J. Biomed. Eng. Res : 6-13, 2006

Fast Volume Visualization Techniques for Ultrasound Data

Koo-Joo Kwon, Byeong-Seok Shin

Department of Computer Science and Information Engineering, Inha University
(Received Sepetmber 21, 2005. Accepted January 17, 2006)

Abstract

Ultrasound visualization is a typical diagnosis method to examine organs, soft tissues and fetus data. It is difficult to visualize ultrasound data because the quality of the data might be degraded by artifact and speckle noise, and gathered with non-linear sampling. Rendering speed is too slow since we can not use additional data structures or procedures in rendering stage. In this paper, we use several visualization methods for fast rendering of ultrasound data. First method, denoted as adaptive ray sampling, is to reduce the number of samples by adjusting sampling interval in empty space. Secondly, we use early ray termination scheme with sufficiently wide sampling interval and low threshold value of opacity during color compositing. Lastly, we use bilinear interpolation instead of trilinear interpolation for sampling in transparent region. We conclude that our method reduces the rendering time without loss of image quality in comparison to the conventional methods.

Key words : ultrasound volume data, adaptive ray sampling, space leaping

I. INTRODUCTION

Ultrasound imaging is a widely known diagnosis method to visualize the distribution of ultrasonic echo throughout a volume of interest inside a patient. Although its image quality is worse than that of CT and MR images, it is widely used for diagnosis for its cost-effectiveness. Recently, 3D ultrasound devices are introduced, which acquire volumetric dataset by scanning an object using motor-driven probe array [1-4]. However, two major problems for the 3D ultrasound visualization are lower signal-to-noise ratio than CT and MRI and the fuzzy nature of the boundary surfaces in the ultrasound image. Sanches et al. describes several techniques to improve the efficiency of the surface reconstruction based multi-scale principles and based on the expansion of the likelihood function in a Taylor series [5]. This method takes long processing time. Kim et al. proposed filtering method using truncated- median filter in 2D ultrasound image [6] but Kim's method needs preprocessing time. Ultrasound data typically contains more speckle noise and fuzzy boundaries than other volumetric imaging method. Volume ray casting is a famous software-based rendering method. Although it produces high-quality images, it takes a long time. In order to speed up, several optimized methods have been proposed [7-9]. They have mainly concentrated on efficiently skipping over transparent or homogeneous regions using coherent data structures such as octrees [10], Run-Length Encoded (RLE) volume [11] and distance map [8]. However these space leaping methods require long preprocessing time and extra storage to maintain the data structures. In the case of ultrasound, it can not use the previous acceleration method since 3D ultrasound device produces more than two volume dataset per second. So we have to devise high-speed volume visualization method.

In this paper, we present several visualization methods for fast volume ray casting of ultrasound data. Adaptive ray sampling that changes the sampling interval adaptively according to spatial distribution of voxels, can reduce the number of samples contributing to final pixel color. We use early ray termination (ERT) method which applies wider sampling interval and low threshold value of opacity. More-over, we exploit bilinear interpolation rather than trilinear one in empty space before reaching the surface. We can improve the rendering speed without preprocessing time.

In Section 2, we explain our methods in detail. Experimental results and remarks are shown in Section 3. Lastly, we summarize and conclude our work.

II. METHODS

One of the significant issues in volume visualization is rendering speed. There are several acceleration methods in this area but most of them are not suitable for ultrasound dataset since it requires long preprocessing time and additional data structure. We deal with the acceleration techniques using

All experimental data are supported by Medison, Co., Ltd. in Korea.
Corresponding Author : Byeong-Seok Shin
Dept. of Computer Science & Information Engineering,
Inha Univ. 253 Yonghyun-Dong, Nam-Gu, Incheon, KOREA
Tel : +82-32-860-7452 / Fax : +82-32-872-7452
E-mail : bsshin@inha.ac.kr

space leaping scheme in near real time. We explain three methods to visualize ultrasound data in this section: adaptive ray sampling, sampling interval adjustment and simplified space leaping. Fig. 1 shows the flowchart representing the difference between previous method and ours. Previous method does trilinear interpolation whether sampling position lies in an object or not. And every sampling position in a ray moves forward as the amount of unit interval. It caused slow down of rendering speed. On the other hand, our method uses bilinear interpolation for determining whether a sample is located object inside or not and it applies sufficient sampling interval during ray tracing. It can reduce the computation time and guarantee fast rendering.

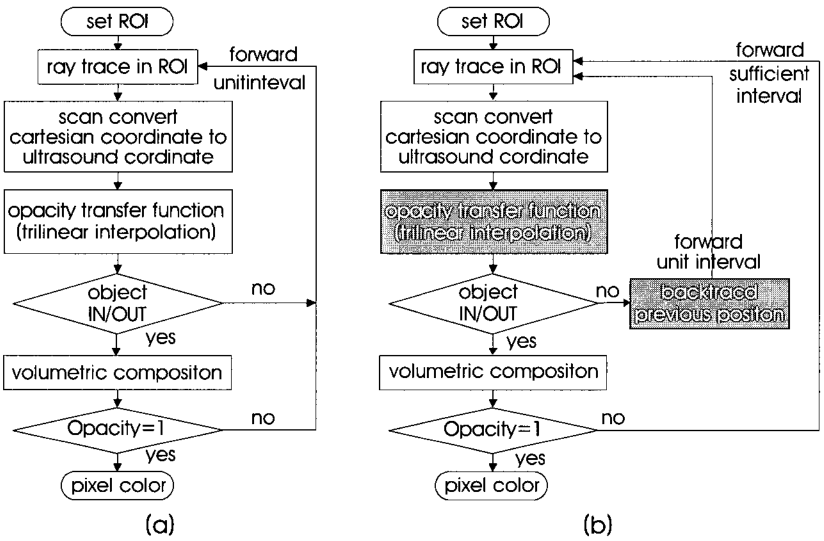

Fig. 1. Flowchart. Previous method calculates trilinear interpolation whether sampling position lies in object or not (a). It caused for slow rendering time. On the other hand, our method uses bilinear interpolation during object in/out test (b).

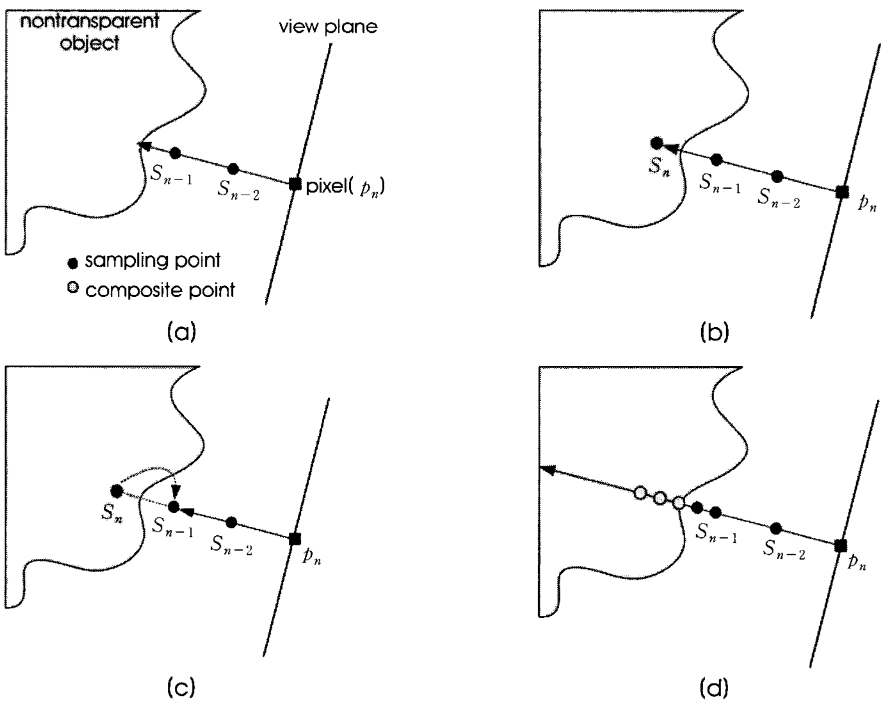

Fig. 2. Adaptive ray sampling scheme begins to sampling with sufficient interval (a), backtracks to previous sampling position when the current sampling position jump into an object (b, c), accumulates color with unit distance from previous sampling position (d)

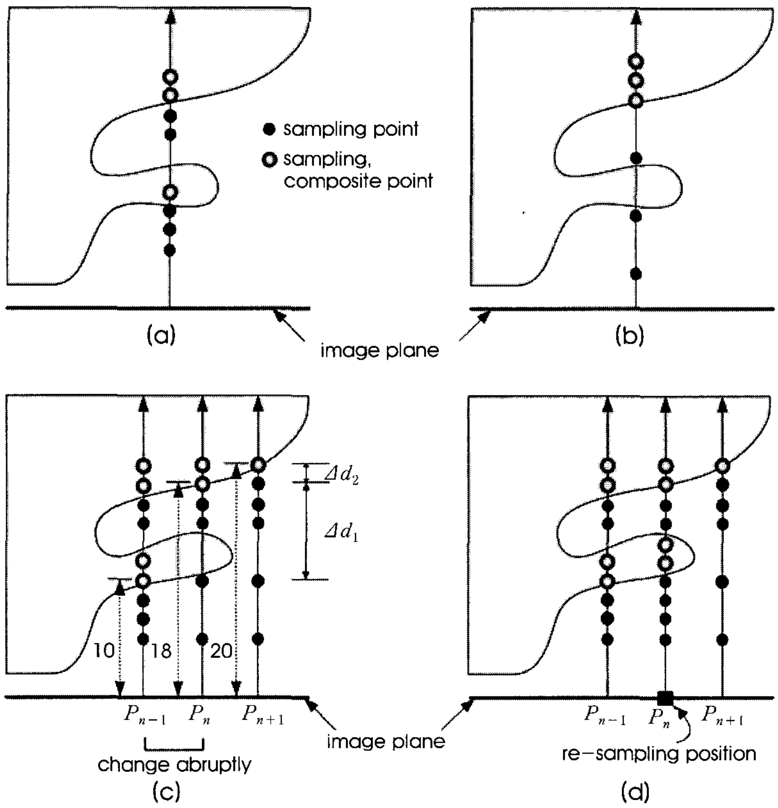

Fig. 3. An example of ray traversal in adaptive ray sampling (a), it can represent tiny structure in some case but it might lose the small structures (b), location of missed structure during volume ray traversing (c), and our re-sampling scheme with depth comparing method from view plane to object surface (d)

A. Adaptive Ray Sampling

Adaptive ray sampling is the method that reduces sampling rate in ray traversing step before a ray reaches object surface. Previous method [12] could not leap out of interesting region composed of transparent voxel. It spends much time since a ray traverse entire volume dataset. We can manipulate region of interest (ROI) to remove the remaining part such as umbilical cord, placenta and the space that filled up amniotic fluid. This means that a ray fired from a pixel traces in empty space for a while. So, we can set the sufficient sampling interval bigger than unit distance in ray traversal. The sampling point is back-tracked to previous position when a ray enters into object inside.

These procedures are as follows: firstly, a ray is fired from a pixel $p_{(i,j)}$ into volume space. Initial sampling interval sets sufficiently large. An optimal sampling distance can be selected by statistical experiment (Fig. 2 (a)). Secondly, in sampling stage, sampling point s_n moves to the previous position s_{n-1} when a ray jumps over the object surface. Thirdly, it computes color value during sampling procedure with unit distance from previous sampling position to object boundary (Fig. 2 (d)). These steps are repeated until the opacity value reaches a threshold or a ray jumps over the volume boundary.

In Fig. 3 (b), adaptive ray sampling cannot visualize a tiny structure if it is smaller than ray interval defined in initial stage. In general, a thin structure smaller than sampling interval appears in object boundary. In this case, we can solve the problem with sampling again using unit distance. Fig. 3 (c), (d) depicts our re-sampling method when a ray missed some structure. Firstly, assume that a view plane P has a resolution of $N \times N$. During rendering we can compute z-value from view plane to object surface for all pixels. Secondly, we define the re-sampling position that is region of contour composed of pixels which has the depth value larger than the threshold ($\Delta d_{th} < \Delta d_l$). In re-sampling position, conventional volume ray-casting method is exploited rather than adaptive ray sampling.

B. Early Ray Termination and Sampling Interval Adjustment

In general, ERT method means that sampling procedure terminates when the opacity value reach 1.0 (fully opaque) during ray traversal. In our method, it terminates ray traversal when accumulated opacity reaches specific value less than 1.0. In this case, rendering image has less brightness since a ray terminates color composition before its opacity becomes 1.0. So we use 1.1 voxel-width as sampling interval instead of 1 voxel-width so as to cover the same ray region of ray traverse

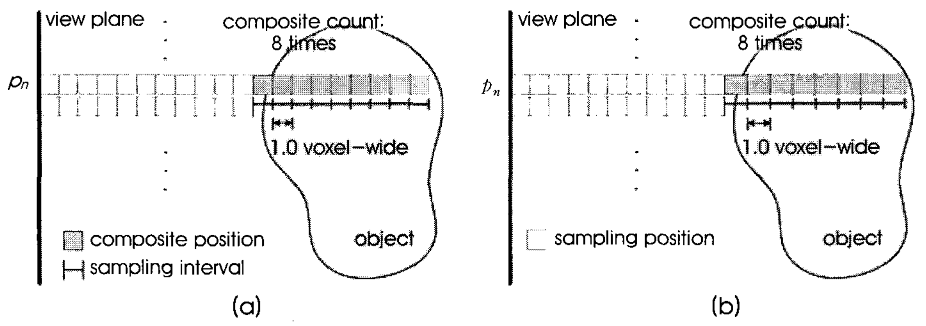

Fig. 4. Sampling interval and composite position in conventional method (a) and ours (b). We use a sampling interval larger than previous one and a composite opacity smaller than 1.0

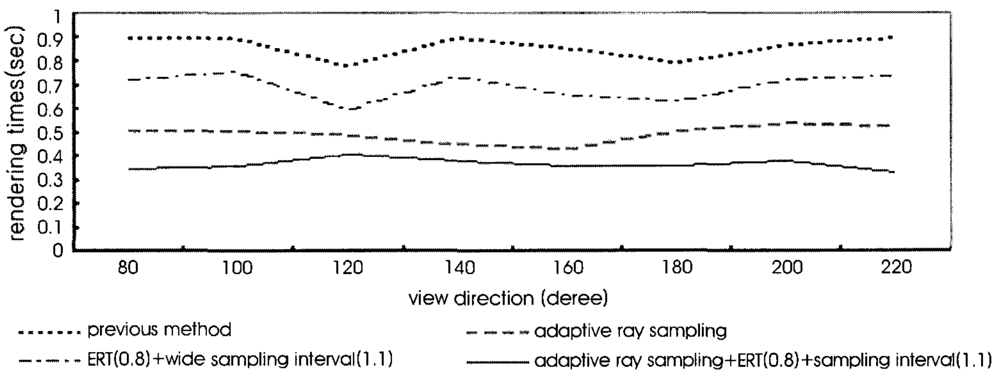

Fig. 5. A comparison of rendering speed. Rendering time with previous method (dotted line), adaptive ray sampling method (dashed line), the method with ERT(0.8) and 1.1 voxel-width sampling interval (dash-dotted line) and the method with all techniques (solid line)

as shown Fig. 4 (b). This results in speed up of ray traversal and there is little difference between two cases.

C. Simplification of Space Leaping

Trilinear interpolation is commonly used method to sample on empty space and to calculate color in volume rendering. We do not use trilinear interpolation in sampling stage since it is not necessary to calculate whether sampling position lie on empty region or not. In our method, we use bilinear interpolation in empty region because it has fast computation time about gradient estimation and color calculation. We can use zero-order interpolation called the nearest neighbor method. However, it makes aliasing such as striped artifact in final image. On the other hand, when we sample a ray using bilinear interpolation on empty space, it can produce high-quality images similar to that of trilinear interpolation and takes less computation time as much as that of zero-order interpolation method.

III. RESULTS AND DISCUSSION

All the methods are implemented on a PC equipped with Pentium IV 2.6 GHz CPU, 1GB main memory. Ultrasound volume dataset used for experiment obtained by scanning a fetus in uterus. We use 37 ultrasound dataset. The comparison of rendering speed (see Fig. 5) is average value. When we show the comparison of image quality, we use four dataset. The first volume (data A) has the resolution of 352×112×70. The second (data B), the third (data C) and the fourth (data D) volume data have 416×80×41, 384×104×48 and 256×60×51 resolution respectively. Image size is 200×200 pixels.

Fig. 5 shows the rendering time with maximum ROI. The results show that our method speeds up the rendering time about 100% faster than the conventional volume rendering method. When we set the maximum ROI, adaptive ray sampling works effectively since the empty space is relatively large. It shows that our method improves the speed about 75% when we use adaptive ray sampling method only (dashed line in Fig. 5). However, adaptive ray sampling is not efficient in some volume dataset that has tight ROI and includes large amount of speckle noise since it is focus on the leaping over empty space. We speed up the rendering time about 25% in this case. Table 1 shows that the rendering speed while changing interval from 2, 4, 8 to 16 in adaptive ray sampling. The result images for each case are shown in Fig. 6. The values reflect the rendering speed (second) in each ray sampling interval. As can be seen from the Table 1 and Fig. 6, if we use a sampling interval less

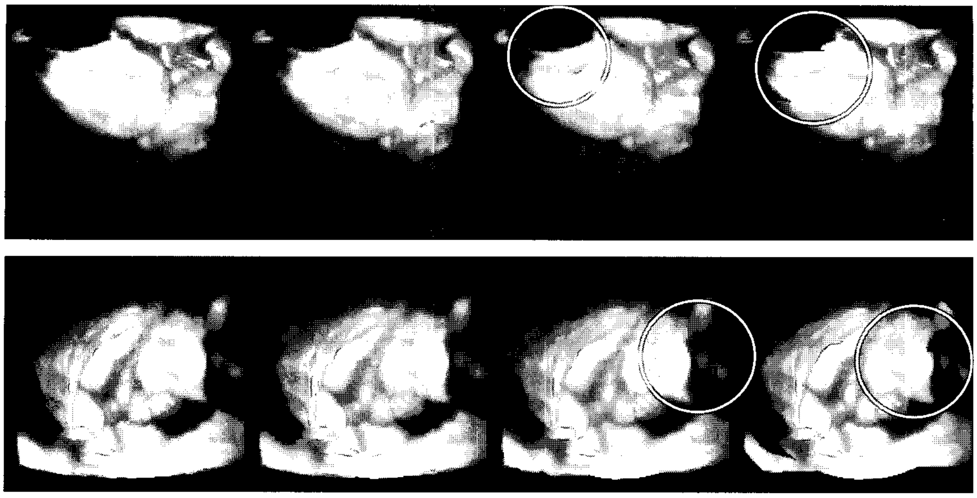

Fig. 6. A comparison of image quality when we use adaptive ray sampling each case of interval 2 (1st), 4 (2nd), 8 (3rd), 16 (4th column)

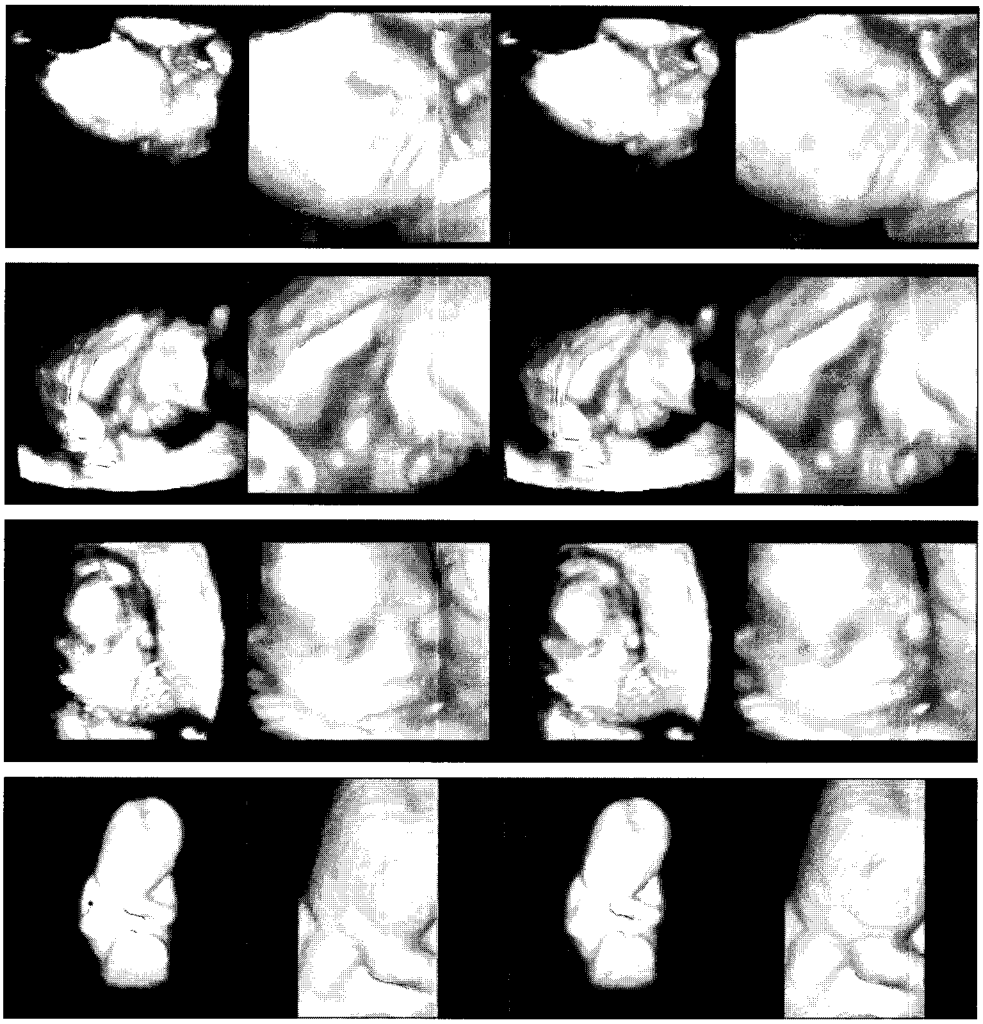

Fig. 7. Comparison of image quality produced by the conventional method (1st, 2nd column) and the adaptive ray sampling method (3rd, 4th column), 2nd and 4th columns are magnified by four

than 4, there is no difference of image quality but takes long time for image generation. Otherwise, it can be possible that a

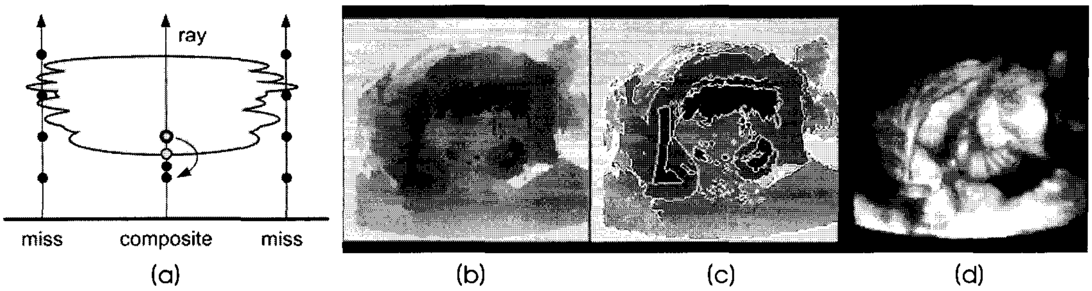

Fig. 8. A re-sampling method based on depth comparison. It cannot visualize a contour region (a). Depth value in depth-buffer (b), contour region (white pixels in (c)) and rendering image (d)

ray lose a tiny structure or sharpen a surface of fetus when we use a sampling interval more than 4 (see white circle in Fig. 6). So, in our experiment using adaptive ray sampling method, we can decide that an optimal value of sampling interval on ultrasound dataset is 4.

Table 1. Rendering speed when we use adaptive ray sampling each case of interval 2, 4, 8, 16(seconds)

ray interval	data A	data B	data C	data D
2	0.513	0.650	0.559	0.575
4	0.417	0.562	0.510	0.451
8	0.372	0.516	0.472	0.384
16	0.363	0.512	0.516	0.363

We apply another method such as ERT using 0.8 as maximum opacity value and 1.1 voxel-widthas sampling interval to the same data set. Dash-dotted line in Fig. 5 depicts a rendering time while applying the ERT and wider sampling interval. It shows that the rendering speed improve as much as 25% of previous method with ERT using 1.0 as maximum opacity value and 1.0 voxel-width. Using adaptive ray sampling and ERT (0.8 opacity) with 1.1 voxel-width sampling interval simultaneously, the rendering speed doubles up.

Fig. 7 shows the rendering results of our adaptive ray sampling method compared with the previous method. The difference of result images is 0.81 in average root mean square error (RMSE) algorithm [20]. So it is very hard to recognize the difference between the images. This implies that our method can effectively speed up the ultrasound dataset without loss of image quality.

Adaptive ray sampling method may miss a thin structure. Fig. 8 (c) illustrates a region of contour of an object where it has possibility of loss sampling position. When we render an image, we can get the distance from image plane to object surface in every pixel on view plane without additional cost (Fig. 8 (b)). We can find contour of object using comparison of depth values in depth buffer. Contour region is depicted as white pixels in Fig. 8 (c). A ray traces again at white pixels on view plane using unit interval. Additional processing time is not required for representing contour region since this process saves depth value to a buffer and calculates difference of depth value in ray casting step. We just require rendering time about 25% for re-sampling step in contour region. However, we cannot find missed region for fetus dataset using our method, so we do not use ray re-sampling method.

Fig. 9 shows a comparison of images when we use a previous method and our ERT method. It shows four kinds of fetus face. Fig. 9 1st and 3rd columns show the result images when we use the threshold value of opacity 1.0 and 1.0 voxel-width sampling interval (previous method). Fig. 9 2nd and 4th columns show the result images when we use ERT value 0.8 and 1.1 voxel-width sampling interval (our method). Fig. 9 shows that our method can speed up the ultrasound data without loss of image quality.

Table 2 shows a comparison of rendering speed when we use those three interpolation methods. Bilinear interpolation takes processing time similar to zero-order interpolation method. There is little difference between bilinear and trilinear interpolation method. Therefore, we use bilinear interpolation method to improve the ray traversal speed in empty region.

Table 2. Rendering speed when we use trilinear, bilinear and zero-order interpolation in sampling step(seconds)

Model	Trilinear	Bilinear	Zero-order
data A	0.897	0.532	0.5
data B	0.719	0.515	0.5
data C	0.891	0.594	0.531
data D	0.531	0.266	0.266

Fig. 10 shows the comparison of quality of final image produced by two methods. Fig. 10 1st and 3rd columns are generated by previous method. The 2nd and 4th columns are produced by our method. The difference of result images is averagely 1.2 with RMSE algorithm. It is very hard to recognize

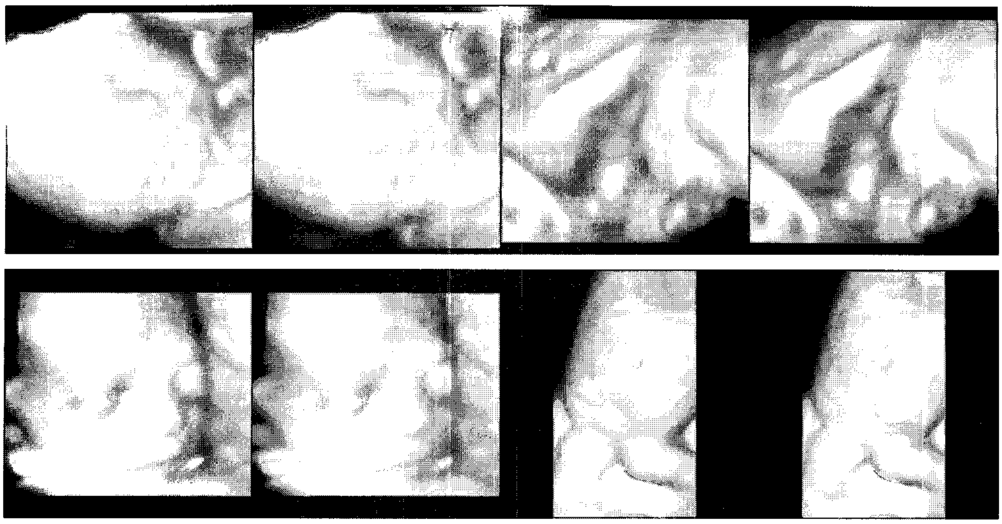

Fig. 9. Rendering images with previous method (1st, 3rd column) using ERT(1.0), 1.0 voxel-width sampling interval and our method (2nd, 4th column) using ERT(0.8), 1.1 voxel-width sampling interval

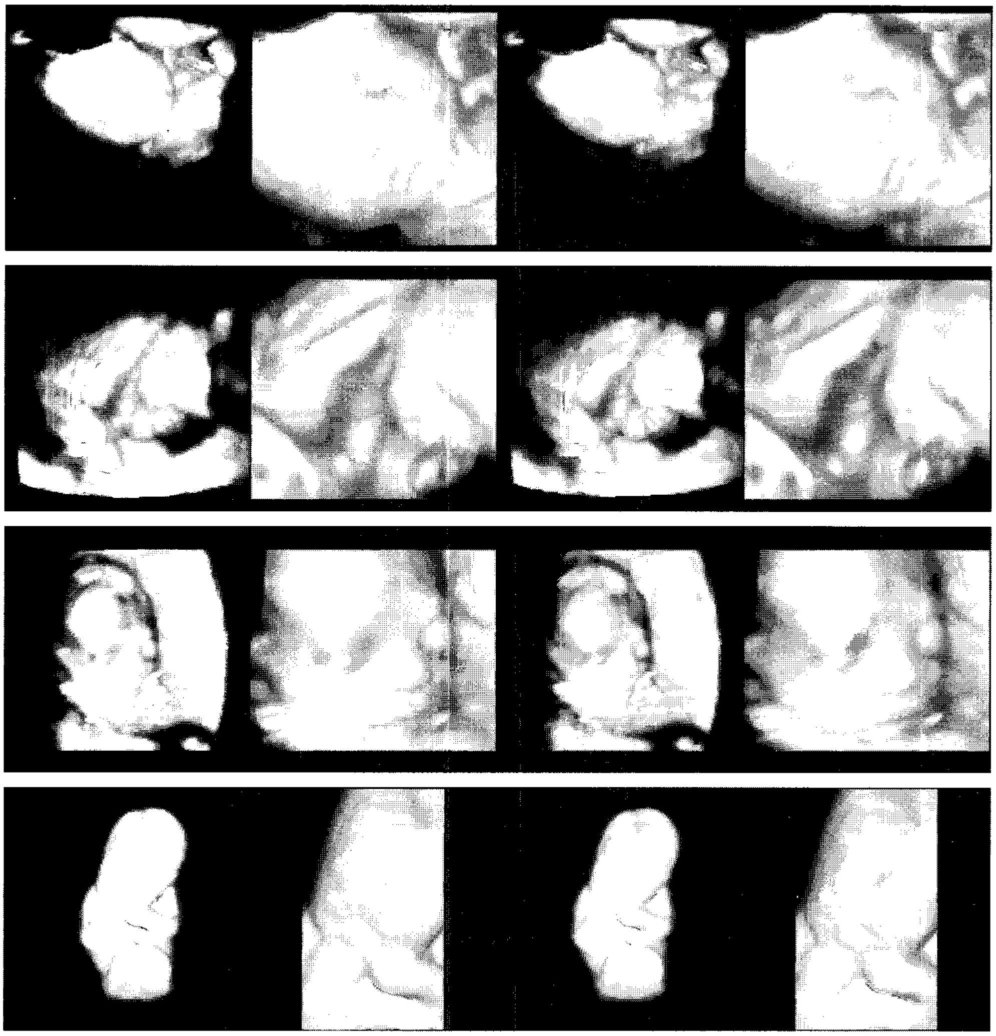

Fig. 10. Rendering images with previous method (1st, 2nd column) and the result images produced by our method (3rd, 4th column), 2nd and 4th columns are magnified by four

the difference between the images. This means that our method can effectively speed up the rendering time by using ultrasound data without loss of image quality.

IV. CONCLUSIONS

In order to achieve interactive rendering speed while preserving image quality, we present several visualization methods: adaptive ray sampling, early ray termination with less opacity and simplification of space leaping. It can reduce the number of sampling operation in empty region. We reduce the rendering time by using bilinear interpolation, and reduce composite count by using the early ray termination method based on less opacity value. When we use each method adaptive ray sampling, ERT (0.8 as maximum opacity value and 1.1 voxel-width) and three method all, the rendering speed is faster than previous method each about 75%, 25% and 100%. Even though our methods are fast, adaptive ray sampling method works efficiently in the state of empty region and the other methods are closely connected with image quality. Experimental results show that our method reduces the rendering time without loss of image quality.

REFERENCES

[1] A. Fenster, S. Tong, D.B. Downey, R.N. Rankin, "*Three-dimensional ultrasound imaging*", Proc. SPIE Medical Imaging, Vol. 2432, pp.176-184, 1995.

[2] T. Nelson, T. Elvins, "*Visualization of 3D ultrasound data*", IEEE Computer Graphics and Applications, Vol. 13, No. 6, pp.50-57, 1993.

[3] G. Sakas, LA. Schreyer, M. Grimm, "*Preprocessing and volume rendering of 3d ultrasonic data*", Computer Graph Appl, IEEE, Vol. 15, No. 4, pp.47-54, 1995.

[4] K. Baba, D. Jurkovic, "*Three-Dimensional Ultrasound in Obstetrics and Gynecology*", Progress in Obstetric and ynecological Sonography Series. The Parthenon Publishing Group p.110, 1997.

[5] João M. Sanches, Jorge S. Marques, "*A 3D Ultrasound System for Medical Diagnosis*", Pattern Recognition and Image Analysis, First Iberian Conference IbPRIA 2003, Springer Verlag LNCS 2652, pp.893-901, 2003.

[6] C. Kim, J. Oh, H. Park, "*Efficient Volume Visualization of 3-D Ultrasound Image*", Proc. SPIE Medical Imaging, Vol. 3658, pp.558-568, 1999.

[7] R. Yagel, Z. Shi, "*Accelerating volume animation by space-leaping*", Proc. of IEEE Visualization, pp.62-69, 1993.

[8] K. J. Zuiderveld, H. J. Koning, M. A. Viergever, "*Acceleration of ray-casting using 3D distance transforms*", Proc. of Visualization in Biomedical Computing, pp.324-335, 1992.

[9] J. K. Udupa, D. Odhner, "*Shell Rendering*", IEEE Computer Graphics and Applications, Vol. 13, No. 6, pp.58-67, 1993.

[10] K. Subramanian, D. Fussell, "*Applying space subdivision techniques to volume rendering*", IEEE Visualization '90, San Francisco, CA, pp.150-159, 1990.

[11] P. Lacroute, M. Levoy, "*Fast volume rendering using a shear-warp factorization of the viewing transformation*", SIGGRAPH 94 Proceedings, Orlando, Florida, pp.451-458, 1994.

[12] M. Levoy, "*Display of surfaces from volume data*", IEEE Computer Graphics and Applications, Vol. 8, No. 3, pp. 29-37, 1988.

[13] M. Levoy, "*Efficient Ray Tracing of Volume Data*", ACM Transactions on Graphics, Vol. 9, pp.245-261, 1990.

[14] E. R. Davies, "*On the noise suppression and image enhancement characteristics of the median, truncated median and mode filter*", Pattern Recog. Lett. Vol. 7, pp.87-97, 1988.

[15] G. Sakas, LA. Schreyer, M. Grimm, "*Preprocessing and volume rendering of 3D ultrasonic data*", IEEE Computer Graphics & Applications, Vol. 15, pp.47-54, 1995.

[16] E. Steen, B. Olstad, "*Volume Rendering in Medical Ultrasound Imaging based on Nonlinear Filtering*", IEEE Winter Workshop on Nonlinear Digital Signal Processing, pp.6.1-6.6, 1993.

[17] D. Hönigmann, J. Ruisz, C. Haider, "*Adaptive design of a global opacity transfer function for direct volume rendering of ultrasound data*", VIS 2003 IEEE, pp.489-496, 2003.

[18] R. Fattal, D. Lischinski, "*Variational Classification for visualization of 3D Ultrasound Data*", VIS '01 Proceedings, pp.403-410, 2001.

[19] G. Sakas, S. Walter, "*Extracting surfaces from fuzzy 3D-Ultrasound data*", SIGGRAPH '95 Proceedings, Annual Conference Series, pp.465-474, 1995.

[20] K. Kim, C.M. Wittenbrink, A. Pang, "*Extended specifications and test data Sets for Data Level Comparisons of Direct Volume Rendering Algorithms*", IEEE Transactions of Visualization and Computer Graphics, pp.299-317, 2001.

[21] V. Shamdasani, U. Bae, R. Managuli, Y. Kim, "*Improving the Visualization of 3D Ultrasound Data with 3D Filtering*", Proc. SPIE Medical Imaging, Vol. 5744, pp.455-461, 2005.