Conditional Density Learning via Regression with Application to Deformable Shape Segmentation

Jingdan Zhang[1,2], Shaohua Kevin Zhou[1], Dorin Comaniciu[1], and Leonard McMillan[2]
[1]Integrated Data Systems Department, Siemens Corporate Research, Princeton, NJ 08540
[2]Department of Computer Science, UNC Chapel Hill, Chapel Hill, NC 27599

Abstract

Many vision problems can be cast as optimizing the conditional probability density function $p(C|I)$ where I is an image and C is a vector of model parameters describing the image. Ideally, the density function $p(C|I)$ would be smooth and unimodal allowing local optimization techniques, such as gradient descent or simplex, to converge to an optimal solution quickly, while preserving significant nonlinearities of the model. We propose to learn a conditional probability density satisfying these desired properties for the given training data set. To do this, we formulate a novel regression problem that finds a function approximating the target density. Learning the regressor is challenging due to the high dimensionality of model parameters, C, and the complexity of relating the image and the model. Our approach makes two contributions. First, we take a multilevel refinement approach by learning a series of density functions, each of which guides the solution of optimization algorithms increasingly converging to the correct solution. Second, we propose a new data sampling algorithm that takes into account the gradient information of the target function. We have applied this learning approach to deformable shape segmentation and have achieved better accuracy than the previous methods.

1. Introduction

Deformable shape segmentation is important to many computer vision applications. Three typical examples are shown in Figure 1: (A) a corpus callosum border is segmented from a mid-sagittal MR image; (B) an endocardial wall of the left ventricle is segmented from an echocardiogram; and (C) facial features are localized in a face image.

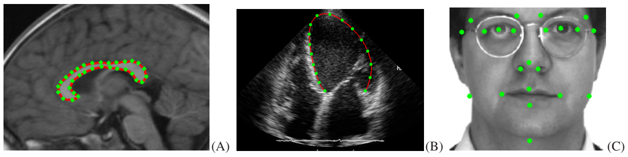


Figure 1. *Three examples of deformable shape segmentation and localization: (A) corpus callosum border segmentation, (B) endocardial wall segmentation, and (C) facial feature localization.*

The object shape in an image I can be represented by a set of continuous model parameters, C, which define the shape and position of the object. Shape segmentation can be considered as optimizing a conditional probability density $p(C|I)$, which represents the probability of the model parameters describing the target object in the image. In any attempt to solve the above problem, there are two questions to be answered: (i) How to construct the density $p(C|I)$; and (ii) How to find the optimal solution for the given $p(C|I)$.

In general, finding the model that maximizes the probability is a difficult optimization problem due to the complexity of image appearance and high dimensionality of the model parameters. Usually, we can have an initial approximation to the correct solution that is provided by some prior knowledge or via rigid object detection [17]. Optimization techniques can then be used to refine the segmentation results. A variety of algorithms are proposed for this purpose, such as active contour model (ACM) [9], active shape model (ASM) [3] and active appearance model (AAM) [1].

It is natural to use general-purpose optimization techniques such as gradient descent or simplex to find an optimal solution. In order to guarantee the optimal convergence of these kinds of algorithms, $p(C|I)$ should be smooth and have only one global maximum, which is the correct solution. Previous approaches have difficulty in guaranteeing this goal.

A common way to construct $p(C|I)$ is to learn this probability density from training data. The density $p(C|I)$ can be constructed by either through generative approaches or discriminative approaches. A generative approach learns a model $p(I|C)$ from the ground truth examples and calculates $p(C|I)$ using Bayes' rule. Although the learned model is sufficient to represent the ground truth examples, it cannot guarantee the proper convergence of the local optimization algorithms at the segmentation stage. When the model C has only one parameter, a typical shape of the learned $p(C|I)$ is shown in Figure 2(A). The global maximum is usually near the ground truth. However, it is not smooth and it could have several local maximums. The cost functions

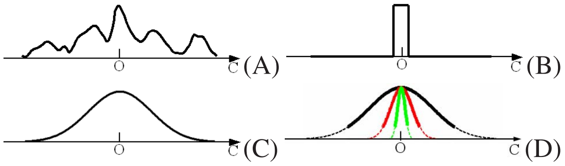


Figure 2. *The learned $p(C|I)$ when C is one dimensional: (A) a generative approach, (B) a classification approach, (C) the regression approach, and (D) the multi-level regression, the functions defined on the feasible regions are drawn using thick lines. The ground truth of the model is O.*

used in energy minimization techniques, such as ACM [9], also suffer the same problem. Thus, the initial solution has to be close enough to the ground truth to make the local optimization algorithm effective. One example of discriminative approaches is the classification method, which directly models $p(C|I)$ by pooling together ground truth examples as positives and the background examples, which are in the complement set of the ground truth examples, as negatives [17]. The learning focus is to build a model to distinguish between foreground and background. The learned $p(C|I)$ is like a boxcar function around the ground truth, e.g., a 1D example shown in Figure 2(B). This model does not provide useful gradient information for local optimization. As a result, the solution is estimated by the exhaustive search, which is computationally prohibitive when the dimensionality of the model C is high.

In this paper, we construct $p(C|I)$ with a desired shape that guarantees the proper convergence of general-purpose local optimization techniques. We learn $p(C|I)$ from annotated training data by formulating a regression task. We constrain the learned density $p(C|I)$ to possess a desired unimodal, smooth shape (such as the bell shape of a normal density) in the model space, which can be used by local optimization algorithms to efficiently estimate the correct solution. Figure 2(C) shows the ideal shape of the 1D $p(C|I)$ learned in this way.

We employ the boosting principle to learn our regressor by selecting relative features to form an additive committee of weak learners. Each weak leaner, based on a Haar-like feature that can be computed rapidly, provides a rough fitness measurement of the object to the image's appearance. The learned regressor computes a robust measurement of fitness by integrating the measurements of selected weak learners.

In most non-rigid segmentation applications, the model C has many parameters, including both pose and shape parameters. It is challenging to learn a function $p(C|I)$ via regression that approximates the target density sufficiently well in the whole parameter space due to insufficient sampling in the high-dimensional model space. To address this 'curse of dimensionality', the number of training examples should be exponential to the dimension of the model space to guarantee training accuracy. Furthermore, appearance variations and noisy imaging artifacts make the problem of insufficient sampling worse by introducing complexity to the regression input.

We make two contributions to tackle the learning challenges. First, we propose a multi-level approach that learns a series of conditional densities, each of which is defined on a feasible region instead of the whole parameter space. By such a design, the regressors defined on feasible regions focus more and more on the region close to the ground truth. A one dimensional multi-level example is shown in Figure 2(D). At the segmentation stage, we perform a series of local optimizations based on the corresponding trained regressors to refine the segmentation result.

Second, when learning an individual regressor, we propose a sampling algorithm for more effective training. The algorithm samples the most important regions in the model space for regression by leveraging the gradient information of the target probability function, which is essential for guiding the local optimization algorithms to find the correct solution.

2. Previous work

In the previous model-based segmentation approaches, a variety of algorithms have been proposed to efficiently search for solutions in a high-dimensional model space. Two typical examples are active shape model (ASM) [3] and active appearance model (AAM) [1]. Our approach focuses on learning a density function with a desired shape tailored for optimization algorithms. Both our approach and AAM search based on a fitness measure of the current hypothesis model. At a hypothesis point in the model space, AAM determines a search direction, which is a vector, based on current match error via a 'difference decomposition' method. While in our approach, the regressor gives a scalar fitness measurement at a point and the search direction is determined from the measurements on local neighborhood points.

To avoid searching in high dimensions, a shape can also be directly inferred from image appearance by searching for the most similar shape from a list of candidate shapes using a sample-based statistical model [7] or a ranking method [19]. This kind of approach needs a large set of training examples for establishing the relationship between appearance and shape. Also, it can only infer the non-rigid shape variation and it relies on other algorithms to determine the global rigid transformation of the shape.

The conditional probability density function can be also constructed using an energy-based model via a convolutional neural network[10]. This approach has been used for detecting similarity transforms of objects [13]. It is unclear how to extend this framework [10] to a high-dimensional model space that includes the parameters for non-rigid deformation.

Regression and boosting [8, 6] are widely used for computer vision applications [17, 7, 18, 20, 21]. Image-based regression has been used to directly estimate the shape deformation [20, 21] and the output of the regressor is more than one dimension. We only tackle a single-output regression problem, which is less complex than the regression problem posed in [20, 21].

Multi-resolution approaches are widely used to improve the efficiency and robustness of the segmentation algorithms [9, 3, 1]. In a typical multi-resolution framework, an image pyramid is built and the optimization is performed in a coarse-to-fine manner. It is generally true that the local maximums of $p(C|I)$, e.g., the local maximums in Figure 2(C), will be smoothed out in the coarse level. However, the useful details might also be lost if the resolution of image is too low. To determine a proper resolution for optimization is usually based on heuristics. We apply the coarse-to-fine principal in a different way, in which a series of functions with nested feasible regions are learned from a single resolution image. By explicitly defining the shape of each function, the multi-level refinements are more controllable.

3. Regression using boosting method

We represent an object in an image I by a set of continuous model parameters $C = (c_1, \ldots, c_D)$, where D is the dimensionality of C. In different applications, C can contain parameters for rigid transformation, or parameters for non-rigid shape deformation, or both. We use regression to learn $p(C|I)$. We determine a target density $q(C|I)$ that possesses a desired unimodal, smooth shape. This can be achieved by defining $q(C|I)$ as a normal density of C:

$$q(C|I) = N(C; \mu(I), \Sigma), \tag{1}$$

where $\mu(I)$ is the ground truth model for the training image I and Σ is an appropriate covariance matrix. In order to simplify computation, we assume the model parameters to be independent. Thus $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_D)$. If there is a dependency among model parameters, a linear transformation can be applied to make them independent.

One additional benefit of using the normal distribution is that it enables the learned regressor to focus on part of the parameter space by setting Σ properly. We discuss how to determine Σ based on an initial error range in section 4.

We apply regression to fit the density $p(C|I)$ to the target density $q(C|I)$. The density $p(C|I)$ learned in this way gives an image match score for each hypothesis model C. This score reflects the distance between the hypothesis and the ground truth.

For a given image I, let $x(I, C)$ be a feature image extracted from the image I using a hypothesis model C. For conciseness, we use x instead of $x(I, C)$ when there is no confusion in the given context. In section 3.2, we address how to obtain x efficiently. Our goal is to find a function $F(x)$ that serves as the density $p(C|I)$. We sample a set of training examples from the input annotated images. A training example is a pair $(x(I_j, C_n), q(C_n|I_j))$, where I is a training image and C_n is a point in the model space. We will discuss how to sample C_n in section 4.1. Regression minimizes the training error while solving the following minimization problem:

$$\hat{F}(x) = \arg\min_{F \in \mathcal{F}} \sum_{n=1}^{N} L(q(C_n|I_n), F(x_n)), \tag{2}$$

where N is the number of training examples, $\mathcal{F}$ is the set of allowed regressors and $L(\circ, \circ)$ is the loss function that penalizes the deviation of the regressor output $F(x)$ from the target probability density $q(C|I)$.

3.1. Boosting

In the boosting method for regression, regressors take the following form:

$$F(x) = \sum_{t=1}^{T} g_t(x); g_t(x) \in \mathcal{G}, \tag{3}$$

where each $g_t(x)$ is a weak learner and $F(x)$ is a strong (more accurate) learner. Further, it is assumed that a weak learner $g(x)$ lies in a *dictionary* set or weak-learner set $\mathcal{G}$.

Boosting iteratively approximates the target function $q(C|I)$ by adding one more weak learner to the regression output:

$$F'(x) = F(x) + g(x). \tag{4}$$

At each round of boosting, we select the learner $\hat{g}$ that most decreases the loss function, by the following greedy choice:

$$\hat{g} = \arg\min_{g \in \mathcal{G}} \sum_{n=1}^{N} L(q(C_n|I_n), F(x_n) + g(x_n)). \tag{5}$$

In this paper, we used the quadratic loss function. The solution of (5) is simply the weak learner that best predicts the current residuals $P(C_n|I_n) - F(x_n)$.

Shrinkage [4, 5] is a technique for reducing overfitting in boosting methods. The idea is very simple: after each round of boosting, we scale the newly selected learner $g(x)$ by a shrinkage factor $\gamma \in [0, 1]$. The resulting update rule is

$$F'(x) = F(x) + \gamma \hat{g}(x), \tag{6}$$

where $\hat{g}$ is the optimal solution found in equation 5. We found that a modest choice of $\gamma = 0.5$ gives good results.

The feature image associated with a model

For a hypothesis model C, the corresponding image patches are sampled from the image I to obtain the feature image $x(I, C)$. We use a set of image patches associated with the current shape C to represent x. Suppose that each shape is represented by M control points, $M + 1$ subimages are extracted from the image as shown in Figure 3. Each subimage has its position, orientation and scale. The first subimage, indicated by the red box in Figure 3, contains the whole object. Its configuration is determined by the object pose. This image patch contains global information to indicate the fitness of the pose parameters. The remaining subimages,

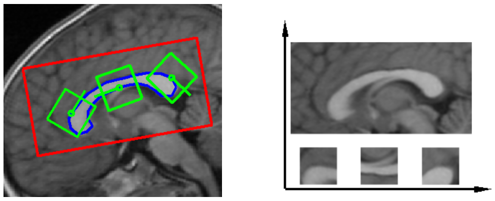


Figure 3. *The feature image x associated with a hypothesis model C. The contour represented by the model C is plotted as blue line. The subimage enclosed by the red box contains global fitness information. The subimges enclosed by the green boxes contains the local fitness information. The image x is composed by subimages with normalized orientation as shown on the right.*

indicated by the green boxes in Figure 3, correspond to the control points, where the configurations are determined by the position, local orientation and scale of the control points. These patches contain local information which is useful for measuring the fitness of the local shape.

How to use the global and local information effectively is application specific. In previous approaches, heuristic rules were used to make decisions [3, 1], such as whether there are strong edges at boundaries or whether the object has an overall unique appearance. In our learning-based approach, we let the algorithm decide which information is most useful by selecting weak learners that mostly decease the regression error. Thus, we form the feature image x to contain both local and global information.

3.2. Weak learner

The dictionary set $\mathcal{G}$ contains weak learner candidates, each of which gives a rough fitness measurement of an object shape to the image's appearance. Intuitively, this set must be sufficiently large to enable rendering of the highly complex output function $F(x)$, through a linear combination of weak learners. Also the computational cost of the feature image x and each $g(x)$ must be low enough for fast evaluation of $F(x)$. We propose an efficient way of computing $g(x)$ from the image I with the model C.

A weak learner $g(x)$ is associated with a Haar filter feature $h(x)$. Each Haar filter $h(x)$ is constrained to be within a subimage of x. It has its own attributes: type, window position, and window size. One can generate a huge number of Haar filters by varying the filter attributes.

All these filters can be evaluated efficiently by precomputing an integral image. See [17] for details. We can use a set of pre-computed integral images with different orientations to compute $h(x)$, eliminating the need to resampling x at run time. This enables the computation of a weak learner very efficiently.

We use the piecewise linear functions as the elements of the dictionary set. A piecewise linear function that is associated with a feature function models the feature response in a piecewise fashion:

$$g(x) = a_{j-1} + \frac{h(x) - \eta_{j-1}}{\eta_j - \eta_{j-1}}(a_j - a_{j-1});\ \ h(x) \in (\eta_{j-1}, \eta_j], \tag{7}$$

where $\{\eta_j;\ j = 0, \ldots, J\}$ divide the range of the feature response equally with $\eta_0 = \min(h(x))$ and $\eta_J = \max(h(x))$ and $\{a_j;\ j = 0, \ldots, J\}$ are the values of the function at the corresponding nodes. At each boosting round, $\{a_j\}$ is computed through solving a linear equation for an optimal least-square solution.

4. Multi-level regression and gradient-based sampling

It is hard to fit a strong learner, $F(x)$, to $q(C|I)$ across the entire model space due to the complexity of the image appearance and the lack of sufficient training examples for representing the high-dimensional model space. Even if $F(x)$ approximates $q(C|I)$ well, the gradient magnitude of $F(x)$ may be too small in some regions of the model space to facilitate the fast convergence of local optimization algorithms.

We propose a multi-level regression to tackle this difficulty by training a series of regressors $F_k(x)$, $k = 1, \ldots, K$, each of which is defined on a feasible region Ω_k of the model space. The feasible region Ω_k is defined as a D-dimensional ellipsoid:

$$\sum_{d=1}^{D} \frac{(c_d - \mu_d(I))^2}{r_{k,d}^2} = 1 \tag{8}$$

where $R_k = (r_{k,1}, \ldots, r_{k,D})$ defines extreme values of each dimension. Let R_1 be the initial error range prior to the local refinement, which can be determined by statistical analysis of the training data. The feasible regions of all levels have the same center $\mu(I)$, which is the ground truth model for the training image I. In the following discussion, we let $\mu(I)$ be at the origin of the model space without loss of generality.

To gradually decrease segmentation error level-by-level, we construct nested feasible regions that shrink to the ground truth:

$$\Omega_1 \supset \Omega_2 \supset \ldots \supset \Omega_K \ni \mu(I). \tag{9}$$

At each level k, we set the target density $q_k(C|I)$ and train $F_k(x)$ to make sure that, in optimizing $F_k(x)$ at the testing stage, an initial solution in the region $\Omega_k - \Omega_{k+1}$ has a high probability of being pushed into the next feasible region Ω_{k+1}. To achieve this goal, we design the function $q_k(C|I)$ to exhibit a high gradient in the region $\Omega_k - \Omega_{k+1}$ and sample more training examples in this region. This leads to a *gradient-based sampling* approach.

Next, we first illustrate the construction of feasible regions and related gradient-based sampling strategy using a 1D example and then extend it to higher dimensions.

4.1. Multi-level regression in 1D

Figure 4 shows the plots of a 1D target function $q(C|I)$ and its gradient magnitude $|\nabla q(C|I)|$. From the figure we

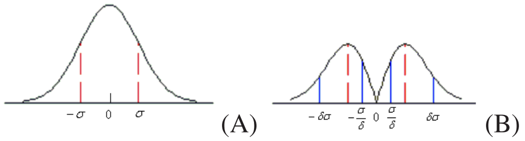


Figure 4. *The plots of (A) 1D normal distribution $q(C|I)$ and (B) the corresponding gradient magnitude $|\nabla q(C|I)|$.*

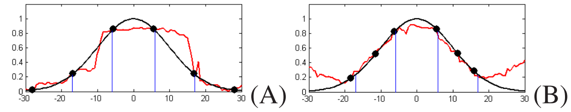


Figure 5. *The target $q(C|I)$ (black line) and the learned $p(C|I)$ (red line) with (A) uniform sampling and (B) gradient-based sampling.*

observe that the gradient magnitude of $q(C|I)$ reaches its maximum at $c_1 = \pm\sigma$ and gradually approaches zero when $c_1 \to \pm\infty$ and $c_1 \to 0$.

Let σ_k be the standard deviation of the function $q_k(C|I)$ and δ is a pre-specified constant (we empirically set $\delta = 1.7$ for all experiments). We construct the nested feasible regions as $\Omega_k = [-\sigma_k\delta, \sigma_k\delta]$ with

$$\sigma_k = r_k/\delta, \ \ r_{k+1} = r_k/\delta^2, \tag{10}$$

where r_1 is the extreme initial value. This construction is based on the fact that the region $[-\sigma_k\delta, -\sigma_k/\delta] \cup [\sigma_k/\delta, \sigma_k\delta]$, centered around the maxima of the gradient magnitude (*i.e.*, $\pm\sigma_k$), contains large gradient magnitude.

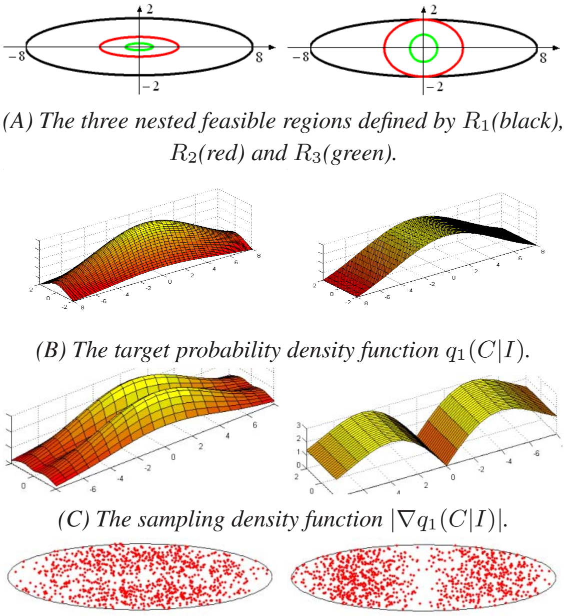
Figure 5 compares two sampling strategies: uniform sampling and sampling based on gradient magnitude. Only six examples are used in the regression to simulate the sparsity of sampling in high dimensions. For the moment, we assume that sampling outside of the feasible region is allowed. The standard deviation is $\sigma = 10$. In the region $[-\sigma\delta, -\sigma/\delta] \cup [\sigma/\delta, \sigma\delta]$, the regressor learned by sampling based on the gradient magnitude (Figure 5(B)) more faithfully captures the shape of the target density function than that learned by uniform sampling (Figure 5(A)).

In summary, the learned regressor $F_k(x)$ should fit the target density $q_k(C|I)$ well enough in the region $\Omega_k - \Omega_{k+1}$ to provide good guidance in optimization. This is achieved by selecting examples primarily in this region that is constructed to have high gradient magnitude. We use the Metropolis sampling algorithm [12] to sample training examples in the feasible region Ω_k with the sampling density function $|\nabla q_k(C|I)|$.

4.2. Multi-level regression in high dimension

As in 1D case, we design a series of target densities with nested feasible regions. The shape of each target density $q_k(C|I)$ is defined by the covariance matrix $\Sigma_k = \text{diag}(\sigma_{1,k}, \ldots, \sigma_{D,k})$. Determining the optimal covariance matrices Σ_k is a hard problem; here we consider two simple approaches.

The first approach is to learn a regressor for decreasing a feasible region in all dimensions uniformly. Similar to Eq. (10), this can be achieved by setting $\sigma_{k,d} = r_{k,d}/\delta$ and $r_{k+1,d} = r_{k,d}/\delta^2$. This approach works well when the elements in R_1, which define the initial error range, are roughly the same. However, in real applications, the range of model parameters varies greatly. Typically, the error range of pose parameters is larger than the error range of shape parameters; hence narrowing down the error ranges in all dimensions at the same rate is usually inefficient. A more practical solution is to first decrease error along the dimensions with larger error ranges. For example, in [7], the pose parameters are determined before estimating shape parameters.



(A) The three nested feasible regions defined by R_1(black), R_2(red) and R_3(green).

(B) The target probability density function $q_1(C|I)$.

(C) The sampling density function $|\nabla q_1(C|I)|$.

(D) The sampled points in the model space based on $|\nabla q_1(C|I)|$.

Figure 6. *Two sampling approaches. Left: the first approach. Right: the second approach.*

The second approach achieves this goal by allowing the target function to have large gradient magnitude along the dimensions with large error ranges. The learned regressor thus focuses on decreasing error along these dimensions. Let $r_k^{\max}$ be the largest element in R_k. We evolve $\sigma_{k,d}$ and $r_{k,d}$ as follows:

$$\begin{array}{lll} \sigma_{k,d} = r_k^{\max}/\delta, & r_{k+1,d} = r_k^{\max}/\delta^2 & \text{if } r_{k,d} > r_k^{\max}/\delta^2 \\ \sigma_{k,d} = \sigma_{\max}, & r_{k+1,d} = r_{k,d} & \text{otherwise} \end{array} \tag{11}$$

where $\sigma_{\max}$ is a constant (typically a large value). The condition $\sigma_{k,d} = \sigma_{\max}$ means that the kth target density varies little along the dth dimension and hence the learned regressor $F_k(x)$ makes no attempt to decrease the model error along this direction. Geometrically, the feasible region gradually shrinks from a high-dimensional ellipsoid to a sphere, and then shrinks uniformly thereafter.

We use a 2D example to illustrate the rationale of this setting. Let $R_1 = (8, 2)$, where the error in the first dimension is much larger than that in the second one. For both approaches mentioned above, Figure 6 shows the three

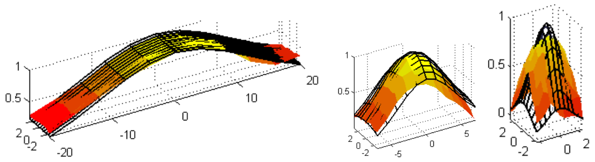


Figure 7. *The 2D slices of $F_1(x)$(left), $F_2(x)$(middle), and $F_3(x)$(right) on a testing image.*

feasible regions, the target function $q_1(C|I)$ for the first level, the sampling density function $|\nabla q_1(C|I)|$, and the corresponding sampling results. Both approaches gradually shrink the feasible region. The shape of the feasible region remains consistent in the first approach while in the second approach, it changes from an ellipse to a circle. Because the first approach samples many points around the short axis, this makes the regression less effective. The second approach based on Eq. (11) is more effective.

5. Experiments

In all experiments, we used control points to represent objects in the image. The training inputs are a set of images and their associated annotations. In training, the generalized Procrustes analysis [3] is invoked to compute the mean shape and rigid alignment of objects in the images. We then apply the PCA to the aligned shape in order to build a shape space, spanned by a reduced set of eigenvectors associated with the largest eigenvalues λ. The initial error ranges of the shape parameters are assumed to be $3\sqrt{\lambda}$.

In testing, we used a standard simplex algorithm [14] as the local optimization algorithm to maximize $F_k(x)$. The simplex method is insensitive to shallow maxima caused by image noise and regression error. The learned $F_k(x)$'s are used sequentially and the converged solution of the current level is the initial value used in the next level. The segmentation errors are measured as the average Euclidean distance between corresponding control points of the segmented shape and the ground truth.

5.1. Corpus callosum border segmentation

Segmentation of the corpus callosum structure in mid-sagittal MR images is a common task in brain imaging [16]. In this data set, the corpus callosum has a clear border making segmentation relatively easy. We collected a total of 148 mid-sagittal MR images with the corpus callosum border annotated by experts using contours with 32 control points. The corpus callosum roughly occupies 120×50 pixels. We used 74 images for training and the remaining 74 images for testing.

The goal of our approach is to locally refine a model C. In a 2D image, the model C has 4 pose and 6 shape parameters. The pose includes 2D translation, rotation, and scale: (t_x, t_y, θ, s). The initial range of the pose is given as $[20, 20, \pi/9, 0.2]$, which means 20 pixels in translation, 20 degrees in rotation and 20% in scale. The six shape parameters account for 85% of the total shape variation. In training, three levels of regressors are trained to approximate the target functions with Σ defined in Eq. (11). We sampled 3000 examples from each image and set the maximum number of weak learners in a regressor to be 500.

(A)	ASM	refine1	refine2	refine3
w/out noise	3.33±1.84 3.07±1.16	2.86±3.05 2.44±0.74	2.02±3.08 1.63±0.46	1.76±3.03 1.37±0.41
with noise	6.77±3.16 6.31±2.20	3.66±4.33 2.85±0.90	2.98±4.63 2.13±0.59	2.94±4.68 2.08±0.64

(B)	ASM	refine1	refine2	
LV	26.20±17.64 23.43±12.03	11.09±4.31 10.43±3.11	10.07±4.52 9.41±3.06	

(C)	AAM	Shape Inf.	RankBoost	Regression
AR face	5.94±2.81 5.50±1.69	5.16±1.26 4.99±1.06	4.24±1.09 4.09±0.88	3.86±0.97 3.72±0.77

Table 1. *The mean and standard deviation of the segmentation errors. In each cell, there are two rows: the first row reports the mean and standard deviation obtained using all testing data and the second row using 95% of testing data (excluding 5% outliers).*

The learned $F_k(x)$'s on a testing data are shown in Figure 7. The overlayed wire-frame meshes are target functions. Because $F_k(x)$'s are high dimensional functions, we plotted the 2D slices by varying the 1st and the 5th parameters of the model in the feasible region while fixing the remaining parameters as the ground truth, where the 1st is a translation parameter and the 5th is a shape parameter corresponding to the largest eigenvalue. The regressors predict the target density well, showing a conspicuous mode, and they are ready to be used in a local optimization algorithm.

In testing, we randomly generated five starting contours for each testing image. The initial shape parameters are set to zero, *i.e.*, using the mean shape. The initial pose parameters are randomly generated within the error range defined in the training. The average error of 370 starting contours is 12.65 pixels. The computational time of our approach is roughly 3 seconds. We compared our approach with ASM. The discussion in [2] explains that ASM works well when the border is supported by strong edges[1]. We also applied multi-resolution searching and carefully tuned the parameters to achieve good performance.

Table 1(A) shows the mean and standard deviation of the testing errors. The proposed approach improves the ASM by 47% in terms of mean error. Figure 8(A1) is a plot of the sorted errors, where points on the curve with the same horizontal position do not correspond to a same testing case. There are outliers in the final segmentation results. If we exclude 5% of the testing data as outliers, then the proposed approach improves the ASM by 55% in terms of mean error and reduces the standard deviation too. Further, each level improves the results from the previous level, proving the effectiveness of the multi-level approach.

We then added noise to make the segmentation more challenging. The Gaussian noise with zero mean and 0.2 variance was added to both the training and testing images

[1] We used the Matlab implementation of ASM by Dr. Hamarneh, which is available at http://www.cs.sfu.ca/~hamarneh/software/asm/index.html.

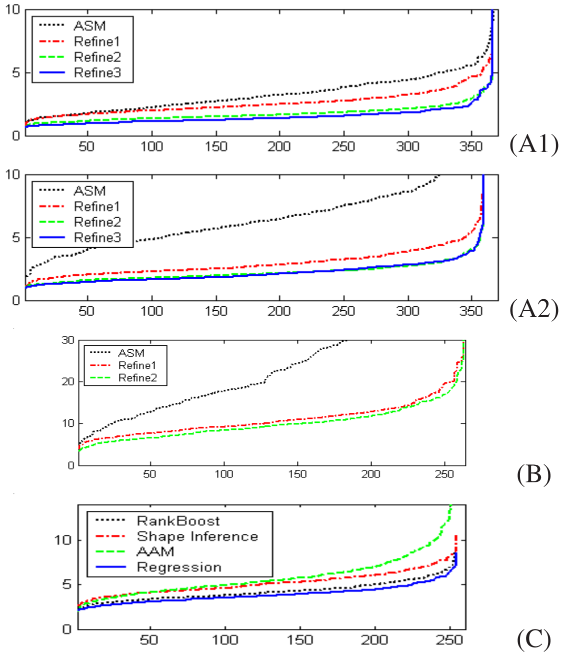


Figure 8. *Sorted errors of the experiment results: (A1) corpus callosum border segmentation on noise free data, (A2) corpus callosum border segmentation on noisy data, (B) LV segmentation, and (C) facial feature localization.*

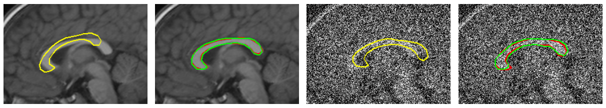


Figure 9. *Two segmentation results obtained by our algorithm. The initial positions are yellow lines. The ground truths are red lines and the segmentation results are green lines.*

with an intensity range $[0, 1]$. Our algorithm and ASM were retrained and tested using the same noise-free setting. The results are shown in Table 1(A) and Figure 8(A2). ASM has a poor performance because ASM cannot estimate accurate intensity profiles normal to the object boundary due to the noise. The performance of our algorithm suffers less since the added noise is considered at the training stage, which proves the effectiveness of the learning approach. Note that in this experiment, the third level refinement cannot improve the segmentation further due to information loss caused by noise.

5.2. Endocardial wall segmentation

In this experiment, we focused on locating the endocardial wall of the left ventricle (LV) in the apical four chamber (A4C) view of echocardiogram. The LV appearance varies significantly across patients and ultrasound images often suffer from signal dropouts and speckle noise. There is no clear edge at the endocardial wall. This combination of factors makes automatic LV segmentation challenging.

In the experiment, we collected a total of 528 A4C images with the wall of the left ventricle annotated by experts using contour with 17 control points. The LV roughly occupies 120×180 pixels. We used half of these data for training and the remaining half for testing. In training, two levels of regressors are trained. The model C has 4 pose with initial error range $[50, 50, \pi/9, 0.2]$ and 5 shape parameters account for 80% of the total shape variation.

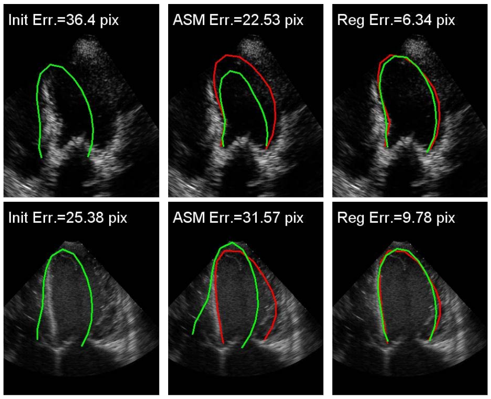


Figure 10. *Two examples on which ASM fails while our algorithm works well. The left column is the initial position. The middle column is the result of ASM. The right column is the result of our algorithm. The ground truths are the red lines and the initial contours and the segmentation results are green lines. The segmentation errors are shown at the top of the images.*

In testing, we randomly generated a starting contour for each testing image. The initial pose parameters were within the error range defined in the training. The initial shape parameters were set to zero. The average initial error is 27.16 pixels. The ASM was also used for comparison. Some detection results are shown in the Figure 10. The segmentation errors are shown in Table 1(B) and in Figure 8(B). The learned regressors outperform the ASM by a large margin, over 60% improvement.

5.3. Facial feature localization

In the third experiment, we tested our approach on facial feature localization using the AR face database [11]. A face image from the database and its annotation of 22 feature points are shown in Figure 1(C). There are a total of 508 images with annotations[2], including 76 males and 60 females with 4 expressions.

We used the exactly same training and testing set as in [19]. In this experiment, half of the data were used for training and half for testing. Examples of the same subject were not used in both training and testing data. The color images were converted to gray-scale images. We also assumed that the face pose is known. The focus is on localizing the nonrigid shape component. The model space is defined by 10 shape parameters, which explain 86% of shape variations. We sampled 1200 examples from an image and trained three levels of regressors. The mean shape is used as the starting point at the testing stage. The average initial error is 5.93 pixels.

[2] The annotations are provided by Dr. Cootes, which is available at http://www.isbe.man.ac.uk/˜bim.

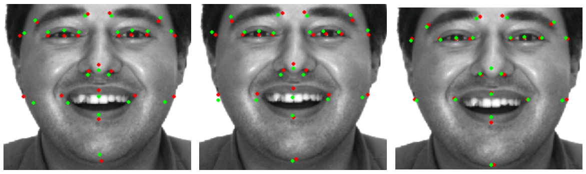


Figure 11. *An example on which our algorithm works better than AAM. The left is the initial position. The middle is the result of AAM. The right column is the result of our algorithm. The ground truth is red dots and the localization result is green dots.*

We compared the performance of our algorithm to the algorithms listed in [19]: AAM[15], shape inference[7], and shape refinement based on rankboost [19]. An example of localization is shown in figure 11. The localization errors of the four algorithms are shown in Table 1(C) and in Figure 8(C). Comparing to shape inference and rankboost, which only consider the candidate shapes in the training set, our algorithm searches in the whole shape space. Again, our approach records the best performance.

6. Discussion and Conclusion

We have presented a regression approach to learning a conditional density function. The target function can be artificially constructed to be both smooth and unimodal and hence easy to optimize. To learn the regressor, the boosting principle has been employed to select relevant features. We have also adopted a multi-level strategy to learn a series of conditional density functions, each of which guides the solution of optimization algorithms increasingly converging to the correct solution. We have then proposed a gradient-based sampling strategy to maximize the gradient contributions in the directions of largest variation. We have successfully applied our approach to segmenting corpora callosa, tracing the endocardial wall of the left ventricle, and localizing facial feature points. Our results consistently outperform those obtained by state-of-the-art methods.

Like all discriminative learning approaches, our approach could suffer the problem of overfitting especially when the variation of training data cannot totally cover that of testing. Because of this, the trained regressor does not have the desired unimodal shape on some testing data and the local optimization algorithm fails to converge to the ground truth. We will analyze the overfitting problem in future work.

7. Acknowledgment

We thank Stephen M. Pizer and J. Stephen Marron for helpful comments. We thank Martin Styner for providing the mid-sagittal MR images and the associated annotations, which are obtained by the support of NIH Roadmap for Medical Research Grant U54 EB005149-01, NIH Conte Center MH064065, NIH RO1 MH61696, NIMH MH64580 and the UNC Neurodevelopmental Disorders Research Center HD 03110T.

References

[1] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):681–685, 2001.

[2] T. Cootes and C. Taylor. Statistical models for appearance for computer vision. In *unpublished manuscript*.

[3] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models–their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[4] H. Copas. Regression, and prediction, and shrinkage. In *J. R. Statist. Soc. B*, volume 45, pages 311–354, 1983.

[5] J. Friedman. Greedy function approxiamtion: A gradient boosting machine. In *The Ann. of Stat.*, volume 28(2), 2001.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistive regression: A statistical view of boosting. In *The Ann. of Stat.*, volume 28, pages 337–374, 2000.

[7] B. Georgescu, X. S. Zhou, D. Comaniciu, and A. Gupta. Database-guided segmentation of anatomical structures with complex appearance. In *Proc. CVPR*, 2005.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, 2001.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. J. Computer Vision*, 1(4):321–331, 1998.

[10] Y. LeCun and F. Huang. Loss functions for discriminative training of energy-based models. In *Proc. AI & Stat.*, 2005.

[11] A. M. Martinez and R. Benavente. The AR face database. 1998.

[12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6).

[13] R. Osadchy, M. Miller, and Y. LeCun. Synergistic face detection and pose estimation with energy-based model. In *NIPS. MIT Press*, 2005.

[14] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. Numerical recipes in C (2nd edition). Cambridge University Press, 1992.

[15] M. B. Stegmann, B. K. Ersboll, and R. Larsen. FAME–a flexible appearance modeling environment. *IEEE Trans. Medical Imaging*, 22(10):1319–1331, 2003.

[16] G. Székely, A. Kelemen, C. Brechbuehler, and G. Gerig. Segmentation of 2D and 3D objects from MRI volume data using constrained elastic deformations of flexible fourier contour and surface models. *MEDIA*, 1(1):19–34, 1996.

[17] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.

[18] J. Zhang, S. Zhou, L. McMillan, and D. Comaniciu. Joint real-time object detection and pose estimation using probabilistic boosting network. In *Proc. CVPR*, 2007.

[19] Y. Zheng, X. S. Zhou, B. Georgescu, S. Zhou, and D. Comaniciu. Example based non-rigid shape detection. In *Proc. European Conf. Computer Vision*, 2006.

[20] S. Zhou and D. Comaniciu. Shape regression machine. In *Proc. IPMI*, 2007.

[21] S. Zhou, B. Georgescu, X. S. Zhou, and D. Comaniciu. Image based regression using boosting method. In *Proc. ICCV*, 2005.