

Article

Rapid Interactive and Intuitive Segmentation of 3D Medical Images Using Radial Basis Function Interpolation [†]

Tanja Kurzendorfer ^{1,2,*}, Peter Fischer ^{1,2}, Negar Mirshahzadeh ¹, Thomas Pohl ², Alexander Brost ², Stefan Steidl ¹ and Andreas Maier ¹

¹ Pattern Recognition Lab, FAU Erlangen-Nuremberg, 91058 Erlangen, Germany; peter.fischer@fau.de (P.F.); n.mirshahzadeh@gmail.com (N.M.); stefan.steidl@fau.de (S.S.); andreas.maier@fau.de (A.M.)

² Siemens Healthcare GmbH, 91301 Forchheim, Germany; thomas.tp.pohl@siemens-healthineers.com (T.P.); alexander.brost@siemens-healthineers.com (A.B.)

* Correspondence: tanja.kurzendorfer@fau.de

[†] This paper is an extended version of our paper published in Annual Conference on Medical Image Understanding and Analysis.

Received: 18 October 2017; Accepted: 28 November 2017; Published: 30 November 2017

Abstract: Segmentation is one of the most important parts of medical image analysis. Manual segmentation is very cumbersome, time-consuming, and prone to inter-observer variability. Fully automatic segmentation approaches require a large amount of labeled training data and may fail in difficult or abnormal cases. In this work, we propose a new method for 2D segmentation of individual slices and 3D interpolation of the segmented slices. The Smart Brush functionality quickly segments the region of interest in a few 2D slices. Given these annotated slices, our adapted formulation of Hermite radial basis functions reconstructs the 3D surface. Effective interactions with less number of equations accelerate the performance and, therefore, a real-time and an intuitive, interactive segmentation of 3D objects can be supported effectively. The proposed method is evaluated on 12 clinical 3D magnetic resonance imaging data sets and are compared to gold standard annotations of the left ventricle from a clinical expert. The automatic evaluation of the 2D Smart Brush resulted in an average Dice coefficient of 0.88 ± 0.09 for the individual slices. For the 3D interpolation using Hermite radial basis functions, an average Dice coefficient of 0.94 ± 0.02 is achieved.

Keywords: smart brush, segmentation, 3D interpolation, Hermite radial basis function

1. Introduction

Segmentation is a common task in the processing of medical images. It is often a pre-requisite for the further image analysis and can be used for therapy planning and guidance [1]. The spectrum of segmentation techniques available to the clinical applications is broad, ranging from manual slice by slice outlining to fully automatic segmentation. Manual segmentation is still widely used for complex segmentation tasks. For example, in structural heart disease, as every heart has a different structure, no learning based approach can be applied. However, manual annotation of every image slice can be very cumbersome and time-consuming, considering the high resolution of the 3D image volumes [2]. A great deal of effort has gone into interactive segmentation tools for 2D segmentations as well as 3D interpolations. Many segmentation techniques have been developed such as Intelligent Scissors, Graph Cuts, and Random Walker [3–5]. There are two important applications that these techniques can speed up. First, manual segmentation, which is still widespread in clinical routine. Second, the generation of ground truth annotations that have to be generated manually, in order to train machine learning algorithms. In particular, deep learning is known to require huge amounts of

annotated data. Furthermore, there are only a few interactive deep learning approaches [6]. Therefore, the challenge is to design a fast, generic, and easy segmentation tool that allows for generating clinical segmentations as well as fast ground truth annotations, in both 2D and 3D medical images and all modalities. The most related 2D segmentation technique is a Smart Brush tool [7,8]. However, the drawback of these methods is that they do not control the boundary smoothness [9].

In surface reconstruction, there is a vast literature that is mainly grouped into direct meshing and implicit approaches. Nowadays, methods based on implicit surface reconstruction are gaining more and more attention. For this approach, first a signed scalar field $f(\cdot)$ is obtained. The value of this scalar field is zero at all control points \mathbf{p} , $f(\mathbf{p}) = 0$, and negative/positive for inside/outside of the surface [10]. Then, the desired surface is reconstructed by extracting the zero-level set of the mentioned field. The radial basis function (RBF) interpolation guarantees having a smooth field, from non-uniformly distributed data points [11–14]. In previous related work [14], this field $f(\cdot)$ is computed in a bilateral domain where the spatial and intensity range domain are joined. The interpolation is done using RBFs with Hermite data, which incorporates normals and gradients of the scalar field directly, $\nabla f(\mathbf{p}) = \mathbf{n}$.

In this work, we propose a new formulation of surface reconstruction that is independent of the 3D intensity gradient information and can make use of both 2D and 3D normal vectors obtained from segmented 2D slices. Therefore, the user input is the interactive segmentation of 2D slices. Hence, a Smart Brush formulation is introduced that can handle medical acquisitions with higher noise level and ambiguous boundaries using adaptive thresholding. From the extracted contour of the segmentation mask, control points are obtained and their normal vectors are estimated based on the curvature of the contour. Furthermore, control points at the intersections of annotated planes from different orientations are fused and the corresponding normals are combined into a 3D normal vector. From this, it follows that the surface is reconstructed using both 2D and 3D normal vectors. In contrast to previous implicit methods, the method is superior for images with a high noise level, as it does not depend on intensity information or well defined borders for the 3D interpolation.

2. Methods

Our approach combines advantages of semi-automatic segmentation methods, as well as the user's high-level anatomical knowledge to generate segmentations quickly and accurately with fewer interactions. Using our method, the user first segments a few slices with the Smart Brush; then, the scattered data points are extracted and the 2D normal information of the annotated slices is computed. Applying our new formulation of Hermite radial basis function (HRBF), the desired surface is reconstructed. In Figure 1, the segmentation pipeline is illustrated.

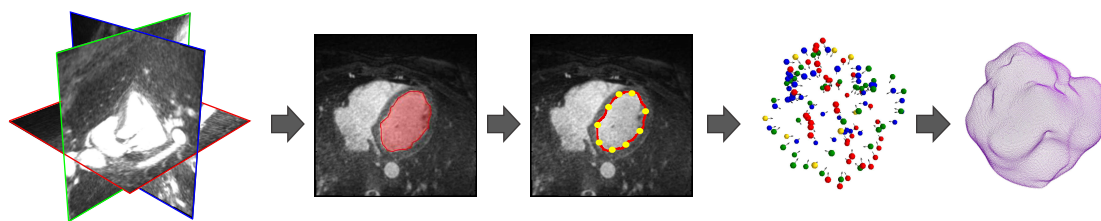


Figure 1. Segmentation pipeline. The first image from the left shows the 3D volume as input. In the next step, single slices are segmented using the Smart Brush functionality. Third, the control points of the contours are extracted. Fourth, the 2D and 3D normal vectors are computed for the Hermite radial basis function interpolation. In the final image, the interpolated surface is visualized ([2] Reproduced with permission).

2.1. Smart Brush

The 2D segmentation functionality classifies pixels into foreground and background based on the intensity information. Prior to the segmentation, pre-processing is required for magnetic resonance images (MRI), as the intensity values are in arbitrary units. Therefore, a normalization is required. For this aim, an interval $I = [v_{\min}, v_{\max}]$ is defined based on the maximum and minimum intensity of the whole DICOM data set $\mathbf{D} \in \mathbb{R}^{w \times h \times l}$. Then, for each 2D image slice, the values outside the interval I are clipped to the interval borders. To have a gray-scale image, the minimum value of the clipped image is set to zero and the maximum to 255. In the next step, to reduce the existing noise, an edge-preserving, denoising filter called bilateral filter is used to suppress the noise in the normalized image. In this method, spatial closeness and radiometric similarity are measured by the Gaussian function of the Euclidean distance between two pixel intensities [15].

The smart brush segmentation is user driven and the interactions start with each displacement of the mouse cursor. The segmentation scheme comprises the following steps: (i) manually initializing a small part of the region of interest (ROI); (ii) improving the segmentation using the Smart Brush functionality; (iii) post-processing the segmentation result and extending the previously segmented region.

First, a small area of the region of interest has to be segmented manually by the user. The mean intensity of this area is required for the initialization of the smart brush functionality. When the user selects a new ROI with the brush, an adaptive threshold is computed using the mean intensity of the ROI $I_{\text{mean}} = \frac{\sum_{i=1}^N I_i}{N}$, where N is the number of pixels in the selected area \mathbf{A}_0 . Afterwards, the user progresses with the smart brush and a new area is selected. For the new selected area, the intensity distribution is investigated. A threshold for pixel-wise classification is derived for the mean values. The pixels of the component whose mean is closer to the mean intensity of the initial area are classified as foreground. Finally, to reduce false positives, the morphological connectivity of each pixel in the ROI to the initial ROI is checked using a 4-connected structuring element. This way, pixels that have the same intensity value but are not connected to the previous segmentation are removed. In Figure 2a, the initialization of the smart brush is shown in red and the brush is moved over a new area, illustrated by the yellow circle. In Figure 2b, the correct segmented area is visualized, using the adaptive thresholding together with the propagation checking. If no propagation checking would be applied for this example, also the right ventricle would be segmented, as it has the same intensity values. In the final step, to get a smooth looking contour, the wholes are filled using binary hole filling.

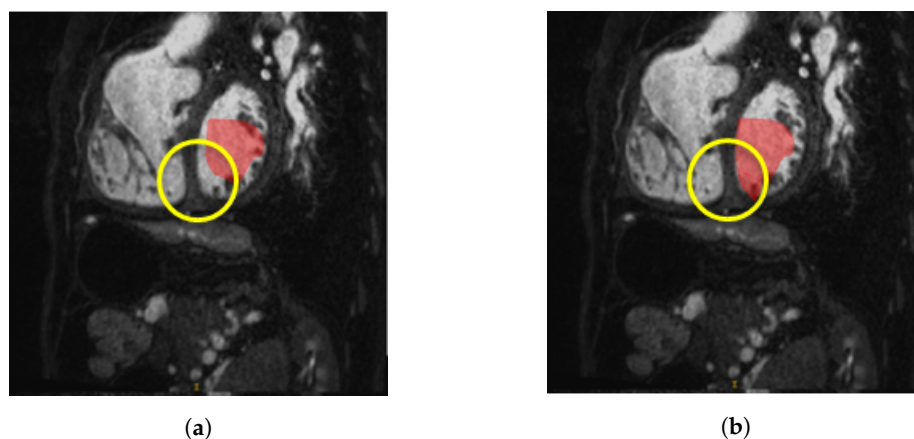


Figure 2. (a) Initialization of the smart brush in red and the smart brush is propagated which is illustrated as yellow circle; (b) Correct segmented area under the brush using adaptive thresholding and propagation checking.

2.2. Control Point Extraction

We assume that multiple slices are segmented in axial, sagittal, and coronal orientation using the smart brush functionality. First, the contours are extracted from the segmentations of the individual slices. Given a structural element, a morphological operation named binary erosion is used to extract the boundary. The structuring element is a square connectivity equal to one. Then, by subtracting the eroded mask from the original mask and applying a threshold, a one-pixel thick edge is extracted. In the next step, the control points (CPs) are computed from the contours adaptively according to the shape of the object.

The contour is sampled equidistantly with a predefined sampling size $\delta \in \mathbb{Z}$. The number of control points $n_e \in \mathbb{N}$ is based on the contour length $l_c \in \mathbb{N}$ and computed as $n_p = \lfloor \frac{l_c}{\delta} \rfloor$. Furthermore, $n_c \in \mathbb{N}$ convexity defect points, where the contour has the maximum distance to its convex hull are added (see the blue points in Figure 3a). To increase the accuracy of the 3D interpolation for complex objects, the number of CPs is increased in rough areas. Therefore, the local curvature $\kappa \in \mathbb{R}$ is checked for all CPs and additional points are added in case of roughness. The curvature κ is dependent on the derivatives of the curve $\mathbf{c}(t) = (x(t), y(t))^T$,

$$\kappa = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}}, \quad (1)$$

where the primes refer to derivatives $\frac{d}{dt}$ with respect to the parameter t . To compare curvature values, a reference quantity $r \in \mathbb{R}$ (global roughness) is defined, which is the ratio of the convex hull area of the curve A_h and the curve area A_c , $r = A_c / A_h$ [16]. New CPs are added at a certain distance to the investigated CP, if the criterion

$$\frac{\kappa}{r} \geq \theta_r \quad (2)$$

is fulfilled, where the threshold $\theta_r \in \mathbb{R}$ is obtained heuristically. The number of additional CPs due to curvature is denoted as n_κ . The total number of CPs is $N = n_e + n_c + n_\kappa$. Figure 3b depicts the total number of extracted control points with the convexity defect points in blue and the additional rough surface points in green.

The subsequent interpolation requires Hermite data, i.e., function values and their derivatives. In this case, we need the normal vector for each control point. The first derivative of the contour approximates the tangent vector of the curve. Having the 2D tangent vector $\mathbf{t} = (d_x, d_y)^T$, the orthogonal normal vector is obtained by $\mathbf{n} = (-d_y, d_x)^T$.

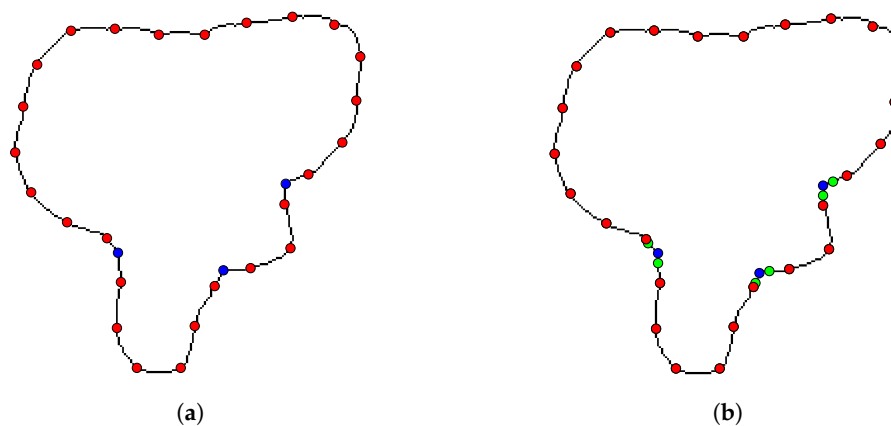


Figure 3. (a) A rough surface with initial equidistant points in red and convexity defect points in blue; (b) A rough surface with increased number of points in green ([2] Reproduced with permission).

2.3. Control Point Merging

The CPs are used to interpolate the 3D surface belonging to one object of interest. To gain more accuracy, it is always better to have cross sections from different orientations of the desired object. Considering a 3D object, the intersection of any two non-parallel image planes will result in a line with at least two intersection points. Figure 4 shows the location of these two points in yellow. It implies that there are joint points in case the selected planes intersect. However, the extracted contour needs to be identical for both planes at the point of intersection to result in the same point in 3D space. In practice, the contour of the segmented mask may not be located precisely at the actual object border, as the annotation is done manually by the user. Consequently, there is no intersection point in 3D. Hence, instead of having one point at each junction, there will be two points at each intersection, corresponding to the annotated planes. These points can then lead to incorrect 3D interpolations, as they have conflicting gradient and zero-level set information. As a result, unwanted holes may appear in the final interpolation result. To prevent this artifact, all possible intersections of cross sections must be detected. Therefore, points in a certain radius r_d are merged into a single 3D point. Apart from eliminating undesired artifacts, merging CPs has another advantage because the 3D normal vector can be computed and this will further improve the accuracy of the 3D interpolation. The calculation of the intersection points and their normal vectors is explained in the following section.

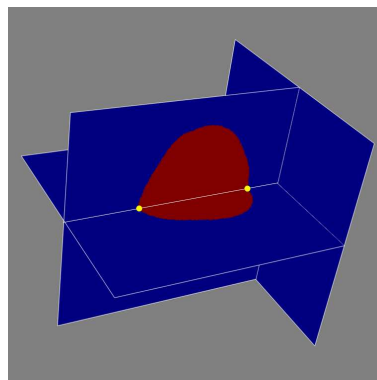


Figure 4. Two orthogonal planes are segmented in red, and the resulting intersection points are depicted in yellow.

2.3.1. Contour Intersection

Since the volumetric 3D image is segmented in arbitrary 2D slices, intersections between segmented slices from different orientations occur. Supposing that closed contours are extracted from the intersecting slices, the intersection should result in two 3D points. The computation of these intersection points is performed iteratively. The iterations are over all non-repetitive 2-permutations of N segmented slices and are comprised of two steps. First, the existence of intersections is checked. In the case of an intersection, the corresponding points are extracted. In some complex objects, i.e., non-convex shapes, more than two intersection points or a set of neighboring points can be extracted. Figure 5 shows two possible cases for extracted intersection points. In the case of multiple intersection candidates, classification is applied in order to distinguish between the different groups of points. In the next step, the average of each group is taken as a final 3D intersection point. The classification of the point groups is described in more details in the following section.

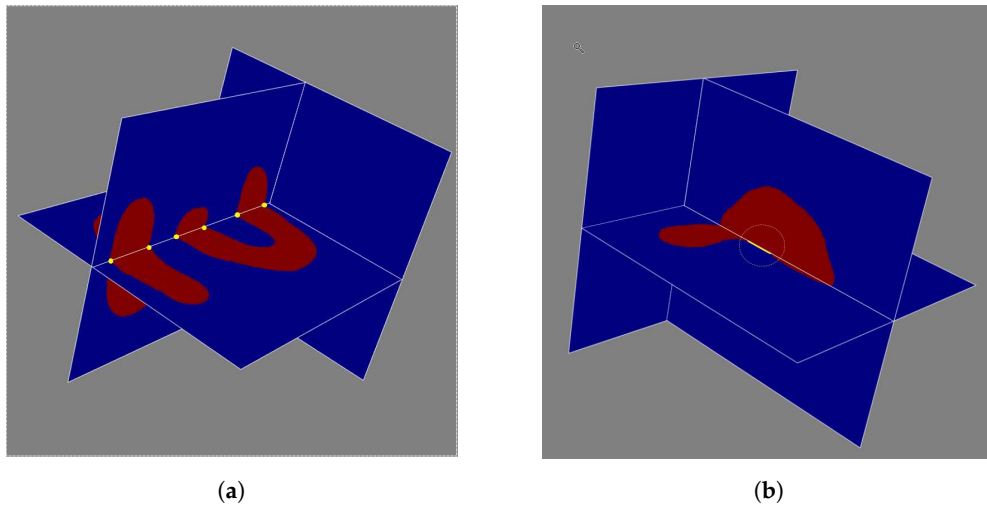


Figure 5. Possible intersection points of annotated non-parallel image slices. (a) The intersection occurs in multiple points; (b) the intersection occurs in the form of one line.

2.3.2. Classification and Merging

Having found the intersection point candidates at each junction, the next step is to merge close intersection points in order to decrease the redundancy. To obtain the neighboring points, a nearest neighbor graph within a given radius r_n is applied to the set of intersection point candidates [17]. According to the size and the complexity of the desired object, the user can change the search radius r_n for the neighboring points. The merging of the 3D points is simply done by averaging, resulting in one 3D point. The merging of the normal vectors is performed such that the final result is a unit vector again. Averaging of a collection of three-dimensional points is described as:

$$\bar{\mathbf{p}} = \frac{1}{m} \sum_{i=1}^m \mathbf{p}_i = \left(\frac{1}{m} \sum_{i=1}^m x_i, \frac{1}{m} \sum_{i=1}^m y_i, \frac{1}{m} \sum_{i=1}^m z_i \right)^T, \quad (3)$$

where \mathbf{p}_i refers to the intersection point candidates in one neighborhood and m is the number candidates. To have a unit vector, the merging is done as

$$\bar{\mathbf{n}} = \left(\frac{1}{N(n_{i,x})} \sum_{i=1}^m n_{i,x}, \frac{1}{N(n_{i,y})} \sum_{i=1}^m n_{i,y}, \frac{1}{N(n_{i,z})} \sum_{i=1}^m n_{i,z} \right)^T, \quad (4)$$

$$\bar{\mathbf{n}}_{\text{normalized}} = \frac{\bar{\mathbf{n}}}{\|\bar{\mathbf{n}}\|}, \quad (5)$$

where operator N counts the number of valid elements, i.e., the intersection points that have an in-plane normal for this dimension. In Figure 6, all the control points extracted from the different image planes are shown, where the control points with a 2D normal vector are visualized in green and all control points with a 3D normal vector are shown in blue.

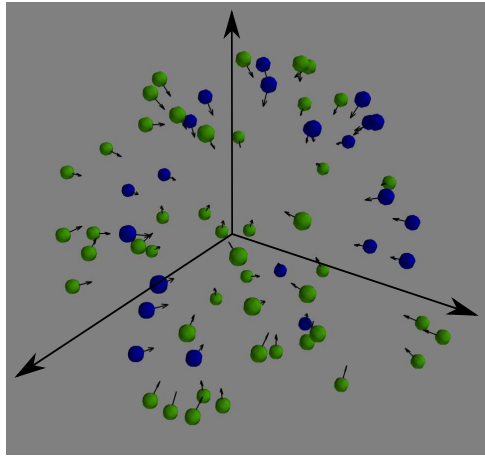


Figure 6. Control points extracted from three different orientations, where N points have a 2D normal vector (green) and M points with 3D normal vector (blue).

2.4. 3D Interpolation

For the 3D interpolation of the scalar field, radial basis functions are used. The RBF function interpolation depends only on the distance of the center \mathbf{x} to a point \mathbf{p}_i [10],

$$\varphi(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{p}_i\|) \quad , \quad (6)$$

where φ is a nonlinear activation function and \mathbf{p}_i is an extracted control point. If we consider N radial basis functions around every control point \mathbf{p}_i , we end up with a system of linear equations

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{p}_i\|) \quad , \quad (7)$$

where α_i is a weighting factor for each control point. To make sure that the equation is always solvable, a low-degree polynomial $g(\mathbf{x})$ is added

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{p}_i\|) + g(\mathbf{x}) \quad . \quad (8)$$

However, this simple RBF formulation requires the definition of inside and outside values. To address this issue, Hermite data is incorporated into the RBF, which directly use derivatives. This method ensures the existence of a non-null implicit surface without the need of additional information [18]. Using the first order Hermite interpolation in combination with RBF, the scalar field can be formulated as follows:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \varphi(\|\mathbf{x} - \mathbf{p}_i\|) - \beta_i \nabla \varphi(\|\mathbf{x} - \mathbf{p}_i\|) + g(\mathbf{x}) \quad , \quad (9)$$

where $\beta_i \in \mathbb{R}^3$ is a weighting factor.

In this work, a new formulation of HRBF is introduced that allows for reconstructing the 3D surface based on scattered control points and their associated 2D and 3D normal vectors. Assume that N Hermite data points $\{(\mathbf{p}_i, \mathbf{n}_i) | \mathbf{p}_i \in \mathbb{R}^3, \mathbf{n}_i^{2D} \in \mathbb{R}^2, i = 1, \dots, N\}$ with a 2D normal vector and M Hermite data points $\{(\mathbf{p}_i, \mathbf{n}_i) | \mathbf{p}_i \in \mathbb{R}^3, \mathbf{n}_i \in \mathbb{R}^3, i = N + 1, \dots, N + M\}$ with a 3D normal vector are generated, where $\mathbf{p}_i \in \mathbb{R}^3$. In RBF interpolation, the final segmentation is given as the zero level set of a scalar field. The scalar field f consists of two components $f = f^{2D} + f^{3D}$. The scalar field f^{2D} for the 2D normal vectors is formulated as

$$f^{2D}(\mathbf{x}) = \sum_{i=1}^N \alpha_i^{2D} \varphi(\|\mathbf{x} - \mathbf{p}_i\|) - \beta_i^{2D^T} s_i^{2D} (\nabla \varphi(\|\mathbf{x} - \mathbf{p}_i\|)) + g(\mathbf{x}) , \quad (10)$$

where $g(\mathbf{x})$ is a low-degree polynomial, $s_i^{2D}(\mathbf{x})$ is a function that selects the 2D gradient direction that is available for control point \mathbf{p}_i , and $\alpha_i^{2D} \in \mathbb{R}$, $\beta_i^{2D} \in \mathbb{R}^2$ are the RBF coefficients. The scalar field f^{3D} for the 3D normal vectors is formulated accordingly:

$$f^{3D}(\mathbf{x}) = \sum_{i=N+1}^{N+M} \alpha_i^{3D} \varphi(\|\mathbf{x} - \mathbf{p}_i\|) - \beta_i^{3D^T} \nabla \varphi(\|\mathbf{x} - \mathbf{p}_i\|) + g(\mathbf{x}) , \quad (11)$$

where $g(\mathbf{x})$ is a low-degree polynomial and $\alpha_i^{3D} \in \mathbb{R}$, $\beta_i^{3D} \in \mathbb{R}^3$ are the RBF coefficients. A 3D gradient selection function similar to $s_i^{2D}(\mathbf{x})$ is not necessary, since all dimensions are specified by the 3D normals. According to previous work [14], the commonly used tri-harmonic kernel $\varphi(t) = t^3$, $t \in \mathbb{R}$, with a linear polynomial $g(\mathbf{x}) = \mathbf{a}^T \mathbf{x} + b$ yields adequate results in terms of shape aesthetics. To determine the coefficients α_i^{2D} , α_i^{3D} , β_i^{2D} , and β_i^{3D} constraints are derived from the CPs [14]:

$$f(\mathbf{p}_i) = 0, \quad (12)$$

$$s_i^{2D}(\nabla f(\mathbf{p}_i)) = \mathbf{n}_i^{2D}, \quad (13)$$

$$\nabla f(\mathbf{p}_i) = \mathbf{n}_i . \quad (14)$$

In addition, the orthogonality conditions

$$\sum_{i=1}^N \alpha_i^{2D} = 0, \quad (15)$$

$$\sum_{i=N+1}^{N+M} \alpha_i^{3D} = 0, \quad (16)$$

$$\sum_{i=1}^N \alpha_i^{2D} s_i^{2D}(\mathbf{p}_i) + \beta_i^{2D} = 0, \quad (17)$$

$$\sum_{i=N+1}^{N+M} \alpha_i^{3D} \mathbf{p}_i + \beta_i^{3D} = 0 , \quad (18)$$

have to be fulfilled [14]. These constraints yield a linear system of equations represented in the matrix form as

$$\begin{bmatrix} 0 & \mathbf{S}_1^T & \cdots & \mathbf{S}_M^T & \mathbf{S}_{M+1}^T & \cdots & \mathbf{S}_{M+N}^T \\ \mathbf{S}_1 & \mathbf{K}_{1,1} & \cdots & \mathbf{K}_{1,M} & \mathbf{K}_{1,M+1} & \cdots & \mathbf{K}_{1,M+N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_M & \mathbf{K}_{M,1} & \cdots & \mathbf{K}_{M,M} & \mathbf{K}_{M,M+1} & \cdots & \mathbf{K}_{M,M+N} \\ \mathbf{S}_{M+1} & \mathbf{K}_{M+1,1} & \cdots & \mathbf{K}_{M+1,M} & \mathbf{K}_{M+1,M+1} & \cdots & \mathbf{K}_{M+1,M+N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_{M+N} & \mathbf{K}_{M+N,1} & \cdots & \mathbf{K}_{M+N,M} & \mathbf{K}_{M+N,M+1} & \cdots & \mathbf{K}_{M+N,M+N} \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_M \\ \mathbf{w}_{M+1} \\ \vdots \\ \mathbf{w}_{M+N} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_M \\ \mathbf{c}_{M+1} \\ \vdots \\ \mathbf{c}_{M+N} \end{bmatrix} , \quad (19)$$

where different colors in the matrix imply the points and the corresponding normal vectors with different dimensionality (3D blue, 2D green, mixed purple). The linear systems of equations can be also written as $\mathbf{D}\mathbf{X} = \mathbf{B}$, with $\mathbf{D} \in \mathbb{R}^{(M+N) \times (M+N)}$, $\mathbf{X} \in \mathbb{R}^{M+N}$ and $\mathbf{B} \in \mathbb{R}^{M+N}$.

The blue block describes the constraints on the 3D variables α_i^{3D} , β_i^{3D} derived from the 3D constraints and orthogonality conditions Equations (12), (14), (16) and (18). Thus, the matrices $\mathbf{K}_{i,j}$, \mathbf{S}_i and the vectors \mathbf{s} , \mathbf{w}_i , and \mathbf{c}_i are defined as:

$$\mathbf{K}_{i,j} = \begin{bmatrix} \varphi(\|\mathbf{p}_i - \mathbf{p}_j\|) & -\nabla\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|)^T \\ \nabla\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|) & -\mathbf{H}\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|) \end{bmatrix}, \quad \mathbf{S}_i = \begin{bmatrix} \mathbf{p}_i^T & 1 \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad (20)$$

$$\mathbf{s} = \begin{bmatrix} \mathbf{a} \\ b \end{bmatrix}, \quad \mathbf{w}_i = \begin{bmatrix} \alpha_i^{3D} \\ \beta_i^{3D} \end{bmatrix}, \quad \mathbf{c}_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{n}_i \end{bmatrix},$$

where $\mathbf{I} \in \mathbf{R}^{3 \times 3}$ is a unit matrix and $\mathbf{H}\varphi \in \mathbf{R}^{3 \times 3}$ is the Hessian matrix of the kernel φ , which arises due to the normal constraint Equation (14) applied to $\nabla\varphi$. The green block describes the constraints on the 2D variables $\alpha_i^{2D}, \beta_i^{2D}$ derived from the 2D constraints and orthogonality conditions Equations (12), (13), (15) and (17). Thus, the matrices $\mathbf{K}_{M+i,M+j}$ and \mathbf{S}_{M+i} and the vectors \mathbf{w}_{M+i} and \mathbf{c}_{M+i} are defined as:

$$\mathbf{K}_{M+i,M+j} = \begin{bmatrix} \varphi(\|\mathbf{p}_i - \mathbf{p}_j\|) & -s_i^{2D} (\nabla\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|))^T \\ s_i^{2D} (\nabla\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|)) & -s_i^{2D} (\nabla^T s_i^{2D} (\nabla\varphi(\|\mathbf{p}_i - \mathbf{p}_j\|))) \end{bmatrix}, \quad \mathbf{S}_{M+i} = \begin{bmatrix} s_i^{2D} (\mathbf{p}_i^T) & 1 \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad (21)$$

$$\mathbf{w}_{M+i} = \begin{bmatrix} \alpha_i^{2D} \\ \beta_i^{2D} \end{bmatrix}, \quad \mathbf{c}_{M+i} = \begin{bmatrix} \mathbf{0} \\ \mathbf{n}_i^{2D} \end{bmatrix}.$$

The mixed blocks are defined analogously. There is always a unique solution to the system of equations, if the points \mathbf{p}_i are pairwise distinct [14,19].

Considering the basis function of the tri-harmonic kernel $\varphi(t) = t^3$, the gradient and the Hessian matrix of the kernel φ is denoted as follows:

$$\nabla\varphi(t) = 3 \|t\| t, \quad (22)$$

$$\mathbf{H}\varphi = \begin{cases} 0 & \text{if } \|t\| = 0 \\ \frac{3tt^T}{\|t\|} + 3 \|t\| \mathbf{I}_k & \text{otherwise} \end{cases}$$

where $\mathbf{I}_k \in \mathbf{R}^{k \times k}$ is a unit matrix. To solve the linear system of equations, it is assumed that all the $M + N$ points have 3D normal vectors. Therefore, the linear system has the size of $3(M + N + 1) \times 3(M + N + 1)$.

The unknown parameters $\alpha_i^{2D}, \alpha_i^{3D}, \beta_i^{2D}, \beta_i^{3D}, \mathbf{a}$ and b can be obtained directly as the matrix is square and non-singular:

$$\mathbf{DX} = \mathbf{B}, \quad (23)$$

$$\mathbf{X} = \mathbf{D}^{-1} \mathbf{B}.$$

The parameter α_i is the weight of each RBF at its center \mathbf{p}_i and β_i is the weight of the normal vector at the same center. The next step is to extract the 3D surface, which is presented in the next section.

2.5. Surface Reconstruction

As mentioned in the introduction, the RBF surface can be interpreted as an implicit surface, as depicted in Figure 7. Therefore, after interpolating the scalar field, in order to get the 3D surface, the zero level of the scalar field has to be extracted. For this aim, the level set method is used and the zero level set is extracted. Using this method, no a priori knowledge is required about the topology of the reconstructed shape. In general, the level set c_0 at time t of a function $\psi(x, y, t)$ is the set of arguments $\{(x, y), \psi(x, y, t) = c_0\}$. In the zero level set, the idea is to define a function $\psi(x, y, t)$ such that, at any time,

$$\gamma(t) = \{(x, y), \psi(x, y, t) = 0\}. \quad (24)$$

The function ψ has many other level sets, in addition to γ , while only γ has a meaning for the segmentation and not for any other level sets of ψ . A very commonly chosen ψ is the signed distance to the front $\gamma(0)$ given as

$$\psi(x, y, 0) = \begin{cases} -d(x, y, 0) & \text{if } (x, y) \text{ inside the front} \\ 0 & \text{if } (x, y) \text{ on the front} \\ d(x, y, 0) & \text{if } (x, y) \text{ outside the front} \end{cases} . \quad (25)$$

The level set method segments the surface iteratively. In the first step, the front $\gamma(0)$ is initialized at a certain position. The second step is to compute $\psi(x, y, 0)$ and then iterate over until convergence:

$$\psi(x, y, t + 1) = \psi(x, y, t) + \Delta\psi(x, y, t) . \quad (26)$$

Lastly, the $\gamma(t_{end})$ is marked as a desired extracted surface.

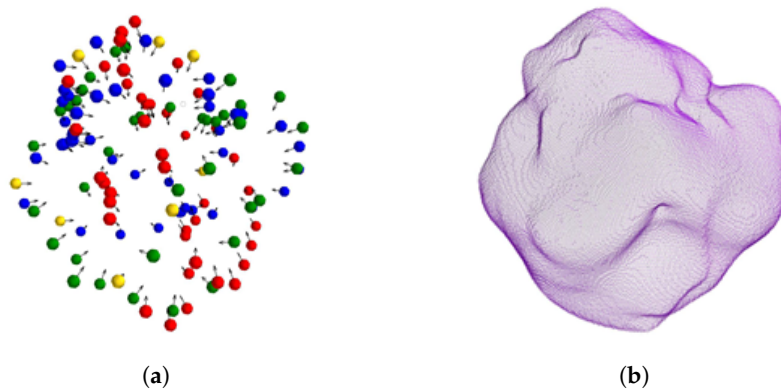


Figure 7. (a) 3D Interpolation using radial basis function (RBF); (b) implicit surface of the RBF.

3. Evaluation and Results

The evaluation was performed on 12 MRI data sets. The data was acquired with a 1.5 T MAGNETOM Aera scanner (Siemens Healthcare GmbH, Erlangen, Germany). Gold standard annotations of the left ventricle were provided by a clinical expert. The Dice coefficient was evaluated as a quantitative score for the segmentation overlap.

3.1. Smart Brush Evaluation

The 2D ground truth annotations were used to assess the 2D segmentation and the complete 3D ground truth for the 3D interpolation scheme. The main problem with evaluating the Smart Brush is that it inherently involves human interaction. Furthermore, there are many parameters that affect the result of the 2D segmentation, such as the size of the brush or the initialization step. Therefore, objective testing without human interaction is difficult. To address this, we mimicked user interactions such as slice selection, mouse movement, brush size, etc. Iteratively, a 2D slice was selected and one patch of the ground truth was for the initialization of the brush. The evaluation of the Smart Brush was performed on a different patch by computing the Dice coefficient per patch. As it is difficult to test all the parameters, we evaluate the performance of the proposed method with a fixed brush size and constant morphological operations such as opening and closing. For each data set, five slices per orientation and five different positions for each slice of the DICOM volume, which leads to 75 patches for each data set, were evaluated.

The results of the 2D evaluation of our Smart Brush are depicted in Figure 8. For most patients, an average Dice coefficient of 0.9 was achieved.

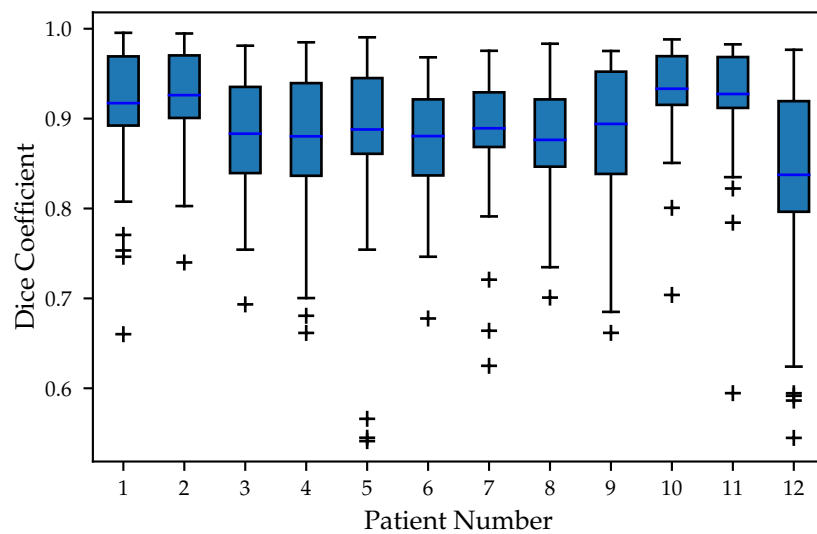


Figure 8. The evaluation results of the 2D segmentation result using the Smart Brush.

3.2. 3D Interpolation Evaluation

For the evaluation of the 3D interpolation, the same 12 data sets were used as for the smart brush evaluation. The Dice coefficient was used to evaluate the quantitative score of overlap of the 3D interpolation compared to the gold standard segmentation. The accuracy of the 2D segmentations, the number of segmented slices, and the distribution of these slices are all criteria that can directly change the result of the 3D interpolation. Therefore, the evaluation is done based on the ground truth segmentation of the 3D volume, where individual slices were extracted to initialize the 3D interpolation. For each data set, the evaluation was performed with a different number of segmented slices per orientation. We evaluated one, three, and five slices per orientation, which means to have a total number of three, nine, and fifteen segmented slices, respectively. The slice selection was randomly; however, for the first three slices, the center slice for each orientation was chosen. The same method of control point extraction was used for both methods. The results of the 3D interpolation are depicted in Figure 9a. It can be seen that, by increasing the number of slices, the Dice coefficient usually increases slightly.

Furthermore, we compare our adapted-HRBF (A-HRBF) method to a reference method that extracts 3D gradients at the control points based on the local intensity (HRBF) [14]. The main difference from our proposed A-HRBF method to the HRBF is that we use a combination of 2D and 3D gradients based on the extracted contour of the 2D segmentation, which makes the interpolation faster. As mentioned, the standard HRBF method uses the 3D intensity gradient for their 3D interpolation. These can lead to errors, especially in the case of ambiguous boundaries, such as the transition between the left and the right ventricle. The differences will be outlined in more detail in the discussion. Comparing the different methods, the Dice coefficient for the proposed A-HRBF is consistently higher than for the HRBF [14], as depicted in Figure 9.

Figure 10 depicts the qualitative results of the A-HRBF 3D interpolation scheme for one example data set, where the result is shown in blue and the ground truth in red.

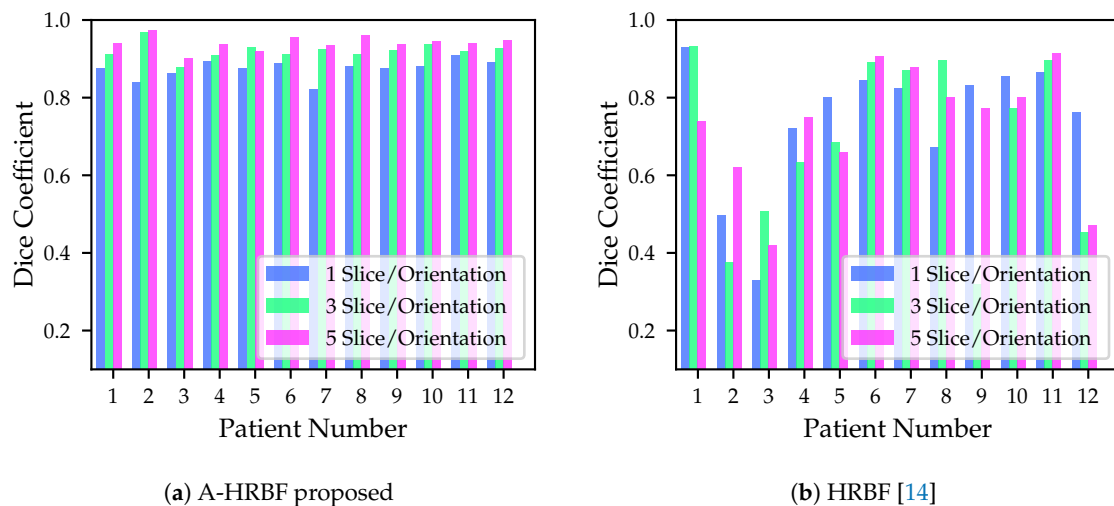


Figure 9. The 3D interpolation evaluation results: (a) the A-HRBF result with average Dice coefficient of 0.91, 0.95, and 0.96 for one, three, and five slices per orientation, respectively; (b) the HRBF result with average Dice coefficient of 0.69, 0.63 and 0.69 for one, three, and five slices per orientation, respectively.

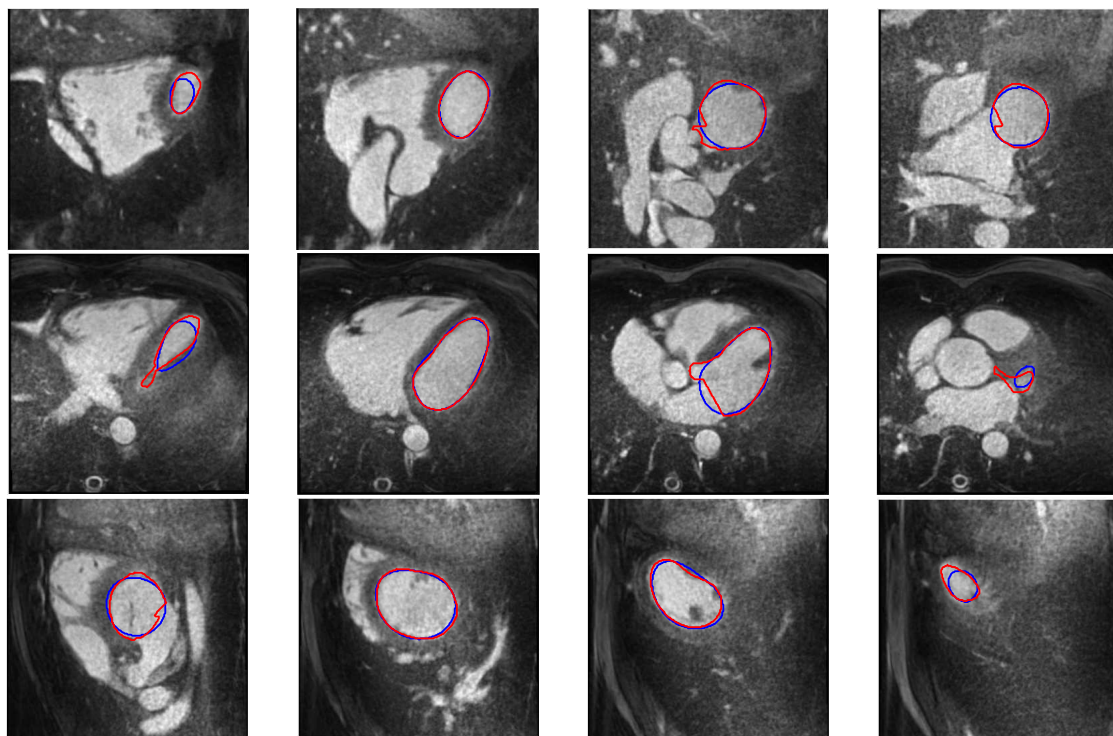


Figure 10. The ground truth (red) and the result of 3D interpolation (blue) are shown. The interpolation is obtained based on only one reference slice per orientation. Each row depicts a different orientation (axial, sagittal, and coronal). It is expected that the closer to the reference slice, the higher Dice coefficient is obtained ([2], Reproduced with permission).

4. Discussion

Our experiments show that three slices per orientation is sufficient to get a good segmentation result. Furthermore, in order to achieve more accurate interpolation results, the user has to segment those slices that have the maximum mismatch with the actual ground truth. In fact, for 3D interpolation,

the user selects those slices that are a good representation of the complete shape. Hence, the actual result of the interpolation is even better than the evaluation result shows.

In contrast to previous implicit methods for 3D interpolation [14], this method can not only be used for high-contrast images, but also for images with high noise level or other confounding factors due to the independence of intensity information for the 3D interpolation. The main advantage happens when there is an ambiguous boundary that only an expert can recognize (e.g., between the left ventricle and left atrium at the left ventricular outflow tract). In this case, the normal vector computation fails based on the previous method [14], whereas, using our method, the normal vectors are oriented based on the contour extracted from the 2D segmentation mask (see Figure 11). This leads to a better segmentation result compared to the standard HRBF method, which is also seen in Figure 9.

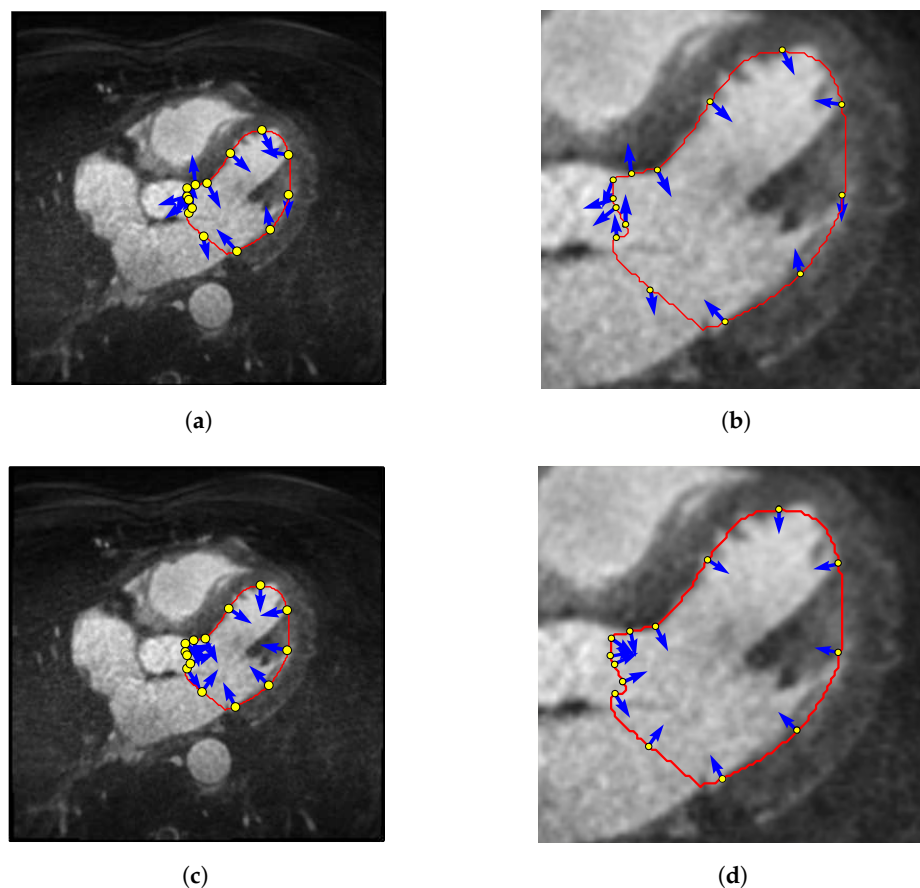


Figure 11. Normal vector orientation for left ventricle segmentation with an ambiguous boundary: (a,b) control points (yellow) and associated normal vectors (blue) based on intensity gradients for the HRBF method; (c,d) control points (yellow) and associated normal vectors (blue) based on the drawn contour (red) for our proposed method ([2], Reproduced with permission).

For the 2D evaluation, the outliers occur mainly because of two reasons. First, when there is no boundary or change in the intensity within the region of interest, in contrast to background or an undesired object, i.e., two different regions have the same intensities. For example, in heart segmentations, the intensities of the left ventricle and the left atrium have the same intensity and only experts can differ between these two objects. In this case, the fully automatic evaluation of the smart brush fails as it considers both objects as a single one (see Figure 12 for an example). Considering this case, the smart brush accuracy decreases as it performs based on the intensity thresholding. The second case of having a low Dice coefficient is when the hole filling is applied on the patch. In a real use of the smart brush, the user fills the holes at the end of the segmentation. In the automatic evaluation, the

filling is done each time after the patch is segmented and, as it is done by using the morphological operations, e.g., opening and closing, it expands the region of interest in this case. As the evaluation is performed patch-wise, the hole filling is done for a single patch each time and, at the end, the ROI is bigger than the ground truth.

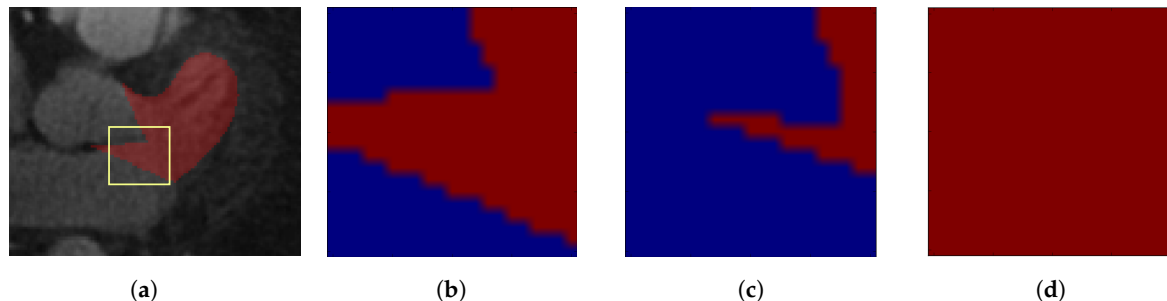


Figure 12. (a) the overlaid ground truth shown in red and the smart brush patch shown with a yellow rectangle; (b) the extracted patch from the smart brush; (c) the pre-segmented mask that is obtained by eroding the extracted patch; (d) the segmentation result by using the smart brush, which is different to the ground truth patch due to the same intensity range values.

For the 3D interpolation, we showed that good results are achieved with only one slice per orientation. However, this was only evaluated for the left ventricle, which is a convex object. For considering more complex objects, more annotations would be necessary. In addition, it would be great to analyze the inter-observer variability. However, we would therefore need the gold standard annotations from at least two clinical experts.

For future work, the method should be extended to use arbitrary orientations of the 2D slices, and not only axial, sagittal, and coronal image slices for the 3D interpolation. Having the possibility to annotate arbitrary orientations, the 2D annotations can be better adopted to the 3D segmentation problem. For the left ventricle, for example, one would annotate the short-axis orientation, and the two long-axis orientations to achieve excellent 3D interpolation results. Therefore, the 3D interpolation with the 2D gradient selector has to be adopted. In addition, the functionality of the 2D smart brush can also be improved, as right now only the intensity distribution and the connectivity is considered. However, for medical images, the texture can also play an important role. Incorporating texture features for the classification of foreground and background could further improve the 2D segmentation result.

5. Conclusions

The benefit of the method is that the user can correct the 3-D segmentation result easily by segmenting an additional 2-D slice with the maximum mismatch. Furthermore, no prior knowledge is involved, which leads to the ability to generate any arbitrary segmentation of any 3D data set, irrespective of image modality, displayed organ, or clinical application.

Acknowledgments: This research project was partly funded by Siemens Healthcare GmbH.

Author Contributions: Tanja Kurzendorfer, Peter Fischer, and Thomas Phol conceived and designed the experiments; Negar Mirshazadeh performed the experiments and analyzed the data; Tanja Kurzendorfer and Peter Fischer wrote the paper. Alexander Brost, Stefan Steidl, and Andreas Maier supervised the work and gave valuable feedback.

Conflicts of Interest: The authors declare no conflict of interest.

Disclaimer: The methods and information presented in this paper are based on research and are not commercially available.

Abbreviations

The following abbreviations are used in this manuscript:

A-HRBF	Adaptive Hermite Radial Basis Function
HRBF	Hermite Radial Basis Function
RBF	Radial Basis Function
MRI	Magnetic Resonance Imaging
ROI	Region of Interest
CP	Control Point

References

1. Kurzendorfer, T.; Forman, C.; Schmidt, M.; Tillmanns, C.; Maier, A.; Brost, A. Fully Automatic Segmentation of the Left Ventricular Anatomy in 3D LGE-MRI. *J. Comput. Med. Imaging Graph.* **2017**, *59*, 13–27.
2. Mirshahzadeh, N.; Kurzendorfer, T.; Fischer, P.; Pohl, T.; Brost, A.; Steidl, S.; Maier, A. Radial Basis Function Interpolation for Rapid Interactive Segmentation of 3D Medical Images. In Proceedings of the Annual Conference on Medical Image Understanding and Analysis, Edinburgh, UK, 11–13 July 2017; Springer: Berlin, Germany, 2017; pp. 651–660.
3. Mortensen, E.N.; Barrett, W.A. Interactive Segmentation with Intelligent Scissors. *Graph. Model. Image Process.* **1998**, *60*, 349–384.
4. Boykov, Y.; Veksler, O.; Zabih, R. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 1222–1239.
5. Grady, L. Random Walks for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1768–1783.
6. Amrehn, M.; Gaube, S.; Unberath, M.; Schebesch, F.; Horz, T.; Strumia, M.; Steidl, S.; Kowarschik, M.; Maier, A. UI-Net: Interactive Artificial Neural Networks for Iterative Image Segmentation Based on a User Model. In *EG VCBM 2017*; Rieder, C., Ritter, F., Hotz, I., Merhof, D., Eds.; Eurographics Association: Aire-la-Ville, Switzerland, 2017; pp. 143–147.
7. Malmberg, F.; Strand, R.; Kullberg, J.; Nordenskjöld, R.; Bengtsson, E. Smart paint a new interactive segmentation method applied to MR prostate segmentation. In Proceedings of the MICCAI Grand Challenge: Prostate MR Image Segmentation, Nice, France, 1–5 October 2012.
8. Parascandolo, P.; Cesario, L.; Vosilla, L.; Pitikakis, M.; Viano, G. Smart brush: A real time segmentation tool for 3D medical images. In Proceedings of the 2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA), Trieste, Italy, 4–6 September 2013; pp. 689–694.
9. Boykov, Y.; Kolmogorov, V. Computing geodesics and minimal surfaces via graph cuts. In Proceedings of the International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 3, pp. 26–33.
10. Morse, B.S.; Yoo, T.S.; Rheingans, P.; Chen, D.T.; Subramanian, K.R. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In Proceedings of the ACM SIGGRAPH 2005 Courses, Los Angeles, CA, USA, 31 July–4 August 2005; ACM: New York, NY, USA, 2005; p. 78.
11. Duchon, J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*; Springer: Berlin, Germany, 1977; pp. 85–100.
12. Carr, J.C.; Beatson, R.K.; Cherrie, J.B.; Mitchell, T.J.; Fright, W.R.; McCallum, B.C.; Evans, T.R. Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 12–17 August 2001; ACM: New York, NY, USA, 2001; pp. 67–76.
13. Turk, G.; O'Brien, J.F. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.* **2002**, *21*, 855–873.
14. Ijiri, T.; Yoshizawa, S.; Sato, Y.; Ito, M.; Yokota, H. Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2013; Volume 32, pp. 123–132.
15. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 7 January 1998; pp. 839–846.
16. Abbena, E.; Salamon, S.; Gray, A. *Modern Differential Geometry of Curves and Surfaces with Mathematica*; CRC Press: Boca Raton, FL, USA, 2006.

17. Elkan, C. Nearest Neighbor Classification. *Encycl. Database Syst.* **2011**, doi:10.1007/978-0-387-39940-9_2920.
18. Macedo, I.; Gois, J.P.; Velho, L. Hermite Radial Basis Functions Implicits. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2011; Volume 30, pp. 27–42.
19. Brazil, E.V.; Macedo, I.; Sousa, M.C.; de Figueiredo, L.H.; Velho, L. Sketching Variational Hermite-RBF Implicits. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, Annecy, France, 7–10 June 2010; Eurographics Association: Aire-la-Ville, Switzerland, 2010; pp. 1–8.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).