Four-Chamber Heart Modeling and Automatic Segmentation for 3D Cardiac CT Volumes Using Marginal Space Learning and Steerable Features

Yefeng Zheng, Adrian Barbu, Bogdan Georgescu, Michael Scheuering, and Dorin Comaniciu

***Abstract*— We propose an automatic four-chamber heart segmentation system for the quantitative functional analysis of the heart from cardiac computed tomography (CT) volumes. Two topics are discussed: heart modeling and automatic model fitting to an unseen volume. Heart modeling is a non-trivial task since the heart is a complex nonrigid organ. The model must be anatomically accurate, allow manual editing, and provide sufficient information to guide automatic detection and segmentation. Unlike previous work, we explicitly represent important landmarks (such as the valves and the ventricular septum cusps) among the control points of the model. The control points can be detected reliably to guide the automatic model fitting process.**

Using this model, we develop an efficient and robust approach for automatic heart chamber segmentation in 3D CT volumes. We formulate the segmentation as a two-step learning problem: anatomical structure localization and boundary delineation. In both steps, we exploit the recent advances in learning discriminative models. A novel algorithm, marginal space learning (MSL), is introduced to solve the 9-dimensional similarity transformation search problem for localizing the heart chambers. After determining the pose of the heart chambers, we estimate the 3D shape through learning-based boundary delineation. The proposed method has been extensively tested on the largest dataset (with 323 volumes from 137 patients) ever reported in the literature. To the best of our knowledge, our system is the fastest with a speed of 4.0 seconds per volume (on a dual-core 3.2 GHz processor) for the automatic segmentation of all four chambers.

***Index Terms*— Heart modeling, heart segmentation, 3D object detection, marginal space learning**

I. Introduction

Compared with other imaging modalities (such as ultrasound and magnetic resonance imaging), cardiac computed tomography (CT) can provide detailed anatomic information about the heart chambers, large vessels, and coronary arteries [1]. Therefore, CT is an important imaging modality for diagnosing cardiovascular diseases. Complete segmentation of all four heart chambers, as shown in Fig. 1, is a prerequisite for clinical investigations, providing critical information for quantitative functional analysis for the whole heart [2], [3]. In this paper, we propose an automatic 3D heart chamber segmentation system using a surface-based four-chamber heart model. There are two major tasks to develop such an automatic segmentation system: heart modeling (shape representation) and automatic model fitting (detection and segmentation). Due to the complexity of cardiac anatomy, it is not trivial to represent the anatomy accurately while keeping the model simple enough for automatic segmentation and manual correction if necessary. The proposed heart model, as shown in Fig. 1 has the following advantages.

1) The heart valves are explicitly modeled as closed contours along their borders in our model. Therefore, our model is more accurate concerning the anatomy, compared to previous closed-surface mesh models [4]–[6].
2) Important landmarks (e.g., valves and ventricular septum cusps) are explicitly represented in our model as control points[1]. These landmarks can be detected reliably to guide the automatic model fitting process.
3) Our model is flexible. Chambers are coupled at atrioventricular valves and it is easy to extract each chamber from the whole heart model.
4) The proposed model is expandable. Our current work focuses on the addition of extra elements, such as dynamic valve modules [7].
5) We propose two approaches to enforce the mesh point correspondence, namely the rotation-axis based and parallel-slice based methods. With such built-in correspondence, we can easily learn a statistical shape model [8] to enforce shape constraints in our automatic model fitting approach.

Using the proposed heart model, we present an automatic heart segmentation method based on machine learning to exploit a large database of annotated CT volumes. As shown in Fig. 2, the segmentation procedure has two stages: automatic heart localization and control point guided nonrigid deformation estimation. Automatic heart localization is largely ignored in early work on heart segmentation [9]–[12]. Recently, learning based approaches have been successfully demonstrated on many 2D object detection problems [13], [14]. However, there are two challenges in applying these techniques to 3D object detection: 1) the exponential computation demands by the use of exhaustive search and 2) lack of efficient features that can

Manuscript received Oct. 8, 2007; revised July 10, 2008; accepted July 11, 2008. Y. Zheng, B. Georgescu, and D. Comaniciu are with the Integrated Data Systems Department at Siemens Corporate Research, Princeton, NJ, USA. A. Barbu is with the School of Computational Science, Florida State University, Florida, USA. M. Scheuering is with the Computed Tomography Division, Siemens Healthcare, Forchheim, Germany.

A. Bardu contributed to this work while he was with Siemens Corporate Research.

Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

[1]"Landmark" is a term used in relation with the anatomy and a "control point" is a mesh point, representing the corresponding landmark in the mesh.

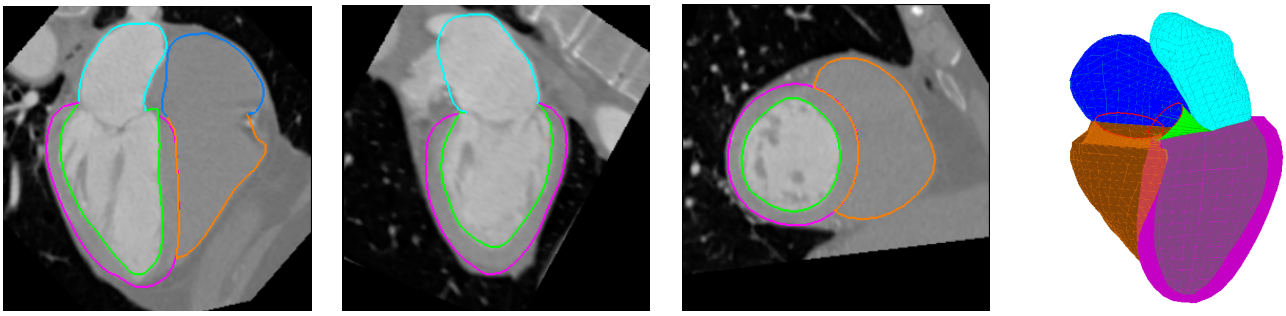


Fig. 1. A four-chamber heart model with green for the left ventricle (LV) endocardium, magenta for the LV epicardium, cyan for the left atrium (LA), brown for the right ventricle (RV), and blue for the right atrium (RA). The left three columns show three orthogonal cuts from a 3D volume data and the last column shows the triangulated surface mesh.

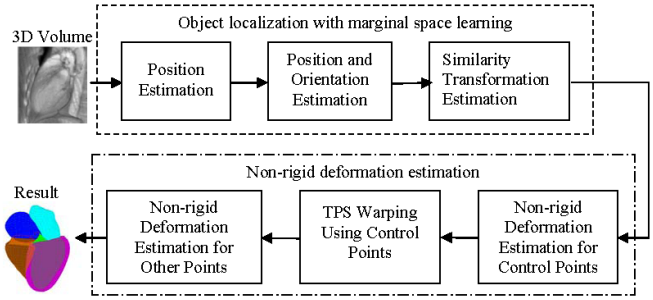


Fig. 2. Diagram for 3D heart chamber detection and segmentation.

be calculated efficiently under rotation.

In this paper, we present an efficient 3D object detection method based on two simple but elegant techniques, marginal space learning (MSL) and steerable features. The idea of MSL is not to learn a classifier directly in the full similarity transformation space but to incrementally learn classifiers on projected sample distributions. As shown in the top dashed box in Fig. 2, we split the estimation into three problems: position estimation, position-orientation estimation, and full similarity transformation estimation. To attack the second challenge, we introduce steerable features, which constitute a very flexible framework. Basically, we sample a few points (e.g., 125 points) from the volume under a sampling pattern and extract a few local features (e.g., intensity and gradient) for each sampling point. The efficiency of steerable features comes from the fact that much fewer points are needed for manipulation, compared to the whole volume. After similarity transformation estimation, we get an initial estimate of the nonrigid shape. We then use a learning-based 3D boundary detector to guide the shape deformation in the active shape model (ASM) framework [8].

In summary, we make two major contributions to the automatic model fitting approach.

1) We propose MSL to search the similarity transformation space efficiently, which reduces the number of testing hypotheses by about six orders of magnitude.
2) We introduce steerable features, which can be evaluated efficiently under any orientation and scale without rotating the volume.

Part of this work has been reported in our conference publications [15], [16]. The remaining of the paper is organized as follows. In Section II we review the previous work on heart modeling and segmentation. Our four-chamber heart model and two schemes to establish point correspondence are presented in Section III. In Section IV, we present an efficient 3D object localization approach, using marginal space learning and steerable features. Nonrigid deformation estimation is discussed in Section V. We demonstrate the robustness of the proposed method in Section VI. This paper concludes with Section VII.

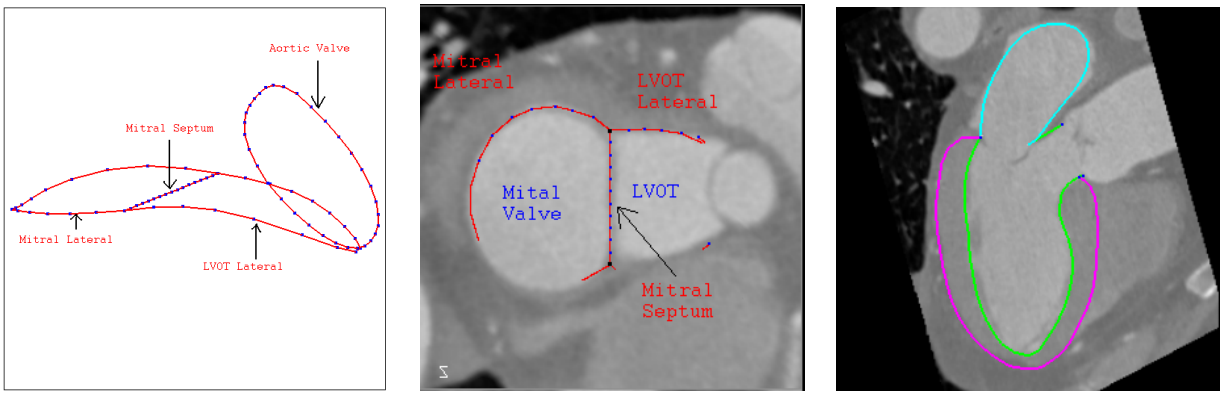
II. Related Work

The research presented in this paper is related to previous work on heart modeling and segmentation.

A. Heart Modeling

Except for a four-chamber heart model from [17] and [10], most of the previous work focused on the left ventricle (LV) and/or the right ventricle (RV). A closed mesh was often used to represent heart chambers [4]–[6]. Nevertheless, it is not clear how the atria interacted with the ventricles around the mitral and tricuspid valves in [4]. The heart model in [10] was more accurate in anatomy and it also included trunks of the major vessels connected to heart chambers. However, artificial patches were added at all valves to close the chamber meshes. These artificial patches were not specifically processed in the segmentation algorithm, therefore, they could not be delineated accurately [17]. In our heart model, we keep the mesh open at a valve. Mesh points around the valves are labeled as control points, which are treated differently to the normal mesh points during automatic segmentation.

The statistical shape model [8] is widely used in nonrigid object segmentation to enforce shape constraints and make the system more robust. However, to build a statistical shape model, it is necessary to establish point correspondence among a group of shapes [8]. There are a few papers on building a statistical 3D shape model automatically using pair-wise or group-wise registration based approaches [18], [19], which are complicated and time consuming. Another approach is to establish correspondence among shapes during the manual labeling process. Though this is difficult for a generic 3D shape, we can consistently resample the surface to establish correspondence for a few simple shapes (e.g., a tube and a parabola) [5], [9].



(a) (b) (c)

Fig. 3. Delineating the mitral and aortic valves. (a) A 3D view of the control points around the valves. (b) Control points around the mitral valve embedded in a CT volume. Since the curves are 3D, they are only partially visible on a specific plane. (c) Annotated LV/LA meshes embedded in a CT volume.

B. Heart Segmentation

Given the heart model, the segmentation task is to fit the model onto an unseen volume. Since the heart is a nonrigid shape, the model fitting (or heart segmentation) procedure can be divided into two steps: object localization and boundary delineation. Most of the previous approaches focused on boundary delineation based on active shape models (ASM) [20], active appearance models (AAM) [21], [22], and deformable models [12], [17], [23]–[26]. These techniques suffer from the following limitations: 1) Most of them are semi-automatic and proper manual initialization is demanded. 2) Gradient based search in these approaches are likely to get stuck in a local optimum.

Object localization is required for an automatic segmentation system, a task largely ignored by previous researchers. Recently, the discriminative learning based approaches have been proved to be efficient and robust to detect 2D objects [13], [14]. In these methods, object detection or localization was formulated as a classification problem: whether an image block contains the target object or not [13]. The parameter space was quantized into a large set of discrete hypotheses. Each hypothesis was tested by the trained classifier to get a detection score. The hypothesis with the highest score was taken as the final detection result. This search strategy is quite different from other parameter estimation approaches, such as deformable models, where an initial estimate is adjusted (e.g., using the gradient descent technique) to optimize a predefined objective function.

Exhaustive search makes the system robust under local optima, however there are two challenges to extend the learning based approaches to 3D. First, the number of hypotheses increases exponentially with respect to the dimensionality of the parameter space. For example, there are nine degrees of freedom for the anisotropic similarity transformation[2], namely three translation parameters, three rotation angles, and three scales. Suppose each dimension is quantized to n discrete values, the number of hypotheses is n^9 (for a very coarse estimation with a small n=5, n^9=1,953,125). The computational demands are beyond the capabilities of current desktop computers. The second challenge is that we need efficient features to search the orientation spaces. To estimate the object orientation, one has to rotate either the feature templates or the volume. Haar wavelet features can be efficiently computed under translation and scaling [13], [27], but no efficient way is available to rotate the Haar wavelet features. Previously, time-consuming volume rotation has been performed to estimate the object orientation [28].

[2]The ordinary similarity transformation allows only isotropic scaling. In this paper, we search for anisotropic scales to cope better with the nonrigid deformation of the heart.

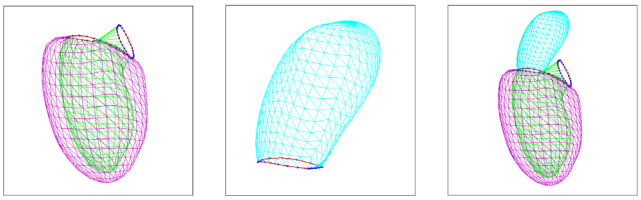


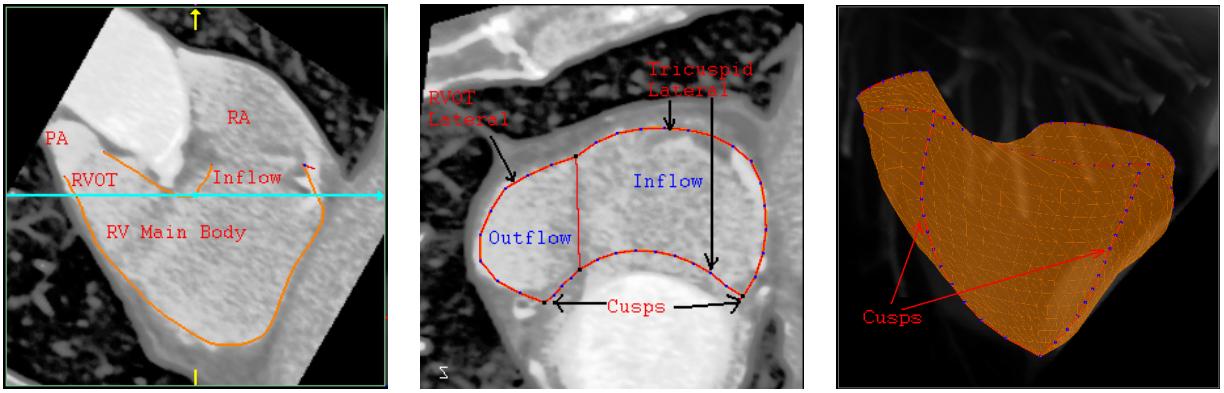
Fig. 4. LV/LA meshes with green for the LV endocardium and left ventricular outflow tract (LVOT), magenta for the LV epicardium, and cyan for the LA. The control points are shown as blue dots and appropriately connected to form the red contours. From left to right are the LV, LA, and combined meshes, respectively.

III. Four-Chamber Heart Modeling

In this section, we first describe our four-chamber heart model, and then present our consistent resampling techniques to establish point correspondence, which is demanded to build a statistical shape model [8]. In the model, some mesh points are special and correspond to distinctive anatomical structures (e.g., those around the valve holes). We label these points as control points. Control points are integral part of the mesh model in the sense that they are also connected to other mesh points with mesh triangles.

A. LV and LA Models

A closed mesh has been used to represent the LV [4]–[6]. Due to the lack of object boundary on the image, it is hard to consistently delineate the interfaces among the LV main body, the left ventricular outflow tract (LVOT), and the basal area around the mitral valve. The mesh often cuts the LVOT and the



(a) (b) (c)

Fig. 5. Delineating the RV control points. (a) The divergence plane (indicated by the cyan line) of the RV inflow and outflow tracts. (b) Delineating the divergence plane. (c) The RV mesh with ventricular septum cusps indicated.

mitral valve at an arbitrary position [4]–[6]. In our heart model, the mitral valve is explicitly represented as a closed contour along its border. Since we exclude the moving valve leaflets (which are hardly visible) from the model, the basal area can be delineated more consistently. Both the endo- and epicardiums are delineated for the LV. The commissure contour of both surfaces corresponds to the mitral valve annulus on one side and the aortic valve level (lying at the bottom edge of the Valsalva sinuses) on the other side, as shown in Fig. 3c. As shown in Fig. 3, three curves are formed by control points around the mitral valve, namely, the mitral lateral (16 points), the mitral septum (15 points), and the LVOT lateral (16 points). They define two closed regions, one for the interface between the LV and the LA, and the other for the LV and the LVOT. The aortic valve (annotated with 32 points) is approximated as a plane, which cuts the valve at the bottom of the Valsalva sinuses. The LA is represented as an open mesh with an open area enclosed by the mitral septum and the mitral lateral control points (as shown in Fig. 3b). Fig. 4 shows the LV/LA meshes with the control points (blue dots connected by red contours). The LV endocardium, epicardium, and LA meshes are represented with 545 points and 1056 triangles each. The LVOT mesh is represented with 64 points and 64 triangles.

B. RV and RA Models

The right ventricle (RV) has a complicated shape with separate inflow and outflow portions. Using a plane (indicated by a cyan line in Fig. 5a) passing the divergence point of the inflow and outflow tracts, we can naturally split the RV into three parts with the RV main body lying below the cutting plane. We will call this plane as "the RV divergence plane." During the manual labeling of ground truth, the RV divergence plane is determined in the following way. We first determine the tricuspid valve, which is approximated as a plane. We then move the plane toward the RV apex to a position where the RV inflow and outflow tracts diverge. As shown in Fig. 5b, two curves, tricuspid lateral (23 points) and right ventricular outflow tract (RVOT) lateral (15 points), are annotated on the RV divergence plane to define the inflow and outflow connections. On a short axis view, the RV main

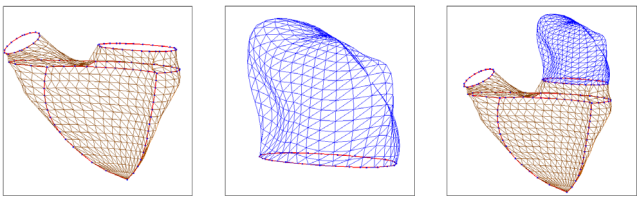


Fig. 6. RV/RA meshes with brown for RV and blue for RA. The control points are shown as blue dots and appropriately connected to form the red contours. From left to right are the RV, RA, and combined meshes, respectively.

body is a crescent (as shown in Fig. 5b). Two cusp points on the intersection are important landmarks, and an automatic detection algorithm should be able to detect them reliably. They are explicitly represented in our model (Fig. 5c). The tricuspid (28 points) and pulmonary (18 points) valves are approximated as a plane. Similar to the LA, the right atrium (RA) is represented as an open mesh with the open area defined by the tricuspid valve. Fig. 6 shows the RV/RA meshes with the control points (blue dots connected by red contours). In our model, the RV is represented with 761 points and 1476 triangles and the RA is represented with 545 points and 1056 triangles.

C. Establishing Point Correspondence

Since the automatic segmentation algorithm (discussed in Section V) exploits a statistical shape model, we need to establish point correspondence. This task is difficult for a generic 3D shape. Fortunately, for a few simple shapes, such as a tube or a parabola, we can consistently resample the surface to establish this correspondence. Since it is easy to consistently resample a 2D curve, we use a few planes to cut the 3D mesh to get a set of 2D intersection curves. The resulting 2D curves are uniformly sampled to get a point set with built-in correspondence. Using different methods to select cutting planes, we develop two resampling schemes, the rotation-axis based method for simple parabola-like shapes such as the LV, LA, and RA, and parallel-slice based method for the more complicated RV. In both methods, the long axis of a chamber needs to be determined from the annotated mesh. Generally,

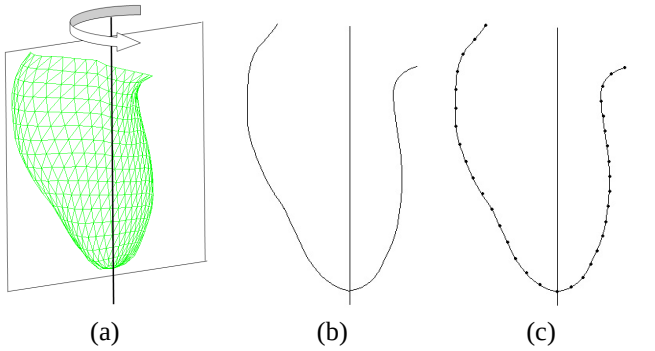


Fig. 7. The rotation-axis based resampling method demonstrated for the LV endocardium mesh. (a) The LV endocardium mesh with its long axis. A cutting plane passing the long axis is also illustrated. (b) The intersection of the mesh with the cutting plane. (c) Resampled points indicated as dots.

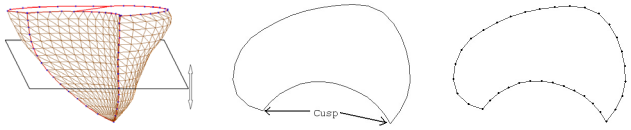


Fig. 8. The parallel-slice based resampling method for the RV main body. (a) The RV main body mesh. A cutting plane perpendicular to the RV long axis is also illustrated. (b) The crescent-shaped intersection of the mesh with the cutting plane. Two cusps separate the intersection into two parts and each can be uniformly resampled independently. (c) Resampled points indicated as dots.

we define the long axis as the line connecting the center of a valve and the mesh point farthest from the valve. For example, the LV long axis is the line connecting the mitral valve center and the LV apex. The RV long axis is determined as a line passing the RV apex and perpendicular to the divergence plane. Compared to other approaches [18], [19], [29] for establishing point correspondence, our solution is simple and each shape is processed independently. No time-consuming and error-prone 3D registration is necessary.

The rotation-axis based method is appropriate for a roughly rotation symmetric shape, which contains a rotation axis. Cutting the mesh with a plane passing the axis, we get a 2D intersection. As shown in Fig. 7b, the rotation axis separates the intersection into two parts, which can be uniformly resampled independently. We then rotate the plane around the axis to get another intersection and resample the intersection in the same way. Repeating the above process, we achieve a set of points with built-in correspondence. We use this approach to resample the LV, LA, and RA, and it is also used to resample the LVOT, and RV inflow and outflow tracts, which can be approximated as tubes.

The rotation-axis based method is not applicable to the RV main body since it is not rotation symmetric. Instead, a parallel-slice based method is developed for the RV, where we use a plane perpendicular to the RV long axis to cut the 3D mesh, as shown in Fig. 8a. The shape of an RV short-axis intersection is a crescent containing two cusp points (which have a high curvature and can be reliably determined from the intersection contour). They split the contour into two parts and each can be uniformly resampled, as shown in Fig. 8c.

IV. 3D Object Localization

In this section, we present our machine learning based 3D object localization technique using two novel techniques, marginal space learning (MSL) and steerable features. The work is built upon recent progress on machine learning based object detection [13], [14], [28].

A. Obtaining Ground Truth from a Mesh

To train object localization classifiers, we need a set of CT volumes. For each volume, we need a nine dimensional vector of the ground truth about the position, orientation, and scaling of a heart chamber in the volume. The ground truth for each chamber is determined from the annotated mesh using a bounding box. The long axis of a chamber defines axis x. The perpendicular direction from a predefined anchor point to the long axis defines axis y. For different chambers, we have freedom to select the anchor point, as long as it is consistent. For example, for the LV, we use the center of aortic valve as the anchor point. The third axis z is the cross-product of axes x and y. This local coordinate system can be represented as three Euler angles, ψ^t, ϕ^t, and θ^t. (Among several well-known Euler angle conventions, the ZXZ convention is used.) We then calculate a bounding box (which is aligned with the local coordinate system) for the mesh points. The bounding box center gives us the position ground truth X^t, Y^t, and Z^t, and the box size along each side defines the ground truth of scaling S_x^t, S_y^t, and S_z^t.

Previously, the Procrustes analysis has been widely used to calculate the mean shape of a set of training samples [8]. We cannot use it in our case since the Procrustes analysis only allows isotropic scaling, but not anisotropic scaling. For each heart chamber, using the method discussed in the previous paragraph, we can calculate its position ($T = [X, Y, Z]'$), orientation (represented as a rotation matrix R), and anisotropic scaling (S_x, S_y, S_z). Since we know the orientation of each shape, we just need to transform each point from the world coordinate system, M_{world}, to the object-oriented coordinate system, m_{object}, and calculate the average over the whole training set. The transformation between M_{world} and m_{object} is

$$M_{world} = R \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} m_{object} + T. \quad (1)$$

Reversing the transformation, we can calculate the position in the object-oriented coordinate system as

$$m_{object} = \begin{bmatrix} \frac{1}{S_x} & 0 & 0 \\ 0 & \frac{1}{S_y} & 0 \\ 0 & 0 & \frac{1}{S_z} \end{bmatrix} R^{-1} \left(M_{world} - T \right). \quad (2)$$

The mean shape is the average over the whole training set

$$\bar{m} = \frac{1}{J} \sum_{j=1}^{J} m_{object}^j, \quad (3)$$

where J is the number of training samples.

B. Marginal Space Learning

Fig. 9 shows the basic idea of machine learning based object detection. First, we train a classifier, which can assign a score (in range [0, 1]) for each input hypothesis. We then quantize

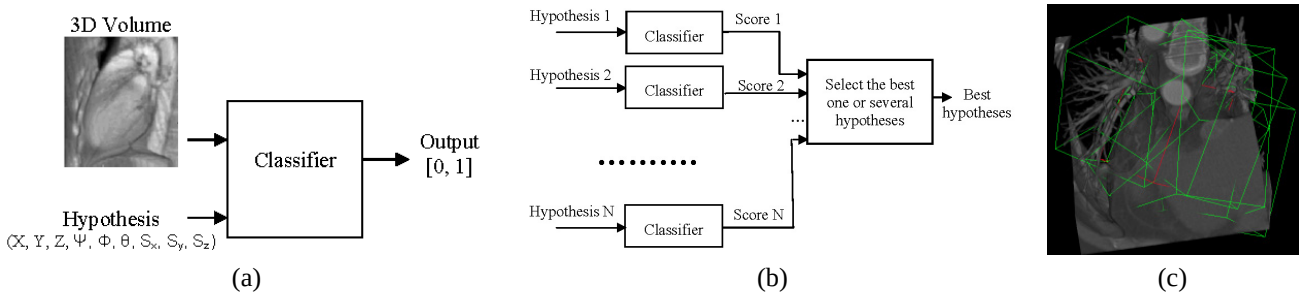


Fig. 9. The basic idea of a machine learning based 3D object detection method. (a) A trained classifier that assigns a score to a hypothesis. (b) The parameter space is quantized into a large number of discrete hypotheses and the classifier is used to select the best hypotheses in exhaustive search. (c) A few hypotheses of the left ventricle (represented as boxes) embedded in a CT volume. The red box shows the ground truth and the green boxes show only a few hypotheses.

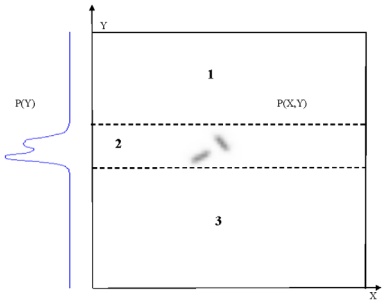


Fig. 10. Marginal space learning. A classifier trained on a marginal distribution $p(y)$ can quickly eliminate a large portion (regions 1 and 3) of the search space. Another classifier is then trained on a restricted space (region 2) for the joint distribution $p(x, y)$.

the full parameter space into a large number of hypotheses. Each hypothesis is tested with the classifier to get a score. Based on the classification scores, we select the best one or several hypotheses. Unlike the gradient based search in deformable models or active appearance models (AAM) [30], the classifier in this framework acts as a black box without an explicit closed-form objective function.

One drawback of the learning based approach is that the number of hypotheses increases exponentially with respect to the dimension of the parameter space. We observed that, in many real applications, the posterior distribution is clustered in a small region in the high dimensional parameter space. Therefore, the uniform and exhaustive search is not necessary and wastes the computational power. Fig. 10 illustrates a simple example for 2D space search. A classifier trained on $p(y)$ can quickly eliminate a large portion of the search space. We can then train a classifier in a much smaller region (region 2 in Fig. 10) for joint distribution $p(x, y)$. Based on this observation, we propose a novel efficient parameter search method, marginal space learning (MSL), to search such clustered spaces. In MSL, the dimensionality of the search space is gradually increased. As shown in the top dashed box in Fig. 2, we split 3D object localization into three steps: position estimation, position-orientation estimation, and full similarity transformation estimation. After each step we keep a limited number of candidates to reduce the search space. To increase the speed further, we use a pyramid-based coarse-to-fine strategy such that object localization is performed on a low-resolution (3 mm) volume.

To train a classifier, we need to split a set of hypotheses into two groups, positive and negative, based on their distance to the ground truth. The error in object position and scale estimation is not comparable with that of orientation estimation. Therefore, a normalized distance measure is defined by normalizing the error in each dimension to the corresponding search step size,

$$E = \max_{i=1,\ldots,D} |P_i^e - P_i^t| / \text{SearchStep}_i, \tag{4}$$

where P_i^e is the estimated value for parameter i, P_i^t is corresponding the ground truth, and D is the dimension of the parameter space. For similarity transformation estimation, the parameter space is nine dimensional, $D = 9$. A sample is regarded as a positive one if $E \leq 1.0$ and all the others are negative samples.

C. Training of Position Estimator

In this step, we want to estimate the position of the object and learning is constrained in a marginal space with three dimensions. Given a hypothesis (X, Y, Z), the classification problem is formulated as whether there is an object centered at (X, Y, Z). Haar wavelet features are fast to compute and have been shown to be effective for many applications [13], [27]. Therefore, we use 3D Haar wavelet features for learning in this step. Readers are referred to [13], [27] for more details, and [28] for a description of 3D Haar wavelet features.

The search step for position estimation is one voxel. According to Eq. (4), a positive sample (X, Y, Z) should satisfy

$$\max\{|X - X^t|, |Y - Y^t|, |Z - Z^t|\} \leq 1 \text{ voxel}, \tag{5}$$

where (X^t, Y^t, Z^t) is the ground truth of the object center. Given a set of positive and negative training samples, we extract 3D Haar wavelet features and train a classifier using the probabilistic boosting-tree (PBT) [31]. After that, we test each voxel in a volume one by one as a hypothesis of the object position using the trained classifier. As shown in Fig. 9a, the classifier assigns each hypothesis a score, and we preserve a small number of candidates (100 in our experiments) with the highest detection score for each volume.

D. Steerable Features

Before discussing our technique for the position-orientation and full similarity transformation estimation, we present another major contribution of this paper, steerable features.

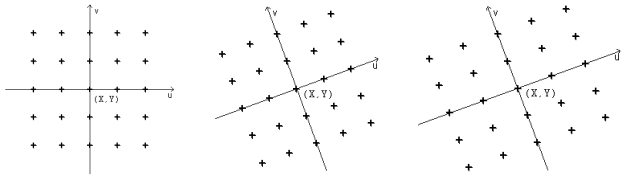


Fig. 11. Using a regular sampling pattern to incorporate a hypothesis (X, Y, ψ, S_x, S_y) about a 2D object pose. The sampling points are indicated as '+'. (a) Move the pattern center to (X, Y). (b) Align the pattern to the orientation ψ. (c) The final aligned sampling pattern after scaling along each axis, proportional to (S_x, S_y).

Global features, such as 3D Haar wavelet features, are effective to capture the global information (e.g., orientation and scale) of an object. To capture the orientation information of a hypothesis, we should rotate either the volume or the feature templates. However, it is time consuming to rotate a 3D volume and there is no efficient way to rotate the Haar wavelet feature templates. Local features are fast to evaluate but lose the global information of the whole object.

In this paper, we propose a new framework, steerable features, which can capture the orientation and scale of the object and at the same time be very efficient. In steerable features, we sample a few points from the volume under a sampling pattern. We then extract a few local features for each sampling point (e.g., voxel intensity and gradient) from the original volume. The novelty of our steerable features is that we embed the orientation and scale information into the distribution of sampling points, while each individual feature is locally defined. Instead of aligning the volume to the hypothesized orientation, we steer the sampling pattern. This is where the name "steerable features" comes from.

Fig. 11 shows how to embed a hypothesis in steerable features using a regular sampling pattern (illustrated for a 2D case for clearance in visualization). Suppose we want to test if hypothesis $(X, Y, Z, \psi, \phi, \theta, S_x, S_y, S_z)$ is a good estimation of the similarity transformation of the object. A local coordinate system is defined to be centered at position (X, Y, Z) (Fig. 11a) and the axes are aligned with the hypothesized orientation (ψ, ϕ, θ) (Fig. 11b). A few points (represented as '+' in Fig. 11) are uniformly sampled along each coordinate axis inside a box. The sampling distance along an axis is proportional to the scale of the shape in that direction (S_x, S_y, or S_z) to incorporate the scale information (Fig. 11c). The steerable features constitute a general framework, in which different sampling patterns [15], [32] can be defined.

At each sampling point, we extract a few local features based on the intensity and gradient from the original volume. A major reason to select these features is that they can be extracted fast. Suppose a sampling point (x, y, z) has intensity I and gradient $g = (g_x, g_y, g_z)$. The three axes of object-oriented local coordinate system are n_x, n_y, and n_z. The angle between the gradient g and the z axis is $\alpha = arccos(n_z.g)$, where $n_z.g$ means the inner product between two vectors n_z and g. The following 24 features are extracted: I, $\sqrt{I}$, $\sqrt[3]{I}$, I^2, I^3, $\log I$, $\|g\|$, $\sqrt{\|g\|}$, $\sqrt[3]{\|g\|}$, $\|g\|^2$, $\|g\|^3$, $\log \|g\|$, α, $\sqrt{\alpha}$, $\sqrt[3]{\alpha}$, α^2, α^3, $\log \alpha$, g_x, g_y, g_z, $n_x.g$, $n_y.g$, $n_z.g$. In total, we have 24 local features for each sampling point. The first six features are based on intensity and the remaining 18 features are transformations of gradients. Feature transformation, a technique often used in pattern classification, is a process through which a new set of features is created [33]. We use it to enhance the feature set by adding a few transformations of an individual feature. Suppose there are P sampling points, we get a feature pool containing $24 \times P$ features. (In our case, the $5 \times 5 \times 5$ regular sampling pattern is used for object localization, resulting in $P = 125$ sampling points.) These features are used to train histogram-based weak classifiers [34] and we use the probabilistic boosting-tree (PBT) [31] to combine them to get a strong classifier. Following are some statistics about the selected features by the boosting algorithm. Combining features in all object localization classifiers, overall, there are 3696 features selected. We found each feature type was selected as least once. The intensity features, I, $\sqrt{I}$, $\sqrt[3]{I}$, I^2, I^3, $\log I$, counted about 26% of the selected features, while, the following four gradient-based features, g_x, g_y, g_z, and $\|g\|$, counted about 34%.

E. Training of Position-Orientation Estimator

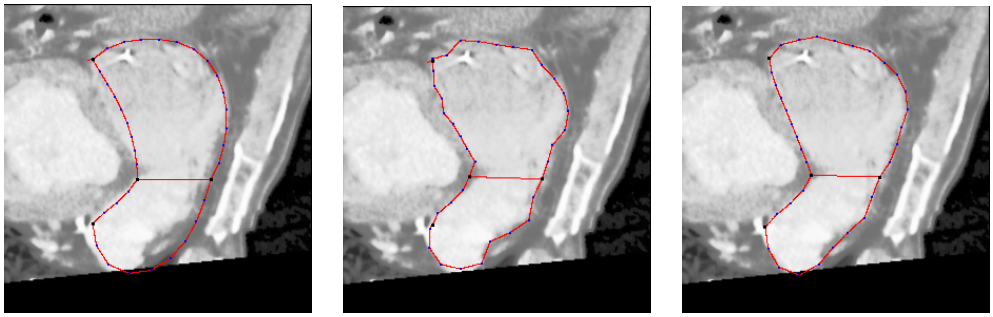
In this step, we want to jointly estimate the position and orientation. The classification problem is formulated as whether there is an object centered at (X, Y, Z) with orientation (ψ, ϕ, θ). After object position estimation, we preserve the top 100 candidates, (X_i, Y_i, Z_i), $i = 1, \ldots, 100$. Since we want to estimate both the position and orientation, we need to augment the dimension of candidates. For each position candidate, we quantize the orientation space uniformly to generate hypotheses. The orientation is represented as three Euler angles in the ZXZ convention, ψ, ϕ, and θ. The distribution range of an Euler angle be estimated from the training data. Each Euler angle is quantized within the range using a step size of 0.2 radians (11 degrees). For each candidate (X_i, Y_i, Z_i), we augment it with N (about 1000) hypotheses about orientation, $(X_i, Y_i, Z_i, \psi_j, \phi_j, \theta_j)$, $j = 1, \ldots, N$. Some are close to the ground truth (positive) and others are far away (negative). The learning goal is to distinguish the positive and negative samples using trained classifiers. Using the normalized distance measure of Eq. (4), a hypothesis $(X, Y, Z, \psi, \phi, \theta)$ is regarded as a positive sample if it satisfies both Eq. (5) and

$$\max\{|\psi - \psi^t|, |\phi - \phi^t|, |\theta - \theta^t|\} \leq 0.2, \tag{6}$$

where $(\psi^t, \phi^t, \theta^t)$ represent the orientation ground truth. All the other hypotheses are regarded as negative samples. To represent the orientation information, we have to rotate either the volume or feature templates. We use the steerable features, which are efficient under rotation. Similarly, the PBT is used for training and the trained classifier is used to prune the hypotheses to preserve only a few candidates (50 in our experiments).

F. Training of Similarity Transformation Estimator

The similarity transformation (adding the scales) estimation step is analogous to position-orientation estimation except learning is performed in the full nine dimensional similarity



(a) (b) (c)

Fig. 12. Nonrigid deformation estimation for control points (the tricuspid lateral and the right ventricular outflow tract lateral) on the RV divergence plane. (a) Detected mean shape. (b) After boundary adjustment. (c) Final result by projecting the adjusted shape onto a shape subspace (25 dimensions).

transformation space. The dimension of each candidate is augmented by searching the scale subspace uniformly and exhaustively. The search step is set to 2 voxels (6 mm).

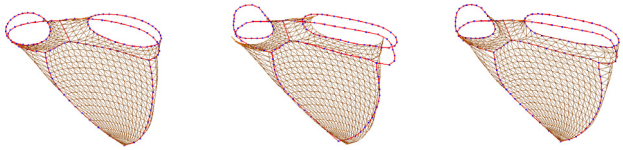
G. Object Localization on Unseen Volume

This section provides a summary about the testing procedure on an unseen volume. The input volume is first converted to the 3 mm isotropic resolution. All voxels are tested using the trained position estimator and the top 100 candidates, (X_i, Y_i, Z_i), $i = 1, \ldots, 100$, are kept. Each candidate is augmented with N (about 1000) hypotheses about orientation, $(X_i, Y_i, Z_i, \psi_j, \phi_j, \theta_j)$, $j = 1, \ldots, N$. Next, the trained position-orientation classifier is used to prune these $100 \times N$ hypotheses and the top 50 candidates are retained, $(\hat{X}_i, \hat{Y}_i, \hat{Z}_i, \hat{\psi}_i, \hat{\phi}_i, \hat{\theta}_i)$, $i = 1, \ldots, 50$. Similarly, we augment each candidate with M (also about 1000) hypotheses about scaling and use the trained classifier to rank these $50 \times M$ hypotheses. The average of the top K ($K = 100$) candidates is taken as the final aggregated estimate.

In terms of computational complexity, for position estimation, all voxels are tested (about 260,000 for a small $64 \times 64 \times 64$ volume at the 3 mm resolution) for possible object position. There are about 1000 hypotheses for orientation and scale each. If the parameter space is searched uniformly and exhaustively, there are about 2.6×10^{11} hypotheses to be tested! However, using MSL, we only test about $260,000 + 100 \times 1000 + 50 \times 1000 = 4.1 \times 10^5$ hypotheses and reduce the testing by almost six orders of magnitude.

V. Nonrigid Deformation Estimation

After automatic object localization, we align the mean shape with the estimated pose using Eq. (1). We then deform the mean shape to fit the object boundary. Active shape models (ASM) are widely used to deform an initial estimate of a nonrigid shape under the guidance of the image evidence and the shape prior. The non-learning based generic boundary detector in the original ASM [8], [9] does not work in our application due to the complex background and weak edges. Learning based methods have been demonstrated to have better performance on 2D images [35]–[37] since they can exploit more image evidences to achieve robust boundary detection. In the previous work [35], [37], a detector was trained to detect boundary with a specific orientation (e.g., horizontal boundary). In order to detect boundary with different orientations, we need to perform detection on a set of rotated images.



(a) (b) (c)

Fig. 13. RV mesh warping using control points. Blue dots indicate control points (which are connected by red contours for visualization) and brown shows the RV mesh. (a) Mean shape using the estimated RV pose. (b) After control point refinement, the mesh is not consistent. (c) Warped mesh, where the control points and the mesh are consistent again.

In this paper, we extend learning based methods to 3D and completely avoid time-consuming volume rotation using our efficient steerable features. Here, boundary detection is formulated as a classification problem: whether there is a boundary passing point (X, Y, Z) with orientation (O_x, O_y, O_z). This problem is similar to the classification problem we solved for position-orientation estimation: whether there is an object centered at (X, Y, Z) with orientation (ψ, ϕ, θ). Therefore, the same approach is used to train a boundary detector using the PBT [31] and steerable features.

Control points in our mesh representation have different image characteristics and should be treated specially. As shown in [17], without special processing, the connection of different chambers around the mitral or tricuspid valve cannot be delineated well. Our nonrigid deformation has three steps (as shown in Fig. 2). We first estimate the deformation of control points. The thin-plate-spline (TPS) model [38] is then used to warp the initial mesh toward the refined control points for better alignment. Last, the normal mesh points are deformed to fit the image boundary. Note that, at this step, the control points are kept unchanged.

In what follows, we illustrate the refinement of the control points on the RV divergence plane. All other control points are refined in a similar way. First, MSL is used to detect the pose of the divergence plane. After that, we get an aligned mean shape for the control points. Fig. 12a shows the aligned mean shape under the estimated pose. The boundary detectors

(a) (b) (c)

Fig. 14. Nonrigid deformation estimation for the LV with green for the endocardium and magenta for the epicardium. (a) Mean shape. (b) After boundary adjustment. (c) Final delineation by projecting the adjusted shape onto a shape subspace (50 dimensions).

are then used to move each control point along the normal direction to the optimal position, where the score from the boundary detector is the highest. After adjustment, the control points fit the boundary well, but the contour is not smooth (Fig. 12b). Finally, we project the deformed shape onto a shape subspace [8]. In all our experiments, to determine the dimension of the subspace, we demand it to capture 98% variations. As shown in Fig. 12c, the statistical shape model is very effective to enforce the prior shape constraint.

The refined control points can be used to warp a mesh to make it fit the image better (as shown in Section VI-C). Fig. 13a shows the mean shape aligned with the detected RV pose. Fig. 13b shows the refinement of the control points, which fit the data more accurately, but inconsistent with the mesh. Using the original and refined control points as the anchor points, we can estimate the nonrigid deformation of the TPS model and use it to warp the mesh points. As shown in Fig. 13c, the mesh points and the control points are consistent again after warping. Since the control points are clustered around the aortic and mitral valves for the LV, we add the point farthest from the mitral valve (which is the LV apex) as an anchor point in the TPS model to warp the LV. A similar treatment is applied to warp both atria.

Due to the large variation introduced by cardiac motion, each chamber is processed separately since the variation of a chamber is smaller than that of a whole heart. After chamber pose estimation, the initial mesh of atria and ventricles have conflict around the mitral and tricuspid valves. Using the control points around the valves as anchor points in TPS warping, we can resolve such mesh conflict. After the whole segmentation procedure, further mesh conflict can be resolved through a post-processing step.

After TPS warping, the mesh points are closer to the chamber boundary. To further reduce the error, we train again a boundary detector for each mesh surface. The boundary detectors are then used to adjust each point (the control points are kept unchanged in this step). Fig. 14a shows the aligned LV in a cardiac CT volume. Fig. 14b shows the adjusted shape. Shape constraint is enforced by projecting the adjusted shape onto a shape subspace to get the final result, as shown in Fig. 14c. The above steps can be iterated a few time. Based on the trade-off between speed and accuracy, we use one iteration for LV/LA, and two iterations for RV/RA since the right side of the heart has typically much lower contrast.

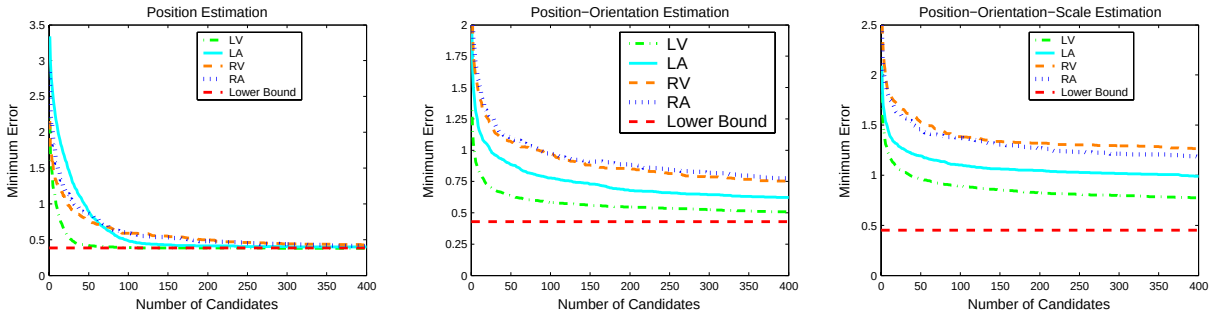
VI. Experiments

A. Data Set

Under the guidance of cardiologists, we manually annotated all four chambers in 323 cardiac CT volumes (with various cardiac phases) from 137 patients with various cardiovascular diseases. The specific disease information for a patient has not been captured. Since the LV is clinically more important than other chambers, to improve the system performance on LV detection and segmentation, we annotated extra 134 volumes. In total, we have 457 volumes from 186 patients for the LV. The annotation is done by several of the authors. However, for each volume, there is only one annotation. Therefore, we cannot study the intra- and inter-observer variabilities, this being a limitation of the dataset. The number of patients used in our experiments is significantly larger than those reported in the literature, for example, 13 in [17], 18 in [39], 27 in [10], and 30 in [9]. The data was collected from 27 institutes over the world (mostly from Germany, the USA, and China) using Siemens Somatom Sensation or Definition scanners. The imaging protocols are heterogeneous with different capture ranges and resolutions. A volume may contain 80 to 350 slices, while the size of each slice is the same with 512×512 pixels. The resolution inside a slice is isotropic and varies from 0.28 mm to 0.74 mm for different volumes. The slice thickness (distance between neighboring slices) is larger than the in-slice resolution and varies from 0.4 mm to 2.0 mm for different volumes. We use four-fold cross validation to evaluate our algorithm. Data from the same patient may have similar shapes and image characteristics since they were often captured on the same CT scanner with the same scanning parameters. If such data appear in both the training and test sets during cross-validation, the result is biased toward a lower segmentation error. To remove such bias, we enforce the constraint that the volumes from the same patient can only appear in either the training or test set, but not in both.

B. Experiments on Heart Chamber Localization

In this section, we evaluate the proposed approach for heart chamber localization. The error measure defined in Eq. (4) is used since we can easily distinguish optimal and non-optimal



(a) (b) (c)

Fig. 15. The error, defined in Eq. (4), of the best candidate with respect to the number of candidates preserved after each step. (a) Position estimation. (b) Position-orientation estimation. (c) Full similarity transformation estimation. The red dotted lines show the lower bound of the detection error.

estimates, compared to other error measures (e.g., the weighted Euclidean distance). The optimal estimate is up-bounded by 0.5 search steps under any search grid. However, a non-optimal estimate has an error larger than 0.5.

The efficiency of MSL comes from the fact that we prune the search space after each step. One concern is that since the space is not fully explored, it may miss the optimal solution at an early stage. In the following, we demonstrate that accuracy only deteriorates slightly in MSL. Fig. 15 shows the error of the best candidate after each step with respect to the number of candidates preserved. The curves are calculated on all volumes based on cross validation. The red dotted lines show the error of the optimal solution under the search grid. As shown in Fig. 15a for position estimation, if we keep only one candidate, the average error may be as large as 3.5 voxels. However, by retaining more candidates, the minimum errors decrease quickly. We have a high probability to keep the optimal solution when 100 candidates are preserved. We observed the same trend in different marginal spaces, such as the position-orientation space as shown in Fig. 15b. Based on the trade-off between accuracy and speed, we preserve 50 candidates after position-orientation estimation. After full similarity transformation estimation, the best candidates we get have an error ranging from 1.0 to 1.4 search steps as shown in Fig. 15c. Using the average of the top K ($K = 100$) candidates as the final single estimate, we achieve an error of about 1.5 to 2.0 search steps for different chambers. Our approach is robust and we did not observe any major failure. For comparison, the heart localization modules in both [17] and [9] failed on about 10% volumes.

C. Experiments on Boundary Delineation

In this section, we evaluate our approach for boundary delineation. As a widely used criterion [9], [10], [17], the symmetric point-to-mesh distance, E_{p2m}, is exploited to measure the accuracy in surface boundary delineation. For each point on a mesh, we search for the closest point (not necessarily mesh triangle vertices) on the other mesh to calculate the minimum Euclidean distance. We calculate the point-to-mesh distance from the detected mesh to the ground-truth and vice versa to make the measurement symmetric.

In our experiments, we estimate the pose of each chamber separately. Therefore, we use $4 \times 9 = 36$ pose parameters to align the mean shapes. As shown in the second column of Table I, the mean E_{p2m} error after heart localization is 3.17 mm for the LV endocardium, 2.51 mm for the LV epicardium, 2.78 mm for the LA, 2.93 mm for the RV, and 3.09 mm for the RA. Alternatively, we can treat the whole heart as one object in heart localization, then we use only nine pose parameters. In this way, the mean E_{p2m} error achieved is 3.52 mm for the LV endocardium, 3.07 mm for the LV epicardium, 3.95 mm for LA, 3.94 mm for the RV, and 4.64 mm for the RA. Obviously, treating each chamber separately, we can obtain a better initialization.

In our nonrigid deformation estimation, control points and normal mesh points are treated differently. We first estimate the deformation of control points and use TPS warping to make the mesh consistent after warping. As shown in the third column in Table I, after control point based alignment, we slightly reduce the error for the LV, LA, and RA by 5% and significantly reduce the error by 17% for the RV since the control points are more uniformly distributed in the RV mesh. After deformation estimation of all mesh points, the final segmentation error ranges from 1.13 mm to 1.57 mm for different chambers. The LV and LA have smaller errors than the RV and RA due to the use of contrast agent in the left heart (as shown in Fig. 16).

We compare our approach to the baseline ASM using non-learning based boundary detection scheme [8]. The comparison is limited to the last step on normal mesh point deformation. Input for both algorithms are the same initialized mesh. The iteration number in the baseline ASM is tuned to give the best performance. As shown in Table I, the baseline ASM actually increase the error for weak boundaries (e.g., the LV epicardium and RV). It performs well for strong boundaries, such as the LV endocardium and the LA, but it is still significantly worse than the proposed method.

Fig. 16 shows several examples for heart chamber segmentation using the proposed approach. It performs well on volumes with low contrast (as shown in the second row of Fig. 16) and it is robust even under severe streak artifacts (as shown in the third example). Since our system is trained on volumes from all phases from a cardiac cycle, we can process volumes from the end-systolic phase (which has a significantly small blood pool for the LV) without any difficulty, as shown in the last example in Fig. 16. Fig. 17

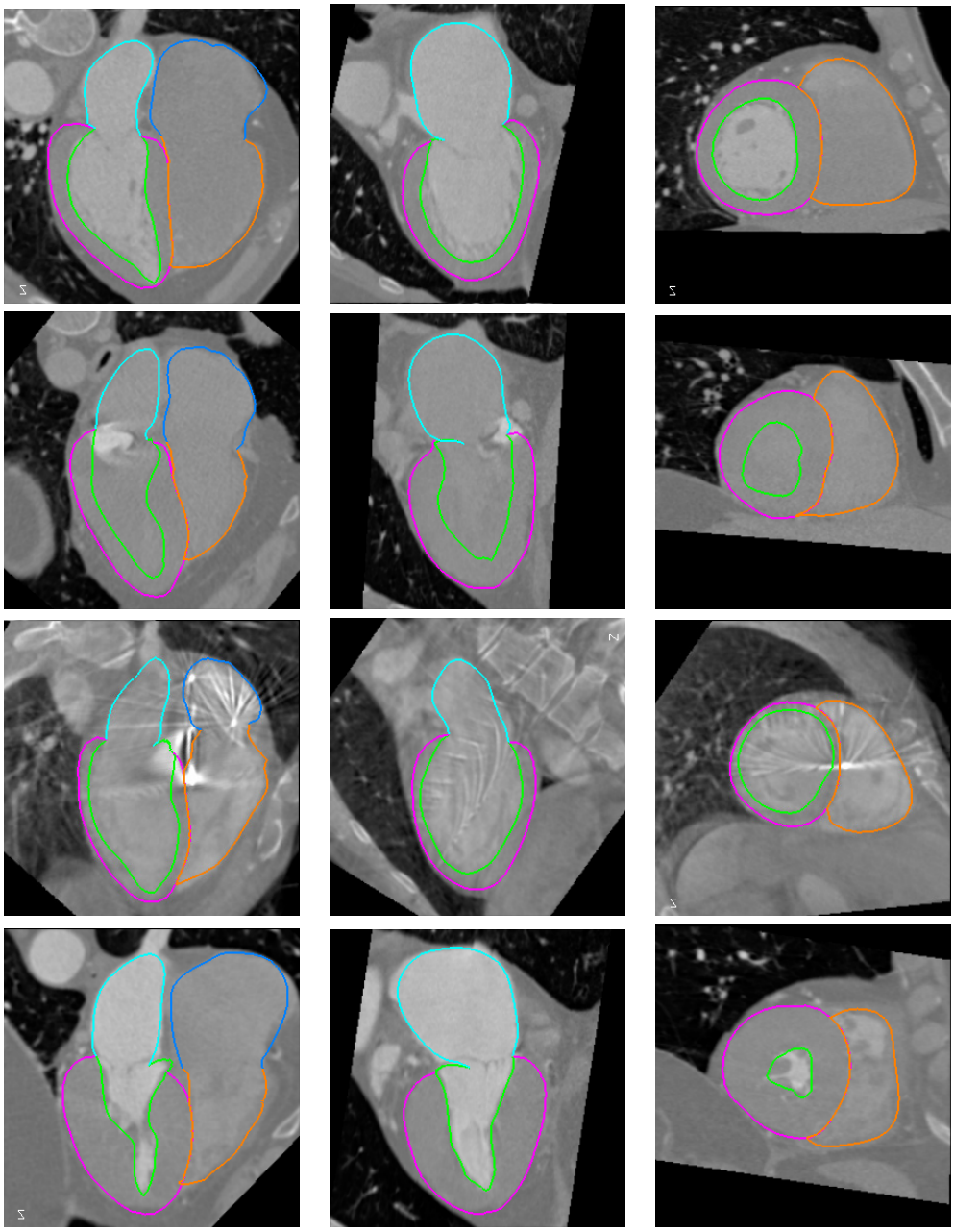


Fig. 16. Examples of heart chamber segmentation in 3D CT volumes with green for the LV endocardium, magenta for the LV epicardium, cyan for the LA, brown for the RV, and blue for the RA. Each row represents three orthogonal views of a volume.

shows the segmentation result on a full torso CT volume, where no contrast agent nor electrocardiogram-based gating is applied. This volume is challenging for thresholding based region growing techniques [9], [39]. However, our machine learning based approach can deal with this case quite well.

After code optimization and using multi-threading techniques, we achieved an average speed of 4.0 seconds for the automatic segmentation of all four chambers on a computer with a dual-core 3.2 GHz processor and 3 GB memory. The computation time is roughly equally split on the MSL based similarity transformation estimation and the nonrigid deformation estimation.

In Table II, we presented a brief summary of the previous work on heart segmentation in 3D CT volumes. It is obvious, our approach is faster compared to other reported results, e.g., 5 seconds for left ventricle segmentation in [39], 15 seconds for nonrigid deformation in [40], and more than 1 minute in [11], [12]. Compared with results on other imaging modalities [5], [22], to the best of our knowledge, our approach is also the fastest. Most of the previous approaches are semiautomatic, except [17]. In general, we cannot compare error measures reported in different papers directly due to the difference in heart models and datasets. In the literature, we noticed two papers [10], [17] reporting better results than ours, both on much smaller datasets. Both used the same heart model, one automatic [17] and one semi-automatic [10]. Different from our four-chamber model, their heart model also included major vessel trunks. Both papers only gave overall errors for

TABLE I

MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF THE POINT-TO-MESH ERROR (IN MILLIMETERS) FOR THE SEGMENTATION OF HEART CHAMBERS BASED ON CROSS VALIDATION.

	After Rigid Localization	After Control Point Deformation and Warping	Baseline ASM [8]	Proposed Approach
Left Ventricle Endocardium	3.17 (1.10)	3.00 (1.11)	2.24 (1.21)	**1.13 (0.55)**
Left Ventricle Epicardium	2.51 (0.78)	2.35 (0.73)	2.45 (1.02)	**1.21 (0.41)**
Left Atrium	2.78 (0.98)	2.67 (1.01)	1.89 (1.43)	**1.32 (0.42)**
Right Ventricle	2.93 (0.75)	2.40 (0.82)	2.69 (1.10)	**1.55 (0.38)**
Right Atrium	3.09 (0.86)	2.90 (0.92)	2.81 (1.15)	**1.57 (0.48)**

TABLE II

COMPARISON WITH PREVIOUS WORK ON HEART SEGMENTATION IN 3D CT VOLUMES.

	Patients/ Subjects	Volumes	Chambers	Automatic	Speed	Point-to-Mesh Error (mm)
Neubauer and Wegenkiltl [11]	N/A	N/A	Left ventricle	No	>1 min	N/A
McInerney and Terzopoulos [12]	1	16	Left ventricle	No	100 min[a]	N/A
Fritz et al. [9]	30	30	Left ventricle	No	N/A	1.5
Jolly [39]	18	36	Left ventricle	No[b]	∼5 s	N/A
Ecabert et al. [17]	13	28	Four chambers and vessel trunks	Yes[c]	N/A	0.85[d]
Lorenz and von Berg [10]	27	27	Four chambers and vessel trunks	No	N/A	0.81-1.19
von Berg and Lorenz [40]	6	60	Four chambers and vessel trunks	No	15 s	N/A
Our approach	137+	323+	Four chambers	Yes	4.0 s	1.13-1.57

[a] This was the time used to process the whole sequence of 16 volumes.
[b] The long axis of the left ventricle needed to be manually aligned. All other steps were automatic.
[c] The success rate of automatic heart localization was about 90%.
[d] Gross failures in heart localization were excluded from evaluation.

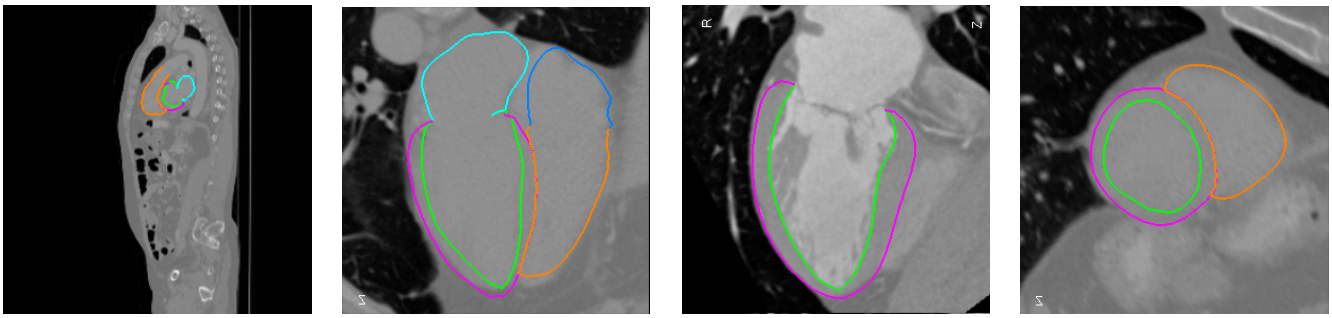


Fig. 17. Heart chamber segmentation result for a low-contrast full-torso CT volume. The first column shows a full torso view and the right three columns show close-up views.

the whole heart model (including the major vessel trunks), without any break-down error measure for each chamber. Some care needs to be taken to compare our approach with these two papers. 1) Ecabert et al. [17] admitted that it was hard to distinguish the boundary of different chambers. For example, it was likely to include a part of the LA in the segmented LV mesh and vice verse. Such errors were only partially penalized in both [17] and [10] since they did not provide break-down error measure for each chamber. However, in our evaluation, we fully penalize such errors. 2) About 8% mesh points around the connection of vessel trunks to heart chambers were excluded for evaluation in [17]. In their model, all chambers and vessel trunks were artificially closed. Since there are no image features around these artificial caps, these regions cannot be delineated accurately even by an expert. Based on this consideration, they were removed from evaluation. In our model, all valves are represented as closed contours along their borders in our heart model. We only need to delineate the border of the valves and this can be done more accurately. Therefore, no mesh part is excluded from evaluation. 3) In [17], the automatic heart localization module failed on about 10% volumes and such gross failures were also excluded for evaluation. 4) In [10], only volumes from the end-diastolic phase were used for experiments. However, our dataset contains 323+ volumes from all cardiac phases. The size and shape of a chamber change significantly from the end-diastolic phase to the end-systolic phase. Therefore, there is much more variance in our dataset.

D. Heart Chamber Tracking

The size and shape of a heart chamber (especially, the LV) change significantly from an expansion phase to a contraction phase. Since our system is trained on volumes from all phases in a cardiac cycle, we can reliably detect and segment the heart from any cardiac phase. By performing heart segmentation frame by frame, the heart motion is tracked in the robust tracking-by-detection framework. To make the motion more consistent, mild motion smoothing is applied after the segmentation of each frame. Fig. 18 shows the tracking results on one

TABLE III

THE EJECTION FRACTION (EF) ESTIMATION ACCURACY FOR ALL SIX DYNAMIC SEQUENCES IN OUR DATASET.

	Patient #1	Patient #2	Patient #3	Patient #4	Patient #5	Patient #6	Mean Error	Standard Deviation
Ground Truth	68.7%	49.7%	45.8%	62.9%	47.4%	38.9%	2.3%	1.6%
Estimation	66.8%	51.8%	42.8%	64.4%	42.3%	38.5%		

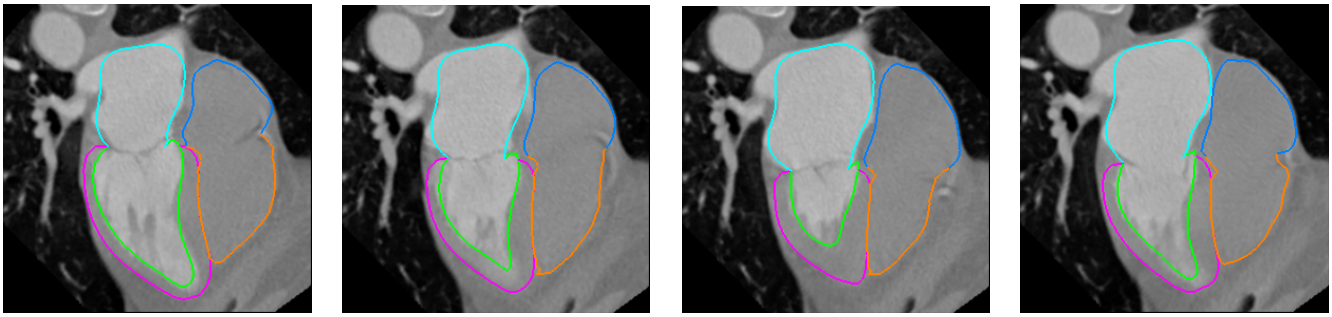


Fig. 18. Tracking results for the heart chambers on a dynamic 3D sequence with 10 frames. Four frames (1, 2, 3, and 6) are shown here.

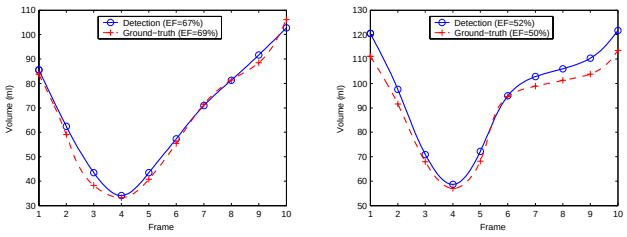


Fig. 19. The left ventricle volume-time curves for two dynamic 3D sequences.

sequence. To further improve the system performance, we can exploit a motion model learned in an annotated dataset [41], but it is out of the scope of this paper.

The motion pattern of a chamber during a cardiac cycle provides many important clinical measurements of its functionality, e.g., the ventricular ejection fraction, myocardium wall thickness, and dissynchrony within a chamber or between different chambers [2]. Given the tracking result, we can calculate the ejection fraction (EF) as follows,

$$\text{EF} = \frac{\text{Volume}_{ED} - \text{Volume}_{ES}}{\text{Volume}_{ED}}, \tag{7}$$

where Volume_{ED} and Volume_{ES} are the volume measures of the end-diastolic (ED) and end-systolic (ES) phases, respectively. In our dataset, there are six patients each with 10 frames from the whole cardiac cycle. Due to the space limit, Fig. 19 shows the LV volume-time curves for two dynamic sequences. Table III shows the EF estimation accuracy for all six sequences. The estimated EFs are close to the ground truth with a mean error of 2.3%.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel four-chamber surface mesh model for a heart. In heart modeling, the following two factors are considered and traded-off: 1) accuracy in anatomy and 2) easiness for both annotation and automatic detection. To more accurately represent the anatomy, important landmarks such as valves and ventricular septum cusps are explicitly represented in our model. These landmarks can be detected reliably to guide the automatic model fitting process.

Using this model, we develop an efficient and robust approach for automatic heart chamber segmentation in 3D CT volumes. The efficiency of our approach comes from the two new techniques, marginal space learning and steerable features. We achieved an average speed of 4.0 seconds per volume to segment all four chambers. Robustness is achieved by using recent advances in learning discriminative models and exploiting a large annotated dataset. All major steps in our approach are learning-based, therefore minimizing the number of underlying model assumptions. According to our knowledge, this is the first study reporting stable results on a large cardiac CT dataset. Our segmentation approach is general and we have extensively tested it on many challenging 3D detection and segmentation tasks in medical imaging (e.g., ileocecal valves, polyps [42], and livers in abdominal CT [32], brain tissues [43] and heart chambers in ultrasound images [41], [44], and heart chambers in MRI).

VIII. ACKNOWLEDGES

The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] P. Schoenhagen, S.S. Halliburton, A.E. Stillman, and R.D. White, "CT of the heart: Principles, advances, clinical uses," *Cleveland Clinic Journal of Medicine*, vol. 72, no. 2, pp. 127–138, 2005.

[2] A.F. Frangi, W.J. Niessen, and M.A. Viergever, "Three-dimensional modeling for functional analysis of cardiac images: A review," *IEEE Trans. Medical Imaging*, vol. 20, no. 1, pp. 2–25, 2001.

[3] A.F. Frangi, D. Rueckert, and J.S. Duncan, "Three-dimensional cardiovascular image analysis," *IEEE Trans. Medical Imaging*, vol. 21, no. 9, pp. 1005–1010, 2002.

[4] J. Lötjönen, S. Kivistö, J. Koikkalainen, D. Smutek, and K. Lauerma, "Statistical shape model of atria, ventricles and epicardium from short- and long-axis MR images," *Medical Image Analysis*, vol. 8, no. 3, pp. 371–386, 2004.

[5] W. Hong, B. Georgescu, X.S. Zhou, S. Krishnan, Y. Ma, and D. Comaniciu, "Database-guided simultaneous multi-slice 3D segmentation for volumetric data," in *Proc. European Conf. Computer Vision*, 2006, pp. 397–409.

[6] D. Fritz, D. Rinck, R. Dillmann, and M. Scheuring, "Segmentation of the left and right cardiac ventricle using a combined bi-temporal statistical model," in *Proc. of SPIE Medical Imaging*, 2006, pp. 605–614.

[7] R. Ionasec, B. Georgescu, E. Gassner, S. Vogt, O. Kutter, M. Scheuering, N. Navab, and D. Comaniciu, "Dynamic model-driven quantitative and visual evaluation of the aortic valve from 4D CT," in *Proc. Int'l Conf. Medical Image Computing and Computer Assisted Intervention*, 2008.

[8] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active shape models—their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[9] D. Fritz, D. Rinck, R. Unterhinninghofen, R. Dillmann, and M. Scheuring, "Automatic segmentation of the left ventricle and computation of diagnostic parameters using regiongrowing and a statistical model," in *Proc. of SPIE Medical Imaging*, 2005, pp. 1844–1854.

[10] C. Lorenz and J. von Berg, "A comprehensive shape model of the heart," *Medical Image Analysis*, vol. 10, no. 4, pp. 657–670, 2006.

[11] A. Neubauer and R. Wegenkiltl, "Analysis of four-dimensional cardiac data sets using skeleton-based segmentation," in *Proc. Int'l Conf. in Central Europe on Computer Graphics and Visualization*, 2003.

[12] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis," *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, pp. 69–83, 1995.

[13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2001, pp. 511–518.

[14] B. Georgescu, X.S. Zhou, D. Comaniciu, and A. Gupta, "Database-guided segmentation of anatomical structures with complex appearance," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005, pp. 429–436.

[15] Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu, "Fast automatic heart chamber segmentation from 3D CT data using marginal space learning and steerable features," in *Proc. Int'l Conf. Computer Vision*, 2007.

[16] Y. Zheng, B. Georgescu, A. Barbu, M. Scheuering, and D. Comaniciu, "Four-chamber heart modeling and automatic segmentation for 3D cardiac CT volumes," in *Proc. of SPIE Medical Imaging*, 2008.

[17] O. Ecabert, J. Peters, and J. Weese, "Modeling shape variability for full heart segmentation in cardiac computed-tomography images," in *Proc. of SPIE Medical Imaging*, 2006, pp. 1199–1210.

[18] A.F. Frangi, D. Rueckert, J.A. Schnabel, and W.J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling," *IEEE Trans. Medical Imaging*, vol. 21, no. 9, pp. 1151–1166, 2002.

[19] C. Lorenz and N. Krahnstover, "Generation of point based 3D statistical shape models for anatomical objects," *Computer Vision and Image Understanding*, vol. 77, no. 2, pp. 175–191, 2000.

[20] H.C. van Assen, M.G. Danilouchkine, A.F. Frangi, S. Ordas, J.J.M. Westernberg, J.H.C. Reiber, and B.P.F. Lelieveldt, "SPASM: A 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data," *Medical Image Analysis*, vol. 10, no. 2, pp. 286–303, 2006.

[21] A. Andreopoulos and J.K. Tsotsos, "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI," *Medical Image Analysis*, vol. 12, no. 3, pp. 335–357, 2008.

[22] S.C. Mitchell, J.G. Bosch, B.P.F. Lelieveldt, R.J. van Geest, J.H.C. Reiber, and M. Sonka, "3-D active appearance models: Segmentation of cardiac MR and ultrasound images," *IEEE Trans. Medical Imaging*, vol. 21, no. 9, pp. 1167–1178, 2002.

[23] Z. Bao, L. Zhukov, I. Guskov, J. Wood, and D. Breen, "Dynamic deformable models for 3D MRI heart segmentation," in *Proc. of SPIE Medical Imaging*, 2002, pp. 398–405.

[24] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti, "Left ventricular volume estimation for real-time three-dimensional echocardiography," *IEEE Trans. Medical Imaging*, vol. 21, no. 9, pp. 1202–1208, 2002.

[25] O. Gerard, A.C. Billon, J.-M. Rouet, M. Jacob, M. Fradkin, and C. Allouche, "Efficient model-based quantification of left ventricular function in 3-D echocardiography," *IEEE Trans. Medical Imaging*, vol. 21, no. 9, pp. 1059–1068, 2002.

[26] K. Park, A. Montillo, D. Metaxas, and L. Axel, "Volumetric heart modeling and analysis," *Communications of the ACM*, vol. 48, no. 2, pp. 43–48, 2005.

[27] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 193–199.

[28] Z. Tu, X.S. Zhou, A. Barbu, L. Bogoni, and D. Comaniciu, "Probabilistic 3D polyp detection in CT images: The role of sample alignment," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1544–1551.

[29] R.H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton, and C.J. Taylor, "A minimum description length approach to statistical shape modeling," *IEEE Trans. Medical Imaging*, vol. 21, no. 5, pp. 525–537, 2002.

[30] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[31] Z. Tu, "Probabilistic boosting-tree: Learning discriminative methods for classification, recognition, and clustering," in *Proc. Int'l Conf. Computer Vision*, 2005, pp. 1589–1596.

[32] H. Ling, S.K. Zhou, Y. Zheng, B. Georgescu, M. Suehling, and D. Comaniciu, "Hierarchical, learning-based automatic liver segmentation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[33] A. Kusiak, "Feature transformation methods in data mining," *IEEE Trans. Electronics Packaging Manufacturing*, vol. 24, no. 3, pp. 214–221, 2001.

[34] R.E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[35] P. Dollár, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2006, pp. 1964–1971.

[36] B. van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Haar Romeny, and M.A. Viergever, "Active shape model segmentation with optimal features," *IEEE Trans. Medical Imaging*, vol. 21, no. 8, pp. 924–933, 2002.

[37] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color and texture cues," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 5, pp. 530–549, 2004.

[38] F.L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 6, pp. 567–585, 1989.

[39] M.-P. Jolly, "Automatic segmentation of the left ventricle in cardiac MR and CT images," *Int. J. Computer Vision*, vol. 70, no. 2, pp. 151–163, 2006.

[40] J. von Berg and C. Lorenz, "Multi-surface cardiac modelling, segmentation, and tracking," in *Proc. Functional Imaging and Modeling of the Heart*, 2005, pp. 1–11.

[41] L. Yang, B. Georgescu, Y. Zheng, P. Meer, and D. Comaniciu, "3D ultrasound tracking of the left ventricles using one-step forward prediction and data fusion of collaborative trackers," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[42] L. Lu, A. Barbu, M. Wolf, J. Liang, M. Salganicoff, and D. Comaniciu, "Accurate polyp segmentation for 3D CT colongraphy using multi-staged probabilistic binary learning and compositional model," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[43] G. Carneiro, F. Amat, B. Georgescu, S. Good, and D. Comaniciu, "Semantic-based indexing of fetal anatomies from 3-D ultrasound data using global/semi-local context and sequential sampling," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.

[44] X. Lu, B. Georgescu, Y. Zheng, J. Otsuki, R. Bennett, and D. Comaniciu, "Automatic detection of standard planes from three dimensional echocardiographic data," in *Proc. IEEE Int'l Sym. Biomedical Imaging*, 2008.