Intuitive and Smart Editing of Three-Dimensional Geometric Heart Valve Apparatus Models from Cardiac CT Data

Félix Lades[1,2], Michael Wels[1], Stefan Steidl[2], and Michael Suehling[1]

[1] Siemens Healthcare GmbH, Forchheim, Germany
[2] Pattern Recognition Lab, Friedrich-Alexander University, Erlangen, Germany

Abstract. In cases of aortic valve diseases, diagnosis and subsequent interventional planning are highly facilitated by deriving exact three-dimensional geometric models of a patient's aortic valve apparatus from Cardiac Computed Tomography Angiography data. Fully-automatic approaches to do so however lack in absolute reliability and manual editing of the initially detected geometric model is often required. We therefore present an interactive editing method for this scenario – in particular for editing the aortic root model – based on the As-Rigid-As-Possible (ARAP) surface modeling paradigm, which allows efficient, robust, intuitive and physically plausible deformations of three-dimensional geometric models. The user constrains a model's surface by setting and moving handles – the so-called constraints – while the remaining, and only the remaining, free part of the surface is deformed automatically in a way such that the global shape is preserved in real-time. We extended the classical ARAP approach for our scenario by an energy smoothness regularization to overcome non-smooth artifacts at constrained positions. We furthermore incorporated the use of image evidence-based cues in the interactive workflow such that handles can be made "snap" into visible 3-D surface indicators. We evaluated our method in a user study regarding intuitiveness, achievable accuracy, inter-user variability, required time and robustness. The participants started with an initial average mesh-to-mesh surface error of 1.65 mm and achieved after 50 mouse clicks on average and less than 3.5 minutes an average error of 0.48 mm with respect to an expert-defined ground truth. The inter-user variability was 0.43 mm.

1 Introduction

Cardiac Computed Tomography Angiography (CCTA) is a widely used acquisition technique to provide insights into a patient's heart anatomy. In order to fully leverage the rich anatomical information contained in the data for interventional planning, e. g., for planning Trans-Aortic Valve Implacement (TAVI) procedures, or for further usage like 3-D printing for interventional training, it is highly desirable to use the data also for deriving exact patient-specific three-dimensional geometric models of the underlying anatomy. While fully-automatic approaches

exist, their clinical acceptance may be hampered by non-intuitive and therefore time-consuming editing tools for subsequent exact adjustment. These may even prevent the user from eventually achieving his ultimate goal: having a 3-D model of a patient's anatomy meeting the clinical needs with respect to accuracy.

We therefore present a novel solution for the associated problem of semi-automatically deriving an accurate patient-specific three-dimensional geometric model of the aortic valve apparatus – in particular of the aortic root model – from 3-D CCTA data. Our solution is a two-step approach: first a fully automatic detection is attempted, then we adopt the As-Rigid-As-Possible (ARAP) paradigm for realistic global shape editing in a new hybrid approach that additionally can take into account local shape evidence from the underlying image data. It is embedded in a user-friendly interactive real-time workflow that is equally transparent, robust and flexible with regards to the finally achieved geometric models.

Work related to our research can roughly be subdivided into three fields: 1) manual segmentation and surface reconstruction, 2) fully-automatic detection of the heart valve apparatus, and 3) segmentation refinement in medical images. On the one hand, completely manual methods like the one presented by Liu et al. [4], where a 3-D geometric model can be built from scratch by drawing contours on several individual slices, tend to be very time-consuming and challenging when trying to achieve highly accurate 3-D models. Additionally, these are not restricted by any physical or other constraints, which may yield unrealistic shapes. On the other hand, totally neglecting manual editing, Ionasec et al. [3] proposed the first fully automatic system for patient-specific modeling of the aortic valve apparatus in 4-D Cardiac CT images. The output of this detection algorithm is a set of 3-D geometric models, which are represented by triangle meshes. Grbić et al. [1] later extended the method to the whole valvular heart apparatus, containing the ascending aorta and the left and right heart valves. After the initial detection process, it could however be necessary to perform an editing step to reach better segmentation accuracy. Being a representative of segmentation refinement in medical images, Ijiri and Yokota [2] presented an interface for refining volume segmentation based on triangle meshes, which can be seen as the most comparable one to our work. The user draws several cut strokes in 3-D, which intersect the current 3-D model and generate contours on it. These contours can then be refined on 2-D cross-sectional views using a set of specific tools. Based on these new contours, the 3-D model is deformed using a global free-form surface editing method called Laplacian surface editing [7]. Ijiri and Yokota [2] applied their approach to medical image data, but not for the heart valve apparatus. Laplacian surface editing [7] can be seen as a predecessor of ARAP surface editing [6], which forms the basis of our method, and – to the best of our knowledge – has not been applied for generating and refining 3-D geometric models of the heart valve apparatus from CCTA data.

2 Methods

We consider a triangle surface mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{E}, \mathcal{F}\}$, where $\mathcal{V}$ represents the set of vertices, $\mathcal{E} \subset \mathcal{V}^2$ the set of actually non-directed edges, i. e., $(i,j) \in \mathcal{E} \Leftrightarrow (j,i) \in \mathcal{E}$, notated as a set of directed edges for convenience, and $\mathcal{F} \subset \mathcal{V}^3$ the set of triangular faces. The geometry of the surface is defined by the vertex positions $\boldsymbol{v}_i \in \mathbb{R}^3, i \in \mathcal{V}$. Surface deformation implies manipulating the geometry while preserving the topology. That means, the graph structure remains the same, while only the vertex positions are displaced.

2.1 As-Rigid-As-Possible Surface Modeling

Surface deformation is applied by moving vertex positions $\boldsymbol{v}_i, i \in \mathcal{V}$, of the initial shape $\mathcal{S}$ to new vertex positions $\boldsymbol{v}'_i$ of the deformed shape $\mathcal{S}'$. For a non-rigid deformation, Sorkine et al. [6] defined the deviation from rigidity by an energy function

$$E_{\mathrm{ARAP}}(\mathcal{S}') = \sum_{(i,j)\in\mathcal{E}} w_{ij} \|(\boldsymbol{v}'_i - \boldsymbol{v}'_j) - \boldsymbol{R}_i(\boldsymbol{v}_i - \boldsymbol{v}_j)\|^2 \tag{1}$$

where $\boldsymbol{R}_i \in \mathbb{R}^{3\times3}$ are the rotation matrices which best align the initial edge vectors $(\boldsymbol{v}_i - \boldsymbol{v}_j)$ with the deformed ones $(\boldsymbol{v}'_i - \boldsymbol{v}'_j)$. The cotangent weights $w_{ij} = \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij})$ equalize varying edge lengths, where α_{ij} and β_{ij} denote the opposite angles corresponding to the edge (i,j). Given the initial vertex positions $\boldsymbol{v}_i$ and certain constrained vertex positions $\boldsymbol{v}'_c$, ARAP surface deformation is applied by minimizing the ARAP energy, which results in solving a nonlinear optimization problem. This is applied in an iterative scheme by first solving for the best-aligning rotations, and then for the new vertex positions $\boldsymbol{v}'_i$ (please see [6] for details).

The global energy formulation of (1) is a weighted sum of squared lengths of local error (vector) terms $\boldsymbol{a}_{ij} = (\boldsymbol{v}'_i - \boldsymbol{v}'_j) - \boldsymbol{R}_i(\boldsymbol{v}_i - \boldsymbol{v}_j)$ over all edges (i,j). Let $k : \mathbb{N}^2 \to \mathbb{N}, (i,j) \to k$, be a mapping and $\boldsymbol{a}_k = \boldsymbol{a}_{ij}$. We define the matrix $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{E}|\times3}$ containing each local energy term $\boldsymbol{a}_k$ in a separate row k, and we use the diagonal edge weighting matrix $\boldsymbol{W} = \mathrm{diag}(w_1, \ldots, w_{|\mathcal{E}|})$ to reformulate (1):

$$E_{\mathrm{ARAP}}(\mathcal{S}') = \|\boldsymbol{A}\|^2_{\boldsymbol{W}} = \mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{A}) \tag{2}$$

This matrix-based reformulation was presented in a similar way by Martinez-Esturo et al. [5] who provided a triangle-centric formulation using the triangle gradient operator. We applied their formulation to the vertex one-ring representation of the original ARAP surface modeling formalism [6]. According to [5], we use the notation $\|\boldsymbol{A}\|^2_{\boldsymbol{W}}$ as defined above for a squared matrix norm that is induced by a symmetric and positive definite matrix $\boldsymbol{W}$. For the identity matrix $\boldsymbol{W} = \boldsymbol{I}$ (which corresponds to uniform weights), the defined norm is identical to the squared Frobenius norm $\|\cdot\|^2_{\mathrm{F}}$. Now we decompose the matrix $\boldsymbol{A} = \boldsymbol{E}\boldsymbol{V}' - \boldsymbol{C}$, where $\boldsymbol{E} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{V}|}$ is the edge matrix, which contains one row k for each directed edge e_k. The components are 1 in the column of the vertex index where

e_k starts, -1 for the vertex index where e_k ends, and 0 for the rest. The matrix $\boldsymbol{C} \in \mathbb{R}^{|\mathcal{E}|\times 3}$ is the matrix of rotated edges, containing the term $\boldsymbol{c}_k = \boldsymbol{R}_i(\boldsymbol{v}_i - \boldsymbol{v}_j)$ in each row k. The whole linear system can now be expressed as follows:

$$\boldsymbol{E}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{E}\boldsymbol{V}' = \boldsymbol{E}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{C} \tag{3}$$

Most surface editing methods that are based on solving a linear system like the one above are challenged by the fact that the energy always has a maximum at the constrained vertex positions. This usually yields non-smooth results. We therefore aim for smoothing their contributions to the overall energy while solving the system. Martinez-Esturo et al. [5] defined the smoothness energy as follows:

$$E_{\mathrm{SMOOTH}}(\mathcal{S}') = \|\boldsymbol{D}\boldsymbol{A}\|^2_{\boldsymbol{G}} = \|\boldsymbol{D}(\boldsymbol{E}\boldsymbol{V}' - \boldsymbol{C})\|^2_{\boldsymbol{G}} \tag{4}$$

where $\boldsymbol{G} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a diagonal matrix of edge lengths and $\boldsymbol{D} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{E}|}$ is a discrete differential operator on the local energy components $\boldsymbol{a}_k$ with elements

$$d_{ab} = \begin{cases} 1, & \text{if } e_a = (i,j) \in \mathcal{E} \wedge e_b = (k,i) \in \mathcal{E}. \\ -1, & \text{if } e_a = (i,j) \in \mathcal{E} \wedge e_b = (j,k) \in \mathcal{E}. \\ 0, & \text{otherwise.} \end{cases}$$

The definition of energy smoothness in (4) measures the squared differences of the non-rigidity energies along each directed edge, weighted by the edge length. We can now combine the non-rigidity energy of (2) and the energy smoothness of (4) and weight them with a regularization parameter $\lambda \in [0, 1[$:

$$E(\mathcal{S}') = (1-\lambda)E_{\mathrm{ARAP}}(\mathcal{S}') + \lambda E_{\mathrm{SMOOTH}}(\mathcal{S}') = \|\boldsymbol{A}\|^2_{\boldsymbol{X}}$$

with

$$\boldsymbol{X} = (1-\lambda)\boldsymbol{W} + \lambda\boldsymbol{D}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{D} .$$

According to [5], appropriate reformulation of the linear system of (3) yields:

$$\boldsymbol{E}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{E}\boldsymbol{V}' = \boldsymbol{E}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{C} \tag{5}$$

It is a system of equations with a system matrix $\boldsymbol{M} = \boldsymbol{E}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{E} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ and a right-hand side $\boldsymbol{B} = \boldsymbol{E}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{C} \in \mathbb{R}^{|\mathcal{V}|\times 3}$. Building and factorizing the system matrix leads to a non-scaling computational overhead compared to the non-smooth ARAP version. This however is negligible as the system matrix must only be built when a new vertex is constrained or freed. To incorporate the constraints into the linear system, we first have to remove all rows and columns from $\boldsymbol{M}$ which belong to a constrained vertex. We get the free part $\boldsymbol{M}_{\mathrm{F}} \in \mathbb{R}^{|\mathcal{V}'_{\mathrm{F}}|\times|\mathcal{V}'_{\mathrm{F}}|}$ and the constrained part $\boldsymbol{M}_{\mathrm{C}} \in \mathbb{R}^{|\mathcal{V}'_{\mathrm{F}}|\times|\mathcal{V}'_{\mathrm{C}}|}$ of the system matrix. Then, we have to erase the corresponding rows in the right-hand side, resulting in $\boldsymbol{B}_{\mathrm{F}} \in \mathbb{R}^{|\mathcal{V}'_{\mathrm{F}}|\times 3}$. This yields

$$\boldsymbol{M}_{\mathrm{F}}\boldsymbol{V}'_{\mathrm{F}} = \boldsymbol{B}_{\mathrm{F}} - \boldsymbol{M}_{\mathrm{C}}\boldsymbol{V}'_{\mathrm{C}},$$

where $\boldsymbol{M}_{\mathrm{C}} \in \mathbb{R}^{|\mathcal{V}'_{\mathrm{F}}|\times|\mathcal{V}'_{\mathrm{C}}|}$ is the constrained part of $\boldsymbol{M}$, and $\boldsymbol{V}'_{\mathrm{C}} \in \mathbb{R}^{|\mathcal{V}'_{\mathrm{C}}|\times 3}$ is the matrix including the constrained vertex positions as row-vectors. In Figure 1, we illustrate the decomposition of $\boldsymbol{M}$ into $\boldsymbol{M}_{\mathrm{F}}$ and $\boldsymbol{M}_{\mathrm{C}}$. On the left there is an example mesh with $\mathcal{V}'_{\mathrm{C}} = \{5, 9\}$. In the middle there is the corresponding system matrix (for simplicity it is the Laplacian matrix $\boldsymbol{L}$ as considered in [6], with uniform weights) and, on the right, there is the permuted system matrix containing $\boldsymbol{L}_{\mathrm{F}}$ as upper left part and $\boldsymbol{L}_{\mathrm{C}}$ as upper right part.

The sparse symmetric positive definite system matrix allows us to use a sparse Cholesky factorization, such that the system can be solved in an efficient manner.

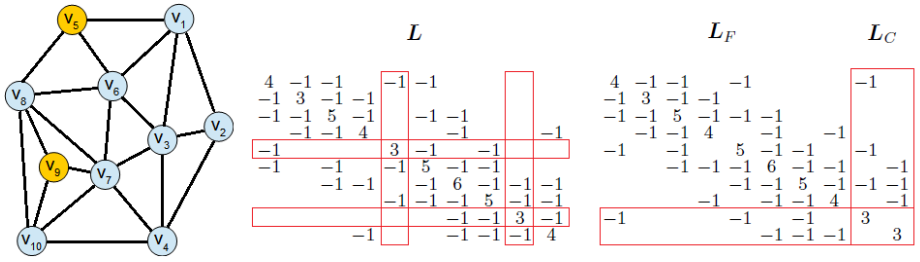

Fig. 1. Decomposition of the system matrix into free and constrained part. From left to right: an example mesh, the corresponding system matrix, and the permuted system matrix.

In Figure 2 we see in the first picture the mean shape model of the aortic root which is smaller than the actual aortic root. Therefore we used five constraints to adjust the mesh. In the second picture, ARAP editing with the parameter $\lambda = 0$, resulting in the non-smooth version with artifacts at the constrained positions, was applied. In the third picture, the smooth ARAP version with $\lambda = 0.33$ (as suggested in [5]) was applied. As can be seen, the non-smooth artifacts disappear and the mean shape model is optimally scaled with only five constraints. In the bottom row, the corresponding meshes in 3-D are shown with the non-rigidity energy color coded. Smooth ARAP yields a more even distribution of the energy over the shape surface.

2.2 Image Evidence-Based Boundary Suggestion

For our scenario, we want the surface to correlate with the image data. Therefore, we have to identify the correct object boundaries inside the image, where the handle vertices can be made "snap" into. While moving a handle vertex $\boldsymbol{v}'_i$ in the direction $\boldsymbol{d}$ of a boundary or edge inside a 3-D volume an edge may be located where the magnitude of the image gradient ∇ has a local maximum. If the sharpest boundary is desired, one may look for the highest local maximum. Therefore we sample all voxels $p_k \in \{p_1, ..., p_K\} = \mathcal{P}_K$ along the moving direction

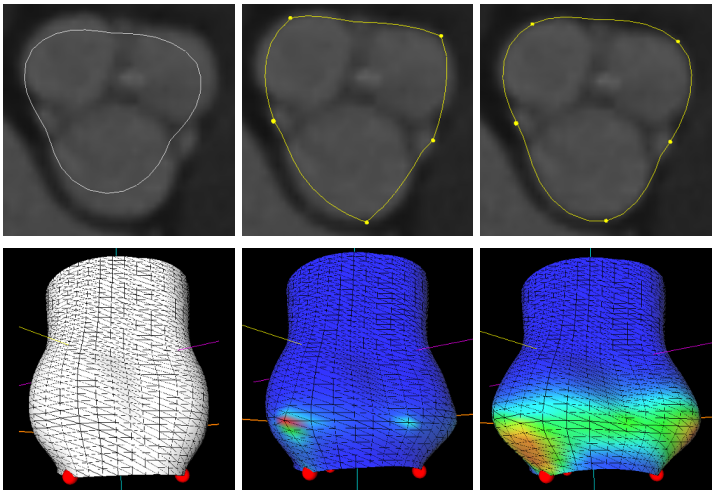

Fig. 2. Initialization of editing workflow (left), result obtained by classical ARAP (middle), and result obtained by smooth ARAP ($\lambda = 0.33$) (right).

$\boldsymbol{d}$ inside a certain observation radius R:

$$\mathcal{P}_K = \{p_k : \|\boldsymbol{v}_i - \boldsymbol{p}_k\| \leq R \wedge \exists\, l : \boldsymbol{v}_i + l\boldsymbol{d} \text{ lies inside voxel } p_k\}$$

where $\boldsymbol{p}_k$ is the world position of the center of the voxel p_k. Then, we compute for each sampled voxel the gradient magnitude $\|\nabla_k\|$. The edge aimed for is located inside the voxel $p_{\hat{k}}$ where p_k has the highest local maximum in gradient magnitude:

$$\hat{k} = \underset{k}{\operatorname{argmax}}\{\|\nabla_k\| : \|\nabla_{k-1}\| < \|\nabla_k\| \wedge \|\nabla_{k+1}\| < \|\nabla_k\|\} \tag{6}$$

Once we have the voxel center $\boldsymbol{p}_{\hat{k}}$, we have to locate the correct position $\boldsymbol{v}_i'$ inside the voxel, since we want $\boldsymbol{v}_i'$ to reside on the line $\boldsymbol{v}_i + l\boldsymbol{d}$. Therefore, we compute the orthogonal projection of the voxel center $\boldsymbol{p}_{\hat{k}}$ onto the line $\boldsymbol{v}_i + l\boldsymbol{d}$:

$$\boldsymbol{v}_i' = \boldsymbol{v}_i + \frac{(\boldsymbol{p}_{\hat{k}} - \boldsymbol{v}_i) \cdot \boldsymbol{d}}{\|\boldsymbol{d}\|}\boldsymbol{d}$$

This results in real-time "snapping" suggestions located in the moving direction of the mouse once the user has decided to start constraining an individual surface vertex. These suggestions can either be accepted or rejected if they do not meet the user's expectation.

We have used the gradient magnitude as the low-level image feature to locate object boundaries in our current system prototype. This can be extended to higher-level edge indicating features in a straightforward manner, e. g., to features computed by machine learning techniques [1] or by more sophisticated higher-order filtering techniques.

3 Experimental Setting and Results

We have evaluated our system (smooth ARAP with $\lambda = 0.33$), which has been implemented relying on MeVisLab 2.6.1 (64 Bit), on a standard notebook with Intel Core i7-4700MQ CPU (2×2.40 GHz) and 4 GB RAM. Regarding performance of our system we worked on an aortic root mesh with 1440 nodes. After fully-automatic detection of the initial geometric model [1] our editing method could be initialized in less than 100 ms and response time to a user input was less then 10 ms making our prototype a real-time system. Setting and moving the constraints was performed in best fitting short-axis views (see top row of Figure 2), exploiting the regular, sliced structure of the mesh.

In order to evaluate achievable accuracy, required editing time, and inter-user variability of our new editing workflow we used six data sets depicting the appropriate heart phase. In cases the automatic detection was already sufficiently good (i.e. for DS 1-5), we replaced the detected model by a mean shape model to simulate initial deviation. For all the six data sets we carried out a user study, where four non-expert respondents had to edit the aortic root mesh until they felt satisfied with the achieved result. In all cases the initial average mesh-to-mesh error could be decreased from 1.64 mm to 0.48 mm with 50 mouse clicks within 207 s on average. The maximum mesh-to-mesh error could be decreased from 7.74 mm to 2.06 mm. The inter-user variability was 0.43 mm. The low initial value of 1.64 mm at the beginning of our editing may suggest that rigid initialization is already sufficiently good. From a clinical perspective it however is not good enough as Figure 3 and the maximum mesh-to-mesh error show. Figure 3 shows some examples for one of the data sets. In the first row, we see the initialized mean shape mesh (from left to right: short-axis, long-axis, and 3-D view), in the second row we see one of the results after editing.

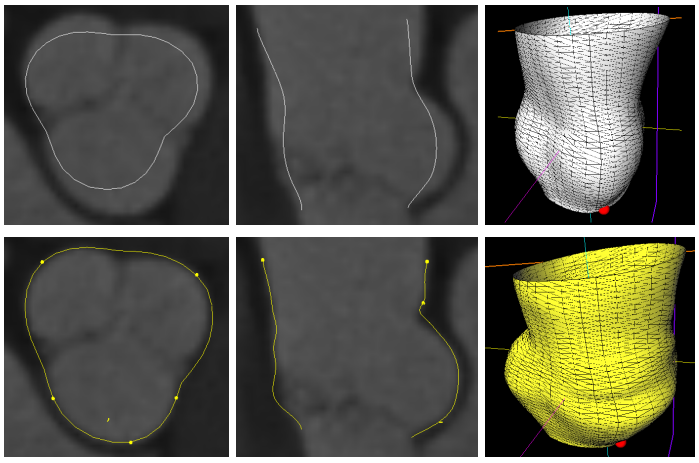

Fig. 3. Examples: initialization (top) and results after editing (bottom).

Following usual practice we used ground-truth annotations from a purely manual tool for our experiments. This way we avoided a possible bias regarding the amount of shape detail that can be handled by our ARAP-based method. Due to its lack of intuitiveness and usability the manual tool is likely to lead to higher inter-/intra-user variability and more inaccuracies than our approach. This strengthens the fact that we cannot achieve maximum accuracy in our experiments. However, our findings suggest that our approach may lead to even more accurate ground-truth annotations if used appropriately in this scenario.

4 Conclusions

We presented a method for interactive, semi-automatic, and image information-assisted editing of 3-D geometric models, which are fully-automatically detected in an initial step, and applied it to accurately modeling the aortic root from CCTA data. We used the concept of ARAP surface editing to deform a surface while preserving its global shape. To also provide smooth results, we extended the classical ARAP surface editing formalism by a smoothing regularization. We further incorporated image information to provide "snapping" suggestions, which are associated with surface cues to be found in the data at hand. We evaluated our interactive editing workflow in a user study, where it was shown to be intuitive, fast and robust. Our method allows experts to derive exact geometric models of a patient's aortic valve apparatus in terms of the surrounding aortic root including the interventionally relevant aortic annulus with reduced effort. This is crucial for diagnosis and interventional planning of aortic valve diseases.

References

1. S. Grbić, R. I. Ionasec, D. Vitanovski, I. Voigt, Y. Wang, B. Georgescu, N. Navab, and D. Comaniciu. Complete valvular heart apparatus model from 4D cardiac CT. In *Int. Conf. Med. Image Comput. Comput.-Assist. Interv., Beijing, China*, pages 218–226. Springer Berlin Heidelberg, 2010.
2. T. Ijiri and H. Yokota. Contour-based interface for refining volume segmentation. In *Comput. Graph. Forum*, volume 29, pages 2153–2160, 2010.
3. R. I. 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 *Int. Conf. Med. Image Comput. Comput.-Assist. Interv., New York, USA*, pages 686–694. Springer Berlin Heidelberg, 2008.
4. L. Liu, C. Bajaj, J. O. Deasy, D. A. Low, and T. Ju. Surface reconstruction from non-parallel curve networks. In *Comput. Graph. Forum*, volume 27, pages 155–163, 2008.
5. J. Martinez Esturo, C. Rössl, and H. Theisel. Smoothed quadratic energies on meshes. *ACM Trans. Graph.*, 34(1):2:1–2:12, 2014.
6. O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *EUROGRAPHICS Symp. Geom. Process., Barcelona, Spain*, pages 109–116, 2007.
7. O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *EUROGRAPHICS Symp. Geom. Process., Nice, France*, pages 179–188, 2004.