# 3D Cardiac MRI Data Visualization Based on Volume Data Preprocessing and Transfer Function Design

F Yang[1], WM Zuo[1], KQ Wang[1], H Zhang[2]

[1]Harbin Institute of Technology, Harbin, China
[2]University of Manchester, Manchester, UK

## Abstract

*The diversity and complexity of cardiac tissues makes it very challenging to visualize 3D cardiac magnetic resonance image (MRI) data. In this paper, we proposed an adaptive volume data preprocessing approach, and used a semi-automated transfer function for volume rendering of the pre-processed cardiac MRI data. To improve the quality of 3D cardiac MRI data, we first used a three-dimensional median filtering method for data denoising, and then proposed an adaptive ellipsoidal Gaussian filtering scheme for local-feature-preserving data smoothing. For effective visualization, we implemented a volume rendering pipeline with ray casting and the semi-automated transfer function design scheme, which allow user create transfer function to visualize the information which he is interested in. Finally, the efficiency of the proposed 3D cardiac MRI data visualization method is verified using the sheep heart MRI data.*

## 1. Introduction

Medical imaging and visualization provides efficient means to reveal the relations between the function, behavior and organization of the biological organs, and thus play a crucial role in medical diagnosis and surgery. The magnetic resonance imaging (MRI) method can non-invasively acquire the physiological and biochemical information with high spatial resolution and outstanding discrimination, and now have been developed as an important research area and as a diagnostic technique in medical practice. Visualization allows physicians to examine MRI data more easily and effectively, and is of great value in medical diagnosis and surgery planning.

Visualization of 3D cardiac magnetic resonance imaging data is a technically difficult problem [1]. For 2D MRI data, it is natural to display using images. For 3D MRI data, the observation and interaction would be more complicated. Volume rendering techniques, which generate meaning 2D images from 3D volume data, can be applied to visualization of 3D cardiac MRI data. However, the variety and complexity of cardiac tissue makes it very challenging to visualize 3D cardiac MRI data.

Currently, there are two classes of volume rendering techniques which could be applied to visualization of 3D MRI data, classification-based volume rendering and direct volume rendering. In classification-based volume rendering, a segmentation procedure is introduced to assign each voxel a specific object category [2]. Since of the structure complexity and individual variation, automated segmentation of cardiac MR images usually is challenging and time-consuming [3].

With the help of transfer function, direct volume rendering gives an approach to explore large 3D data sets, and thus could alleviate the difficulty in cardiac tissue segmentation. Transfer functions, which assign appropriate optical properties to the voxels, are of great importance for direct volume rendering [4]. However, transfer function design is difficult. By far, there is no one ideal solution which could automatically generate satisfactory transfer functions for any volume data. Thus, several semi-automated [5, 6] or interactive transfer function design scheme [7] have been proposed to address this. Besides, heart has many kinds of tissues and pathological reactions may cause the change of some tissues. All these factors would also make it very challenging to design transfer function for 3D cardiac MRI data.

In this paper, we proposed an adaptive volume data preprocessing approach, and used a semi-automated transfer function for volume rendering of 3D cardiac MRI data. To improve the quality of 3D cardiac MRI data, we first used a three-dimensional median filtering method for data denoising, and then proposed an adaptive ellipsoidal Gaussian filtering scheme for local-feature-preserving data smoothing. For effective visualization, we implemented a volume rendering pipeline with ray casting and the semi-automated transfer function design scheme.

The reminder of the paper is organized as follows. Section 2 presents an adaptive pre-processing approach. Section 3 describes the volume rendering scheme,

ISSN 0276−6574

including ray casting and semi-automated transfer function design. Section 4 provides experimental results and discussions. Finally, Section 5 gives our conclusion and future work.

## 2. Data pre-processing

In this section, we proposed an adaptive volume data preprocessing approach to improve the quality of 3D cardiac MRI data. The proposed method includes two parts, 3D median filtering and adaptive ellipsoidal Gaussian filtering.

In the median filtering stage, we adopt the recursive median filtering method rather than the simple median filtering approach. In the median operation, simple median filtering uses the following rule

$$y(i,j) = \text{MEDIAN}[x(i-k,j-k),\cdots,x(i,j),\cdots x(i+k,j+k)] \tag{1}$$

to calculate the filtering output. If the filtering output $y(i',j')$ of pixel $x(i',j')$ have been derived, recursive median filtering replace the $x(i',j')$ in the right hand of Eq. (1) with $y(i',j')$,

$$y(i,j) = \text{MEDIAN}[y(i-k,j-k),\cdots,x(i',j'),\cdots,x(i,j),\cdots x(i+k,j+k)]. \tag{2}$$

Compared with simple median filtering, recursive median filtering has a number of advantages, such as better noise suppression and less signal distortion. Further, recursive median filtering can attain similar performance with a smaller window size than simple median filtering, and thus is computationally more efficient [8]. All these are the reasons that we use the recursive median filtering method in the first stage of the volume data preprocessing.

In the second part, we proposed an adaptive ellipsoidal Gaussian filtering scheme for local-feature-preserving data smoothing. Circular Gaussian filter is an ideal filter in the sense that it reduces the magnitude of high spatial frequency in an image and has been widely applied in image smoothing and denoising [9]. Circular Gaussian filtering, however, would also cause the blurring of the local image features, such as edges and lines. Thus, we propose to use adaptive ellipsoidal Gaussian (EG) filtering for local-feature-preserving smoothing.

The adaptive ellipsoidal Gaussian filtering scheme has two parameters (variance $\sigma$ and the reference gradient $g_{\text{ref}}$), and includes three major steps, computation of the first derivatives, EG parameters estimation, and filtering.

In the first step, the gradient vector is represented as

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}\right]^T = \left[f_x, f_y, f_z\right]^T, \tag{3}$$

and the magnitude of the gradient vector is

$$\left|\nabla f\right| = \sqrt{f_x^2 + f_y^2 + f_z^2}\ . \tag{4}$$

The ellipsoidal Gaussian filter can be represented as,

$$G(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}\left|\mathbf{V}\right|^{1/2}}\exp(-\frac{\mathbf{x}^T\mathbf{V}\mathbf{x}}{2}), \tag{5}$$

where $\mathbf{x}=[x, y, z]^T$, and $|\mathbf{V}|$ denotes the determinant of $\mathbf{V}$. Further, we assume

$$\mathbf{V} = \left[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\right]^T \begin{bmatrix} \sigma & & \\ & \sigma_1 & \\ & & \sigma_1 \end{bmatrix} \left[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\right], \tag{6}$$

where $\mathbf{u}_i$ is the $i$th eigenvector of $\mathbf{V}$. Because the second and the third eigenvalues are the same, the parameter estimation problem is to determine the value of $\mathbf{u}_1$ and $\sigma_1$. In the second step, we let

$$\sigma_1 = \frac{\sigma}{(1 + \left|\nabla f\right| / g_{ref})}, \tag{7}$$

$$\mathbf{u}_1 = \frac{\nabla f}{\left|\nabla f\right|}. \tag{8}$$

In the third step, for a specific voxel, we use the ellipsoidal Gaussian filter defined in Eqs. (7) and (8) to convolve with the volume data. It is obvious that, when the voxel is close to the boundary, the value of $\left|\nabla f\right|$ is large, and the anisotropy would be significant. If the value of $\left|\nabla f\right|$ is small, then $\sigma_1 \approx \sigma$, we will use a circular 3D Gaussian filter for image smoothing.

## 3. Volume rendering

### 3.1. Ray casting

Ray casting is a classical volume rendering algorithm which performs the rendering integral along one viewpoint ray. For each pixel, the method uses the resampling and compositing operations to compute the final pixel color. For the details of ray casting, please refer to [10]. During compositing, both the color value and the opacity are accumulated according to the compositing equation. Transfer function is a mapping from voxels to opacity, and thus has a direct consequence on ray casting rendering quality.

### 3.2. Semi-automated transfer function design

Transfer functions play a crucial role for direct volume rendering. The semi-automated transfer function design scheme is based on the ideal derivative of Gaussian (DoG) boundary model, and includes four major steps: computation of the first and the second derivatives, histogram volume creation, mapping data value to position, and opacity function generation. In the following, we will briefly introduce these four steps:

(1) Computation of the derivatives: we calculate the first

and the second derivatives of each voxel. The magnitude of the first derivative |∇*f*(*x*, *y*, *z*)| is defined in Eqs. (3) and (4). We obtain the magnitude of the second derivative of each voxel using Hessian method,

$$\mathbf{H}f = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{pmatrix}, \tag{9}$$

$$|\mathbf{H}f| = \frac{1}{|\nabla f|}(\nabla f)^T \mathbf{H}f(\nabla f). \tag{10}$$

(2) Histogram volume creation: we first normalized the value of |∇*f*| to [0, 255], and then convert it to 8-bit integer. Analogously, we convert the value of |**H***f*| to 8-bit integer. Using the normalized value of *f*, |∇*f*|, and |**H***f*|, we create a histogram value **H**(*i*, *j*, *k*) (0 ≤ *i*, *j*, *k* ≤ 255) to represent the distribution of the voxel value, the first and the second derivatives, where **H**(*i*, *j*, *k*) = *m* denotes that there are *m* voxels with *f* = *i*, |∇*f*| = *j*, |**H***f*| = *k*.

(3) Mapping data value to position: we first calculate the variance

$$\sigma = \frac{2\sqrt{e}\,|\nabla f|_{\max}}{|\mathbf{H}f|_{\max} - |\mathbf{H}f|_{\min}}, \tag{11}$$

and then the position function p(v) that maps the data value to position is defined as

$$p(v) = \frac{-\sigma^2 h(v)}{g(v)}, \tag{12}$$

where *h*(*v*) denotes the average |**H***f*| at value *v*, and *g*(*v*) denotes the average |∇*f* | at value *v*.

(4) Opacity function generation: after obtaining *p*(*v*), the opacity function generation is to create a mapping termed as a boundary emphasis function from position to opacity.

$$\alpha = b(p(v)) = \begin{cases} \frac{1}{\sigma}(p(v)+\sigma), p(v) \in [-\sigma, 0] \\ -\frac{1}{\sigma}(p(v)-\sigma), p(v) \in [0, \sigma] \end{cases}, \tag{13}$$

## 3.3. Visualization toolkit

The Visualization ToolKit (VTK) is a free open source software system and is mainly used in 3D computer graphics and visualization. VTK is an object library rather than a system. The object library can be implanted in application program. Users can develop their own libraries based on the original libraries of VTK. General visualization algorithms are packaged in VTK, including ray-casting algorithm used in this paper which is packaged as classes provided to users.

VTK provides three kinds of transfer function formulations (vtkVolumeRayCastComposite class of VTK) in Compositing Volume Rendering. We adopt one of them where values of volume data are transformed to optical properties, such as color, opacity and so on by transfer function. Then these properties are composited to each pixel on screen to generate result image.

## 4. Experimental results

In this section, we use the MRI data of a sheep heart (http://www9.informatik.uni-erlangen.de/External/vollib/) to evaluate the performance of the proposed 3D cardiac MRI data visualization method. The size of data set is 352×352×256, and the data contain some damaged tissue caused by the partial circulation blocking of the heart.

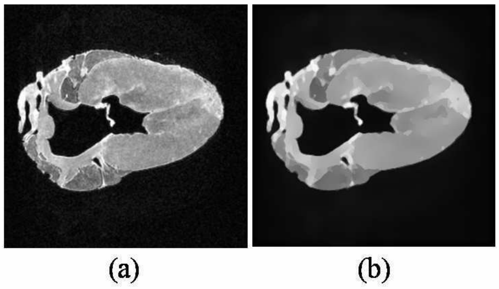

Fig. 1 Preprocessing of the sheep heart data: (a) original, and (b) preprocessed.

We first present an example of the pre-processing result. Fig. 1 shows an original slice of the volume data and the corresponding preprocessing image. From Fig. 1, it is obvious that preprocessing is effective in removing noise.

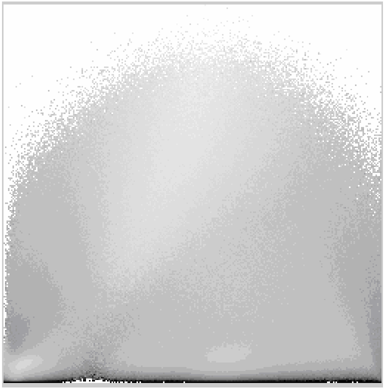

Fig. 2 The (*v*, |∇*f*|) scatterplot.

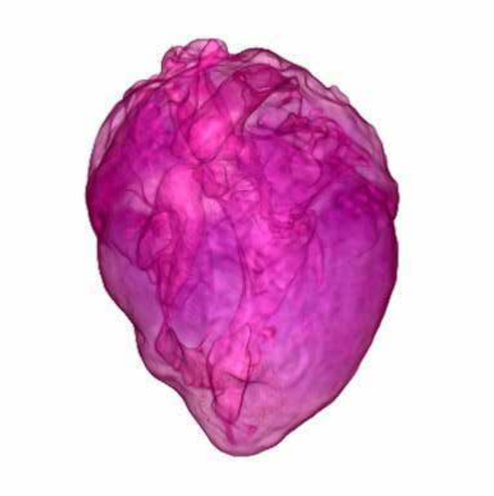

Fig. 3 Volume rendering of the sheep heart data.

Fig. 2 shows the ($v$, $|\nabla f|$) scatterplot created from the sheep heart data set. In the semi-automated design, we can further post-processing the histogram to edit the transfer function for better rendering. Finally, we show the volume rendering of the preprocessed data, as depicted in Fig. 3.

## 5. Conclusions and discussion

In this paper, we proposed an adaptive volume data preprocessing approach, and used a semi-automated transfer function for volume rendering of 3D cardiac MRI data. To improve the quality of 3D cardiac MRI data, we first used a 3D median filtering method for data denoising, and then proposed an adaptive ellipsoidal Gaussian filtering scheme for local-feature-preserving data smoothing. For effective visualization, we implemented a volume rendering pipeline with ray casting and the semi-automated transfer function design scheme.

In the future, we will evaluate the method using the human heart data, implement other transfer function design method [7], and explore the effective interactive ways to allow users tune the visualization results for medical analysis.

## Acknowledgements

The work is partially supported by the NSFC fund under Contract No. 60571025, and the Chinese 863 fund under Contract No. 2006AA01Z308.

## References

[1] Solaiyappan M, Poston T, Heng PA, McVeigh ER, Guttman MA, Zerhouni EA, Interactive visualization for rapid noninvasive cardiac assessment, Computer, 1996; 29(1):55-62.

[2] Bullitt E, Aylward SR, Volume rendering of segmented image objects, IEEE Trans. Medical Imaging, 2002; 21(8): 998-1002.

[3] Hautvast G, Lobregt S, Breeuwer M, Gerritsen F, Automatic contour propagation in cine cardiac magnetic resonance images, IEEE Trans. Medical Imaging, 2006; 25(11): 1472-1482.

[4] Pfister H, Lorensen B, Bajaj C, Kindlmann G, Schroeder W, Avila LS, Martin K, Machiraju R, Lee J, The transfer function bake-off, IEEE Computer Graphics and Applications, 2001; 21(3):16-22.

[5] Kindlman G, Durkin J, Semi-automatic generation of transfer functions for direct volume rendering, Proc. 1998 IEEE Symp. Volume Visualization, IEEE CS Press, Los Alamitos, Calif., Oct. 1998:79-86.

[6] Kniss J, Kindlmann G, Hansen C, Multidimensional transfer functions for interactive volume rendering, IEEE Trans. Visualization and Computer Graphics, 2002; 8(3): 270-285.

[7] Wu Y, Qu H, Interactive transfer function design based on editing direct volume rendered images, IEEE Trans. Visualization and Computer Graphics, 2007; 13(5):1027-1040.

[8] Bovik A, Handbook of image and video processing, 2nd edn. Singapore. Amsterdam: Elsevier; 2005.

[9] Pratt WK, Digital image processing, 3nd edn. New York. New York: Wiley; 2001.

[10] Levoy M, Display of surfaces from volume data, IEEE Computer Graphics and Applications, 1988; 8(3):29-37.

Address for correspondence

Wangmeng Zuo
Mailbox 332, Harbin Institute of Technology
Harbin 150001, China
E-mail: cswmzuo@gmail.com