GPU-Based Visualization and Synchronization of 4-D Cardiac MR and Ultrasound Images

Qi Zhang, *Member, IEEE*, Roy Eagleson, and Terry M. Peters, *Fellow, IEEE*

***Abstract*—In minimally invasive image-guided interventions, different imaging modalities, such as magnetic resonance imaging (MRI), computed tomography (CT), and 3-D ultrasound (US), can provide complementary, multispectral image information. Dynamic image registration is a well-established approach that permits real-time diagnostic information to be enhanced by placing lower-quality real-time images within a high quality anatomical context. For the guidance of cardiac interventions, it would be valuable to register dynamic MRI or CT with intra-operative US. However, in practice, either the high computational cost prohibits such real-time visualization, or else the resulting image quality is not satisfactory for accurate interventional guidance. Modern graphics processing units (GPUs) provide the programmability, parallelism and increased computational precision to address this problem. In this paper, we first outline our research on dynamic 3-D cardiac MR and US image acquisition, real-time dual-modality registration and US tracking. Next, we describe our contributions on image processing and optimization techniques for 4-D (3-D + time) cardiac image rendering, and our GPU-accelerated methodologies for multimodality 4-D medical image visualization and optical blending, along with real-time synchronization of dual-modality dynamic cardiac images. Finally, multiple transfer functions, various image composition schemes, and an extended window-level setting and adjustment approach are proposed and applied to facilitate the dynamic volumetric MR and US cardiac data exploration and enhance the feature of interest of US image that is usually restricted to a narrow voxel intensity range.**

***Index Terms*—Cardiac imaging, dynamic texture binding, extended window-level, GPU, image compositing, real-time visualization, synchronization, transfer function.**

I. Introduction

Cardiac diseases include a broad range of conditions that affect the human heart, such as myocardial infarction, vascular spasms (angina pectoris), cardiomyopathy, heart rhythm problems (arrhythmias), and surgery is one of the most effective treatment options [1]. However, the invasive nature of conventional open-heart surgery carries the risk of embolism and stroke, and associated healthcare costs due to such complications [2], [3]. In recent years, noninvasive imaging and cardiac procedures have coevolved with imaging technologies, which play an essential role in the precise assessment of cardiac anatomy and function, as well as image guidance during the treatment procedure and follow-up. Furthermore, imaging capabilities have not only increased in resolution and dimension, but also in the variety of modalities. These technological advancements have led to the development of minimally invasive image-guided cardiac intervention, and have been employed to reduce the risk of open-heart surgery [4], [5], such as lower risk of bleeding and blood transfusion, better recovery and improved rehabilitation, as well as fewer rehospitalizations and complications from heart failure [6].

Minimally invasive interventions on the beating heart require real-time imaging for the guidance procedure. 3-D ultrasound (US) provides radiation-free, cost-effective imaging with high temporal resolution, and is an ideal imaging modality for these applications, providing dynamic information during the guidance of the procedure, so that the surgeons can monitor the beating heart as well as the interventional tools in real time [7]. However, in clinical practice, due to the poor image quality and low spatial resolution, coupled with a limited field of view, the sole imaging modality US cannot be used to reliably perceive anatomical structural information. It is, therefore, difficult to guide cardiac interventions using US alone. The low image quality can be improved by integrating real-time intra-operative 3-D US with preoperatively acquired high-quality MR or CT volumes [8]. Real-time display and synchronization of high-quality registered 4-D multimodal cardiac images is a key addition in the evolution of a platform for planning and guiding beating heart surgery. With such a display, the US images may be visualized within the context of a high quality preoperative volumetric image. Fused 4-D real-time imaging offers this contextual information using the best possible image quality to allow surgeons to assess cardiac anatomy, morphology, and function during the entire intervention process [9], [10].

Visualization plays a central role in the presentation of fused medical image data, providing clinicians with a complete view of the anatomy, increasing the reliability of conventional 2-D visualization techniques. In recent years, the graphics processing unit (GPU) has seen wide applications for medical data visualization and simulation [11]. For example, Xie *et al.* [12] presented a GPU-based volume rendering algorithm to interactively visualize large CT data with high image quality, Wang and Good [13] employed GPUs to stereoscopically

Manuscript received November 10, 2011; revised March 1, 2012 and May 4, 2012; accepted June 2, 2012. Date of publication June 18, 2012; date of current version September 20, 2012. This work was supported in part by the Ontario Graduate Scholarship, in part by the Ontario Research & Development Challenge Fund, in part by the Canadian Institutes of Health Research, in part by the Canada Foundation for Innovation, and in part by the Natural Sciences and Engineering Research Council of Canada.

Q. Zhang is with the MR Research & Development Research Group, National Research Council Institute for Biodiagnostics, Winnipeg, MB R3B 1Y6, Canada (e-mail: qi.zhang@nrc-cnrc.gc.ca).

R. Eagleson is with the Department of Electrical and Computer Engineering, Western University, London, ON N6A 5K8, Canada (e-mail: eagleson@uwo.ca).

T. M. Peters is with the Imaging Research Laboratories, Robarts Research Institute, London, ON N6A 5K8, Canada, and also with the Department of Biomedical Engineering, Western University, London, ON N6A 5K8, Canada (e-mail: tpeters@robarts.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITB.2012.2205011

display large 3-D medical data sets in real time to explore lung images for cancer screening. Kutter *et al.* [14] used a GPU-accelerated approach to render medical images interactively, and simulate US image from CT data. A web-based medical image visualization and processing platform was developed by Mahmoudi *et al.* [15], where graphics hardware accelerated visualization toolkit (VTK) libraries and their virtual reality modeling language (VRML) exporting capacity were used for volume rendering. Dai *et al.* [16] developed a real-time 3-D US reconstruction algorithm implemented on a GPU and volume rendered the reconstructed results to provide interactive visual feedback. However, all of these medical volume rendering techniques have not addressed the multimodality issues that arise when combining different medical datasets in a synchronized real-time visualization environment.

The advances provided by new visualization techniques have significantly accelerated the use of multimodality images in image-guided interventions [17], [18]. Researchers such as Kim *et al.* [19] have developed a dual-modality PET/CT image visualization platform through the integration of interactive fuzzy thresholding segmentation, and several reports in the literature outline the display of multimodality 3-D medical images [20]–[23]. Some of these approaches employ software-based volume-rendering techniques [20], [21], which are too slow for real-time clinical applications in image-guided therapy, while others [22], [23] use rendering approaches based on texture mapping. The introduction of “striping” or “flipping” artifacts with texture mapping based volume visualization techniques results in low image quality, making these techniques unsuitable for surgical guidance and surgical tool tracking. When texture mapping techniques are applied to visualize 4-D cardiac multimodal images [24], [25], the depth sorting computation of the slices from different imaging modalities reduces the rendering efficiency, and compositing these slices derived from different imaging modalities further compromises the rendered quality of the fused 3-D images.

To address the aforementioned challenges, we have developed a GPU-accelerated 4-D medical image visualization system for stereoscopic beating heart displays, which can provide interactive feedback on cardiac behaviors and physiological parameters [26]. In this paper, we first describe the extension of our research on dynamic medical image visualization to real-time multimodality 4-D cardiac image rendering, where the image compositing and ray casting computations are performed directly on the GPU vertex and fragment processors and used with an artifact-free rapid voxel classification algorithm [27]. Subsequently, we present two texture binding schemes to render and synchronize dynamic dual-modality images of the beating heart. The first scheme is based on dynamic texture binding, which maintains the image volumes within the GPU during the entire rendering procedure, while the second uses a multitexture identification tag (ID) to update the textures dynamically. Furthermore, we analyze the spatial and temporal resolution of the MR and US images, and propose a super-sampling scheme on the CPU that is optimized for real-time rendering and synchronizing 4-D MR and US images on the GPU fragment shader. To minimize graphics memory usage and to accelerate the texture loading speed, we propose techniques to change the internal texture storage format of the MR and US images. For the purpose of improving the visualization results and to extend the surgeons’ perceptual abilities, an extended window-level technique is presented, and techniques such as volume shading, multiple clipping, and weighting factor adjustments are proposed and implemented to enhance the rendered dual-modality images and facilitate their exploration. Finally, the resulting fused cardiac images are evaluated and analyzed.

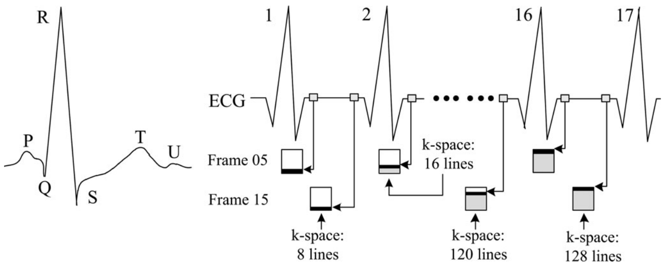


Fig. 1. (Left) Form of an ECG signal over a single cardiac cycle with 6 features labeled according to the standard convention. (Right) Pipeline of segmentation-based cine slab MR image acquisition over 16 cardiac cycles.

II. Volumetric Multimodal Image Processing

In this section, we first review the electrocardiogram (ECG) signal and its application to 4-D cardiac MR image acquisition. Next, technologies of dual-modality image registration and tracking are depicted, followed by an introduction of methodologies for processing and optimizing the cardiac images in CPU main memory.

A. Dataset Acquisition

Dynamic MR cardiac image acquisition is typically accompanied by the recording of an ECG signal [see Fig. 1 (left)] to synchronize the slices of the 3-D data needed to reconstruct the final dynamic 3-D cardiac images. During the acquisition of MR images, raw data are first acquired in the spatial frequency domain, i.e., *k*-space[1], and then are reconstructed via a discrete Fourier transform to form the image. The dimensionality of the image and *k*-space should be the same, i.e., 2-D data must be acquired to generate a 2-D image.

In practice, cardiac MRI is commonly performed using a segmentation-based approach, where data acquisition is segmented over several cardiac cycles as measured by the ECG (see Fig. 1). During imaging the subjects hold their breath to minimize artifacts from respiratory motion. The cardiac cycle is usually split into 20 frames. The right-hand side of Fig. 1 shows two example frames, i.e., Frame 05 and Frame 15, in a cardiac R–R interval. Each of these 20 frames represents one of 20 equally spaced time points in a cardiac cycle during a single R–R interval. During the MR data acquisition process, every MR scan fills 8 lines in each of these 20 frames in *k*-space. For

[1] A temporary image space in spatial frequency domain, in which raw data acquired from digitized MR signals are stored during data acquisition process. When *k*-space is full (at the end of the scan), the data are mathematically processed through using a discrete Fourier transform to produce a final reconstructed MR image.

every frame, 128 lines are acquired to fill the *k*-space array and then the data in the filled *k*-space are used to reconstruct a 2-D slab image with a ∼6-mm-thick slice. Therefore, to reconstruct a 2-D slab image, we require a total of 16 cardiac cycles, as shown in the right-hand side of Fig. 1. The data acquired using the protocol described in the Fig. 1 illustrates the cardiac motion in only one slab of tissue, i.e., 2-D cine slab image. A full 4-D cardiac representation can be obtained by repeating the aforementioned procedure for each slice location in the heart [28], [29].

To demonstrate the efficiency of our system, high-quality dynamic gradient echo T_1-weighted 3-D MR images of a volunteer were acquired in the coronal plane using a 1.5-T GE CVi scanner (GE medical systems, Milwaukee). The entire sequence comprised twenty ECG-gated, 4-D MR images, each depicting a snapshot of the heart at a different time-point within the cardiac cycle. Each volume has 115×170×75 voxels, with a voxel size of 1.48×1.48×1.5 mm^3. In addition, fourteen real-time 3-D US images of the same subject were acquired on a Philips SONOS 7500 US machine. The dimension of each US volume (USV) was 160×208×144 with a voxel size of 1.24×1.26×0.8 mm^3.

B. Dual-Modality Image Registration

Registration is an important step to align intra-operative real-time images with the preoperatively acquired high-quality images to provide the user with real-time guidance information during the intervention procedure. There are two principal methods, surface- and intensity-based, employed in medical image registration. Due to the fact that surface-based registration relies on the accurate segmentation of anatomical structures in the images to be registered, it is not generally considered to be suitable for registering 3-D US to volumetric MR images. It is also difficult to segment 3-D US images in real time. Intensity-based mutual information (MI) registration methods make no assumptions about the intensity values in the images, but match the two images based on statistical relationships between the pixel values contained within them. Using MI, it is possible to provide satisfactory registration results for multimodality image registration of 3-D US and MR images, and therefore, MI-based registration algorithm was employed in our system [30].

To explain the registration pipeline, let two discrete volumes be A and B with individual probability density distributions $p_A(a)$ and $p_B(b)$, and joint probability density distribution $p_{AB}(a,b)$ in the overlapping regions of A and B. The MI $I(A,B)$ of volume dataset A and B measures the degree of dependence of A and B as the distance between the joint distribution and the distribution associated with the case of complete independence $p_A(a) \cdot p_B(b)$, by means of the following Kullback–Leibler measure [31]:

$$I(A,B) = \sum_{a,b} p_{AB}(a,b) \log \frac{p_{AB}(a,b)}{p_A(a) \cdot p_B(b)}. \tag{1}$$

This MI algorithm involves iteratively transforming the floating template image until it is optimally aligned with the reference image, based on maximizing the MI metric. For the purpose of registering 4-D MR images to 3-D US images, a Polaris optical tracking system (OTS) (NDI, Waterloo, Canada) was employed to continuously track an US probe while ECG signals were simultaneously acquired from the patient. This tracking information was used to register the US images to the world coordinate system (WCS) [9].

For the 20 cardiac volumes over one cardiac cycle MRV_i $(i = 0, 1, 2, \ldots, 19)$, first, intensity-based MI registration algorithms were used to generate 19 nonlinear transformation volumes MRTV_k $(k = 0, 1, 2, \ldots, 18)$ for registering the MRV_0 to MRV_1; MRV_1 to $\mathrm{MRV}_2, \ldots$, and MRV_{18} to MRV_{19}. Every voxel in these 19 nonlinear transformation volumes stores a transformation vector.

Because the temporal frame rate of the US image volume depends on the field-of-view (FOV) selected for a particular study, the US images are not typically acquired at the same time-points within the cardiac cycle as the MR images. The ECG signals associated with the US image are therefore employed to align the US and MRI frames in the temporal domain by phase-stamping the US images with timing information derived from the ECG signals. These acquired phase data are used as parameters to interpolate the vectors stored in the voxels of the MR transformation volumes MRTV_k $(k = 0, 1, 2, \ldots, 18)$, and the generated new MRTVs are used to transform the MRVs to generate MR images at new spatial location between the two MR cardiac volumes used in the interpolation, so that phase-coincident US and MR images can be identified. Finally, the US images are transformed into the WCS and registered with the MR images, providing fourteen US transformation matrices, along with the identification numbers of the corresponding MR images. Fig. 5 describes an example of the interpolation process.

C. 3-D Medical Image Optimization

In our implementation, the real-time 3-D US images are first loaded to the CPU main memory, and then spatially transformed using the matrix calculated from the registration procedure. Due to the imaging time constraints for dynamic MR image acquisition, the resulting 4-D image volume is generally of lower resolution than that of the US images, particularly in the slice direction. Therefore we employ a super-sampling scheme for MR images on the CPU. The MRVs are first loaded to the CPU main memory, then resampled to the same voxel size as the USVs using cubic interpolation. The right-hand side of Fig. 2 illustrates this image processing procedure.

The intensity values of each voxel in the raw 3-D data from the MR or US scanner are loaded to the graphics memory as a set of 3-D textures during the rendering pipeline, with the α channel being used to store the intensity values. The GPU has 128 bit texture capability (*R*, *G*, *B* and α channels, each with 32 bits). However, since the raw MR and US images are monochrome, the internal voxel storage format of the 3-D texture is set to one single alpha component (32 bits), resulting in one-fourth of the graphics memory usage and a fourfold increase in texture transfer rate. Fig. 3 shows this procedure, which demonstrates that the loaded intensity value in the 3-D texture is dynamically mapped to *R*, *G*, *B*, α through color lookup table during the ray casting process, resulting in colorized volume-rendered images.

As discussed in our previous work [32], in the case of preintegrated volume shading, we need luminance and α channels

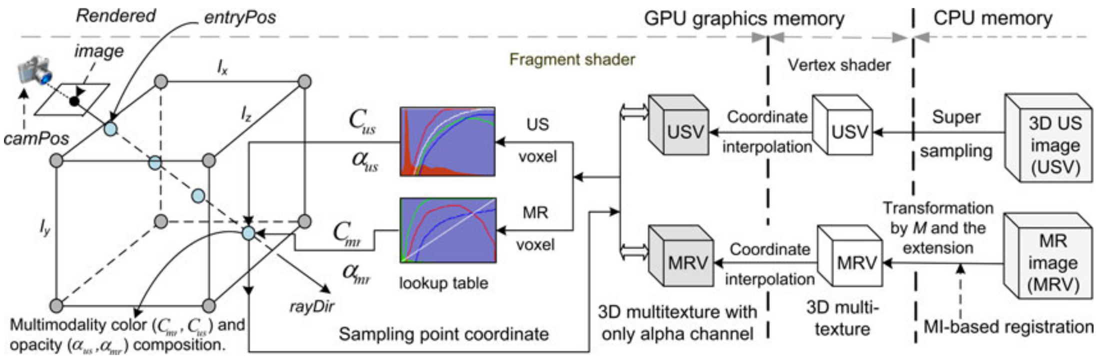


Fig. 2. Flowchart depicting real-time dual-modality cardiac MR and US image rendering pipeline: MR volumes (MRVs) are first super-sampled using cubic interpolation, and US volumes (USVs) are transformed using matrices derived from mutual information (MI)-based registration. The two volumes are first loaded to GPU vertex shader, where their coordinates are interpolated as data streams, and then transferred to the GPU fragment shader, where they are rendered using ray casting algorithm and image compositing techniques.

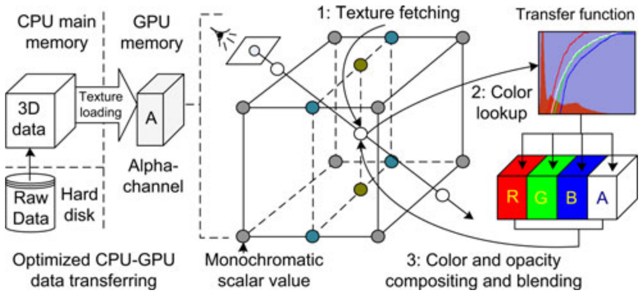


Fig. 3. Optimized texture loading and storage pipeline, resulting in one-fourth memory usage and a fourfold increase in texture transfer rate.

to store precomputed shading components and intensity values, respectively. For fragment-based dynamic shading, we still need the *R*, *G*, and *B* channels to store the precalculated voxel normal vector, and the α channel is employed for the intensity.

III. Visualization and Synchronization Pipelines

In this section, we first introduce the optical model for volumetric data rendering, which is the basis for dual-modality medical image visualization. Then, the procedure for fused cardiac image ray casting on a GPU architecture, whose working pipeline is illustrated in left-hand side of Fig. 2, is presented. This pipeline depicts the computation process of a GPU-accelerated scalar value sampling and compositing, which is based on the introduced optical model. Finally, we present several methodologies for real-time synchronization and display of the dual-modality 4-D cardiac images.

A. Optical Model

The optical model for volume rendering describes a numerical simulation of light passing through, and being reflected by, cloud-like regions within the volume, as a metaphor for the volume visualization integral. Optical models can be based on emission or absorption individually, or both, depending on the applications [33], [34]. To reduce the computational cost and balance between realism and computational complexity, a low albedo environment is usually assumed [35].

Suppose a viewing ray $\mathbf{r}(t) = \text{ray}(x(t),y(t),z(t))$, where $s(t_i) = s(\mathbf{r}(t_i))$ is the scalar (intensity) value at the sampling point $\mathbf{r}(t_i)$ on the viewing ray $\mathbf{r}(t)$. Equation (2) demonstrates the computational procedure of the light emission–absorption model, and the solution to this equation is given in the (3), showing the intensity of each pixel

$$\frac{dI(s(t))}{ds(t)} = c(s(t))\tau(s(t)) - I(s(t))\tau(s(t)) \tag{2}$$

$$I(D) = I_0 T(D) + \int_0^D c(s(t))\tau(s(t))T(t)dt. \tag{3}$$

Here $T(t) = \exp(-\int_0^t \tau(s(\omega))d\omega$ describes the volume transparency. The first term I_0 represents light coming from the background, and D is the extent of the ray over which light is emitted. The last term models the behavior of the incoming light. The source term $\tilde{c}(s(t)) = c(s(t))\tau(s(t))$ indicates the color change, and is referred to as associated color. The extinction coefficient $\tau(s(t))$ defines the occlusion of light per unit length due to light scattering or absorption. At each point $\mathbf{r}(t_i)$ $(i = 1, 2, \ldots, N)$ with intensity value $s(t_i)$ on the viewing ray $\mathbf{r}(t)$ (N is the total number of the sampling points along this ray), the function $\tilde{c}(s(t))$ describes the color emission, while $\tau(s(t))$ represents the color absorption. In the actual ray casting computation process, these two color parameters are not specified, so they cannot be used directly to calculate the visual appearance of the continuous scalar field. The scalar value $s(t)$ must be mapped to *RGB* color $C(t)$ and opacity $\alpha(t)$. This mapping process is called scalar value classification and the resulting function is referred to as a transfer function.

At each point $s(\mathbf{r}(t_i))$ on the casting rays, the transfer function mapped color $C(t_i)$ and opacity $\alpha(t_i)$ must be composited to generate the final volume rendered images. Alpha blending is commonly employed as an optical blending technique in this process, and is implemented by using the Riemann sum to discretize the continuous function (3), resulting in front-to-back and back-to-front alpha blending, which depends on the compositing order. Equation (4) illustrates the discretized version of the front-to-back alpha-blending function, which is used in the volume compositing procedure during the ray casting process of the software system illustrated in this paper

$$I(D) = \sum_{i=0}^{n} \alpha(t_i)C(t_i)\prod_{j=0}^{i-1} T(t_j). \tag{4}$$

Here, the premultiplied $\alpha(t_i)C(t_i)$ is the total light emitted from a point $t_i, T(t_i) = 1 - \alpha(t_j)(0 \leq T(t_i) \leq 1)$ is the transparency of a point, and $\alpha(t_i)$ is its opacity. During the ray casting process, for every cast ray $\mathbf{r}(t)$, the computation based on (4) must be performed at every point $\mathbf{r}(t_i)$. This is a computationally expensive procedure, and software-based computational algorithms are too slow for real-time applications. Modern GPUs are highly parallel single instruction multiple data (SIMD) processors, which can be employed to significantly accelerate the ray-volume intersection speed and the rendering efficiency. New generation graphics cards are designed to operate on large continuous streams of vertex and fragment data on GPUs and provide programmability to control the graphics pipelines at a low level. The pixel depth is also increased from 32 to 128 bits, allowing for higher precision scientific computing and graphics processing.

To deliver high-quality volume rendered images in real time, we designed an optimized data transformation strategy for transferring data between the main memory of CPU and the GPU graphics memory in real time, whose working pipeline is described in Section II-C and Fig. 3: since the raw MR and US data are monochrome, in our system, the internal voxel storage format of the 3-D texture for storing the MR and US images is set to be 32 bit, i.e., a single α component for storing the intensity values of the MR and US images, instead of the original 128 bit four components, i.e., *R*, *G*, *B*, and α channels, resulting a fourfold increase in texture transfer rate between CPU main memory and GPU.

B. GPU-Based Volume Visualization and Blending

First the sizes of the two datasets are equalized as described in Section II-C and Fig. 2. Then, these two raw volumetric cardiac datasets are loaded to the graphics memory as two 3-D textures, along with two 4-channel (*R*, *G*, *B*, α) color lookup tables which are similarly loaded as two 2-D textures. Two transfer functions are then constructed for each of the color tables. The opacity and color mapping pipeline of the transfer function is based on our artifact-free postcolor attenuated classification algorithm [27]: for every two adjacent points $\mathbf{r}(t_i)$ and $\mathbf{r}(t_{i+1})$ with the corresponding intensity values s_f and s_b on the casting ray $\mathbf{r}(t)$, assuming the segment length is d and the ray $\mathbf{r}(t)$ is uniformly sampled. Equations (5) and (6) represent the postcolor attenuated classification algorithm that is used to compute the mapped color $C = C(s_f, s_b, d)$, and opacity $\alpha = \alpha(s_f, s_b, d)$, which are employed to build the color and opacity lookup table

$$C \approx \left[1 - \frac{1}{2}\alpha(s_f, s_b, d)\right] \times \frac{d}{s_b - s_f}\left(\int_0^{s_b} \tau(\mu)c(\mu)d\mu - \int_0^{s_f} \tau(\mu)c(\mu)d\mu\right) \quad (5)$$

$$\alpha \approx 1 - \exp\left[-\frac{d}{s_b - s_f} \times \left(\int_0^{s_b} \tau(\mu)d\mu - \int_0^{s_f} \tau(\mu)d\mu\right)\right]. \quad (6)$$

The dual-modality (MR and US) volume visualization is computed on GPU, and the computation procedure is described in the following.

1) Scaling factors $F_\lambda = \max(l_x, l_y, l_z) \,/\, l_\lambda$, where l_λ is the volume side length along λ (λ = x, y, z) direction, are first computed for the x, y and z directions and then loaded to the GPU fragment shader.
2) In graphics memory, the two 3-D textures have the same texture coordinates, which are first loaded to the vertex shader, interpolated, and then output to the fragment shader to compute the casting ray entry points: entryPos = texCoord $\times F_{\vec{\lambda}}$, where $F_{\vec{\lambda}} = (F_x, F_y, F_z)$.
3) The camera position camPos is obtained by transforming the origin point with the inverse modelview matrix, which is employed with the entry point to calculate the ray direction: rayDir = (entryPos − camPos)$\times F_{\vec{\lambda}}$ / |entryPos − camPos|.
4) These two 3-D texture volumes are sampled at each sampling point along the casted ray $\mathbf{r}(t)$ with a specified stepsize d to acquire the corresponding voxel values $s_{\mathrm{mr}}(\mathbf{r}(t_i))$ and $s_{\mathrm{us}}(\mathbf{r}(t_i))$, which are then used to index into the corresponding color lookup table to obtain *R*, *G*, *B* color C_{mr}, C_{us} and opacity α_{mr}, α_{us} (the subscripts mr and us indicate the MRV and USV, respectively).
5) The mapping rules for each of these four channels are determined by the corresponding transfer function described in (5) and (6).
6) The derived colors and opacities from the two imaging modalities are composited together using the rules described by (7)–(9), and alpha blending is then employed to assign the derived color value to the destination color.
7) If the opacity has exceeded a preset threshold, or the casting ray has left the 3-D texture volume, the sampling computation is terminated; otherwise, the algorithm proceeds to the next sampling step.
8) The derived destination color is then alpha blended with the background color, and the result is set as the output fragment color.

Equations (7)–(9) illustrate the blending schemes used in the image fusing process computed on GPU fragment shaders

$$\begin{aligned} \alpha &= \gamma_{\mathrm{mr}} \cdot \alpha_{\mathrm{mr}} + \gamma_{\mathrm{us}} \cdot \alpha_{\mathrm{us}} \\ C &= \gamma_{\mathrm{mr}} \cdot C_{\mathrm{mr}} + \gamma_{\mathrm{us}} \cdot C_{\mathrm{us}} \end{aligned} \quad (7)$$

$$\begin{aligned} \alpha &= \gamma_{\mathrm{mr}} \cdot (1.0 - \alpha_{\mathrm{us}}) \cdot \alpha_{\mathrm{mr}} + \gamma_{\mathrm{us}} \cdot (1.0 - \alpha_{\mathrm{mr}}) \cdot \alpha_{\mathrm{us}} \\ C &= \gamma_{\mathrm{mr}} \cdot (1.0 - \alpha_{\mathrm{us}}) \cdot C_{\mathrm{mr}} + \gamma_{\mathrm{us}} \cdot (1.0 - \alpha_{\mathrm{mr}}) \cdot C_{\mathrm{us}} \end{aligned} \quad (8)$$

where α_{mr}, α_{us}, and α indicate the opacities of MR, US, and the fused dual-modality images, respectively, while C_{mr}, C_{us}, and C are the corresponding colors. The γ_{mr} and γ_{us} are rendering effect adjustment factors of the MR and US images, respectively. Unlike the blending algorithms described in (7) and (8), in which the blending computation is performed at every sampling step during the ray casting procedure, in the third blending method, the MR and US images are blended separately during the ray casting process, and the dual-modality blending is performed at the last step before the final image is created

$$\begin{aligned} \alpha_f &= \alpha_{\mathrm{bg}} \cdot \zeta + \gamma_{\mathrm{mr}} \cdot \lambda_{\mathrm{us}f} \cdot \alpha_{\mathrm{mr}f} + \gamma_{\mathrm{us}} \cdot \lambda_{\mathrm{mr}f} \cdot \alpha_{\mathrm{us}f} \\ C_f &= C_{\mathrm{bg}} \cdot \zeta + \gamma_{\mathrm{mr}} \cdot \lambda_{\mathrm{us}f} \cdot C_{\mathrm{mr}f} + \gamma_{\mathrm{us}} \cdot \lambda_{\mathrm{mr}f} \cdot C_{\mathrm{us}f}. \end{aligned} \quad (9)$$

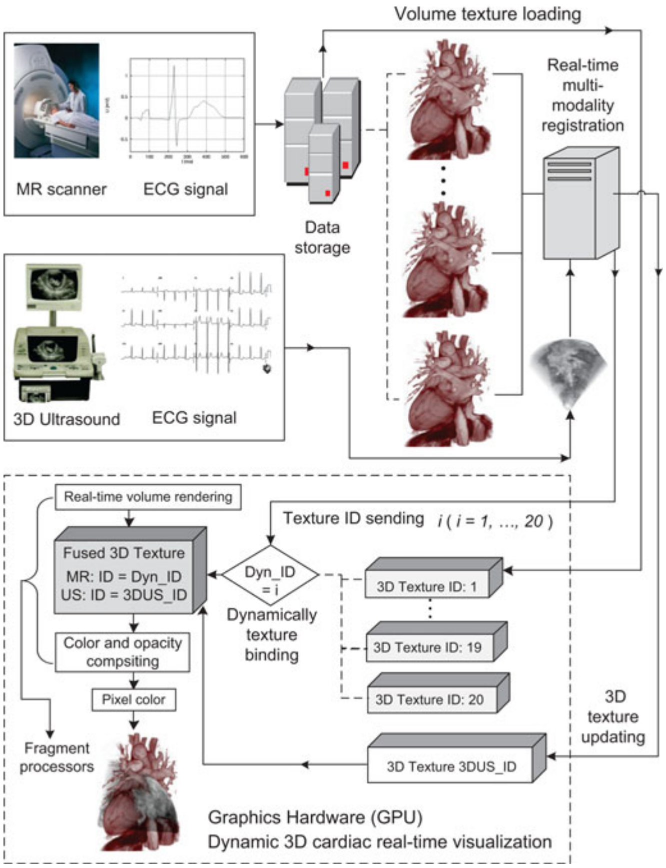


Fig. 4. Flowchart illustrating the process of cardiac dynamic real-time visualization. Twenty MR images are acquired during a single cardiac cycle with a 1.5-T GE CVi scanner, while timing information is derived from the ECG signal. Twenty MRVs are loaded into graphics cards as a set of volume texture, a dynamic binding technique is employed to real-timely bind different volume texture ID with our ray casting based 3-D rendering engine, which directly runs on the fragment processors.

Here, $\lambda_{\mathrm{us}f} = \max(1-\alpha_{\mathrm{us}f}/2.0, 0.0)$, $\lambda_{\mathrm{mr}f} = \max(1 - \alpha_{\mathrm{mr}f}$ / 2.0, 0.0), and $\zeta = 1.0 - (\gamma_{\mathrm{mr}} \cdot \alpha_{\mathrm{mr}f} + \gamma_{\mathrm{us}} \cdot \alpha_{\mathrm{us}f})$. C_f, $C_{\mathrm{mr}f}$, and $C_{\mathrm{us}f}$ are the colors of the final output blended image, MR, and US, respectively, while α_f, $\alpha_{\mathrm{mr}f}$, and $\alpha_{\mathrm{us}f}$ are their corresponding opacities. C_{bg} and α_{bg} are the background color and opacity.

C. Dual-Modality Cardiac Data Synchronization

As described in Fig. 4, prior to registering dynamic MR and US images, ECG-gated gradient echo T1-weighted 3-D MR images are first acquired at different phases over a single cardiac cycle. These data are stored in a server prior to the rendering process.

A 3-D US image of the heart then is acquired along with the tracking information and ECG data throughout the cardiac cycle. Since the US and MR datasets are derived with different acquisition rates, their frames are not temporally synchronized. ECG information is therefore employed to determine the two nearest MRVs corresponding to the cardiac phase at which the US image is acquired. These two MRVs are interpolated to produce a new MRV, to which the 3-D US image is registered to obtain a US transformation matrix M. The MRV and corresponding USV are processed with the techniques illustrated in Section II-C and the right-hand side of Fig. 2. These two registered 3-D images, along with their corresponding color lookup tables, are then loaded into the graphics memory to be visualized statically, as is described in the left-hand side of Fig. 2. The operator adjusts the transfer functions and the volume clipping planes to capture a volume of interest within this fused cardiac image, and this adjustment information is used to calibrate the initial transfer function and clipping plane settings during the dynamic dual-modality volume rendering procedure.

To register the US and MR images in real time, the dynamic 3-D US images acquired from a tracked US transducer are transformed to MR coordinates, and delivered to the display system along with timing information extracted from the ECG signal. To simulate an actual clinical application, we developed two approaches to synchronize the registered dual-modality 4-D cardiac images in real time.

1) Multitexture Preloading Approach: During the period between the calibration and the interventional procedure, the previously acquired MRVs are processed with the methods described in Section II-C, and then loaded into the graphics memory as a set of 3-D textures, MRV$_i$, where i is the texture number ($1 \leq i \leq N$, and N is the number of MRVs). Due to the fact that this approach requires the US images to be updated in real time, we therefore also create a 3-D texture USV$_{N+1}$ for the 3-D US image in the graphics memory, with all the texture elements initialized to zero before the procedure.

During the intervention, the dynamic 3-D US images with their associated ECG signals are acquired continuously. For each USV, the corresponding ECG signal is used to find the phase-coincident MRV, whose corresponding 3-D texture number can be assumed to be k ($1 \leq k \leq N$). These two 3-D images are registered in real time, providing us with a transformation matrix M. Note that because we cannot at present acquire a 3-D US image stream online, we perform offline registration with cardiac phases synchronized by the ECG signal. The 3-D US is first embedded within a large array in the CPU main memory, and then transformed by M. Image embedding is required to ensure that the US image remains within the extended volume after transformation.

Finally, the transformed 3-D US image is loaded into the graphics memory to update the 3-D texture USV$_{N+1}$. The texture number k is also loaded to the GPU to activate the corresponding MR 3-D texture by using dynamic texture binding. These two synchronized 3-D textures are composited and displayed by the GPU in real time, as illustrated in the left-hand side of Fig. 2.

2) ECG-Gated MRV Interpolation: In this dual-modality 4-D cardiac image synchronization approach, the preoperative MRV sequence MRV$_i$ ($i = 0, 1, \ldots, 19$) is not directly loaded into the graphics memory in advance. Instead, unique identity tags ID$_{\mathrm{mr}}$ and ID$_{\mathrm{us}}$ are set and loaded into GPU graphics memory to point to two texture memory blocks reserved to store the 3-D textures of MR and US, respectively. The two MRVs MRV$_i$ and MRV$_j$ (i, $j = 0, 1, \ldots, 19$) with the cardiac phase closest to that at which the USV$_{\mathrm{new}}$ is acquired are used to generate a new MRV$_{\mathrm{new}}$ using the acquired ECG information.

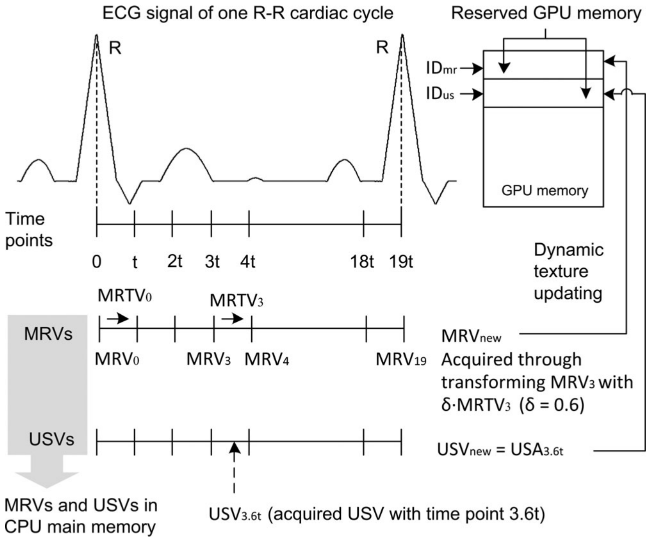

ECG signal of one R-R cardiac cycle
Reserved GPU memory
R
R
IDmr
IDus
GPU memory
Time points
0
t
2t
3t
4t
18t
19t
Dynamic texture updating
MRTV0
MRTV3
MRVs
MRV0
MRV3
MRV4
MRV19
MRVnew
Acquired through transforming MRV3 with δ·MRTV3 (δ = 0.6)
USVs
USVnew = USA3.6t
MRVs and USVs in CPU main memory
USV3.6t (acquired USV with time point 3.6t)

Fig. 5. Diagram illustrating the ECG-Gated MRV transformation and interpolation, which are used at the "dynamically texture binding" and 'fused 3-D texture" steps of Fig. 4.

Fig. 5 describes the detailed working pipeline with an example of a USV$_{\text{new}}$ being acquired at the time point 3.6t during a cardiac cycle from 0 to 19t. Assume the MRV dimension is $L \times M \times N$, as described in Section II-B, the MR transformation volume MRTV$_3$ for transforming MRV$_3$ to MRV$_4$ is used in the volume interpolation. For a voxel v^3_{abc} at the location (a, b, c) (a, b, and c are integers, and $0 \le a \le L, 0 \le b \le M$, and $0 \le c \le N$) in the MRV$_3$, the corresponding transformation vector (v^3_a, v^3_b, v^3_c) stored at the location (a, b, c) in the MR transformation volume MRTV$_3$ is used to transform v^3_{abc} to a new location ($\tilde{a}$, $\tilde{b}$, $\tilde{c}$) in the MRV$_4$ through transformation $a + v^3_a = \tilde{a}$, $b + v^3_b = \tilde{b}$, and $c + v^3_c = \tilde{c}$. Now for generating the new MRV$_{\text{new}}$ located between MRV$_3$ and MRV$_4$, the volume interpolation factor $\delta = 0.6$, which is used to transform v^3_{abc} to a new location ($\hat{a}$, $\hat{b}$, $\hat{c}$) in the MRV$_{\text{new}}$ through transformation $a + \delta{\cdot}v^3_a = \hat{a}$, $b + \delta{\cdot}v^3_b = \hat{b}$, and $c + \delta{\cdot}v^3_c = \hat{c}$.

Finally, the newly generated MRV$_{\text{new}}$ and the corresponding USV$_{\text{new}}$ are loaded into the GPU graphics memory, updating the two 3-D textures pointed by ID$_{\text{mr}}$ and ID$_{\text{us}}$, respectively, which are fused and visualized in real time by using our GPU-accelerated volume rendering engine.

D. Extended Window Level

MR images are typically recorded with 4096 intensities with each pixel of the image represented by a 12-bit value. However, since the diagnostic information is typically contained within a much smaller range of values, it is necessary to display only a selected window width of these values at a particular level. The top two images in Fig. 6 describe the window width and level mapping operation.

During the rendering process, from the analysis of the histogram of 3-D US, we note that the features of interest of US image are restricted to a narrow voxel intensity range, as illustrated in Fig. 6 (middle-left). Since it is difficult to display these structures clearly or enhance them, we proposed an extended window-level algorithm to address this issue. This method is similar to the standard window-level approach [36], but it is designed for 3-D image mapping and contrast enhancement. As illustrated in the middle-right panel of Fig. 6, this algorithm expands the intensity range of interest within the window width, while shrinking the intensity range of other regions, instead of simply mapping them to white or black.

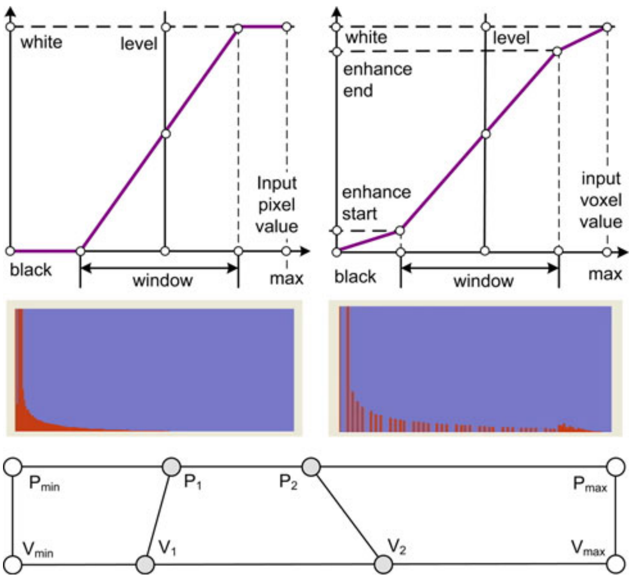


Fig. 6. Window-level (top) with corresponding mapped 3-D US histogram (middle): traditional (left) and extended (right). Bottom: data mapping in extended window-level for rendering 3-D medical images.

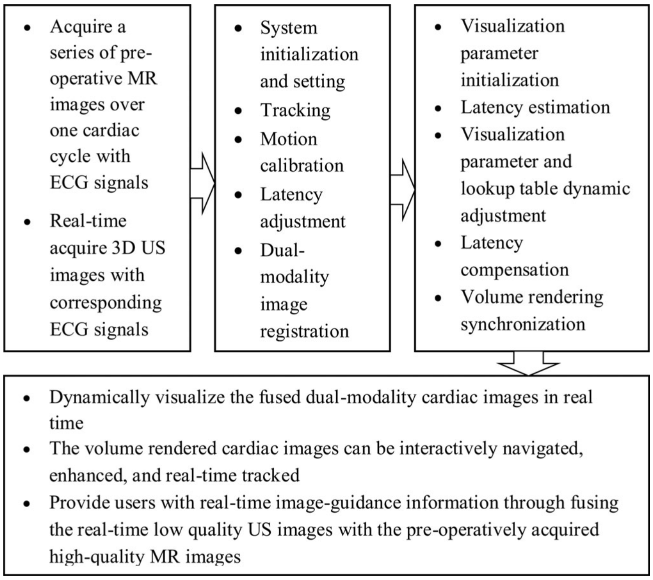
As demonstrated in the bottom row of Fig. 6, we first calculate the minimum and maximum value of the 3-D US image, obtaining $P_{\min}$ and $P_{\max}$, and then select the intensity range of interest $[P_1, P_2]$. Next, map it to an enlarged intensity range $[V_1, V_2]$, and finally, every voxel in the 3-D US image is mapped to a new intensity value using (10).

$$\hat{P} = \begin{cases} (P - P_1) \times \alpha_1 + V_1, & \text{if } P_1 \le P \le P_2 \\ V_{\min} + (P - V_{\min}) \times \alpha_0, & \text{if } P_{\min} \le P \le P_1 \\ V_2 + (P - P_2) \times \alpha_2 & \text{if } P_2 < P < P_{\max} \end{cases} \quad (10)$$

Here $P_{\max} \ge P_2 > P_1 \ge P_{\min}$, $P_{\min} = V_{\min}$, $P_{\max} = V_{\max}$, and $\alpha_0 = (V_1 - V_{\min})/(P_1 - V_{\min})$, $\alpha_0 = 0$ when $P_1 = V_{\min}$. $\alpha_1 = (V_2 - V_1)/(P_2 - P_1)$, and $\alpha_2 = (V_{\max} - V_2)/(V_{\max} - V_{\min})$, $\alpha_2 = 0$ when $V_{\max} = P_2$. The images at the bottom row of Fig. 13 and the middle and right of Fig. 14 show the extended histogram of the 3-D US images, from which we observe that the intensity range of interest is extended.

E. Possible Diagnostic and Interventional Applications

Fig. 4 illustrates the workflow of dual-modality cardiac image real-time visualization pipeline. To further explain the possible clinical application, the flowchart (see Fig. 7) describes a possible application pipeline.



- Acquire a series of pre-operative MR images over one cardiac cycle with ECG signals
- Real-time acquire 3D US images with corresponding ECG signals

- System initialization and setting
- Tracking
- Motion calibration
- Latency adjustment
- Dual-modality image registration

- Visualization parameter initialization
- Latency estimation
- Visualization parameter and lookup table dynamic adjustment
- Latency compensation
- Volume rendering synchronization

- Dynamically visualize the fused dual-modality cardiac images in real time
- The volume rendered cardiac images can be interactively navigated, enhanced, and real-time tracked
- Provide users with real-time image-guidance information through fusing the real-time low quality US images with the pre-operatively acquired high-quality MR images

Fig. 7. Workflow illustrating the possible clinical applications of the GPU-based dual-modality cardiac image real-time visualization and synchronization system.

IV. Experimental Results and Analysis

We have implemented our real-time 4-D multimodal cardiac image display system in such a manner that it can run on a system which is minimally equipped with a GeForce 8800 GTS graphics card with 640-MB memory. In our implementation, we employed an Intel Core 2 Duo E8400 Dual Core 3.0-GHz Processor and 4-GB main memory.

Programming languages Qt, C++, OpenGL, and OpenGL Shading Language (GLSL) were employed in the software development and algorithm implementation. To improve the visibility of the anatomical structures in the volume rendered images, and to satisfy clinical applications, the image display window dimension of this real-time medical image display system is set 800 × 800, and the camera is at (0, 0, 2.5) in the OpenGL coordinate system with respect to the original reference point (0, 0, 0).

The visualization pipeline is based on the rendering optical model described in (3), and the compositing order is front to back as illustrated in (4). The volume visualization procedure is as described in Section II-C, and the color lookup table is built using the post-color attenuated classification algorithm as presented in (5) and (6). The real-time process to synchronize the two image volumes is depicted in Section III-C1, and the results are demonstrated in the fused cardiac images in Figs. 8, 10, 11, as well as 14, and validated by experts.

A. Image Quality Analysis

To evaluate the effect of different dual-modality cardiac image visualization and fusion strategies, we invited twenty observers to rank the volume rendered dual-modality images based on criteria such as image quality, anatomic information delivery, the intuitive nature of the fused images, and personal preference. The observers include medical imaging researchers, clinicians, as well as radiologists and cardiologists with knowledge of cardiac anatomy and medical imaging. In this section, the results are rank ordered with one being the best.

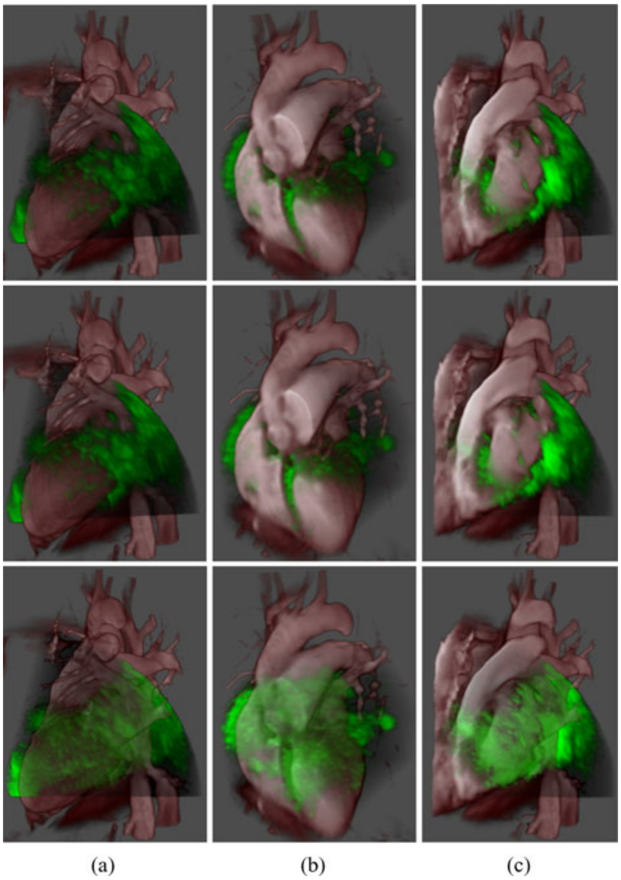


Fig. 8. Illustration of rendered dual-modality (MR and US) cardiac volumes of the registered real-time transthoracic 3-D US and preoperative MR images using different voxel compositing algorithms during the visualization pipeline. The images in the first, second, and third rows are created with the blending (7), (8) and (9), respectively.

TABLE I
Image Quality and Fusion Result Comparisons of the Images in Fig. 8

Rank	a_1	a_2	a_3	b_1	b_2	b_3	c_1	c_2	c_3
No.1	10	15	4	9	13	4	6	14	3
No.2	7	4	2	9	7	0	12	5	0
No.3	3	1	14	2	0	16	2	1	17

1) Volume Composition: We implemented and evaluated the three dual-modality volume compositing algorithms expressed by (7)–(9). The results are demonstrated by the images in the first, second, and third row of Fig. 8, respectively. Table I shows the evaluation results. The subscript i $(i = 1, 2, 3)$ of the letters a, b and c represents the corresponding row in Fig. 8, indicating that the corresponding images are generated with (7)–(9), respectively.

Fig. 9 shows the average ranking numbers of the three algorithms with standard deviation (SD) (top). To quantify the precision of the mean, we also calculate the 95% confidence interval (CI) (Fig. 9, bottom), from which we can see that the images generated with Alg3 are ranked lowest. The average highest ranking numbers of the images created with Alg1 and

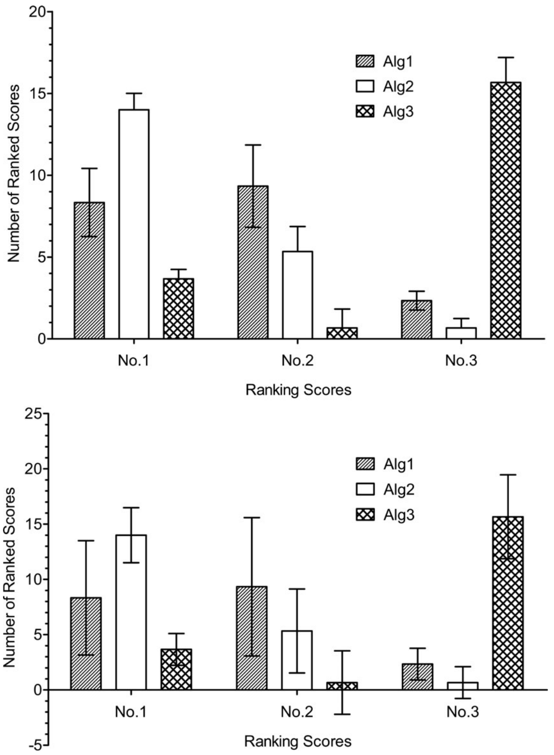


Fig. 9. The average ranking of the visualization results with standard deviation (SD) (top) and 95% confidence interval (CI) (bottom) of the three groups of images, i.e., group (a), (b), and (c), in Fig. 8 using algorithms Alg1 (images in the first row), Alg2 (images in the second row) and Alg3 (images in the third row), which correspond to the (7)–(9), respectively.

Alg2, whose rankings are around 2 and 4 times of the images created with Alg3. The images with Alg2 are ranked highest, having an average 30% and 50% higher than those of the images with Alg1 and Alg3, respectively, and have the lowest SD in the highest ranking in the three groups of images. However, its 95% CI is narrower than that of Alg1 in the most highly ranked group.

Generally, the observers believed that the dual-modality images composed with Alg2 deliver more naturally fused imaging results, and around 70% of the reviewers cannot visually distinguish the difference between the first two rows of images generated with Alg1 and Alg2. In addition, observers, who ranked the images created with Alg3 most highly, preferred the higher prominence of the US image, while those ranking it lowest thought that the depth of information was lost. Furthermore, observers who ranked the images generated with Alg1 the highest believed that those images delivered the optimal information.

As described in Sections IV-A2 and IV-A3, adding volume shading and extended window level will enhance the reality and feature display of the images visualized in Fig. 8; however, we did not find an obvious difference in the overall observer ranking of the image qualities.

2) Volume Shading and Weighting Factor: As part of our visualization platform, we supplied the options of adding volume shading to the dual-modality volume rendering pipeline, as shown in Fig. 10. In the bottom row, volume shading was added to the MR images of the four dual-modality images in the columns (a)–(d), and the US images of the two dual-modality images in the columns (b) and (d). As a comparison, in the top row of Fig. 10, the four dual-modality images in the columns (a)–(d) were rendered without volume shading on either image modality. Furthermore, the image display methods include clipping both image modalities, e.g., images in the columns (a) and (b) of Fig. 10, or only partly clipping the MR images and rendering the clipped MR images semitransparently, e.g., images in the columns (c) and (d).

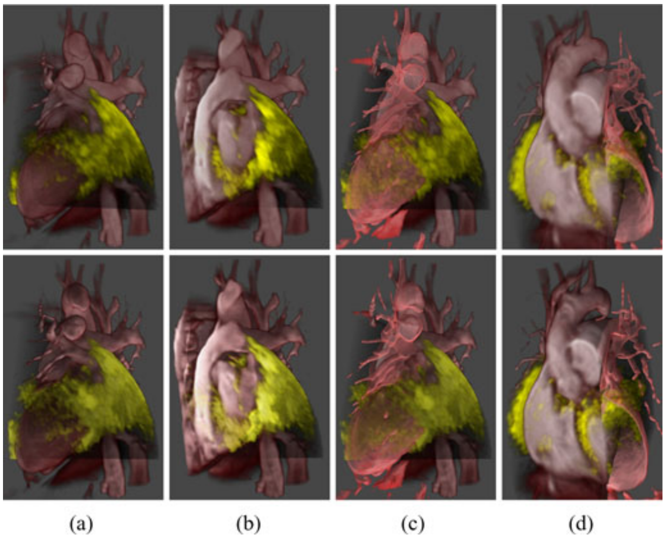


Fig. 10. Illustration of volume shading: in the top row, no volume shading is added to either US images or MR images of the four dual-modality images in the columns (a)–(d); in the bottom row, volume shading is added to the MR images of the four dual-modality images in the columns (a)–(d), and the US images of the two dual-modality images in the columns (b) and (d).

TABLE II
VOLUME SHADING AND WEIGHTING FACTOR: a_1–d_1 FOR IMAGES WITHOUT VOLUME SHADING, WHILE a_2–d_2 FOR IMAGES WITH SHADING. IMAGES e_1 and f_1 ARE RENDERED WITHOUT WEIGHTING FACTOR ADJUSTMENT, WHILE IMAGES e_2 and f_2 ARE GENERATED USING WEIGHTING FACTOR MANIPULATION

	Volume Shading								Weighting Factor			
Ranking	a_1	a_2	b_1	b_2	c_1	c_2	d_1	d_2	e_1	e_2	f_1	f_2
No. 1	10	10	8	12	12	8	8	12	10	10	11	9
No. 2	10	10	12	8	8	12	12	8	10	10	9	11

Even though improved realism can be observed in the volume shaded MR images, the diagnostic details in the US images are lost within the volumetric shading, which decreases the overall diagnostic value of the volume shaded dual-modality images. Table II and Fig. 11 confirm this conclusion, i.e., MR shading enhance the overall rankings by ∼10.5%. Furthermore, both the groups with and without shading have the same SD $\approx$ 1.9. Therefore, in clinical applications, we recommend adding shading to one image modality (MR image) while leaving the US image unshaded in the dual-modality display.

Fig. 12 shows the rendering results of fused images created with weighting factor adjustment. (e_1) and (f_1) are rendered with weighting factors equal to 1.0; while (e_2) and (f_2) are displayed with weighting factor adjustment. For (e_2), the weighting factor of MR image γ_{mr} is 0.6, while US adjustment factor $\gamma_{ur} = 1.4$. In (f_2), $\gamma_{\mathrm{mr}} = 0.58$, and $\gamma_{\mathrm{us}} = 1.3$. In Fig. 12 ($e_1$) and

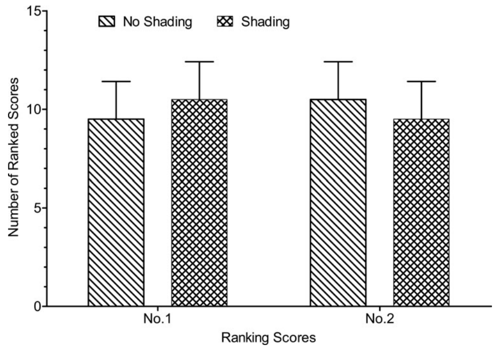


Fig. 11. The average ranking of the images with SD in Fig. 10.

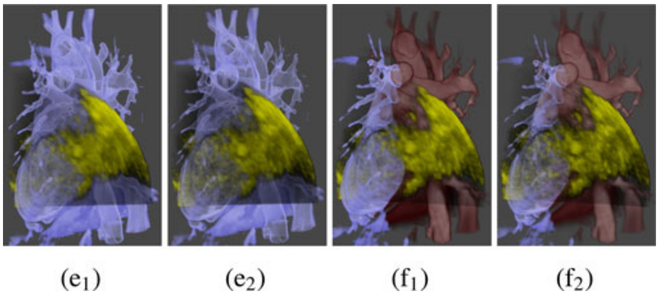


Fig. 12. Illustration of weighting factor adjustment. For images (e$_1$) and (f$_1$), the adjustment factors $\gamma_{\rm mr} = 1.0$, $\gamma_{\rm us} = 1.0$. (e$_2$): $\gamma_{\rm mr} = 0.6$, $\gamma_{\rm us} = 1.2$; and (f$_2$) $\gamma_{\rm mr} = 0.58$, $\gamma_{\rm us} = 1.3$.

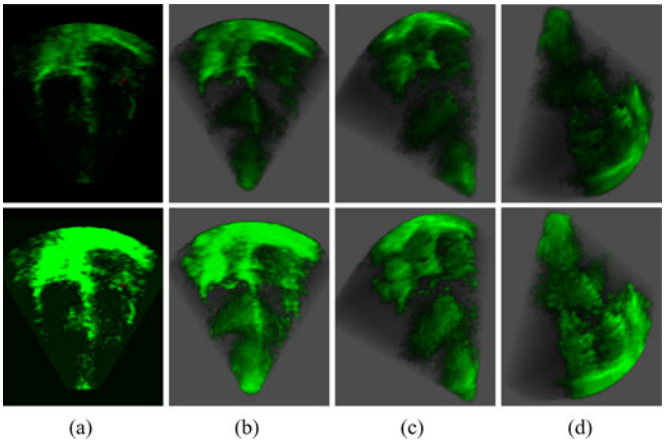


Fig. 13. 2-D and 3-D visualization of US images. In the second row, the images are enhanced with extended window-level (W-L) mapping.

(e$_2$), the MR is rendered semitransparently, so we can observe the US image inside the MR. In Fig. 12 (f$_1$) and (f$_2$), only the MR images are clipped, and the clipped regions are rendered semitransparently. Furthermore, the interactive weighting factor adjustment can enhance a specific imaging modality and emphasize anatomical features, as well as providing users with manipulation flexibility. Table II tabulates the adjustment factors, showing their variable effects on the final results, depending on the rendering images and specific adjustment factors applied.

3) Extended Window Level: Fig. 13 shows US images with and without structure enhancement. Column (a) is a 2-D display of a slice through the 3-D volume, where the window and level are 254 and 127 in the top row image, while in the bottom row image, the window and level are 74 and 37, respectively, which demonstrates the image structure enhancement. The images in the last three columns are the volume rendered US images, and in the bottom row, the extended window-level (W-L) setting is employed for tissue structure enhancement. In the image in the bottom row of column (b), the parameters $P_{\min}$, $P_{\max}$, $V_{\min}$, and $V_{\max}$ in (10) are set to be 3, 80, 0, and 180, respectively, while in the bottom row of the columns (c) and (d), the parameters are $P_{\min} = 2$, $P_{\max} = 140$, and $V_{\min} = 0$, $V_{\max} = 230$.

Generally, the bottom row images of columns (a) and (b) are too bright and the structural details are lost, while in the top row the images in these two columns are too dim to show the anatomical structures clearly. However, in the last two columns, even though the bottom images are also adjusted with the extended W-L, different W-L parameters are used, which have larger mapping range than those employed in the column (b), resulting in enhanced tissue structures without losing anatomical detail. As shown in Table III and Fig. 15 (top), appropriate settings of extended W-L can enhance the display of anatomic information.

TABLE III
WINDOW-LEVEL (W-L): IMAGES a$_1$ ∼d$_1$, AND e$_1$ ARE RENDERED WITHOUT W-L ADJUSTMENT; IMAGES a$_2$–d$_2$, e$_2$ AND e$_3$ ARE RENDERED WITH W-L ADJUSTMENT

	US Image								US+MR Image		
Ranking	a$_1$	a$_2$	b$_1$	b$_2$	c$_1$	c$_2$	d$_1$	d$_2$	e$_1$	e$_2$	e$_3$
No. 1	10	10	11	9	5	15	4	16	3	2	15
No. 2	10	10	9	11	15	5	16	4	9	9	2

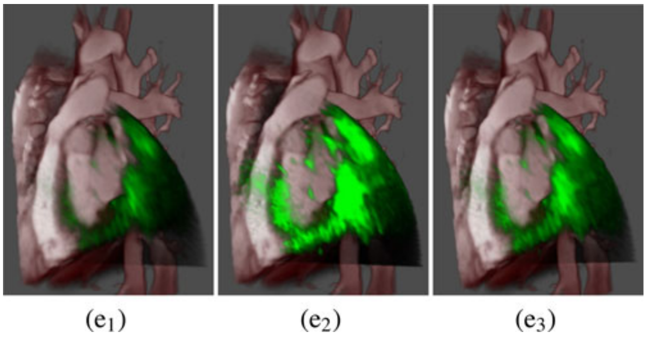


Fig. 14. Illustration of the extended window-level (W-L) effects in the volume rendered registered dual-modality (MR and US) cardiac volumes. The US images in columns (e$_2$) and (e$_3$) are emphasized with extended window-level mapping.

Fig. 14 demonstrates the display results of using extended W-L in the dual-modality volume rendering for US image structure enhancement. Column (e$_1$) shows image displayed without W-L adjustment, while the images in the last two columns are created with the extended W-L setting. The W-L parameters used in column (e$_2$) are the same as those employed in the column (b) of Fig. 13, while the parameter settings of (e$_3$) are the same as (c) and (d) of Fig. 13. In Fig. 14, we note that the US image is too weak in column (a) to effectively show any anatomical detail, while it is too strong in (b), resulting in the loss of some small structure information. In column (c), the US image is

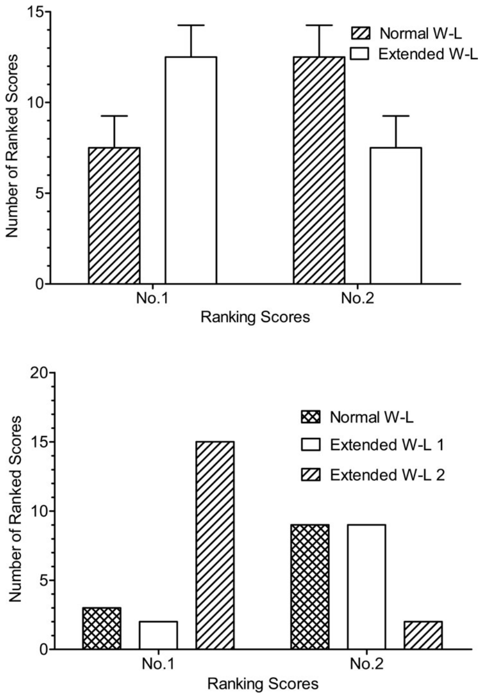


Fig. 15. The average ranking of images generated with and without using extended window level. Top: monomodality (US). Bottom: dual-modality (US+MR).

suitably enhanced without losing structure details. The survey result in Table III and Fig. 15 (bottom) confirms our conclusion, image (e_3) receives the highest ranking. We therefore conclude that with appropriate W-L parameter adjustments, the user can enhance the multimodality image display results to optimize the visualization of anatomical information.

B. Performance Evaluation

We have evaluated the 4-D dual-modality cardiac image volume rendering system using a series of 20 cardiac MR and 14 US images, and the two set images are registered together using MI based algorithm described in (1). Tables IV and V show the experimental results. For dynamic rendering, we used the multitexture preloading approach that was based on dynamic texture binding, and found that there was ~10% performance degradation in the the ECG-gated MRV interpolation method, a difference we attributed to the operation of physically uploading the MR and Vs from CPU main memory to GPU per frame in the MRV interpolation-based approach. In the following, we use *algorithm*+SH to represent a volume rendering algorithm (such as MR, US, or US+MRV rendering) with shading; and employ *algorithm*-SH to indicate a rendering algorithm without shading. In addition, we use *algorithm* to denote an algorithm, for which it is not specified whether or not shading is applied.

When we tested the static rendering algorithm with shading, we found that the frame-rates of both US+SH and US+MR+SH were around 30 fps, however, MR+SH can achieve 60 fps. At the same time, for static rendering without shading, US+MR-SH could achieve around 53 fps, while the other two data set rendering methods US-SH and MR-SH performed at around 30 fps. We attributed these two performance anomalies to the early ray termination technique[2] and the voxel intensity mapping pipelines, i.e., mapping voxel intensity to color and opacity through color lookup table, used in the rendering procedure.

For dynamic rendering, we did not observe any visible performance degradation caused by volume shading, with frame rates of around 15 fps in all cases with the exceptions of MR+SH and US+MR-SH, which achieved and ~19 fps, respectively. When we compared dynamic rendering with the static case, we noted ~50% (~15 fps) performance degradation in the US±SH, US+MR±SH, and MR+SH renderings. In addition, in the MR+SH and US+MR+SH renderings, the performance decline was up to ~66% (40 fps) and ~65% (34 fps), respectively. We attributed this performance loss to the MRV swapping in the GPU fragment shader and the USV updating from CPU to GPU during every rendering frame, and the transformation speed restrictions between CPU and GPU caused by the bandwidth limitations.

We had expected a performance degradation when comparing the dual-modality US+MR volume rendering with single-modality US or MR rendering, since in US+MR, both datasets were rendered simultaneously. However, in our experiment, we found that the US+MR-SH had higher performance than both US-SH and MR-SH rendering by ~78% and ~22%, respectively. In addition, in the case of volume rendering with shading, the speed of US+MR+SH and US+SH were almost the same, both of which were slower than the MR+SH rendering with the percentages of ~25% and ~49%, respectively.

For volume shading, we used preintegrated shading, during which the light position was fixed and the shading component was computed in the CPU main memory and then loaded into GPU. We found that there was only ~1% performance decline when employing the dynamic shading, even though this approach computes shading components on the GPU and consumes more graphics memory. On the contrary, in the US+MR case, we found ~3% speed increase when shading was included, a result we attributed to the early ray termination technique and the efficient volume shading component calculation speed on the GPU fragment shader. Finally, when we increased the sampling step by 50%, we were able to achieve ~100% performance increase without significant image quality degradation.

V. Discussion and Conclusion

In this paper, we presented methodologies and a software platform for visualizing, enhancing, and synchronizing the dual-modality 4-D cardiac volumes in real time using technologies such as GPU-based ray casting, interactive artifact-free voxel classification, as well as dynamic texture binding, overcoming the limitations of the traditional volume rendering approaches. To enable real-time display and manipulation of high quality 4-D dual-modality cardiac images and deliver optimal diagnostic information, we have integrated advanced techniques such as multiple transfer functions, texture sampling and storage

[2] For every casting ray during volume rendering process, stopping the color and opacity compositing computation when the accumulated opacity is larger than a preset threshold.

TABLE IV
STATIC AND DYNAMIC VOLUME RENDERING SPEED (FRAMES PER SECOND, FPS) COMPARISONS OF US, MR, AND US+MR IMAGES WITH AND WITHOUT VOLUME SHADING: INCLUDING PERFORMANCE DEGRADATIONS IN BOTH RENDERING SPEED (FPS) AND PERCENTAGE (%) DUE TO USING DYNAMIC VOLUME RENDERING.

	US		MR		US+MR	
	Shading	No Shading	Shading	No Shading	Shading	No Shading
Static (speed, fps)	30.3	30.0	59.9	30.0	30.6	53.4
Dynamic (speed, fps)	15.3	15.2	20.2	15.2	15.2	18.6
Degradation (speed, fps)	15.0	14.8	39.7	14.8	15.4	34.8
Degradation (percentage, %)	49.5	49.3	66.3	49.3	50.3	65.2

TABLE V
STATIC AND DYNAMIC VOLUME RENDERING PERFORMANCE (FRAMES PER SECOND, FPS) COMPARISONS OF US, MR, AND US+MR IMAGES WITH AND WITHOUT VOLUME SHADING: INCLUDING PERFORMANCE DEGRADATIONS IN BOTH RENDERING SPEED (FPS) AND PERCENTAGE (%) DUE TO USING DUAL-MODALITY VOLUME RENDERING

Rendering Mode	Shading	US	MR	US+MR				
		Speed	Speed	Speed	Degradation Comparison with US		Degradation Comparison with MR	
					Percentage (%)	Speed (fps)	Percentage (%)	Speed (fps)
Static	Yes	30.3	59.9	30.6	−1.00	−0.30	+48.9	+29.3
	No	30.0	30.0	53.4	−78.0	−23.4	−78.0	−23.4
Dynamic	Yes	15.3	20.2	15.2	+1.30	+0.20	+25.3	+05.1
	No	15.2	15.2	18.6	−22.4	−3.40	−22.4	−03.4

format optimization, texture transfer acceleration, volume shading, multiple volume clipping, and three dual-modality volume compositing algorithms. Furthermore, an extended window-level setting and adjustment approach was presented and applied to enhance the anatomical structure representations of US images. Finally, the resultant image quality and different multimodality image enhancement techniques were evaluated and compared, providing some useful information regarding the optimized display methods for dual-modality medical image volume visualization.

While we have presented a number of different multimodality visualization schemes for presenting fused 3-D US and MR data and statistically analyzed the results, it is still unclear which of these approaches will be the most optimal and effective in a clinical setting. To determine this, a task-oriented multiuser study must be performed to determine the effectiveness of each of these procedures for accomplishing specific tasks in both diagnostic and interventional contexts. In our approach, we have provided user-interface based controls on the display platform, so that users can tailor the visualization parameters according to their individual preferences.

Due to the access restrictions of live stream of US image volumes in current 3-D US scanners, preoperatively acquired data were used in the system. In the near future, when the live stream US data are available, a data capture card will be employed to acquire the live data in real time and store them in the CPU main memory for processing and visualization. Since there can be a latency of up to 100 ms in the process of data acquisition, registration, transformation, and visualization, before actual guidance, the system must be initialized through acquiring data one cardiac cycle in advance and storing them in CPU main memory and constructing and initializing a multistep color lookup table [26]. During an actual image guidance procedure, the real-time ECG signal and the corresponding data acquired one cardiac cycle in advance will be used for the dual-modality cardiac image registration and visualization. The new lookup table will be employed to compensate the latency through dynamically adjusting the volume rendering speed [26].

Using the preoperative 4-D MR scans with intra-operative 3-D US images in image-guided cardiac intervention has the potential benefit of providing surgeons with real-time high-quality guidance information to enable a new class of minimally-invasive cardiac interventions. We believe that this approach will ultimately greatly facilitate the interpretation of dynamic multi-modality medical image data in general, and play a pivotal role in minimally invasive image-guided procedures, particularly those aimed at therapy inside the beating heart.

ACKNOWLEDGMENTS

The authors would like to thank Dr. G. Guiraudon for clinical validation of the experimental results and software platform, Dr. X. Huang for providing the registration results, Mr. J. Moore for providing the cardiac MR and US images, and the insightful discussions with Drs. L. Ryner and M. Alexander. The authors would also like to thank for the constructive comments of the reviewers and editors for improving our research work described in this paper.

REFERENCES

[1] K. Ferdinand and D. Ferdinand, "Cardiovascular disease disparities: Racial/ethnic factors and potential solutions," *Curr. Cardiovasc. Risk Rep.*, vol. 3, pp. 187–193, 2009.

[2] G. Bajwa, T. Mihaljevic, M. Gillinov, and L. Svensson, "Minimally invasive aortic valve replacement," *Operat. Techn. Thoracic Cardiovasc. Surg.: Comparat. Atlas*, vol. 15, no. 4, pp. 307–320, 2010.

[3] J. D. Schmitto, S. A. Mokashi, and L. H. Cohn, "Minimally-invasive valve surgery," *J. Amer. College Cardiol.*, vol. 56, no. 6, pp. 455–462, 2010.

[4] M. Figl, D. Rueckert, D. J. Hawkes, R. Casula, M. Hu, O. Pedro, D. P. Zhang, G. P. Penney, F. Bello, and P. J. Edwards, "Image guidance for robotic minimally invasive coronary artery bypass," *Comp. Med. Imag. Graphics*, vol. 34, no. 1, pp. 61–68, 2010.

[5] K. Cleary and T. M. Peters, "Image-guided interventions: Technology review and clinical applications," *Ann. Review Biomed. Eng.*, vol. 12, no. 1, pp. 119–142, 2010.

[6] N. Noiseux, M. Ruel, and T. M. Hemmerling, "Minimally invasive cardiac surgery: new challenges for the surgeon and the anesthesiologist," *Techn. Regional Anesthesia Pain Manage.*, vol. 12, no. 1, pp. 72–79, 2008.

[7] S. G. Yuen, S. B. Kesner, N. V. Vasilyev, P. J. Nido, and R. D. Howe, "3-D ultrasound-guided motion compensation system for beating heart mitral valve repair," in *Proc. 11th int. conf. Med. Image Comput. Comput.-Assist. Intervent.—Part I*, (MICCAI '08 Series).. Berlin, Germany: Springer-Verlag, 2008, pp. 711–719.

[8] I. W. Wells, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Med. Image Anal.*, vol. 1, no. 1, pp. 35–52, 1996.

[9] X. Huang, J. Ren, G. Guiraudon, D. Boughner, and T. M. Peters, "Rapid Dynamic Image Registration of the Beating Heart for Diagnosis and Surgical Navigation," *IEEE Trans. Med. Imag.*, vol. 28, no. 11, pp. 1802–1814, Nov. 2009.

[10] F. P. P. Markelj, D. Tomaževič, B. Likar, "A review of 3-D/2D registration methods for image-guided interventions," *Med. Image Anal.*,vol. 16, no. 3, pp. 642–661, 2010.

[11] Q. Zhang, R. Eagleson, and T. M. Peters, "Volume visualization: A technical overview with a focus on medical applications," *J. Digit. Imag.*, vol. 24, no. 4, pp. 640–664, 2011.

[12] K. Xie, J. Yang, and Y. M. Zhu, "Real-time visualization of large volume datasets on standard pc hardware," *Comput. Methods Programs Biomed.*, vol. 90, no. 2, pp. 117–123, 2008.

[13] X. H. Wang and W. F. Good, "Real-time stereographic rendering and display of medical images with programmable gpus," *Comput. Med. Imag. Graphics*, vol. 32, no. 2, pp. 118–123, 2008.

[14] O. Kutter, R. Shams, and N. Navab, "Visualization and gpu-accelerated simulation of medical ultrasound from ct images," *Comput. Methods Programs Biomed.*, vol. 94, no. 3, pp. 250–266, 2009.

[15] S. E. Mahmoudi, A. Akhondi-Asl, R. Rahmani, S. Faghih-Roohi, V. Taimouri, A. Sabouri, and H. Soltanian-Zadeh, "Web-based interactive 2D/3-D medical image processing and visualization software," *Comput. Methods Programs Biomed.*, vol. 98, pp. 172–182, May 2010.

[16] Y. Dai, J. Tian, D. Dong, G. Yan, and H. Zheng, "Real-time visualized freehand 3D ultrasound reconstruction based on GPU," *IEEE Trans. Informat. Technol. Biomed.*, vol. 14, no. 6, pp. 1338–1345, Nov. 2010.

[17] R. A. Robb and D. P. Hanson, "Biomedical image visualization research using the visible human datasets," *Clin. Anatomy*, vol. 19, no. 3, pp. 240–253, 2006.

[18] H. Jelinek, D. Cornforth, and K. Blackmore, "Visualisation in biomedicine as a means of data evaluation," *J. Visualization*, pp. 1–7, 2011.

[19] J. Kim, W. Cai, S. Eberl, and D. D. Feng, "Real-time volume rendering visualization of dual-modality PET/CT images with interactive fuzzy thresholding segmentation," *IEEE Trans. Inf. Technol. Biomed.*, vol. 11, no. 2, pp. 161–169, Mar. 2007.

[20] X. Hu, K. K. Tan, D. N. Levin, S. G. Galhotra, C. A. Pelizzari, G. T. Y. Chen, R. N. Beck, C.-T. Chen, and M. D. Cooper, "Volumetric rendering of multimodality, multivariable medical imaging data," in *Proc. Chapel Hill Workshop Volume Visualization*. New York: ACM Press, 1989, pp. 45–49.

[21] D. J. Valentino, J. C. Mazziotta, and H. K. Huang, "Volume rendering of multimodal images: Application to MRI and PET imaging of the human brain," *IEEE Trans. Med. Imag.*, vol. 10, no. 4, pp. 554–562, Dec. 1991.

[22] K. J. Zuiderveld and M. A. Viergever, "Multi-modal volume visualization using object-oriented methods," in *Proc. Symp. Volume Vis.* New York: ACM Press, 1994, pp. 59–66.

[23] H. Hong, J. Baeb, H. Kye, and Y. G. Shin, "Hardware-accelerated multimodality volume fusion," in *SPIE Med. Imag.* R. L. Galloway and K. R. C., Jr., Eds., vol. 5744, San Diego, CA: SPIE, Feb. 2005, pp. 629–635.

[24] F. Enders, M. Strengert, S. Iserhardt-Bauer, U. E. Aladl, and P. J. Slomka, "Interactive volume rendering of multimodality 4-D cardiac data with the use of consumer graphics hardware," in *SPIE Medical Imaging*, vol. 5029, R. L. Galloway, Ed. San Diego, CA: SPIE, Feb. 2003, pp. 119–128.

[25] D. Levin, U. Aladl, G. Germano, and P. Slomka, "Techniques for efficient, real-time, 3D visualization of multi-modality cardiac data using consumer graphics hardware," *Comput. Med. Imag. Graphics*, vol. 30, no. 6, pp. 463–475, Sep. 2005.

[26] Q. Zhang, R. Eagleson, and T. M. Peters, "Dynamic real-time 4D cardiac MDCT image display using GPU-accelerated volume rendering," *Comput. Med. Imag. Graphics*, vol. 33, no. 6, pp. 461–476, 2009.

[27] Q. Zhang, R. Eagleson, and T. M. Peters, "Rapid scalar value classification and volume clipping for interactive 3D medical image visualization," *Visual Comput.*, vol. 27, no. 1, pp. 3–19, 2011.

[28] J. Moore, M. Drangova, M. Wierzbicki, and T. M. Peters, "A high resolution dynamic heart model based on averaged MRI data," in *Med. Image Comput. Comput.-Assisted Intervention Conf.* (Lecture Notes in Computer Science Series), R. E. Ellis and T. M. Peters, Eds. Berlin, Germany: Springer-Verlag, vol. 2878, 2003, pp. 549–555.

[29] M. Wierzbicki, M. Drangova, G. Guiraudon, and T. Peters, "Validation of dynamic heart models obtained using non-linear registration for virtual reality training, planning, and guidance of minimally invasive cardiac surgeries," *Med. Image Anal.*, vol. 8, no. 3, pp. 387–401, 2004.

[30] T. Mäkelä, P. Clarysse, O. Sipilä, N. Pauna, Q. Pham, T. Katila, and I. E. Magnin, "A review of cardiac image registration methods," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1011–1021, Sep. 2002.

[31] S. Kullback, *Information Theory and Statistics*. New York: Dover, 1968.

[32] Q. Zhang, R. Eagleson, and T. Peters, "High-quality cardiac image dynamic visualization with feature enhancement and virtual surgical tool inclusion," *Vis. Comput.*, vol. 25, pp. 1019–1035, 2009.

[33] N. Max, "Optical models for direct volume rendering," *IEEE Trans. Vis. Comput. Graphics*, vol. 1, no. 2, pp. 99–108, Jun. 1995.

[34] M. Hadwiger, J. M. Kniss, C. Rezk-salama, D. Weiskopf, and K. Engel, *Real-Time Volume Graphics*, 1st ed. Natick, MA: A. K. Peters, 2006.

[35] J. F. Blinn, "Light reflection functions for simulation of clouds and dusty surfaces," *SIGGRAPH Comput. Graphics*, vol. 16, pp. 21–29, Jul. 1982.

[36] R. B. Paranjape, "Fundamental enhancement techniques," in *Handbook of Medical Imaging*. I. N. Bankman, Ed. Orlando, FL: Academic, 2000, pp. 3–18.

Qi Zhang (S'08–M'12) received the B.Sc. and M.Sc. degrees in computational mathematics and applied software from the Jilin University, Changchun, China, the M.Sc. degree in computer science from the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree in biomedical engineering with computer science and engineering option from the Western University, London, ON, Canada.

He had worked as Postdoctoral Researcher at the Imaging Research Laboratories, Robarts Research Institute, and then as a Natural Sciences and Engineering Research Council Research Fellow at the Defense Research & Development Canada, Ottawa, ON, Canada. He is currently an Associate Research Officer at the National Research Council Institute for Biodiagnostics, Winnipeg, MB, Canada. His research interests include medical data visualization, computer graphics, medical image processing and analysis, high-performance computing, human-computer interaction, and GPU programming.

Roy Eagleson received the Ph.D. degree in 1992, supervised by Zenon Pylyshyn at the University of Western Ontario Centre for Cognitive Science.

He is an Associate Professor in the Faculty of Engineering, Western University, London, ON, Canada, an Associate Scientist at the Robarts Research Institute, London, and a Scientist and Principal Investigator at Canadian Surgical Technologies and Advanced Robotics Research Center, London Health Sciences Centre, Lawson Health Research Institute, London. His research interest includes 3-D biomedical visualization and surgical simulation, human-computer interface design for telerobotics and telesurgery, haptic interfaces and programmable graphical interfaces (GPU programming).

Terry M. Peters (F'09) received the B.Eng. and the Ph.D. degrees from the University of Canterbury, Ilam, New Zealand, in 1970 and 1974, respectively.

He is currently a Scientist in the Imaging Research Laboratories, Robarts Research Institute, London, ON, Canada, and a Professor in the Department of Medical Imaging, Medical Biophysics and Biomedical Engineering, Western University, London. From 1978 to 1997, he was at the Montreal Neurological Institute, Montreal, QC, before moving to his current position. His research interests include image-guided intervention where the focus of his laboratory is the development of techniques that integrate preoperative and intraoperative imaging modalities to guide minimally invasive therapy in the brain, heart, and abdominal cavity.

Dr. Peters is a Fellow of the Canadian College of Physicists in Medicine, the American Association of Physicists in Medicine, and the Institute of Electrical and Electronics Engineers.