Interactive 3D Navigation System for Image-guided Surgery

Huy Hoang Tran[1], Kiyoshi Matsumiya[1], Ken Masamune[1], Ichiro Sakuma[2, 3], Takeyoshi Dohi[1] and Hongen Liao[2, 3]

[1] *Graduate School of Information Science and Technology, The University of Tokyo.*
[2] *Graduate School of Engineering, The University of Tokyo.*
[3] *Translational Systems Biology and Medicine Initiative, The University of Tokyo.*

Abstract—**This paper presents a novel surgery navigation system based on a three-dimensional (3D) imaging technique, integral videography (IV). In our system, the 3D structure of the object of interest is reconstructed using surface rendering and corresponding pixel distribution methods. We developed a high-speed algorithm that renders high-quality IV images from the surface model in real time and allows interactions like rotating and scaling to be done smoothly. Using the patient-image registration method, IV images can be displayed with the correct size and relative position with respect to the surgical instruments. Experiments were carried out with various anatomical models, and the results show that our system could be useful in many clinical situations such as orthopedic surgery and neurosurgery.**

Index Terms—**Surgical Navigation, Integral Videography, Real-time Rendering, Registration.**

I. INTRODUCTION

Navigation systems based on pre-operative images (CT, MRI) and intra-operative images (ultrasound, fluoroscopy) have shown their usefulness in many clinical situations [1, 2]. There are especially useful in minimally invasive surgery (MIS) where specialized surgical instruments and arthroscopes are inserted through a small incision. In most of these systems, the 3D anatomical structures can be reconstructed properly by using advanced computer graphics algorithms but are often *rasterized* into 2D arrays of pixels for display. Using a normal 2D display may lead to difficulties in defining objects of interest or planning surgical paths, especially in orthopedic surgery [3]. Therefore, 3D display methods for surgery have become a focus of development in the field of navigation systems.

The stereo method is the one of the main 3D display technique used in surgical navigation systems [4, 5]. However, the need for a head mounted device and the lack of motion parallax have limited the performance of systems based on this method. In this paper, we introduce a novel navigation system based on the 3D integral videography (IV) [6, 7] visualization method. Unlike other methods, IV can display full-color, full-parallax 3D images without the need for special glasses; hence, it enables simultaneous observation by many persons [8]. However, the IV image costs much more computationally than a single 2D image, and therefore, interactions such as rotating

and scaling are comparatively much more difficult to perform. In this paper, we mainly focus on a high-speed IV rendering method that allows interactions to be done in almost real-time, while maintaining the rendered result at an acceptable quality. We also built a navigation system based on the open source software 3D-Slicer [9] open source software so that many tasks can be done in a single application.

This paper is organized as follows. Section II briefly describes the navigation system and patient-image registration method. Section III discusses the IV rendering algorithm. Experiments are described in section IV. Section V concludes this paper.

II. SYSTEM AND METHODS

2.1 Navigation system

Fig. 1 shows the configuration of the navigation system. The system consists of an image acquisition device (CT or MRI), optical position tracking device and main PC for data segmentation, communication and other computational tasks. The patient is scanned by X ray CT or MRI and 3D voxel data is sent to the main PC through RS232C. The data is then segmented and reconstructed as a 3D surface model. Patient to image registration is done prior to the operation. During the operation, the tracking device reads the positions of optical markers on the surgical tool and sends them to the main PC where the tool's orientation and the position of its tip are computed so that surgeons can know its correct position during operation. These processes are all controlled by the 3D Slicer module. Another module separately handles the IV rendering so that the IV image can be updated as the surgeons manipulate the surface model.
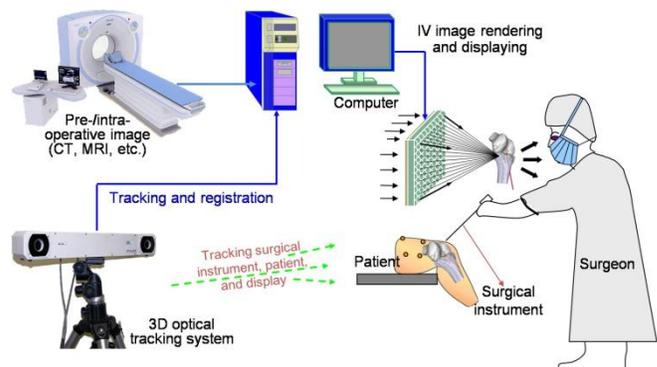

Fig. 1. System configuration.

2.2 Surgery's flow

Prior to the operation, a 3D model to be used for navigation is created from the CT data. Objects of interest are segmented from the data set on the basis of the pixels' values. Depending on the complexity of the object interest, the segmentation can be done automatically or semi-automatically. At this point, surgeons can locate the target or critical tissues (vessels, nerves) and plan the surgical path to avoid those parts.

The next task is to register the 3D model to the patient, which is called patient-to-image registration. The most common way to do this is to plant markers directly on the patient's body and perform multi-point registration. This method is effective but planting a marker on patient increases the invasive aspect of the surgery. Here, we shall assume that a minimally invasive registration method is to be used. By recording a set of surface data and comparing it with the surface model, patient-to-image registration can be done accurately and less invasively.

During the operation, images are displayed on normal and IV displays. Although the 3D structure is properly reproduced in the IV images, we still need a normal display for other information such as the distance to the target and the current position and orientation of the tool. Furthermore, and the spatial resolution of the IV image is limited, and small tissues might be properly visualized. For that reason, we built a synchronized display system that simultaneously displays normal and IV images.

2.3 Patient-image registration

When the registration tool's tip touches the bone surface, 3D information about the surface is acquired and compared with the 3D surface model of the bone. A registration mat1ix between the two surfaces is computed using iteration closest point (ICP) algorithm developed by Besl and Mckay [10] after the initial registration is done. After the transformation, the positional relation between the bone and surgical tool will be accurate enough so that surgeons can know the correct position of the tool during the operation.

The ICP algorithm is a common method to register two given surfaces or point clouds. For two point sets $S = \{S_1,...,S_N\}$ (the source or points acquired by the tracking device), and $T = \{T_1,...,T_M\}$ (the target surface model reconstructed on the PC), the closest function $C(S_i)$ returns the index of the closest point to S_i ($T_{c(S_i)}$ is closest to S_i). In each iteration step, the transformation (rotation and translation) (R,t) is the one that minimizes the error

$$d_k = \frac{1}{n} \times \Sigma_{i=1}^{N} \left\| T_{c\left(S_{K-1},I\right)} - \left(R_{K-1}S_{K-1,I} + t_{k-1}\right) \right\| \quad (1)$$

Here, K is the number of completed iteration, and $S_{k,i} = R_{k-1}S_{k-1,i} + t_{k-1}, S_{0,i} = S_i$. The iteration ends when the error converges to a given small value or the maximum number of iterations is reached. Although the ICP algorithm always monotonically converges to a local minimum, it needs an initial alignment of the two point sets to achieve the best performance;

otherwise the iteration would end up in a completely unexpected position. In this paper, making this initial alignment is called rough registration, meaning that a more accurate registration is done afterwards.

We built a 3D Slicer-based GUI for the rough registration task. The point cloud recorded by the tool is displayed in the main navigation viewer. Operators rotate and translate the point cloud to the position that best matches the surface model. Fig. 2 shows a model of a human tibia and the recorded point cloud that is to be used later for the ICP computation, together with the GUI for rough registration.
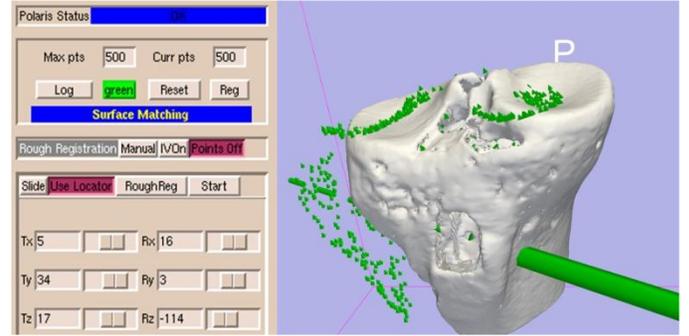

Fig. 2. GUI for initial alignment. (See Color Plate 3)

2.4 3D integral videography

IV records and reproduces 3-D images by using a micro convex lens array and a high-pixel-density flat display, e.g., an LCD display. This display is usually placed at the focal plane of the lens array so that light rays from the corresponding pixels will converge and form a single dot in physical space (Fig. 3). Many types of data can be processed to produce IV images of 3D objects. Here, we discuss the two main methods of making IV, the volume ray-casting method and the pixel distribution method [8].
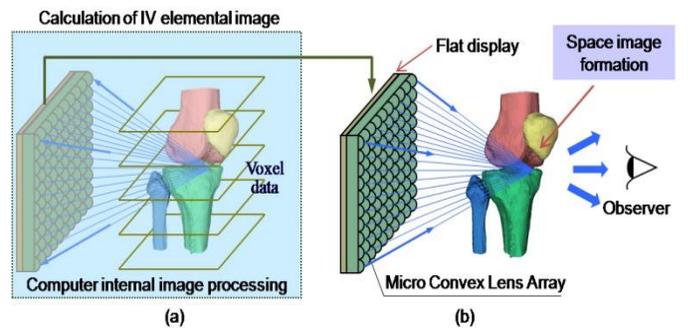

Fig. 3. Principle of integral videography; (a) Computer generated elemental images, (b) IV image spatial formation.

The volume ray-casting method directly processes the volume data (CT, MRI, US). It is basically an extended volume rendering method in which a light ray must go through a micro lens before intersecting with the screen.

The pixel distribution method constructs an IV image from a set of multi-view images acquired by geometrically based surface rendering. This method processes CG surface models, and therefore, it can produce high-quality images with many

visual effects. In addition, peripheral devices such as surgical tools can be visualized in the IV image as a simple CG model. For these reasons, our system uses pixel distribution as the primary method for rendering IV images.

Fig. 4 shows the principle of pixel distribution. Suppose that there are $L_X \times L_Y$ micro lenses in the lens array and each micro lens covers approximately $P_X \times P_Y$ pixels on the background image. We will need to capture $P_X \times P_Y$ images at $P_X \times P_Y$ viewpoints corresponding to the pixel number behind one lens. Each image must be at least of size $L_X \times L_Y$ so that each pixel will go to one corresponding micro lens.
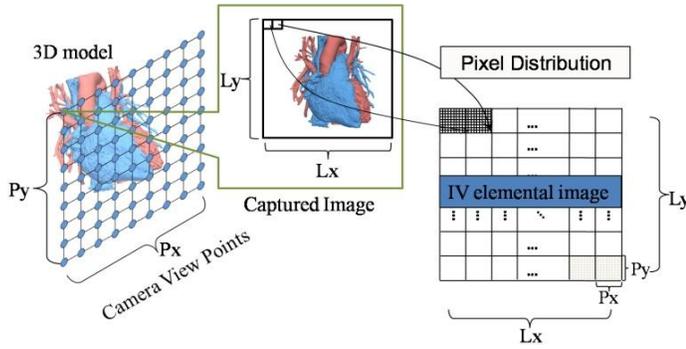

Fig. 4. IV rendering using the pixel distribution method.

Let us denote the pixel $(u, v)(u = 0..L_X, v = 0..L_Y,)$ of the image at viewpoint $(i, j)(i = 0..P_X, j = 0..P_Y,)$ as $P_{(i,j,u,v)}$ and the pixel (i, j) behind lens number (u, v) as $L_{(u,v,i,j)}$. Accordingly, the following relation holds,

$$P_{(i,j,u,v)} = L_{\left(u,v,P_{X-i},P_{Y-j}\right)} \tag{2}$$

Because of the number of pixels behind each lens, the number of required images varies from 50 to several hundreds. As the complexity of the rendered object increases, the computational cost becomes extremely high and conventional geometrically based CGs become unsuitable for real-time rendering, even with today's high-performance graphics processing units. A solution to the problem is to use the image-based rendering method in which a certain view of the object is generated from pre-acquired images [11, 12].

III. HIGH SPEED IV RENDERING USING THE PIXEL DISTRIBUTION

3.1 Light field formation and data acquisition

The main concept of image-based rendering is that each light ray is considered to be a function of position (u, v, w) and direction (ϕ, θ) (Fig. 5). Because the direction of the light ray can be defined by the *source* and the *target,* we can represent each ray as a 5D function $R(u, v, w, i, j)$ where (u, v, w) is the position of a viewpoint and (i, j) is the index of the pixel of the image captured at that viewpoint. This representation forms a

5D light space (light field) in which each captured image is a 2D slice [13, 14]. New images are rendered by extracting a bundle of light from the space by using interpolation. For convenience, the (u, v, w) plane is called the camera plane and the (i, j) plane is called the focal plane on which image is rendered.

The camera plane can be of any shape as long as it covers all the directions entering the light field. In fact it should be a closed surface surrounding the rendered object. Naturally, we chose a sphericface aal surnd data is sampled over this sphere.

Since the quality of the extracted image depends on the sampling rate on the camera plane, the density of sample cameras over the spherical surface should be uniform. This fact leads to a problem of how to uniformly sample a spherical surface. A solution to this problem is to distribute cameras over a surface based on a 20-face polyhedron, i.e., an icosahedrons (Fig. 6).
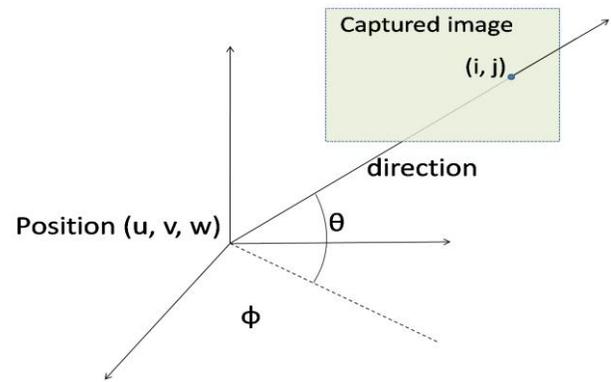
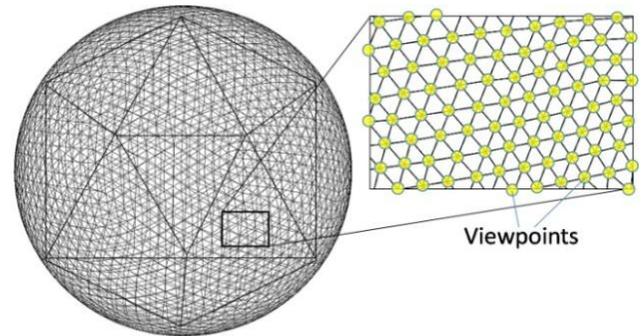

Fig. 5. Distribution of sample viewpoints.

Fig. 6. Distribution of sample viewpoints.

As shown in Fig. 6, each face of the icosahedrons is divided into smaller regular triangles and sample cameras are placed at their vertexes. This process continues until the sample cameras are dense enough. How dense is *dense enough* will be discussed in section 3.3.

3.2 Image extraction

Images are reconstructed by computing a bundle of light or an array of pixels with a given viewpoint. These computations are performed by interpolating the values of neighboring light rays. Given a new viewpoint and a new focal plane to render to, the interpolation is done as follows (Fig. 7)
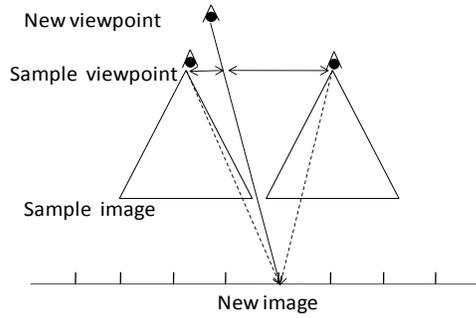
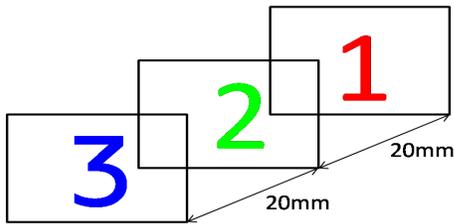Fig. 7.   Extraction of light rays by interpolating neighboring pixel values.

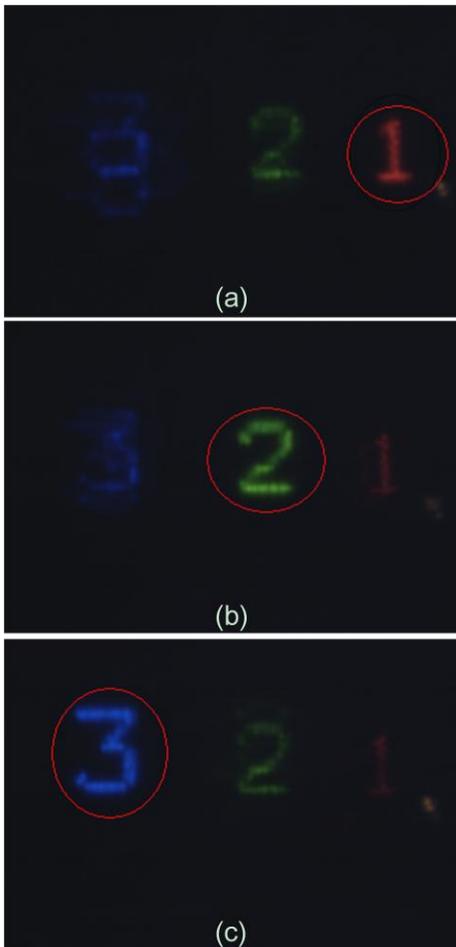Fig. 8. 3D scene with objects distributed in the depth direction.

Fig. 9.   IV images rendered with different focal planes. (a) Focal plane at position 1; (b) Focal plane at position 2; (c) Focal plane at position 3 as shown in Fig.8.

Fig. 10. Displayed 2D image (left) and corresponding IV image (right).

● Search for neighboring viewpoints

● Find the intersection points of all light rays and the new focal plane

● Perform interpolation on the focal plane and camera plane

3.3 IV image construction and user interaction

An IV image is constructed by rendering images from a set of viewpoints as discussed in section 2.2. The geometry of the viewpoint set is defined by the specifications of the lens array and the display. Light rays from two adjacent viewpoints must end up at two adjacent pixels through a single lens. In other words, the viewpoint set is the inverse magnification of the pixel set behind one lens, and the following equation holds:

$$\frac{d_p}{f} = \frac{d_v}{D} \qquad (3)$$

Here, d_P is the pixel pitch, d_V is the viewpoint pitch, f is the focal length of each lens in the lens array, which also means the gap between the lens array and the display, and D stands for the standard viewing distance from where we actually look at the IV image. From equation (3), we can compute d_v because the other parameters are constants in most cases. Since the viewpoint set is distributed at a spatial frequency $of 1/d_v$, the answer to the question in 3.1 is that cameras should be sampled at a rate of not less than $2/d_v$ (see the remark about the Nyquist-Shanon sampling theorem [15]).

Interactions by users, like rotating and scaling, are translated into viewing parameters. To be more specific, we make the following associations.

● Rotation → viewpoints position

● Scaling → unit length of the new image's focal plane

● Focusing → relative position of focal plane to object

One feature of our system is that users can focus on the desired location. Because of the configuration of the lens array, objects that are away from the image's focal plane (or the

display) will appear blurred in the IV image. This is because the density of light rays coming from the display is lower in the outer area. So with a scene with objects distributed in the depth direction, the user can set the important part to be "in focus" and the less important part to be "out of focus". Fig. 8 shows a scene with objects distributed in the depth direction, and Fig. 9 shows IV images rendered with a focal plane placed at different locations.

The user interface for interacting with the IV image is integrated in a 3D slicer. We use a high-resolution display for IV and a normal display for a 3D Slicer (Fig. 10). Both are connected to a PC through a DVI connection. Every change applied to the model is updated in the IV image so that interaction with the IV image can be done exactly the same way as with the CG model.

IV.   EXPERIMENTS AND RESULTS

4.1 IV rendering speed

The efficiency of our system was evaluated in terms of the rendering speeds of our method and the the conventional geometrically based method. We used an Intel Core 2 6600@2.4GHz CPU with 4GB of memory and a Geforce 8800 GTX with 768 MB memory GPU. The specifications of the IV display and the lens array are as follows.

- Display: XGA (1024×768), 203 dpi

- Pixel pitch: 0.125mm

- Lens array: Fly's eye lens

- Focal length: 3mm

- Lens pitch: 1.001mm×0.876mm

With the above configuration, each lens in the array covers about 8 pixels horizontally and 7 pixels vertically, which means we need to render images from 8x7 viewpoints for one IV frame. Furthermore, with the standard viewing distance of 300 mm away from the display, equation (3) implies that the viewpoint pitch should be 37.5 mm in both directions, requiring the sample rate of cameras to be greater than 2/37.5=0.053 samples per 1 mm. For this reason, each face of the icosahedron described in section 3.1 is divided into smaller triangles until the minimum sampling rate is satisfied, and the total number of sample cameras is 2562.

We used two models of different complexities: a human head model with 55424 vertexes and a heart model with 401587 vertexes (Fig. 11). Capturing images from sampling cameras took about 11 seconds for the head model and 18 seconds for the heart model and used 500MB of memory. Fig. 12 shows the motion parallax of the IV image of the heart model.

TABLE 1 show the average required times for 10 times for renderings of one frame. The rendering time of the conventional method increased significantly as the complexity of model increased, whereas the rendering time of the proposed method stayed almost constant regardless of the model. For an IV image of size 1024x768 pixels, the proposed method takes about 90 ms for 1 frame, meaning a frame rate of 11 fps. This frame rate enables smooth interaction regardless of the scene's complexity.

TABLE 1: RENDERING TIME FOR 1 FRAME (ms). (n=10)

Model	Head	Heart
Conventional method	346.8 ± 2.4	625.3 ± 3.9
Proposed method	92.1 ± 2.4	95.4 ± 3.7

4.2 Evaluation of image quality

Since images are extracted using interpolation in image-based rendering, there should be a reduction in image quality in comparison with that of the conventional method. We quantitatively evaluated the quality of the IV image made by the proposed method.

We used the modulation transfer function (MTF) to evaluate the spatial frequency of IV images. IV images made by the conventional and new method were captured under the same conditions with a digital camera (Nikon D1X, Micro-Nikkor 60mm f/2.8D). After that, 256x256 pixels in the center of each photo were cropped out and taken for evaluation (Fig. 13). They were then Fourier transformed into a frequency space with a horizontal frequency u and vertical frequency v (Fig. 14). Let the signal at (u,v) in the conventional method be $S(u,v)$ and the signal in the proposed one be $M(U,V)$, the relative MTF at (u,v) is defined as follow.

$$MTF(u,v) = \frac{M(u,v)}{S(u,v)} \qquad (4)$$

Because each lens in the lens array has a diameter of 1mm, the maximum spatial frequency that an IV image can produce is 1 cycle/mm. Therefore, we evaluated the relative MTF within the domain from 0 to 1 cycle/mm in both directions.
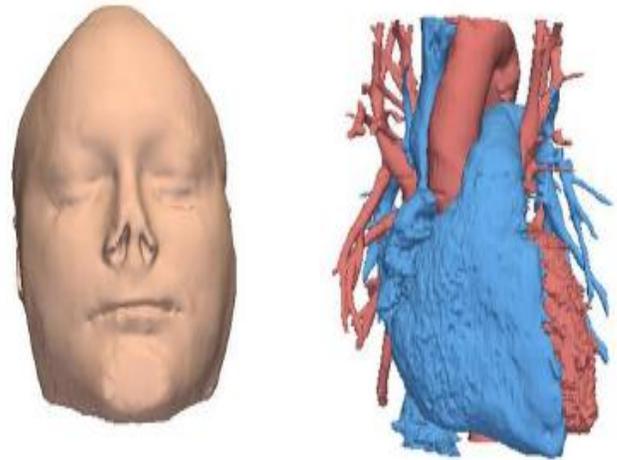

Fig. 11. Two medical datasets of different complexities. (See Color Plate 4)
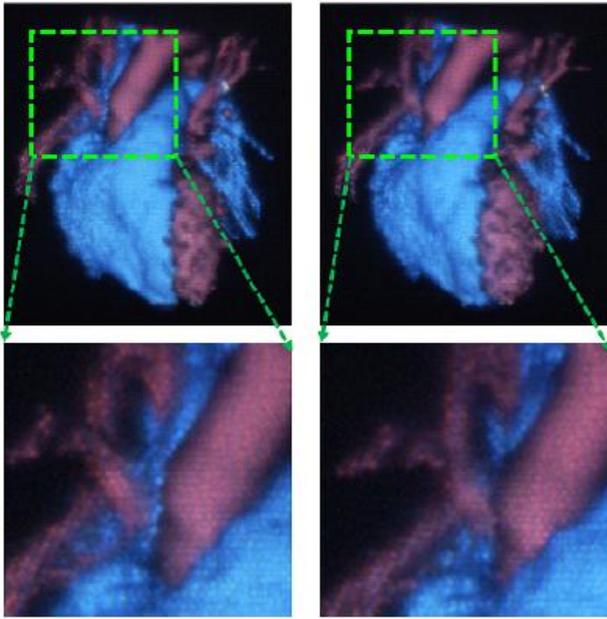
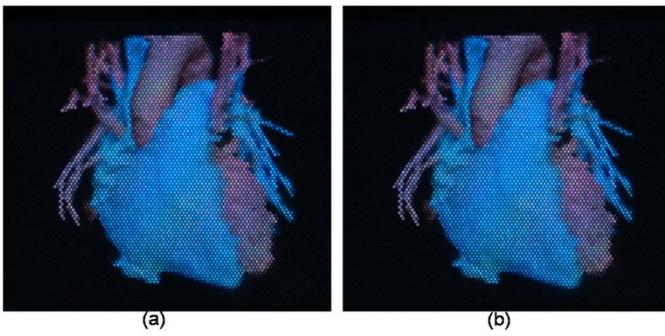Fig. 12. Motion parallax of IV image.

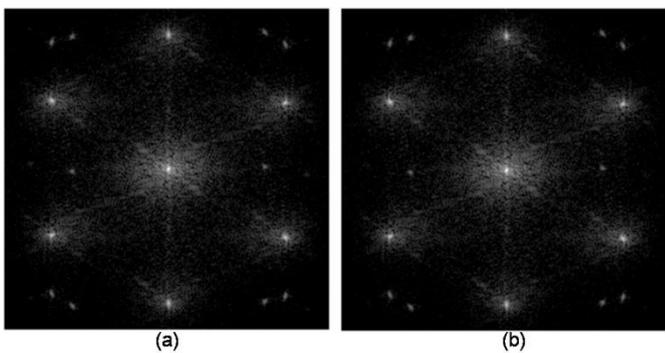Fig.13. IV images rendered with (a) conventional method and (b) proposed method.

Fig. 14. Fourier transformation of IV images. (a): conventional method, (b): proposed method.

Fig. 15 and 16 show the relative MTF when *u* and *v* are set to 0. While the horizontal frequency is almost the same in both methods (MTF=1), the vertical frequency is not properly reproduced. This is caused by the lens pitch of the array not being the same in the vertical and horizontal directions (Fig. 17).

From the experimental results, we can see that IV images rendered with the image-based technique show almost no loss in quality in comparison with the image rendered with the conventional CG. Therefore, our method is a promising way to render real-time IV images and can be utilized in various medical applications because the total memory required is only around 500MB.
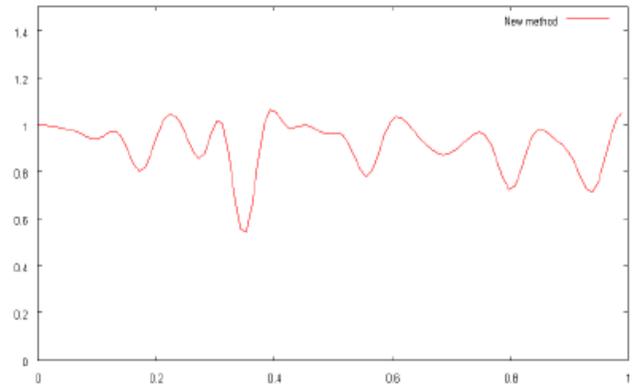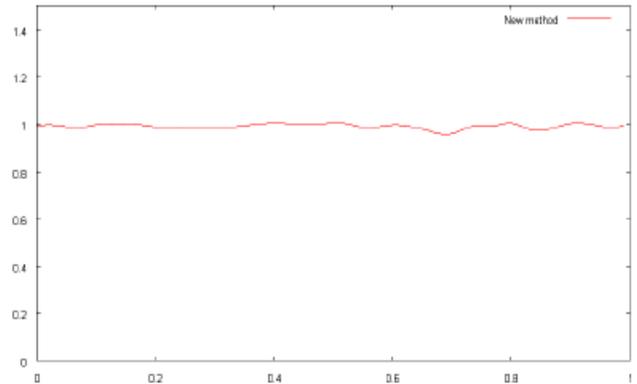

Fig. 15. Relative MTF of proposed method, u=0.
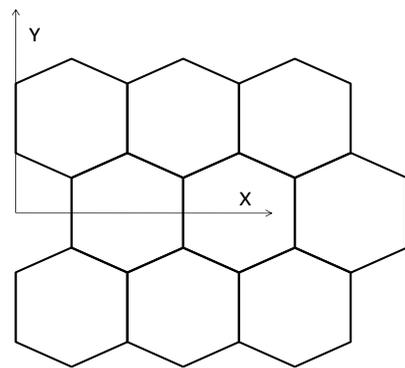

Fig. 16. Relative MTF of proposed method, v=0.

Fig. 17. Configuration of lens array

V.   CONCLUSION AND FUTURE WORK

We developed a high-speed, high-quality 3D IV surgical navigation system. By using image-based rendering, the system can render IV XGA images (1024x768 pixels) at approximately 11 frames per second while keeping the quality of the image at a high level. We also built a GUI interface through which

interactions with the IV image, like rotating, scaling, and focusing, can be smoothly done. By developing each part of the system as a 3D Slicer module, many tasks can be done in a single application, and therefore, our system is a compact yet powerful navigation system.

Regarding future improvements to our system, we think that most of the computational tasks can be processed in parallel and the rendering speed would be significantly increased by improving the GPU computing technique. Instead of displaying only still images, we intend to develop a movie-like IV display system for displaying periodical motion of an organ.

ACKNOWLEDGEMENT

REFERENCES

[1] S. Plaweski, J. Cazal, P. Rosell and P. Merloz. Anterior Cruciate Ligament Reconstruction Using Navigation, *The American Journal of Sports Medicine*, vol.34, pp.542-552, 2006.

[2] I. M. Germane, H. Villalobos, A. Silvers, K. D. Post. Clinical use of the optical digitizer for intracranial neuronavigation, *Neurosurgery*, vol.45, no.2, pp.261-269, 1999.

[3] F. S. Azar, N. Perrin, A. Khamene, S. Vogt, F. Sauer. User performance analysis. Of different image-based navigation systems for needle placement procedures, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006, pp.373-380, 2006.

[4] P. M. Novotny, S. K. Jacobsen, N. V. Vasilyev, et al.. 3D ultrasound in robotic surgery: performance evaluation with stereo displays, *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, pp. 279-285, 2006.

[5] W. Birkfellner, M. Figl, C. Matula, J. Hummel, R. Hanel, H. Imhof, F. Wanschitz, A. Wagner, F. Watzinger, H. Bergmann. Computer-enhanced stereoscopic vision in a head-mounted operating binocular, *Physics in Medicine and Biobogy*, vol. 48, pp. 49-57, 2003.

[6] H. Liao, S. Nakajima, M. Iwahara, E. Kobayashi, I. Sakuma, N.Yahagi, T. Dohi. Intra-operative Real-Time 3-D Information Display System based on Integral Videography, Fourth International Conference on Medical Image Computing and Computer assisted Intervention --MICCAI 2001, LNCS 2208, pp.392-400, 2001.

[7] H. Liao, N. Hata, S. Nakajima, M. Iwahara, I. Sakuma, T. Dohi. Surgical Navigation by Autostereoscopic Image Overlay of Integral Videography, *IEEE Trans. Information Technology in Biomedicine*, vol.8, no.2, pp.114-121, 2004.

[8] M. G. Lippmann. Epreuves reversibles donnant la sensation du relief, *J. de Phys.*, vol.7, pp.821-825, 1908.

[9] www.slicer.org.

[10] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes, *IEEE transactions on pattern analysis and machine intelligence*, vol.14, no.2, pp.239-256, 1992.

[11] T. Naemura, T. Yoshida and H. Harashima. 3-D Computer graphics Based on Integral Photography, *Optics Express*, vol.8 ,no.4, 2001.

[12] T. Naemura , J. Tago , H. Harashima. Real-Time Video-Based Modeling and Rendering of 3D Scenes, *IEEE Computer Graphics and Applications*, vol.22 ,no.2, pp.66-73, March 2002.

[13] M. Levoy, P. Hanrahan. Light field rendering, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp.31-42, 1996.

[14] W. C. Chen, J. Y. Bouguet, M. H. Chu, R. Grzeszczuk. Light field mapping: efficient representation and hardware rendering of surface light fields, *ACM Transactions on Graphics*, vol.21, no.3, July 2002.

[15] C. E. Shannon. Communication in the presence of noise, *Proc. Institute of Radio Engineers*, vol. 37, no.1, pp.10-21, Jan. 1949.

Huy Hoang Tran received his BS degree in mechano-informatics from the University of Tokyo in 2007, and is currently a MS student in the Graduate School of Information and Technology, The University of Tokyo. His research interests include 3D image processing, medical imaging and surgical navigation.

Kiyoshi Matsumiya received the B.S. degree from Waseda University, Japan, in 1999, and the M.S. and Ph.D. degrees in engineering from The University of Tokyo, Tokyo, Japan, in 2001 and 2004, respectively. He was a Research Associate with The University of Tokyo from 2004–2007. He is currently an instructor with the Okayama University of Science, Okayama, Japan. His research interest covers different biomedical engineering applications and surgical robotics for minimally invasive surgeries.

Ken Masamune was born in Tokyo, Japan, in 1970. He received the Ph.D. degree in precision machinery engineering from The University of Tokyo in 1999. From 1995 to 1999, he was a Research Associate in the Department of Precision Machinery Engineering, The University of Tokyo. From 2000 to 2004, he was an Assistant Professor in the Department of Biotechnology, Tokyo Denki University, Saitama, Japan, and between 2004 and 2005, he was an Associate Professor in the Department of Intelligent and Mechanical Engineering. Since August 2005, he has been an Associate Professor with the Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo. His main interest includes computer-aided surgery and, in particular, medical robotics, image guided surgery, and visualization systems for assisting surgical procedures to achieve minimally invasive surgery using MRI and other image information.

Ichiro Sakuma received the B.S., the M.S., and the Ph.D. degrees in precision machinery engineering from the University of Tokyo, Tokyo, Japan, in 1982, 1984, and 1989, respectively. He was Research Associate at the Department of Precision Machinery Engineering in Faculty of Engineering, the University of Tokyo from 1985 to 1987. He was an Associate Professor at the Department of Applied Electronic Engineering, Tokyo Denki University, Saitama, Japan, from 1991 to 1999 and an Associate Professor and a Professor at the Institute of Environmental Studies, Graduate School of Frontier Sciences, The University of Tokyo, from 1999 to 2001 and from 2001 to 2006. He is currently a professor at Department of Precision Engineering, School of Engineering, The University of Tokyo.
His research interests are in biomedical instrumentation, simulation of biomedical phenomena, computer-assisted intervention, and surgical robotics. He was the voice president of Japanese Society for Medical and Biological Engineering from 2006 to 2007. He is board member of Medical Image Computing and Computer Assisted Intervention (MICCAI) Society, Japan Society of Computer Aided Surgery, and Japanese Society of Electro cardiology.

Takeyoshi Dohi received the B.S., the M.S. and the Ph.D. degrees in Precision Machinery Engineering from the University of Tokyo, Tokyo, Japan in 1972, 1974 and 1977, respectively.

After a brief research fellowship in the Institute of Medical Science, The University of Tokyo, he joined Tokyo Denki University, Tokyo, Japan, in 1979 as a Lecturer and then became Associate Professor in 1981. From 1981 to 1988, he was an Associate Professor with The University of Tokyo in precision machinery engineering. Since 1988, he holds a full Professor position in The University of Tokyo, where he presently is a Professor of Information Science and Technology. His research interests include computer-aided surgery, rehabilitation robotics, artificial organs, and neuro-informatics.

Dr. Dohi has served as a president of numerous domestic society and international professional societies including the International Society for Computer Aided Surgery (ISCAS), The Japanese Society for Medical and Biological Engineering, The Japan Society of Computer Aided Surgery (JCAS). He was a board member of Medical Image Computing and Computer Assisted Intervention Society.

Hongen Liao received the B.S. degree in mechanics and engineering sciences from Peking University, Beijing, China, in 1996, and the M.E. and Ph.D. degrees in precision machinery engineering from the University of Tokyo, Tokyo, Japan, in 2000 and 2003, respectively. He was a Research Fellow of Japan Society for the Promotion of Science. He has been a faculty member at the Graduate School of Engineering, The University of Tokyo, Japan since 2004.

He is currently an Associate Professor of Department of Bioengineering, Graduate School of Engineering, The University of Tokyo. He is the author and co-author of more than 90 peer-reviewed articles in journals and proceedings, as well as over 150 abstracts and numerous invited lectures. His research interests include medical image, image-guided surgery, medical robotics, computer-assisted surgery, and fusion of these techniques for minimally invasive precision diagnosis and therapy. He has also been involved in long viewing distance autostereoscopic display and 3D visualization.

Dr. Liao was distinguished by receiving the government award [The Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology (MEXT), Japan]. He is also the recipient of more than ten awards including OGINO Award (2007), ERICSSON Young Scientist Award (2006), IFMBE Young Investigators Awards (2006, 2005), and several Best Paper Awards from different academic societies. His research is well funded by the Ministry of Education, Culture, Sports, and Technology (MEXT), Ministry of Internal Affairs and Communications (MIC), New Energy and Industrial Technology Development Organization (NEDO) and Japan Society for the Promotion of Science (JSPS) in Japan. He is an Associate Editor of IEEE-EMBC Conference, Organization Chair of MIAR 2008, Program Chair of ACCAS 2008, and Tutorial co-chair of MICCAI 2009.