Automatic 3D Motion Estimation of Left Ventricle from C-arm Rotational Angiocardiography Using a Prior Motion Model and Learning Based Boundary Detector

Mingqing Chen[1], Yefeng Zheng[1], Yang Wang[1],
Kerstin Mueller[2,3], and Guenter Lauritsch[2]

[1] Imaging and Computer Vision, Siemens Corporate Technology, Princeton, NJ, USA
[2] Healthcare Sector, Siemens AG, Forchheim, Germany
[3] Pattern Recognition Lab, Department of Computer Science,
Friedrich-Alexander-University Erlangen-Nuremberg, Germany
yefeng.zheng@siemens.com

Abstract. Compared to pre-operative imaging modalities, it is more convenient to estimate the current cardiac physiological status from C-arm angiocardiography since C-arm is a widely used intra-operative imaging modality to guide many cardiac interventions. The 3D shape and motion of the left ventricle (LV) estimated from rotational angiocardiography provide important cardiac function measurements, e.g., ejection fraction and myocardium motion dyssynchrony. However, automatic estimation of the 3D LV motion is difficult since all anatomical structures overlap on the 2D X-ray projections and the nearby confounding strong image boundaries (e.g., pericardium) often cause ambiguities to LV endocardium boundary detection. In this paper, a new framework is proposed to overcome the aforementioned difficulties: (1) A new learning-based boundary detector is developed by training a boosting boundary classifier combined with the principal component analysis of a local image patch; (2) The prior LV motion model is learned from a set of dynamic cardiac computed tomography (CT) sequences to provide a good initial estimate of the 3D LV shape of different cardiac phases; (3) The 3D motion trajectory is learned for each mesh point; (4) All these components are integrated into a multi-surface graph optimization method to extract the globally coherent motion. The method is tested on seven patient scans, showing significant improvement on the ambiguous boundary cases with a detection accuracy of 2.87 ± 1.00 mm on LV endocardium boundary delineation in the 2D projections.

1 Introduction

With multi-view or rotational angiocardiography, the 3D shape and motion of the left ventricle (LV) can be estimated for intra-operative analysis of cardiac physiology, e.g., ejection fraction (EF) and myocardium motion dyssynchrony. Previous methods in 3D LV shape/motion estimation from angiocardiography include binary voxel reconstruction based on 2D image intensities [1], elastic matching [2], and 4D B-spline model [3], etc. Most of these methods rely on manual editing of the silhouette of the LV endocardium due to the difficulty in automatic boundary detection. Recently, an automatic

K. Mori et al. (Eds.): MICCAI 2013, Part III, LNCS 8151, pp. 90–97, 2013.

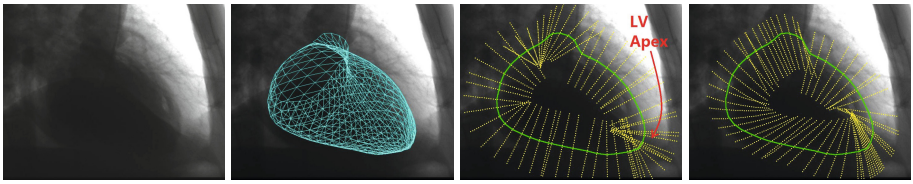


Fig. 1. **(a)** One projection image; **(b)** Projected initial mesh; **(c)** The silhouette (green contour) of the projected mesh and projected trajectory (yellow lines) based on the normal direction; **(d)** The silhouette of the projected mesh and projected trajectory based on the prior model

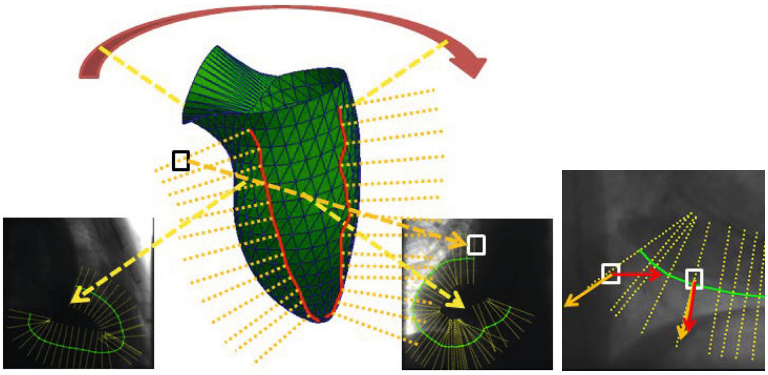
method was proposed for 3D motion estimation of the LV from rotational angiocardiography [4]. First, a 3D LV mesh model is segmented from the 3D volume containing motion blur. The mesh is then forward-projected to each 2D X-ray angiogram as an initial estimate of the LV boundary (Fig. 1b). The silhouette of the projected mesh is extracted and deformed to fit the LV boundary observed on the angiogram. Therefore, the 2D motion vector can be estimated for each silhouette point. With the assumption that the 3D LV motion is along the surface normal, the estimated 2D motion vector can be converted into 3D motion.

In this paper, a new method is proposed to overcome several limitations in the previous work. First, the initialization with a static LV model [4] (which is close to the shape of the end-diastolic phase) is not accurate enough, especially for the end-systolic phase, which has a much smaller LV volume. To address this issue, a prior LV motion model is learned from a set of dynamic cardiac CT sequences. Using the previous 3D static LV model generated from the non-gated reconstruction, the prior motion model is aligned to the patient coordinate system to provide individualized initial mesh for each cardiac phase. Second, the previous assumption of motion along the surface-normal is reasonable for the middle and basal LV segments, but not good for the LV apex (Fig. 1c) with many intersections in the trajectories of mesh points around the apex. In this work, for each mesh point, a 3D motion trajectory is also learned from the dynamic cardiac sequences to replace the assumption of motion along the surface normal. Third, instead of using each individual local image feature [4] for the weak classifiers to train an AdaBoost based boundary detector, principal component analysis (PCA) is applied to a local image patch to create a compact feature representation. The new feature pool is composed of the top principal components (PC) and the combination of a pair of PCs, which are more powerful than the previous local image features. Last, instead of estimating the 3D LV shape of each cardiac phase independently, we exploit the recently proposed multi-surface graph optimization approach [5] to extract the whole motion sequence from one single optimization procedure.

2 Method

2.1 Problem Formulation

In rotational angiocardiography, the C-arm X-ray source/detector pair is rotated around the patient over 200 degrees while acquiring 2D projection images. Imaging starts with



(a) (b)

Fig. 2. **(a)** An illustration of computing cost values along motion trajectory. **(b)** An illustration of computing the dot product between the 2D motion vector (yellow arrow) and the image gradient (red arrow) in two locations indicated by the white rectangle.

a delay of 1s with respect to the injection of contrast agent into the heart chambers. The electrocardiography (ECG) signal is used to record the cardiac phase synchronous to the acquisition of each projection image. The patients were asked to hold breath during the acquisition. In the proposed framework, The ECG signal is analyzed and a relative phase bin is assigned to each projection image. The LV motion is assumed to be periodic and is represented by a sequence of triangulated meshes $\overrightarrow{n}(m,t)$, which indicates the 3D spatial location of a mesh point indexed by $m \in [0, M-1]$ at phase $t \in [0, T-1]$. Here, M is the number of mesh points and T is the number of cardiac phase bins. This representation inherently enforces the identical topology of the LV meshes and point correspondence in different phases. The motion detection framework is established on predetermined motion trajectories from an initial location, which provides a close approximation to the detected motion and is used to create the base of the 4D graph. The motion trajectories determine the possible location for each mesh point before the detection. They are sampled by a limited number of points $n_s(m,t)$, where $n_s \in [0, N-1]$ is the point index. (For an initial estimation based on a static mesh, $n_s(m,t)$ is irrelevant to t.) Each sampled point $n_s(m,t)$ is assigned a cost value, which is used during the optimal detection framework. The optimization aims to minimize the following function,

$$\hat{n}(m,t) = arg \min_{n_s(m,t)} \sum_{t=0}^{T-1} \sum_{m=0}^{M-1} cost(n_s(m,t), m, t), \tag{1}$$

where $\hat{n}$ is selected from all the possible n_s. The details of the graph-search-based motion detection framework can be referred to Chen et al. [5], where both spatial and temporal smoothness constraints are incorporated in the optimization framework.

The assignment of the cost value for motion trajectories on the silhouette is illustrated in Fig. 2a, where the black and the white rectangles represent the original 3D sampled location and the corresponding 2D projected locations. For each sampled point the cost

value is computed by the following functions:

$$cost(n_s, m, t) = -\sum_{p=0}^{P-1} \delta(p,t)\zeta(m,p)w(n_s,m,p), \tag{2}$$

where P is the number of projection images. Function $\delta(p,t)$ equals to one if and only if the projection image p belongs to the phase index t. Otherwise, it equals to zero. Function $\zeta(m,p)$ equals to one if and only if the mesh point m locates on the silhouette in projection p. Otherwise, it equals to zero. Function $w(n_s,m,p)$ computes the cost value for a sampled point n_s of a mesh point m in one projection image p. In [5], the cost function $w(n_s,m,p)$ is assigned based on the dot-product of the projection motion direction and image gradient (Fig. 2b), which works well for lung tumor boundary detection in 2D projections of mega-voltage cone beam CT. In this application, the image boundary of the LV endocardium is much weaker than a nearby confounding boundary of the LV pericardium (the boundary between the heart and lung). Therefore, the deformed silhouette is often attracted to the pericardium. In this work, we propose to use the response of a learning-based boundary detector as the cost function $w(n_s,m,p)$ (Section 2.4).

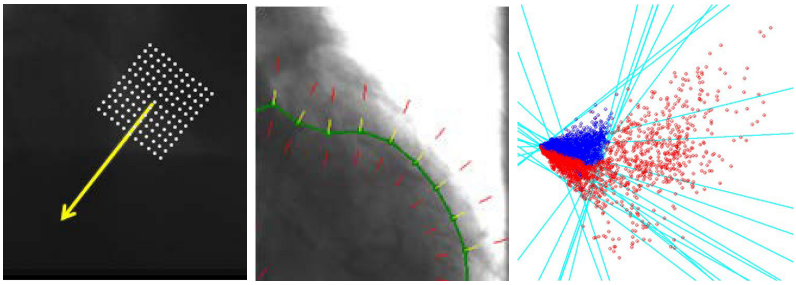
2.2 Prior Motion Model

The prior LV motion model is derived from a training set containing 12 LV motion sequences extracted from dynamic cardiac CT data sets, where each sequence contains 10 LV meshes (10 phases). The mean LV surface is computed after aligning the 4D (3D+t) shapes based on the Procrustes analysis [6]. Given an input dataset of a patient, a static LV mesh is extracted from the non-ECG-gated reconstruction. The static mesh and the prior motion model are represented by the same LV model with built-in point correspondences. The prior motion model is transformed to the patient coordinate system specified by the static mesh based on the thin-plate-spline (TPS) interpolation approach [7] to provide an personalized initial mesh for each phase. The static mesh is found close to the shape of the end-diastolic (ED) phase, instead of the average shape of the whole cardiac cycle. The displacement between ED phase of the mean LV motion model and the static mesh provides the landmarks/anchor points for the registration technique.

2.3 Motion Trajectory

In order to convert the 2D motion vector estimated from the angiogram to 3D, a previous assumption is that the motion occurs mainly along the surface normal [4]. It is reasonable for most parts of the LV where the motion is dominated by the contraction and relaxation of the LV cavity. However, this assumption does not hold well for the LV apical region, where the LV endocardium forms a sharp wedge. As shown in Fig. 1c, the projected normal directions have severe self-intersections around the LV apex. The intersections in the motion trajectory will eventually result in folds in the estimated 3D mesh.

In this work, the prior motion trajectory is fitted by a 3D quadratic curve for each mesh point (m,t). The fitting is based on using the transformed prior motion model



(a) (b) (c)

Fig. 3. **(a)** Sample of an image patch; **(b)** Along a manual contour (green) in one projection image, positive (yellow arrows) and negative (red arrows) locations are selected to train the boundary detector; **(c)** Positive (blue) and negative (red) training data visualized in PCA-reduced 2D space. Each cyan line shows the classification boundary of a weak classifier.

introduced in the previous section, where a parametric curve with the following representation is determined by the least-squared errors (LSE) with 10 points (one from each cardiac phase and the whole cardiac cycle contains 10 phases):

$$x(t) = x_0 + a_1 t + a_2 t^2, \tag{3}$$

$$y(t) = y_0 + b_1 t + b_2 t^2, \tag{4}$$

$$z(t) = z_0 + c_1 t + c_2 t^2. \tag{5}$$

(x_0, y_0, z_0) is assigned with the point location of the initial mesh model.

Fig. 1d shows the projected motion trajectory. Compared to the normal direction, the learned motion trajectory has much fewer self-intersections around the LV apex.

2.4 Boundary Detector Based on Supervised Learning

Instead of computing the image gradient for each projected 2D location [5], a binary classifier is trained from a data set containing manually labeled positive and negative data points. The graph node cost $w(n_s, m, p)$ in Equation (2) is computed based on the posterior probability $p(+1|x)$, where x is a feature vector including the intensity of a sampled 11×11 pixel patch centered on the projected location. The orientation of the patch is determined by the tangent direction of the motion trajectory at its center location. Fig. 3a illustrates one image patch example.

The manually annotated LV contours on the projection images are used to create the training data set, which is illustrated in Fig. 3b. The contour points are used as positive data, while the negative data are located at certain distance outside or inside of the contour. The orientation of both positive or negative patches is set to the normal direction of the nearby annotated contour.

The training stage includes a dimensionality reduction step based on the principal component analysis (PCA) and an AdaBoost training step [8]. During the PCA step, the original 121 dimensional space is reduced to a few principal components (PC) with

largest variations. Two types of features are extracted after PCA to train an AdaBoost classifier. The first type of features contains coefficients for the 50 most significant PCs:

$$f_{T1}(X_r) = \sum_i w_i I_{ri}, \tag{6}$$

where w_i and I_{ri} represents the weight and value of the i^{th} PC, respectively. The second type of features contains a linear combination of the first two PCs since the two most significant PCs cover about 98% of all the variance,

$$f_{T2}(X_r) = cos(\theta) I_{r1} + sin(\theta) I_{r2}, \tag{7}$$

where I_{r1} and I_{r2} are the value of the training point X_r in the reduced dimension and θ sets the relative weight of these two PCs. Parameter θ is quantized with a step size of 1 degree to generate an finite number of 2D linear features. Fig. 3c shows a few selected type-II features by the AbaBoost classifier. The positive and negative training data are shown as blue and red dots, respectively. A cyan line shows a type-II feature with a specific combination of the first two PCs.

For each training point, the feature function f_{T1} or f_{T2} is able to generate a value used for a weak classifier:

$$h_i(X_r) = sign(p_i f_i(X_r) - p_i \gamma_i), \tag{8}$$

where p_i is ± 1, representing the parity of the data, and γ_i is the threshold for the binary classification. The subscript i in those denotations represents the i^{th} feature in the feature pool. A strong classifier is generated by combining a set of selected weak classifiers from the feature pool with a value F:

$$H(X_r) = sign(F(X_r)),\ F(X_r) = \sum_{i=1}^{K} \sigma_i h_i(X_r), \tag{9}$$

where σ_i is a parameter related to the training error of the i^{th} weak classifier and K is the number of selected weak classifiers. The posterior probability of the binary AdaBoost classifier is given by:

$$q(+1|X_r) = \frac{e^{2F(X_r)}}{1 + e^{2F(X_r)}},\ q(-1|X_r) = \frac{e^{-2F(X_r)}}{1 + e^{-2F(X_r)}}. \tag{10}$$

3 Experiments

We collected seven sequences of C-arm rotational angiocardiograms from two clinical sites. A typical imaging protocol was composed with a rotation of the C-arm X-ray source/detector around the patient for 200 degrees. Each rotation took about five seconds and a total of 133 projection images were recorded. Leave-one-out cross validation is performed to quantitatively evaluate the accuracy of the proposed method by computing the distance between the mesh silhouette and a manual contour. The distance of the two contours is computed by measuring the distance between each silhouette point and its closest location on the manual contour:

$$dist(C_1, C_2) = \frac{1}{N_p} \sum_{p_1 \in C_1} min_{p_2 \in C_2} dist(p_1, p_2), \tag{11}$$

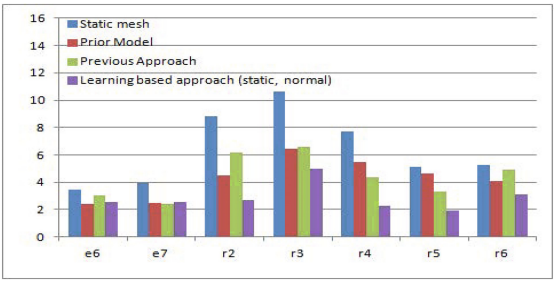


Fig. 4. Mean of the distance (measured in mm) between silhouette derived based on four approaches and manual contour

where p_1 and p_2 are the points belonging to contour C_1 and C_2, respectively. N_p is the number of points in C_1 that are taken to compute the distance metric. To evaluate the performance of the algorithm for each scan, the mean and variance of the contour distance in all the projection images are computed. Fig. 4 shows the mean distance before and after the motion estimation in seven scans. The bars "Static Mesh" show the initialization error using a static mesh extracted from a 3D volume using non-ECG-gated reconstruction, while the bars "Prior Model" show the initialization error using the pre-learned motion model, which is more accurate than the initialization using a static mesh. The performance of the previous approach [4] is shown as the green bar, which is surpassed by the proposed method (the purple bar). The proposed method achieves a mean contour distance of $2.87 \pm 1.00\ mm$. It should be noted that for a gradient-based approach, the difference to the manual contour is even larger than the initial static model, and it is not shown in this figure.

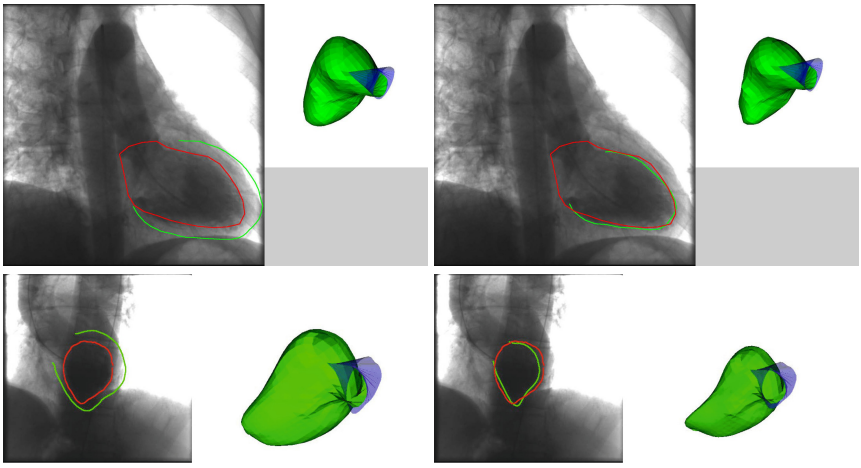


Fig. 5. Detected silhouette (green) using gradient-based boundary detector (left, [5]) and learning-based boundary detector (right) compared with manual contour (red). The 3D mesh of LV (green) and outflow tract (blue) is shown on the right for each 2D projection image.

Fig. 5 compares the detection result of a gradient-based cost function [5] and the proposed learning-based cost function. It can be seen that for the learning-based boundary detector, the detected silhouette is much closer to the manual contour, while the gradient-based approach is attracted to the confounding boundary.

4 Discussions and Conclusions

In this work, a new method was proposed to automatically estimate the 3D motion of the left ventricle from rotational angiocardiography, which incorporated a boundary detector based on the principal component analysis of local image patches. All the components were integrated into a multi-surface graph optimization framework to achieve a coherent 3D motion estimate. The estimated 3D motion can be used for motion compensated reconstruction and intra-operative measurement of the cardiac function, including ejection fraction and dyssynchrony analysis.

References

1. Prause, G.P.M., Onnasch, D.G.W.: 3D reconstruction of the ventricular dynamic shape from the density profiles of biplane angiocardiographic image sequences. In: IEEE Proc. Computers in Cardiology, pp. 193–196 (1994)
2. Lotjonen, J., Magnin, I., Nenonen, J., Katila, T.: Reconstruction of 3-D geometry using 2-D profiles and a geometric prior model. IEEE Trans. Medical Imaging 18(10), 992–1002 (1999)
3. Moriyama, M., Sato, Y., Naito, H., Hanayama, M., Ueguchi, T., Harada, T., Yoshimoto, F., Tamura, S.: Reconstruction of time-varying 3D left ventricular shape from multiview x-ray cineangiocardiograms. IEEE Trans. Medical Imaging, 773–785 (2002)
4. Chen, M., Zheng, Y., Mueller, K., Rohkohl, C., Lauritsch, G., Boese, J., Funka-Lea, G., Hornegger, J., Comaniciu, D.: Automatic extraction of 3D dynamic left ventricle model from 2D rotational angiocardiogram. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011, Part III. LNCS, vol. 6893, pp. 471–478. Springer, Heidelberg (2011)
5. Chen, M., Bai, J., Zheng, Y., Siochi, R.A.C.: 3D lung tumor motion model extraction from 2D projection images of mega-voltage cone beam CT via optimal graph search. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) MICCAI 2012, Part I. LNCS, vol. 7510, pp. 239–246. Springer, Heidelberg (2012)
6. Gower, J.C.: Generalized procrustes analysis. Psychometrika 40(1), 33–51 (1975)
7. Bookstein, F.: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Anal. Machine Intell. 11(6), 567–585 (1989)
8. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1), 119–139 (1997)