# Segmenting the Papillary Muscles and the Trabeculae from High Resolution Cardiac CT through Restoration of Topological Handles

Mingchen Gao[1⋆], Chao Chen[1⋆], Shaoting Zhang[1], Zhen Qian[2], Dimitris Metaxas[1], and Leon Axel[3]

[1] CBIM Center, Rutgers University, Piscataway, NJ 08854
[2] 2 Piedmont Heart Institute, Atlanta, GA 30309
[3] New York University, 660 First Avenue, New York, NY 10016

**Abstract.** We introduce a novel algorithm for segmenting the high resolution CT images of the left ventricle (LV), particularly the papillary muscles and the trabeculae. High quality segmentations of these structures are necessary in order to better understand the anatomical function and geometrical properties of LV. These fine structures, however, are extremely challenging to capture due to their delicate and complex nature in both geometry and topology. Our algorithm computes the potential missing topological structures of a given initial segmentation. Using techniques from computational topology, e.g. persistent homology, our algorithm find topological handles which are likely to be the true signal. To further increase accuracy, these proposals are measured by the saliency and confidence from a trained classifier. Handles with high scores are restored in the final segmentation, leading to high quality segmentation results of the complex structures.

## 1 Introduction

Computed tomography (CT) is a very important imaging modality for diagnosing cardiovascular diseases. Compared with other imaging modalities (such as ultrasound and magnetic resonance imaging), CT is able to show detailed anatomic structures within the cardiac chambers [15]. Recent advances in CT technology allow a 320 multi-detector CT scanner to successfully capture the papillary muscles and trabeculae at a resolution which has not been reached before.

Most of the existing methods to perform cardiac segmentations [3, 20, 13] model the inner heart wall as a smooth surface, which does not include the papillary muscles and the trabeculae at all. Zheng *et al.* [20] proposed an algorithm to automatically segment the four chambers of the heart in four seconds. Ecabert *et al.* [5] presented a learning-based approach based on active shape model (ASM) for the segmentation of four chambers and major vessel trunks. Other models include, but are not limited to graph cut [7], atlas based segmentation [10] and local deformation [11].

⋆ Both authors contributed equally to this work.

These methods, although proven to be successful in various situations, are not designed to accurately segment smaller, complex structures such as the papillary muscles and the trabeculae. Previous attempts [3, 18] were able to capture the papillary muscle, but could not segment trabeculae with satisfying quality. Gao *et al.* [8] manually segmented one frame (at the end-diastole state) of an image sequence of a cardiac cycle, and then deformed the segmentation to match the other frames. Although their method focused on preserving the fine structures during the deformation, it only enforced consistency of geometry [17], not of topology. Accurately segmenting the complex structures of the papillary muscles and the trabeculae is still a challenging task. The reason is threefold. 1) The detailed structures are complex and small, making them hard to be distinguished from noise. 2) Some trabeculae go through the ventricle cavity and are very thin. Existing methods often fail to segment them due to the smoothness prior. 3) Such complex structures have a very different nature from other parts such as free wall and septum. Furthermore, trabeculae have a large variety of geometry and intensity even within the same cardiac image. This requires the segmentation method to be extremely adaptive in terms of parameters, making full automation very difficult.

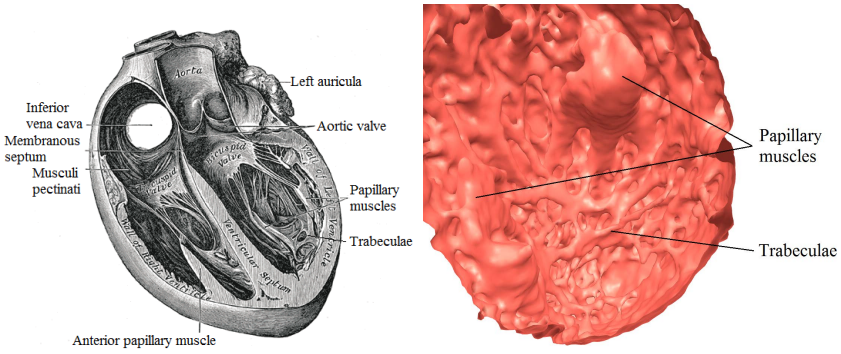
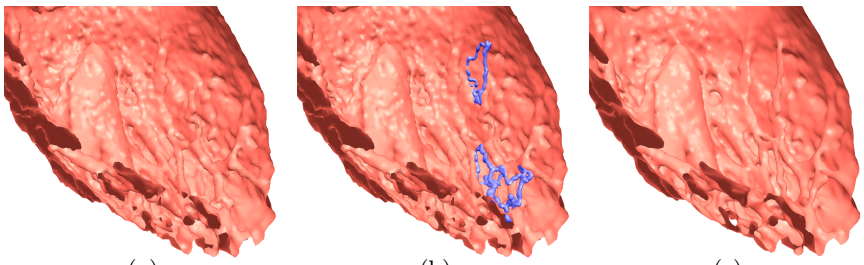



**Fig. 1.** Left: Left ventricle anatomy, picture from [9]. Right: Segmentation results represented as a 3D triangle mesh successfully captured the papillary muscle and the trabeculae.

Accurately segmenting the papillary muscles and the trabeculae is very important and of high interest to doctors. The functions of the papillary muscles and the trabeculae have still not been fully understood. Left ventricle anatomy is show in Fig. 1. The *papillary muscles* are attached to the valves via chordae tendineae. The *trabeculae* project from the inner surface of both ventricles of the heart. Some are completely attached to the wall of the heart. Others are fixed at both ends to the ventricular wall, but the intermediate section is freely mobile within the cavity, forming topological *handles*. There are a number of functional hypotheses for the trabeculation of the heart wall. High quality segmentations of



(a) (b) (c)

**Fig. 2.** Improvement of our method. (a) Baseline segmentation. (b) Proposed restoration handles. (c) Final segmentation.

such structures are useful for further investigating their functions, the mechanics of the heart [12] and geometrical properties of cardiac structures [13].

In this paper, we propose a topological method to restore missing structures of a given segmentation, generated by any existing segmentation tool. It proposes hypotheses of where and how topological handles should be reestablished. On the basis of those topological proposals, a two-step screening is performed to select handles with higher confidence for structure restoration. Our algorithm evaluates each handle independently based on its saliency, rather than absolute intensities. Explicitly restoring selected handles makes the restoration adaptive to each trabecula, thus avoiding a universal threshold in the whole domain. Fig. 2 shows the improvements of our algorithm, with restored trabeculae highlighted. Quality of restored handles can be verified by comparing with the intensity function in Fig. 3.

Using topological information in image segmentation has been studied in both computer vision [2] and medical imaging [16]. As far as we know, in all previous methods, that use either random field energy models (MRF and CRF) [14] or deformable models [19], topological priors such as connectivity or handle-free are enforced as a segmentation constraint. In this paper, instead of enforcing the final segmentation to have an upper bound of the number of components or handles, we restore topological features, as long as we have high confidence in them.

## 2 Methodology

The algorithm flow is illustrated in Fig. 3. An initial segmentation is applied on the image and then we compute handles that need to be restored. Each handle is delineated by a thickened cycle, as illustrated in Fig. 3 (c). The segmentation is fixed accordingly, by enforcing these cycles to appear in the final segmentation.

In this section, we first state the desired properties for the cycles that we should use for handle restoration. Next, we build a connection to a theory of persistent homology [6] in the computational topology community. The output of persistent homology is a set of dots corresponding to handles that appear

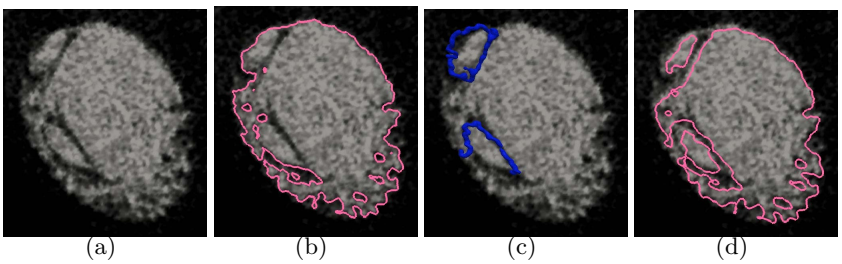



**Fig. 3.** A two-dimensional slice of the data on which we illustrate the workflow. (a) CT image. (b) Initial segmentation. (c) Proposed fixing cycles (partially occluded). (d) Final segmentation with handles restored.

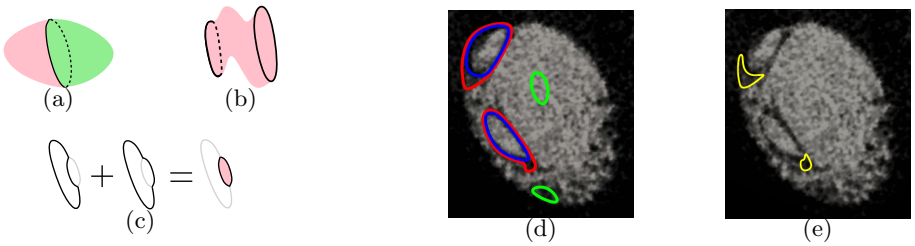



**Fig. 4.** (a) A given cycle is sealed by two different patches (pink and green). (b) The sum of two cycles is sealed by a tube shaped patch (pink), which delineates a way to deform between the two. (c) The mod-2 sum of two cycles, and a sealing patch. (d) Various cycles for a given function. (e) Sum of the corresponding blue and red cycles.

when we threshold the domain using a function value. Based on such theory, we design an algorithm to compute proposal cycles, each of which delineates one handle. We conclude this section by explaining how to choose the promising candidates from all these proposals so that they satisfy the desired properties.

### 2.1 Intuition and the Desired Properties of Cycles for Handle Restoration

We start by introducing some terminology. A closed curve is called a *cycle*. The mod-2 sum (exclusive or) of a set of cycles is also called a *cycle*. A 2-manifold with boundary is called a surface *patch*. A patch $c$ *seals* a cycle $z$ if its boundary is $z$, formally, $\partial(c) = z$. When the sealed cycle is the sum of two cycles, the patch could be considered as the area swept through when we smoothly deform the first cycle into the second. In a 3D image, there could be infinitely many patches that seal a given cycle, and thus infinitely many ways to deform between cycles. See Fig. 4(a)-(c) for illustration.

Given a function defined on the image domain $f : \Omega \to \mathbb{R}$, $\Omega \subseteq \mathbb{R}^3$. To restore missing handles based on the image $f$, the two blue cycles in Fig. 4(d)

are natural choices. Intuitively, a cycle is chosen if the intensity along it is low; yet the intensity between the corresponding handle and the wall is high. On the other hand, we need to propose a set of cycles such that any two of them would not delineate the same trabecula. Furthermore, each trabecula should be covered by a proposal. These intuitions lead to three properties need to be satisfied.

The set of cycles we select should satisfy the following properties. First, we require a high saliency for each selected cycle. A selected cycle $z$ needs to go through points with relatively low function values, and any surface patch sealing this cycle has to have some points with relatively large function values. We measure the *saliency* of the cycle using the difference of maximal function values of the cycle and a sealing patch. There are infinitely many possible patches sealing a given cycle (Fig. 4(a)). Among them, we choose the patch whose maximal value is the smallest, formally,

$$\text{Saliency}(z) = \left( \min_{c:\partial(c)=z} \max_{p\in c} f(p) \right) - \max_{p\in z} f(p) \tag{1}$$

In Fig. 4(d), the blue and red cycles have high saliency, but green ones do not.

Second, we should not select several cycles that in fact correspond to the same trabeculae/handle. Any two selected cycles are required to have a large dissimilarity, i.e., the saliency of their mod-2 sum,

$$\text{Dissimilarity}(z_1, z_2) = \text{Saliency}(z_1 + z_2)$$

The dissimilarity between a cycle and zero is its saliency. In Fig. 4(d), there is a small dissimilarity between each blue cycle and the red cycle surrounding it. We should select only one of them. The sum of of the corresponding cycles, which are represented as yellow cycles in Fig. 4(e), have low saliency.

Third, we should exhaustively select all possible salient cycles. Any given cycle $z$ should have a small dissimilarity from the set of selected cycles, $Z$, which is defined as the minimal dissimilarity between $z$ and the mod-2 sum of a subset of $Z$,

$$\text{Dissimilarity}(z, Z) = \min_{Z'\subseteq Z} \text{Dissimilarity}\left( z, \sum_{z'\in Z'} z' \right)$$

This quantity lowerbounds the saliency of $z$ itself since we allow $Z'$ to be empty. Thus the dissimilarity is small if $z$ has small saliency. In other words, any cycle $z$ either has a low saliency, or has a good approximation from the given set $Z$, expressed as the sum of a subset $Z' \subseteq Z$.

### 2.2 Persistent Homology

In order to compute cycles that serve our purpose, we use persistent homology. The input of the tool is a topological space and a scalar function, e.g., the image domain $\Omega$ and the image function $f$. The output is a set of dots on $\mathbb{R}^2$ corresponding to a set of features.

For a given scalar value $\ell$, we call the set of points with function value no greater than $\ell$ a *sublevel set*, formally, $\Omega_\ell = \{x \in \Omega \mid f(x) \leq \ell\}$. We study the

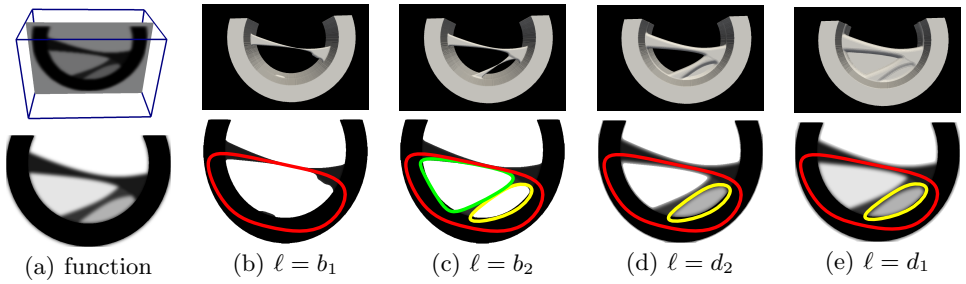



**Fig. 5.** (a) Synthetic function. (b)-(e), Sublevel sets $\Omega_\ell$ at time $b_1 < b_2 < d_2 < d_1$. Bottom row: 2D slices of the sublevel sets. We also show the intensity inside the sublevel sets. The red, yellow and green cycles are $z_1$, $z_2$ and $z_3$ respectively.

topological changes of sublevel sets $\Omega_\ell$ as the parameter $\ell$ increases from $-\infty$ to $+\infty$, during which the sublevel set grows from empty to the whole domain $\Omega$. For convenience, we say a topological event happens *at time* $\ell_0$ if it happens when we grow the sublevel set from $\Omega_{\ell_0-\epsilon}$ to $\Omega_{\ell_0}$.

In this paper, we focus on a specific kind of topological feature, handle. In Fig. 5, at time $b_1$, a new handle (delineated by the cycle $z_1$) is created. This handle is destroyed (becomes trivial) at time $d_1$. The two corresponding function values are called the *birth time* and *death time* of this topological feature. At time $b_2$ and $d_2$, another handle (delineated by the cycle $z_2$) is created and destroyed. For each handle, the difference between its death time and birth time is called the *persistence*.

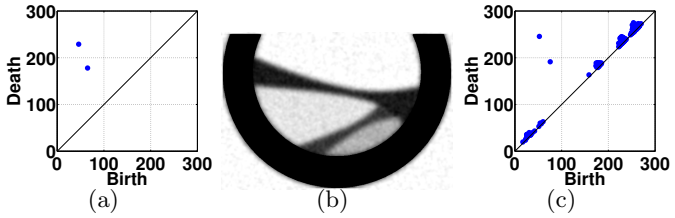



**Fig. 6.** (a) Persistence diagram of the synthetic function $\mathcal{D}_f$. (b) Perturbed function $\hat{f} = f + e$. (c) Persistence diagram of the perturbed function $\mathcal{D}_{\hat{f}}$.

All topological features are recorded in a *persistence diagram*. Each handle corresponds to a dot in $\mathbb{R}^2$, whose $x$ and $y$ coordinates are the birth and death times. The vertical or horizontal distance of a dot from the diagonal $x = y$ is its persistence. Fig. 6 is the persistence diagram of the synthetic function, with the two handles corresponding to two blue dots.

A justification of using the persistence diagram is its stability with regard to perturbations of the function [4]. Formally, the bottleneck distance between the diagrams of a function and the same function with added noise is upperbounded by the $L_\infty$ norm of the noise, $\text{dist}(\mathcal{D}_f, \mathcal{D}_{\hat{f}}) \leq \|f - \hat{f}\|_\infty = \|e\|_\infty$, where $\hat{f} = f + e$. In Fig. 6, after introducing noise $e$ into the synthetic function, the persistence diagram could have many new dots with small persistence ($\leq \|e\|_\infty$). However, no large persistence dots are introduced or removed. The large persistence dots only move in the diagram by at most $2\|e\|_\infty$. In other words, noise in the image only introduces spurious handles that are destroyed right after creation.

In order to compute the persistence diagram, we first discretize the image domain into a *cubical complex* whose basic elements are *cells* of dimension zero to four, i.e., *vertices*, *edges*, *squares* and *cubes*, respectively. The set of vertices corresponds to the set of all voxels in the image. In Fig. 7(a), we show an example complex in 2D, with values of vertices specified. This discretization corresponds to the 4-/6-neighborhood for 2D/3D images, as defined in digital topology.

We build the *boundary matrix* of dimension $d$, whose columns and rows correspond to $d$-dimensional cells ($d$-cells) and $d$-dimensional cells ($(d-1)$-cells) respectively. Columns and rows are indexed from left to right and from top to bottom respectively, corresponding to cells sorted according to function values. An entry of the matrix is set to 1 if the corresponding $(d-1)$-cell belongs to the boundary of the corresponding $d$-cell, and 0 otherwise. The one-dimensional boundary matrix is, in fact, the adjacency matrix of the underlying graph. For the example complex in Fig. 7(a), the sorted cells, and one-and-two-dimension boundary matrices are given in Fig. 7(b). Each column vector of the two-dimensional boundary matrix is a cycle, and the boundary of a 2-cell is a square. Since we use mod-2 addition, the sum of any set of columns is a cycle and the boundary of a patch which is the sum of the set of corresponding 2-cells. Columns of the boundary matrix span the space of all possible cycles of the discretized image domain $\Omega$.

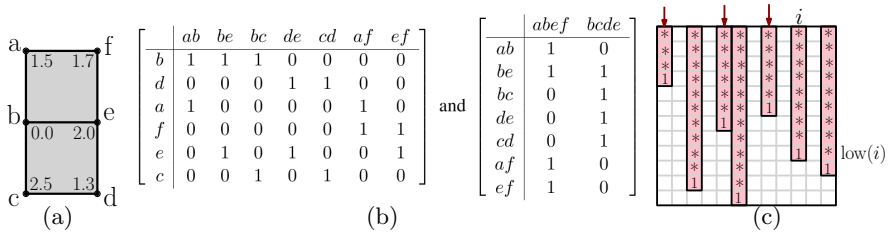



**Fig. 7.** (a) Example cubical complex, with function values given. (b) Boundary matrices of dimension one and two. (c) An indicative example of the reduced matrix $R$. This example does not correspond to boundary matrices in (b).

To compute the one-dimensional persistence diagram, which records features corresponding to handles, we apply a matrix reduction on the two-dimensional

boundary matrix. Note that all additions are mod-2. We reduce columns of the matrix from left to right. For each column, we only use the columns on its left to reduce it. We start from the row index of the lowest nonzero entry of column $i$, called $\text{low}(i)$. If this row index is equal to $\text{low}(j)$ for some column $j$ that has been reduced, we add column $j$ to $i$, and thus reduce $\text{low}(i)$. We repeat until $\text{low}(i)$ is not the lowest nonzero entry of any column $j < i$, or column $i$ becomes zero. In the former case, this reduced column corresponds to a handle in the persistence diagram, whose birth (resp. death) time is the function value of the cell $\text{low}(i)$ (resp. the cell $i$). One property of the reduced matrix is that the lowest nonzero row index $\text{low}(\cdot)$ for all nonzero columns are unique. Fig. 7(c) shows an example of the reduced matrix, denoted by $R$. The edge $\text{low}(i)$ and the square $i$ are where the handle is created and destroyed, called the *creator* and *destroyer*.

### 2.3 Computing Proposal Cycles

We first compute one proposal cycle for each handle from the persistence diagram. For a handle that is born at time $b$ and dies at $d$, we take a cycle that goes through the handle and lies within the sublevel set $\Omega_b$. Furthermore, we choose a cycle which is sealed by a patch with the maximum function value $d$. For example, in Fig. 5, we choose $z_1$ for the handle born at $b_1$. For the handle born at $b_2$, we choose $z_2$ instead of $z_3$, because it is sealed up by a patch with a maximum value $d_2$. We say the computed cycle *delineates* the corresponding handle. We denote by $\bar{Z}$ the set of all proposal cycles, delineating all handles that appeared in some sublevel sets. How to choose from them the salient ones will be discussed later.

To compute elements of $\bar{Z}$, we reuse the output of the algorithm for the persistence diagrams, in particular, the reduced matrix $R$ (Fig. 7(c)). To compute a cycle for the handle corresponding to column $i$, collect the set of columns $R(*, j)$ on $R(*, i)$'s left such that $\text{low}(j) < \text{low}(i)$, e.g., the three marked columns in Fig. 7(c). These columns form a new matrix, $\widehat{R^i}$. The following theorem shows that any cycle that is the sum of the $i$-th column and a set of columns in $\widehat{R^i}$ is a valid cycle representing the handle corresponding to column $i$.

**Theorem 1** *$\forall x$, $y = R(*, i) + \widehat{R^i}x$ is a cycle delineating the corresponding handle.*[4]

A delineating cycle may have freedom to wiggle within a handle, as long as it contains the creator edge $\text{low}(i)$. Thus we prefer computing a cycle with simple geometry.

**Problem 1** *Compute $y = R(*, i) + \widehat{R^i}x$ with the minimal number of nonzero entries.*

Unfortunately, this problem is not only NP-hard, but also NP-hard to approximate within any constant. Alternatively, we propose a heuristic method to

[4] Theoretical proof is omitted due to limited space.

compute $y$ as follows. Starting with the $i$-th column $y = R(*, i)$. Iterate through the row indices from $\text{low}(i) - 1$ to 1. For each row index $k$, if $y(k) \neq 0$, and $k = \text{low}(j)$ for some $j < i$, and adding $R(*, j)$ to $y$ would reduce the number of nonzero entries, then add $R(*, j)$ to $y$.

Over the course of the algorithm, all used columns $R(*, j)$ will belong to $\widehat{R^i}$. So we always get a valid $y$. Furthermore, the number of nonzero entries of $y$ monotonically decreases. The cycle gets shorter after each addition. In practice, the heuristic algorithm generates cycles that are reasonably simple. Trabeculae usually correspond to thin handles, which leave limited space for cycles to wiggle within.

### 2.4 Selecting Proposal Cycles Satisfying Desired Properties

From the set of all proposed cycles, $\bar{Z}$, we select the set of promising ones using a two level screening method. In the first level, we select cycles delineating handles with persistence not less than a threshold $\theta$.

In fact, the saliency of each delineating cycle as defined in Equation (1) is equal to the persistence of the handle. We abuse notations and say a proposed cycle has the same birth time, death time and persistence as its corresponding handle. The following theorem guarantees that the selected set of cycles, namely, $Z_\theta = \{z \in \bar{Z} \mid \text{persistence}(z) \geq \theta\}$, satisfies the three desired properties we discussed in Section 2.1.

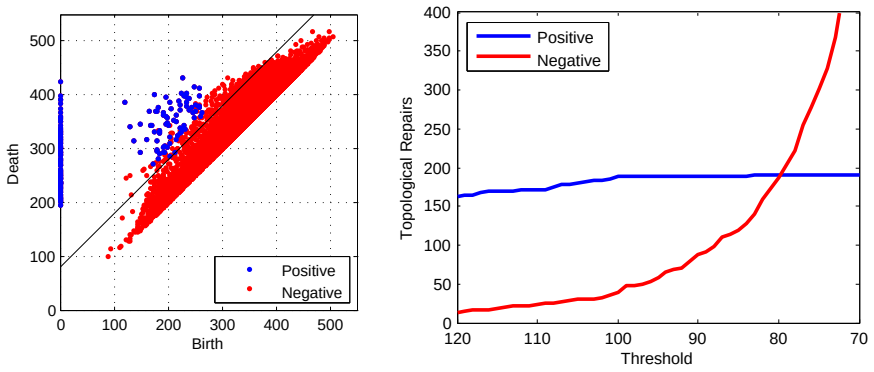
**Theorem 2** *(A) Any cycle in $Z_\theta$ has a saliency at least $\theta$;*
*(B) The dissimilarity between any two cycles of $Z_\theta$ is at least $\theta$;*
*(C) For any cycle $z$, its dissimilarity from $Z_\theta$ is at most $\theta$.*

Although high persistence cycles lead to salient handles that are more likely from trabeculae, in practice, the first screening would inevitable select certain wrong cycles. Therefore, we use a classifier with geometrical features as the second level screening.

## 3 Experiments

The proposed algorithm was employed on 6 cardiac CT image at the end diastolic state, where trabeculae structures are separated the most. The CT data were acquired on a 320-MDCT scanner, using a conventional ECG-gated contrast-enhanced CT angiography protocol. The imaging protocol parameters include: prospectively triggered, single-beat, volumetric acquisition; detector width 0.5 mm, voltage 120 KV, current $200 - 550$ mA.

We used the region competition algorithm [21] to initialize the segmentation. In order to focus on restoring the missed trabeculae, we decreased the function value of a voxel to zero if it was already segmented out. Handles which correspond to the structures that had been successfully captured would have birth time 0 and appear as dots on the $y$ axis of the diagram (Fig. 8(a)). Their cycles were not



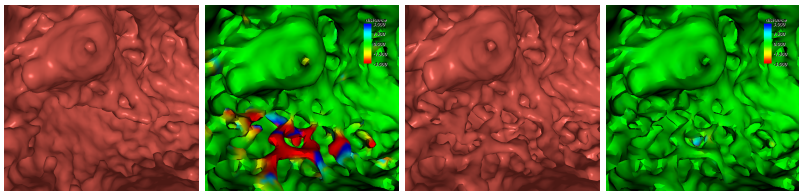

(a) (b)

**Fig. 8.** (a) Persistence diagram of one cardiac image. The persistence threshold is marked as 80. (b) The relationship of persistence threshold and number of cycles.

be used for restoration. It took about 6 to 8 minutes on a commodity machine using 6 to 10 GB memory to compute the persistence diagram and cycles.

We used 5 out of 6 images for selecting persistence threshold, and training, and the remaining image for testing. We had human experts carefully examine proposal cycles and mark them as positive and negative, by studying the image function. (For example, the blue cycles in Fig. 3(c) are considered positive.) We performed the two level screening to select promising proposals. We empirically chose the persistence threshold. In Fig. 8(b) we plotted the number of positive and negative dots with persistence above a threshold, for one training image. We chose $\theta = 80$ so that we included all positive proposal cycles and a reasonable number of negative ones from the training images. In the persistence diagram shown in Fig. 8(a), we drew the line $y = x + \theta$. All dots above this line were selected after the first level screening except the ones on the $y$ axis. Notice the big variation of the birth and death times of positive dots. This implies that it is impossible to detect them using an universal intensity prior.

Next we explain how to train the classifier for the second level. For all six images, we selected 458 positive out of 1095 proposals after persistence screening.



(a) (b) (c) (d)

**Fig. 9.** (a) Baseline segmentation. (c) Proposed segmentation. (b)(d) Distance map from the pseudo-groundtruth to (a)(c) respectively.

Among those selected proposals, we used the ones from the five training images for training and 10-fold cross validation, and the ones belonging to the test image for testing. We used the LIBSVM toolbox [1] to train our classifier. Features used were birth time, death time, persistence, length of the cycle, and the relative position in the ventricle. We achieved 81.69% accuracy in the testing.

After promising proposals were selected, we generated the final segmentation by enforcing these cycles to be included. We reused the region competition algorithm with the same parameters so that the remaining parts of the final segmentation are the same as the initial one. Groundtruth is extremely difficult to get for this kind of data using manual segmentation. We generated the *pseudo-groundtruth* for the testing image by enforcing the human marked positive cycles. We compared the results of our method to that of a baseline segmentation generated using the region competition method (Fig. 9) by showing the distance from the pseudo-groundtruth to each result. Distance was represented by different colors. Green, red and blue represented accurate segmentation, over segmentation and under segmentation, respectively. The trabeculae missing from the baseline segmentation had greater error. Our segmentation, as shown in Fig. 9(b), successfully captured more trabeculae. The distance error of the initial segmentation is $0.2108 \pm 0.4973$ voxel, whereas our segmentation method has distance error $0.1101 \pm 0.3679$ voxel.

## 4 Conclusion

In this paper, we proposed a novel left ventricle segmentation method. Our segmentation approach is generic and could be applied to other topologically complicated segmentation problems with complex topological structures. It would be of theoretical interest if we build a quantitative relationship between the signal-noise ratio of the image and the stability of the persistent diagram.

**Acknowledgement** This work is supported by NIH-R21HL88354-01A1, Multiscale Quantification of 3D LV Geometry from CT. The second author thanks Prof. Herbert Edelsbrunner for helpful discussions.

## References

1. C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.
2. C. Chen, D. Freedman, and C. H. Lampert. Enforcing topological constraints in random field image segmentation. In *CVPR*, pages 2089–2096, 2011.
3. T. Chen, D. Metaxas, and L. Axel. 3D cardiac anatomy reconstruction using high resolution CT data. In C. Barillot, D. Haynor, and P. Hellier, editors, *MICCAI*, volume 3216 of *LNCS*, pages 411–418. Springer, Heidelberg, 2004.
4. D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
5. O. Ecabert, J. Peters, H. Schramm, C. Lorenz, J. von Berg, M. Walker, M. Vembar, M. Olszewski, K. Subramanyan, G. Lavi, and J. Weese. Automatic model-based segmentation of the heart in CT images. *TMI*, 27(9):1189 –1201, sept. 2008.

6. H. Edelsbrunner and J. Harer. *Computational topology: an introduction.* Amer Mathematical Society, 2010.
7. G. Funka-Lea, Y. Boykov, C. Florin, M.-P. Jolly, R. Moreau-Gobard, R. Ramaraj, and D. Rinck. Automatic heart isolation for CT coronary visualization using graph-cuts. In *ISBI*, pages 614 –617, april 2006.
8. M. Gao, J. Huang, S. Zhang, Z. Qian, S. Voros, D. Metaxas, and L. Axel. 4D cardiac reconstruction using high resolution CT images. In D. Metaxas and L. Axel, editors, *FIMH*, volume 6666 of *LNCS*, pages 153–160. Springer, Heidelberg, 2011.
9. H. Gray. *Anatomy of the human body.* Lea & Febiger, 1918.
10. I. Isgum, M. Staring, A. Rutten, M. Prokop, M. Viergever, and B. van Ginneken. Multi-atlas-based segmentation with local decision fusion-application to cardiac and aortic segmentation in CT scans. *TMI*, 28(7):1000 –1010, july 2009.
11. M.-P. Jolly. Automatic segmentation of the left ventricle in cardiac MR and CT images. *IJCV*, 70:151–163, November 2006.
12. S. Kulp, M. Gao, S. Zhang, Z. Qian, S. Voros, D. Metaxas, and L. Axel. Using high resolution cardiac CT data to model and visualize patient-specific interactions between trabeculae and blood flow. In G. Fichtinger, A. Martel, and T. Peters, editors, *MICCAI*, volume 6891 of *LNCS*, pages 468–475. Springer, Heidelberg, 2011.
13. C. Lorenz and J. Berg. A comprehensive shape model of the heart. *Medical image analysis*, 10(4):657–670, 2006.
14. S. Nowozin and C. Lampert. Global connectivity potentials for random field models. In *CVPR*, pages 818–825, 2009.
15. P. Schoenhagen, A. Stillman, S. Halliburton, and R. White. CT of the heart: principles, advances, clinical uses. *Cleveland Clinic journal of medicine*, 72(2):127–138, 2005.
16. F. Ségonne, J. Pacheco, and B. Fischl. Geometrically accurate topology-correction of cortical surfaces using nonseparating loops. *TMI*, 26(4):518–529, 2007.
17. D. Shen, E. Herskovits, and C. Davatzikos. An adaptive-focus statistical shape model for segmentation and shape modeling of 3D brain structures. *TMI*, 20(4):257–270, 2001.
18. L. Spreeuwers, S. Bangma, R. Meerwaldt, E. Vonken, and M. Breeuwer. Detection of trabeculae and papillary muscles in cardiac MR images. In *Computers in Cardiology, 2005*, pages 415 –418, sept. 2005.
19. G. Sundaramoorthi and A. Yezzi. Global regularizing flows with topology preservation for active contours and polygons. *TIP*, 16(3):803–812, 2007.
20. Y. Zheng, A. Barbu, B. Georgescu, M. Scheuering, and D. Comaniciu. Four-chamber heart modeling and automatic segmentation for 3D cardiac CT volumes using marginal space learning and steerable features. *TMI*, 27(11):1668 –1681, nov. 2008.
21. S. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *PAMI*, 18(9):884–900, 1996.