variation reduces the dimensionality of \mathcal{S}_v to resolve this problem. The boundary \mathcal{S}_v can be approximated in the PCA subspace as a vector, \mathbf{v} , with fewer dimensions. As detailed in [8], \mathbf{v} is constructed to be a combination of the boundary coordinates in PCA subspace, \mathbf{b} , and the pose of the boundary, \mathbf{z} , relative to the mean shape calculated during the shape training process (see next section) as follows:

$$\mathbf{v} = \begin{bmatrix} \mathbf{b} \\ \mathbf{z} \end{bmatrix}, \quad (10)$$

where $\mathbf{z} = [s \ \theta \ T_x \ T_y]^T$. Here, s is scale, θ is rotation, and T_x, T_y are the x- and y-translations required to align the boundary with the mean shape. Also we define the length of the PCA shape vector, \mathbf{b} , to be t_s , the length of \mathbf{z} is $t_p = 4$, and the overall length of \mathbf{v} is then $t = t_s + t_p$.

Hence, optimizing the objective function in the PCA sub-space with respect to the more compact vector, \mathbf{v} , is a simpler task. If we substitute the new boundary representation, \mathbf{v} , into Eq. 7, the objective function then takes the form

$$J(\mathbf{v}) = \ln p(\mathbf{G}|\mathbf{v}) + \ln P(\mathbf{v}), \quad (11)$$

where

$$\ln p(\mathbf{G}|\mathbf{v}) = \sum_{j=1}^N \ln p(\mathbf{g}_j | f_{x_j}(\mathbf{v}), f_{y_j}(\mathbf{v})), \quad (12)$$

and $f_{x_j}(\mathbf{v}), f_{y_j}(\mathbf{v})$ are functions that map \mathbf{v} from the PCA subspace back into the image (x, y) -coordinate space.

To use this objective function in practice, we must know all of the individual conditional probability densities, $p(\mathbf{g}_j | x_j, y_j), j = 1, \dots, N$. In the PSAM approach, these conditional densities can be straightforwardly measured from the training data as described later. Finally, the prior probability of the PCA boundary vector, $P(\mathbf{v})$, must also be known. Once again, this can be easily estimated from the training data.

Shape Model Training

Although it was convenient to represent the boundary as a matrix, \mathcal{S}_v , during the compound Bayesian description of the objective function, it is simpler for implementation to represent the entire collection of (x, y) coordinates as one long vector, \mathbf{p} . Let the manually selected boundary for the i^{th} image in an M -image training set be represented by a collection of landmark points (LPs), \mathbf{p}_{train_i} as

$$\mathbf{p}_{train_i} = [x_{i_1} \ x_{i_2} \ \dots \ x_{i_N} \ y_{i_1} \ y_{i_2} \ \dots \ y_{i_N}], \quad i = 1, \dots, M, \quad (13)$$

where the N LPs that make up the boundary for the i^{th} sample are defined by the coordinate pairs

$$(x_{i_j}, y_{i_j}) \quad j = 1, \dots, N. \quad (14)$$

These training boundaries are manually delineated by a qualified person that understands the characteristics of the object in the scene that must be segmented and recognized.

Prior to building the shape model, the alignment of the sets of LPs to a common coordinate frame is done via Procrustes analysis [11] to form the aligned sets of LPs:

$$\hat{\mathbf{p}}_{train_i} = [\hat{x}_{i_1} \ \hat{x}_{i_2} \ \dots \ \hat{x}_{i_N} \ \hat{y}_{i_1} \ \hat{y}_{i_2} \ \dots \ \hat{y}_{i_N}], \quad i = 1, \dots, M. \quad (15)$$

Using this training data set of aligned LPs, $\hat{\mathbf{p}}_{train_i}$, we can straightforwardly create a GSM based on \mathbf{v} by applying PCA as described in [8].

Gray-Level Model Training

Recall that \mathbf{g}_j is a vector of the gray-level values along a profile that passes through the j^{th} LP and is normal to the current boundary estimate. We can write the raw pixel intensity elements of the normal profile, i_j , as

$$i_{j_k} = I \left(x_j - \left(\frac{N_g - 1}{2} - k + 1 \right) \cos \alpha_j, y_j - \left(\frac{N_g - 1}{2} - k + 1 \right) \sin \alpha_j \right) \quad (16)$$

$k = 1, \dots, N_g$ and $j = 1, \dots, N,$

where \mathbf{I} is the image under test, N_g is the number of gray-level samples in each profile, and α_j is the angle of the profile through the j^{th} LP normal to the boundary. As noted in Eq. 4, the final feature vectors are given as $\mathbf{g}_j = h(i_j)$, where h is the gradient operator. Once we have extracted the gray level profile vectors for each LP on the boundary for all M images in the training set, we can (similar to the GSM) create a GLM using PCA as described in [8].

Objective Function Parameterization and Optimization

Once the formulation of the GSM and GLM is complete, all of the required information is available to parameterize and optimize the objective function in Eq. 11. We can rewrite Eq. 11 as functions of J_1 and J_2 as follows:

$$J(\mathbf{v}) = J_1(\mathbf{v}) + J_2(\mathbf{v}) \quad (17)$$

where $J_1(\mathbf{v}) = \ln P(\mathbf{v})$ and $J_2(\mathbf{v}) = \ln p(\mathbf{G}|\mathbf{v})$. By design, J has one term for each of the trained model components: GSM (J_1) and GLM (J_2).

For parameterization purposes, the distributions of both the shape vectors ($\hat{\mathbf{p}}_{train_i}$) and the gray-level vectors for each LP (\mathbf{g}_j) are assumed to be Gaussian. As detailed in [8] and [9], these assumptions are not only valid, but they simplify both the parameterization and optimization of the objective function.

Gradient descent was chosen as the initial approach for optimizing the objective function because of its simplicity and the fact that the calculation of the gradient of Eq. 11 is straightforward. The gradient can be written as:

$$\nabla J(\mathbf{v}) = \nabla J_1(\mathbf{v}) + \nabla J_2(\mathbf{v}). \quad (18)$$

Because we assume Gaussian distributions for the GSM and GLM, the gradient of each can be straightforwardly calculated. The gradient of the shape term, J_1 is given by:

$$\frac{\partial}{\partial v_n} J_1(\mathbf{v}) = \frac{2(v_n - \bar{v}_n)}{\sigma_n^2}. \quad (19)$$

where σ_n^2 is the variance of v_n as measured from the training set.

The gray-level model term is more complex. The gradient of J_2 is given by

$$\frac{\partial}{\partial v_n} J_2(\mathbf{v}) = 2 \sum_{j=1}^N \mathbf{g}_j(\mathbf{v})^T \mathbf{K}_{g_j} \frac{\partial}{\partial v_n} \mathbf{g}_j(\mathbf{v}). \quad (20)$$

Note that $\dot{\mathbf{g}}_j$, the mean-centered profile (see Eq. 21) is being written as $\dot{\mathbf{g}}_j(\mathbf{v})$ to indicate that it is a function of \mathbf{v} , the PCA-based boundary. The partial derivative term in Eq. 20 is given by

$$\frac{\partial}{\partial v_n} \dot{\mathbf{g}}_j(\mathbf{v}) = \frac{\partial}{\partial v_n} (\mathbf{g}_j(\mathbf{v}) - \bar{\mathbf{g}}_j) = \frac{\partial}{\partial v_n} \mathbf{g}_j(\mathbf{v}), \quad (21)$$

where $\bar{\mathbf{g}}_j$ is the mean gray-level profile through the j^{th} LP. The partial derivative of \mathbf{g}_j depends, of course, on how the gray-level profiles are defined during training, as well as on the operator, $h(\mathbf{g}_j)$. The profile samples here are calculated as edge profiles based on a first-forward difference gradient:

$$\mathbf{g}_{j_k}(\mathbf{v}) = \mathbf{I}(Q_{j_k}(\mathbf{v}), R_{j_k}(\mathbf{v})) - \mathbf{I}(Q_{j_{k+1}}(\mathbf{v}), R_{j_{k+1}}(\mathbf{v})), \quad k = 1, \dots, N_g, \quad (22)$$

Here, Q_j and R_j are vectors that contain the values of the (x, y) coordinate locations of the samples along the j^{th} normal profile and are defined as (see Eq. 16)

$$Q_{j_k} = x_j - \left(\frac{(N_g - 1)}{2} - k + 1 \right) \cos \alpha_j, \quad k = 1, \dots, N_g, \quad (23)$$

$$R_{j_k} = y_j - \left(\frac{(N_g - 1)}{2} - k + 1 \right) \sin \alpha_j, \quad k = 1, \dots, N_g. \quad (24)$$

where α_j is the angle of the j^{th} profile. Also note that \mathbf{I} is the image under test and that (q_j, r_j) is the pose-corrected coordinate pair indicating the location of the j^{th} LP within the image. The profile is also normalized by the sum of the absolute values of the profile elements as:

$$\sum_{m=1}^{N_g} |\mathbf{g}_{j_m}|.$$

The expression for the profile in this case is

$$\mathbf{g}_{j_k}(\mathbf{v}) = \frac{\mathbf{I}(Q_{j_k}, R_{j_k}) - \mathbf{I}(Q_{j_{k+1}}, R_{j_{k+1}})}{\sum_{m=1}^{N_g} |\mathbf{I}(Q_{j_m}, R_{j_m}) - \mathbf{I}(Q_{j_{m+1}}, R_{j_{m+1}})|}, \quad (25)$$

for $k = 1, \dots, N_g$. The gradient of the normalized profile in Eq. 25 results in a lengthy but straightforwardly implemented result. The final expression still requires one to calculate the gradient of the image along the profile points—that is,

$$\frac{\partial}{\partial v_n} \mathbf{I}(Q_{j_k}, R_{j_k}).$$

Applying the chain rule, we can write this partial derivative as follows:

$$\frac{\partial}{\partial v_n} \mathbf{I}(Q_{j_k}, R_{j_k}) = \frac{\partial}{\partial Q_{j_k}} \mathbf{I} \frac{\partial}{\partial v_n} Q_{j_k} + \frac{\partial}{\partial R_{j_k}} \mathbf{I} \frac{\partial}{\partial v_n} R_{j_k}, \quad (26)$$

We do not have an analytical form of the image, \mathbf{I} , so we calculate its partial derivatives via a finite-difference operation. Also, we can write the partials of Q_j and R_j as follows:

$$\frac{\partial}{\partial v_n} Q_{j_k} = \frac{\partial}{\partial v_n} x_j + \left(\frac{N_g - 1}{2} - k + 1 \right) \sin \alpha_j \frac{\partial}{\partial v_n} \alpha_j \quad (27)$$

$$\frac{\partial}{\partial v_n} R_{j_k} = \frac{\partial}{\partial v_n} y_j - \left(\frac{N_g - 1}{2} - k + 1 \right) \cos \alpha_j \frac{\partial}{\partial v_n} \alpha_j. \quad (28)$$

The partials of x_j and y_j (the scaled and rotated coordinates of the j^{th} LP) and α_j (the angle of the j^{th} profile) have a straightforward analytic solution. This completes the calculation of the gradient terms in the objective function, J . The next section presents some results using PSAM.

3. Results

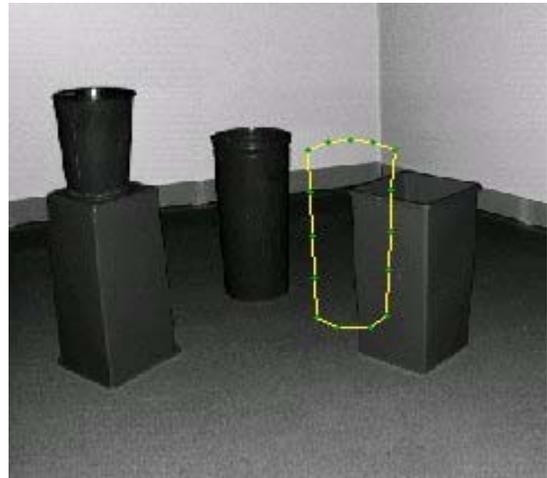
The PSAM algorithm was applied to images of a mock-up industrial scene. Ten different views were captured of the scene in Fig. 1 by varying the horizontal perspective by ± 30 degrees and the distance from the scene by ± 5 feet. The varied viewpoints create object orientation variations and scale variations that must be modeled by the GSM. LPs were manually placed on object boundaries in the 10 views and used as the training set. In the examples shown the test image was left out of the training set used for segmentation.

One result of a segmented cylinder is shown in Fig. 1. A similar result of the same object from a visually obstructed viewpoint is shown in Fig. 2. Note that although a substantial part of the cylinder is occluded, the object was still recognized and segmented successfully with only a slight deformation of the left edge of the cylinder. Also note that the edges of the segmented cylinder are not clearly visible against the dark floor in either example. In these regions the lack of clear edge detail forces the segmentation to be more dependent on the shape model than the gray-level model. In both cases the average LP error (as compared to the ground truth training data) was < 2 pixels. It took ~ 1 - 2 seconds to run the algorithm for each case on a 300MHz PC platform.

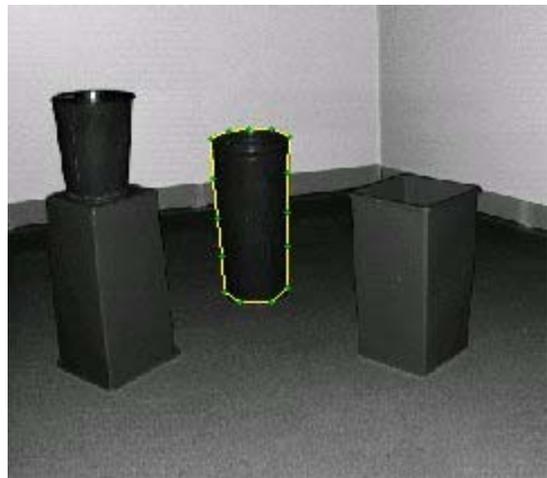
4. Conclusions and Future Work

A theoretical formulation for PSAM has been presented that has several improvements over similar techniques. Most notable is the formulation of an objective function that allows simultaneous optimization of the GSM and GLM. The algorithm has been demonstrated successfully on a simple industrial-type scene.

There is work to be done to validate this new approach on more complex scenes and objects. An earlier version of the PSAM approach has been tested extensively on medical image data with a high degree of complexity [9], so it is quite reasonable to predict that much more complex industrial scenes could be successfully segmented using this approach. This earlier work also provides a detailed comparison of the PSAM and ASM approaches in terms of PSAM's superior performance on medical image data.

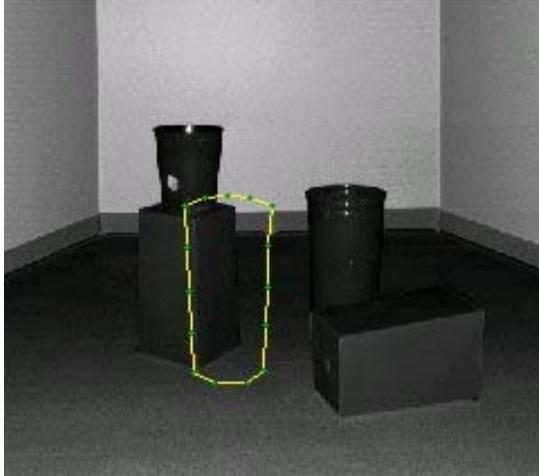


(a)

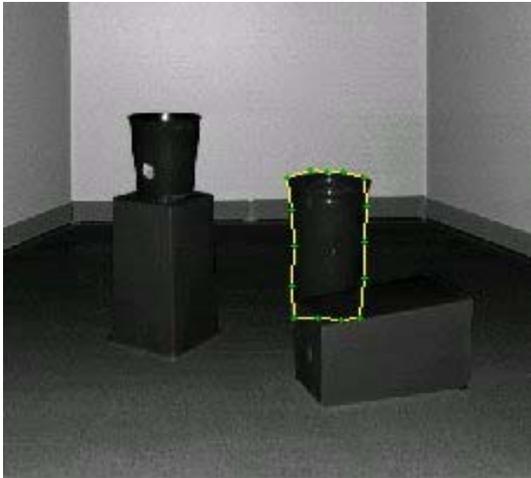


(b)

Figure 1. Preliminary segmentation results using the new shape modeling approach. The initial model position is shown in (a) and (b) shows the final boundary position.



(a)



(b)

Figure 2. Initial model position (a) and segmentation result (b) using the new shape model optimization approach on an occluded object.

In terms of future enhancements to PSAM, optimization strategies other than gradient descent will be explored. Other approaches may be less susceptible to getting trapped in local minima due to spurious edges in a complex scene, for example. Finally, extension of the approach to 2.5-dimensional (e.g. range) and 3-dimensional (volumetric) data is underway. Successful segmentation of 3D synthetic volumetric data has been demonstrated, and segmentation of real-world 3D structures will follow.

5. References

[1]. E. L. Juarez, C. Dumont, M. Abidi, "Object modeling in multiple-object 3D scenes using deformable simplex meshes," *12th Annual Symposium Electronic Imaging 2000: Science and Technology*, SPIE Photonics West, San Jose, California USA, January 23-28, 2000.

[2]. Y. Zhang, J. Price, M. Abidi, "Superquadrics-based object representation of complex scenes from range images". *Conference on Three-Dimensional Image Capture and Applications III*, San Jose, CA, Proceedings of SPIE vol. 4298, 2001.

[3]. T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active shape models — their training and application", *Computer Vision and Image Understanding*, vol. 61, n. 1, pp. 38-59, Jan. 1995.

[4]. Y. Wang and L. Staib, "Boundary finding with correspondence using statistical shape models", *IEEE Proc. Computer Vision and Pattern Recognition*, pp. 338-345, 1998.

[5]. C. Kervrann and F. Heitz, "A hierarchical markov modeling approach for the segmentation and tracking of deformable shapes", *Graphical Models and Image Processing*, vol. 60, n. 3, pp. 173-195, May 1998.

[6]. N. Duta and M. Sonka, "Segmentation and interpretation of MR brain images: an improved active shape model", *IEEE Trans. on Medical Imaging*, vol. 17, no. 6, December 1998.

[7]. S. Gleason, H. Sari-Sarraf, M. Paulus, D. Johnson, M. Abidi, "Automatic screening of polycystic kidney disease in x-ray CT images of laboratory mice," *Proc. SPIE Conf. on Medical Imaging*, 2000.

[8]. S. S. Gleason, H. Sari-Sarraf, M. J. Paulus, D. K. Johnson, M. A. Abidi, "Statistical-based deformable models with simultaneous optimization of object gray-level and shape characteristics," *Proc. Southwest Symposium on Image Analysis and Interpretation*, 2000.

[9]. S. Gleason, "Development of a unified probabilistic framework for segmentation and recognition of semi-rigid objects in complex backgrounds via deformable shape models," A Dissertation for the Doctor of Philosophy Degree, The University of Tennessee, Knoxville, May, 2001.

[10]. M. Paulus, H. Sari-Sarraf, S. Gleason, M. Bobrek, J. Hicks, D. Johnson, J. Behel, and L. Thompson, "A new x-ray computed tomography system for laboratory mouse imaging," *IEEE Trans. on Nucl. Science*, July, 1999.

[11]. C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society B*, 53(2):285-339, 1991.

[12]. R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons, New York, 1973.