Volume Reconstruction of Freehand Three-Dimensional Ultrasound Using Median Filters

Qing-Hua Huang and Yong-Ping Zheng

Department of Health Technology and Informatics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, P.R.China.

Running Title: **3D Ultrasound Using Median Filters**

Abstract

Objectives:

This paper aims to apply median filters for reducing interpolation error and improving the quality of 3D images in a freehand 3D ultrasound (US) system.

Background and Motivation:

Freehand 3D US imaging has been playing an important role in obtaining the entire 3D impression of tissues and organs. Reconstructing a sequence of irregularly located 2D US images (B-scans) into a 3D data set is one of the key procedures for visualization and data analysis.

Methods:

In this study, we investigated the feasibility of using median filters for the reconstruction of 3D images in a freehand 3D US system. The B-scans were collected using a 7.5 MHz ultrasound probe. Four algorithms including the standard median (SM), Gaussian weighted median (GWM) and two types of distance weighted median (DWM) filters were proposed to filter noises and compute voxel intensities. Qualitative and quantitative comparisons were made among the results of different methods based on the image set captured in freehand from the forearm of a healthy subject. A leave-one-out approach was used to demonstrate the performance of the median filters for predicting the removed B-scan pixels.

Results:

Compared with the voxel nearest neighbourhood (VNN) and distance-weighted (DW) interpolation methods, the four median filters reduced the interpolation error by 8.0%~24.0% and 1.2%~21.8%, respectively, when 1/4 to 5 B-scans was removed from the raw B-scan sequence.

Conclusions:

In summary, the median filters can improve the quality of volume reconstruction by reducing the interpolation errors and facilitate the following image analyses in clinical applications**.**

Keywords – Freehand 3D ultrasound, 3D ultrasound imaging, median filter, Gaussian weighted median, volume reconstruction.

Corresponding author: Yongping Zheng, PhD.

Department of Health Technology and Informatics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, P.R.China.

Tel: 852-27667664; Fax: 852-23624365; Email: ypzheng@ieee.org

1. Introduction

During the past decades, two-dimensional (2D) ultrasound (US) imaging has been widely used for many clinical applications in a rapid, inexpensive, non-ionizing, and non-invasive manner. However, it has some limitations for clinicians to visualize and analyze three-dimensional (3D) anatomies for diagnosis. 3D US imaging has proven to be able to overcome the limitations by constructing various 3D data sets of tissues or organs for visualization and analysis [1-3].

Many types of systems that are capable of obtaining 3D US imaging have been developed for various clinical practices in recent years [1, 2]. One of the often-used systems is the freehand imaging with spatial tracking devices, where the US probe is moved by hand in an arbitrary manner. A sequence of irregularly

spaced 2D US images (B-scans) can be captured as the probe is moved over the region of interest, and resampled onto slices along any-plane [4], regular voxel arrays [5] or surfaces [6] based on the positional information recorded from the spatial sensing device during the scanning. In comparison with other 3D systems, such as 3D imaging with 2D transducer arrays and 3D mechanically driven imaging, the freehand protocol is the cheapest and most flexible approach [3], allowing the most freedom for the operator to acquire arbitrarily 3D anatomical structures with different sizes. Due to the arbitrary movement of the US probe in the freehand 3D US imaging, it is important to find a suitable interpolation method to reconstruct 3D image data to preserve the diagnostic information and meanwhile to avoid the introduction of noises and artefacts.

The methods of reconstructing 3D images, producing various slices or approximating surfaces from a number of irregularly spaced 2D B-scans have been investigated by several researchers [4-10]. Rohling et al. [5] reviewed a number of often-used methods for resampling B-scan data into 3D image data in freehand systems, and grouped the previous interpolation methods into three categories: voxel nearest-neighbour (VNN) interpolation, pixel nearest-neighbour (PNN) interpolation, and distance-weighted (DW) interpolation. For the VNN method, the intensity of the nearest pixel in a voxel neighbourhood is assigned to this voxel. Prager et al. [4] used the VNN method to obtain resliced images along any planes directly from the raw B-scan pixels in real-time in their sequential freehand system. Although the VNN is straightforward and fast, it tends to produce reconstruction errors on the image slices, particularly when the scanning has big gaps. The PNN method searches through each pixel in B-scans, and assigns the pixel intensity to its nearest voxel. Multiple contributions to the same voxel are usually averaged. Interpolation methods can be used to fill gaps in the voxel arrays. However, boundaries between the voxels directly assigned by pixels and those gaps filled using interpolation methods can be clearly observed. Unlike the VNN and PNN methods, the DW interpolation computes each voxel value by assigning the weighted average of a set of pixels falling into a predefined 3D region centred about that voxel. All pixels in the region are weighed by their inverse distances to the voxel centre [7]. As this procedure can be regarded as a mean filtering method that can reduce noises and artefacts, DW interpolation normally provides good

results with speckle and shadowing reduction. However, averaging operation can blur image details, especially the boundaries of tissue components with small dimensions. To provide more diagnostic details in 3D US images, Meairs et al. [9] reported a weighted ellipsoid Gaussian convolution kernel to assign the pixel weights. Huang et al. [10] used a square distance weighted (SDW) interpolation in their 3D imaging system. In comparison with the conventional DW method, these modified DW methods could provide improved results, but image blurring could still not be avoided.

On the other hand, as a nonlinear processing technique, the median filter has been successfully used for speckle reduction and edge preservation in ultrasonic images because it has the advantages of removing noises and meanwhile well preserving image details [11, 12]. However, little attention was paid to employing this technique for interpolating voxels, surface contours or intensities on any planes resliced directly from raw B-scan pixels in 3D US systems. In this paper, we propose three weighted median (WM) filters and applied them for the volume reconstruction in our freehand system [10]. A standard median (SM) filter was also used for comparison purposes. A data structure was designed accordingly for storing pixel intensities and their distances to the voxel centre in a spherical region. The detailed methods are described in section 2. We compared the interpolation performances of the median filters with those of the VNN and DW methods. The reconstruction and comparison results are presented in section 3 and the discussion and conclusions in section 4, respectively.

2. METHODS

2.1. System description

A freehand 3D US imaging system was previously developed and used for the imaging of musculoskeletal tissues [10]. This system was comprised of three parts including an electromagnetic spatial sensing device (miniBird, Ascension Technology Corporation, Burlington, VT, USA), a portable US scanner (SonoSite 180PLUS, SonoSite, Inc., Bothell, WA, USA), and a PC with custom-designed software, which was used for data collection, volume reconstruction, visualization and analysis. The receiver of the spatial sensing device was attached to the probe of the US scanner. During the data

acquisition, the video stream of the US scanner was digitized by a video capture card (NI-IMAQ PCI/PXI-1411, National Instruments Corporation, Austin, TX, USA) and then transferred to the PC. The spatial data, including the position and orientation of the receiver of the spatial sensing device, were simultaneously recorded by the system through RS232 serial port. The B-scans were collected at a rate of 20 frames per second. The software was programmed in VC++ and run in a PC with a 2.8 GHz Pentium IV processor and 1 GB of RAM.

2.2. Calibration methods

Temporal and spatial calibration experiments were performed for the freehand system [10]. According to a previously reported method [13], we used a 3D translating device (Parker Hannifin Corporation, Irvine, CA, USA) to control the movement of the US probe during the temporal calibration. The US probe was immersed in a tank filled with water and moved up and down by the 3D translating device. The metal distortions that might be caused by the translating device were not observed and the movement error was insignificant in our experiments. The normalized positions of the lines denoting the bottom of the water tank in B-scan images were correlated with those measured by the spatial sensor. The optimum time delay between the captured images and sensed spatial data could be obtained by finding out the minimum difference between the two data streams. The spatial calibration was conducted using a cross-wire phantom [7]. Two cotton wires were crossed in a water tank. In each experiment, approximately 60 B-scans with corresponding spatial locations were recorded from various directions. The crossing point in each B-scan was manually marked and a Levenberg-Marquardt nonlinear algorithm was performed to determine the spatial relationship between the image plane and the spatial sensor [14].

2.3. Volume reconstruction

Most of the freehand systems employed similar methods for the transformation procedures, as described in [10]. As shown in Fig. 1, the pixels in a sequence of B-scans could be transformed into the 3D coordinate system. The dimensions and grid points of a regular 3D data set should be established in

this coordinate system before the reconstruction. With respect to the conventional DW interpolation, a spherical region centred about each voxel grid point with a radius R should be defined, and all pixels transformed into this region should be weighed according to their inverse distances to the voxel centre, and then averaged to obtain the voxel intensity, as demonstrated in Fig. 2(a). If the radius of the spherical region is set to be large enough, the gaps between raw B-scans shown in Fig. 2(b) can be well filled as illustrated in Fig. 2(c).

In this study, we adopted the same spherical region centred about each voxel for volume reconstruction. However, we applied median filters, which used the median intensity value of the pixels in the predefined region as the voxel value, instead of weighted mean filters used by the DW interpolation. It is well known in the fields of signal and image processing, median filters can provide smoothing function and meanwhile preserve the high frequency components. We designed a data structure to store all pixels falling into the spherical region of each voxel and their corresponding distances to the voxel centre, as shown in Fig. 3, to facilitate the implementation of the median filters. For each voxel, all pixels in its spherical neighbourhood would be stored in the pixel intensity array, and their corresponding distances to the centre would be computed and recorded in the distance array in the same order. Then, a median filter was applied to those stored pixel intensities. Finally, the voxel intensity was assigned to be the output of the median filter. If gaps occurred, the intensity of the empty voxel was assigned to be the output of the median filter applied to the value-assigned voxels in its spherical neighbourhood. Fig. 2(b) shows an typical example of reconstructed volume data with filled gaps.

2.4. Median filters

Four median filters were used for calculating voxel intensities in the 3D image reconstruction. The simplest approach was the SM filter which is defined as:

$$I\left(\vec{V}_C\right) = median\left\{ I\left(\vec{V}_P^{\,k}\right) \mid \vec{V}_P^{\,k} \in \Re; k = 1 \ldots N_c \right\} \quad (1)$$

where $\Re$ is the predefined spherical region containing N_c pixels, $\vec{V}_C$ locates at the centre of $\Re$, $I(\vec{V}_P{}^k)$ is the intensity of the kth pixel falling into $\Re$, and $I(\vec{V}_C)$ is the intensity of the voxel at the centre of $\Re$. The median value of the N_c pixels should be the value of $I(\vec{V}_C)$. If N_c is an even number, the pixel with the largest difference to the mean of the N_c pixels would be removed before the sorting in this study.

According to Loupes et al. [11], the weighted median (WM) filter could be helpful in speckle reduction and signal preservation. Therefore, three WM filters with different mathematical expressions were also proposed in this study to improve the interpolation results. Appendix A provides the definition and an example for the WM filter [15]. In these WM filters, the pixels' distances to the centre of each voxel were inversely related to their weights because smaller distance should indicate that the pixel had more probability to be the output value.

The first WM filter is named as type-1 distance-weighted median (DWM-1) using the squared inverse distance to calculate the weights, expressed as follow:

$$\begin{aligned} I(\vec{V}_C) &= median\left\{ W_k \lozenge I(\vec{V}_P^{\,k}) \mid \vec{V}_P^{\,k} \in \Re; k = 1 \ldots N_c \right\} \\ W_k &= \frac{1}{d_k{}^2} \end{aligned} \tag{2}$$

where d_k is the distance between the kth pixel and the current voxel centre, and $\lozenge$ denotes the duplication operation.

The second WM filter is type-2 distance-weighted median (DWM-2), which is written as:

$$\begin{aligned} I(\vec{V}_C) &= median\left\{ W_k \lozenge I(\vec{V}_P^{\,k}) \mid \vec{V}_P^{\,k} \in \Re; k = 1 \ldots N_c \right\} \\ W_k &= R_0^2 - d_k^2 \end{aligned} \tag{3}$$

where R_0 is the radius of the predefined spherical region, and d_k is the distance between the kth pixel and the current voxel centre.

The two DWM methods were similar but not the same. Let us consider a specific pixel very close to the centre of the spherical region. For the DWM-1, the weight assigned to the pixel at the voxel centre would be extremely large, indicating that final output would be dominantly impacted by this pixel. For the DWM-2, the weight of the pixel was limited and not larger than R^2. Thus the point at the center might

not be able to mainly affect the final output value. This difference would cause the two types of DWM filters to have different interpolation results.

The 3rd WM filter is named as Gaussian weighted median (GWM) filter. A Gaussian function is applied to calculate the weights. The computation is described by

$$I\left(\vec{V}_C\right) = median\left\{ W_k \Diamond I\left(\vec{V}_P^k\right) \mid \vec{V}_P^k \in \Re; k = 1 \ldots N_c \right\}$$
$$W_k = e^{-\frac{d_k^2}{2\sigma^2}} \quad (4)$$

where σ is a parameter that could be adjusted by the operator, and d_k is the distance between the kth pixel and the current voxel centre. The weight for the kth pixel is obtained by the Gaussian function, where the output (W_k) is related to the inverse exponent of the squared distance between the voxel centre and the kth pixel in the spherical region. For the three WM filters, the pixels with smaller distances to the voxel centre would have relatively larger weights. However, the changing rates of weight as a function of the distance from the center were different among the three WM filters. Higher changing rates could better preserve edges and lower rates could provide better smoothing effects. While these typical weight functions allowed us to demonstrate the capabilities of WM filters, it is understandable that there should be other functions for weight assignment.

2.5. Evaluation tests and comparisons

The evaluation approach introduced by Rohling et al. [5] was used in this study. The idea of the approach was to evaluate the ability of a reconstruction technique in preserving true intensity values at the locations where a part of original data was removed. A good reconstruction method should interpolate the removed grid points with values very near to the original data. Following this evaluation method, we scanned the right forearm of a healthy male subject using the 3D US system. The subject gave his informed consent to the investigation which was approved by the Hong Kong Polytechnic University Human Subjects Ethics Committee. A B-scan near the middle of the scanned region was selected for pixel removing. Different percentages of pixels were removed randomly from this selected B-scan. The rest

data was used to reconstruct a voxel array with a voxel size equivalent to the pixel size. The average absolute differences between the interpolated grid points and the missing original pixel values was calculated for evaluating the reconstruction performance using the following equation:

$$V = \frac{1}{N}\sum_{i=1}^{N}\left|p_i - r_i\right| \tag{5}$$

where p_i is the removed original pixel intensity, r_i is the interpolated intensity at the location of p_i, and N is the number of removed pixels. A smaller V indicates a better performance of interpolation.

Eight different data removing ratios were used in the current evaluation tests, i.e. 0%, 25%, 50%, 75%, 100%, 300%, 500% and 700%. For the 0% test, V was calculated using all pixels of the selected B-scan. The tests with the data removing ratios of 25%, 50%, 75% and 100% were performed using the correspondingly selected B-scan n. It cannot happen that only a number of pixels instead of a whole B-scan plane are missing in a real-world environment. The evaluation tests using 25%, 50% and 75% removal percentage aimed to obtain the interpolation performance of different methods when the gaps between B-scans were very small. For the 300% test, the data from B-scan n-1, B-scan n and B-scan n+1 were totally removed. The 500% test further removed all data from B-scan n-2 and B-scan n+2, and the 700% test further removed all data from B-scan n-3 and B-scan n+3. The 100%, 300%, 500% and 700% tests aimed to mimic the gaps existing in the B-scan sequence.

In our experiments, the distance between any two adjacent B-scans ranged from 0.11mm to 0.56 mm, with a mean of 0.29 mm. Therefore, we used 0.40 mm as the radius R of the spherical neighbourhood for the 0%, 25%, 50%, 75% and 100% tests. To cover the larger gaps for the 300%, 500% and 700% tests, the radius R was increased to 0.69 mm, 0.98 mm and 1.27 mm, respectively. The VNN interpolation method and the DW method were implemented and compared with the four median based interpolation methods. In order to investigate the effects of the parameter σ in the GWM, it was empirically set to be 0.01, 0.025, 0.05, 0.075, and 1.0, respectively, for the evaluation tests. Since the computation of median

filters is expensive, we recorded the computation times of above methods at the 100% test for comparison purpose.

3. RESULTS

Table 1 summarizes the averaged interpolation errors and the standard deviations of the evaluation tests using the VNN, DW and the four median filters. According to the values in bold type which indicate the best interpolation result for each test, the best result using the interpolation errors using four median filters were reduced by 8.0%~24.0% compared with the VNN method, and 1.2%~21.8% compared with the DW method. Table 2 gives the averaged computation time at the 100% test for each interpolation method used in this study. Fig. 4 shows the trends of the interpolation errors for the different reconstruction methods. A typical slice reconstructed using 50%, 100% and 300% data removing ratios is illustrated in Figs. 5, 6, and 7. A small window is marked on the figures to illustrate the image content more clearly.

For the 0% data removing ratio, the VNN and DW methods presented no error as all pixels of the selected B-scan were assigned to their original locations on the slice reconstructed (Table 1). For the DWM-1, the interpolation error was also 0 because the weights were infinite for the original pixels overlapping with the voxels and the median outputs for these voxels were therefore the corresponding pixel values. The other median filters resulted in some errors because the median outputs were smoothed in comparison with the original data. The GWM with smaller σ (σ=0.01) produced relatively smaller interpolation error, because the output with very small σ approximated the nearest pixel, making the interpolation act like the VNN method. For the 25%, 50% and 75% tests, the median filters provided similar results to the VNN and DW methods.

According to Table 1 and Fig. 4, the VNN method produced larger interpolation errors than the other methods for the tests with data removing ratios larger than 100%. For the 100% test, the median filters showed performances similar to the DW method, except the GWM with σ=0.01. For the ratios of 300%,

500% and 700% , all the median filters offered smaller errors in comparison with the DW method, with the GWM with σ=0.1 showing the best interpolation results.

For the qualitative comparisons, it is apparent that the slices reconstructed using the median filters could preserve relatively sharper edges and more texture patterns in comparison with the DW method, as demonstrated in Figs. 5, 6, and 7. Although the slices resampled using the VNN method looked sharp and presented the most texture patterns for the 100% and 300% tests, the anatomical information was actually distorted due to the misalignments of pixels, which had been addressed by Rohling et al. [5] and could be observed in some regions of the images shown in Figs. 6(g) and 7(g). This problem appeared more serious for the cases with 500% and 700% data removing ratios. It is an inherent problem of VNN when the image gap is relatively large.

4. DISCUSSION AND CONCLUSION

In this paper, four median filters were studied and implemented for 3D image reconstruction in the freehand 3D US imaging system. Besides the standard median filter, we also proposed three weighted median filters based on the distance between the neighbouring pixels and the voxel center. A leave-one-out approach [5] was used to study the performance of the median filters for predicting the removed B-scan pixels of a 3D image data set collected from the forearm of a healthy subject. The results showed that the median filters would generally produce smaller interpolation errors in comparison with the VNN and DW methods when pixels were removed from more than one B-scan images, especially with 300%, 500% and 700% image removing ratios. In addition, qualitative visualization demonstrated that the median filters could better preserve the tissue boundary information in comparison with the DW method and meanwhile provide smoothing for the speckles.

In general, all the median filters (SM, DWM-1, DWM-2 and GWM) worked better for the 300%, 500% and 700% tests in comparison with the VNN and DW methods, indicating their advantages in filling large gaps. For the 25%, 50% and 75% tests, the weighted median filters (DWM-1, DWM-2 and GWM) also demonstrated improved interpolation in comparison with the DW method. Among the four median filters,

the GWM with relatively larger sigma (such as σ = 0.075) offered the best reconstruction results at the 300%, 500% and 700% tests, and the GWM with relatively smaller signal (such as σ = 0.01) presented the lowest interpolation error at the 0%, 25%, 50% and 75% tests. Theoretically, there is no limit for the assignment of σ. If σ gets much closer to 0, the pixel with the shortest distance to the voxel centre will have much larger weight and the filter tends to choose one of the nearest pixels as the output, acting like the VNN method. If σ is infinitely positive, the filter tends to be a standard median filter. Therefore, the parameter σ in the GWM should be carefully selected for different applications. How to select an optimized σ or adaptively set σ with respect to the gap size for the GWM method should be further investigated.

According to the qualitative comparison results (Figs. 5, 6, and 7), it was obvious that the median filters could better preserve tissue edges in comparison with conventional DW methods. Although the VNN method appeared to provide most details in the reconstructed image, the images were actually distorted for the raw data set with large B-scan gaps. Therefore, quantitative analyses based on the VNN results would not be so reliable when the 3D scanned image slices were not dense enough. The proposed median filters appeared to be able to provide a sort of compensation between the VNN and DW methods. It is well-known that the correct edge preservation is very important for the image segmentation of anatomical structures, such as surface extraction [6], as the enlarged gradient at tissue edges can make the tracing of tissue boundaries easier and more accurate. The use of the proposed median filters in the 3D image reconstruction may potentially provide an approach for exploring automatic and robust segmentation techniques based on the images with better preserved edges. Consequently, more accurate 3D measurements, such as volume estimation, can be realized based on the accurate segmentation of 3D objects. In addition, the improvement of edge preservation will be useful for 3D compounding of multiple sweeps [16-18], as the correctly preserved edges can be considered as tissue landmarks and used for alignment of different sweeps.

No artificial phantom with homogeneous material has been used in the evaluation tests in this study. The reason was that we would like to calculate the interpolation errors globally on the reconstructed slices. As shown in the Fig. 8(a), if the rectangles in the two images are not matched exactly, there would

be interpolation errors only in the edge regions but not in the internal regions because the internal intensities are constant. In contrast, Fig. 8(b) shows another example where the image content is more complicated and the interpolation errors can be computed from internal regions. Therefore, in order to completely evaluate the interpolation performance, more complicated structures of the scanned object will provide more reliable results. Because the phantoms with regular geometry or structures are usually made using homogeneous materials, they may not be so reliable to evaluate the interpolation performance in comparison with real human tissues.

To realize the proposed weighted median filters, a data structure was designed to store the array of pixels falling into the spherical neighbourhood centred about each voxel and the corresponding distances of these pixels to the voxel centre. In comparison with the DW interpolation, the two arrays associated with the voxels require more memory during the reconstruction. The larger spherical region or the more voxel grid points defined, the more memory was required. This problem can be solved by increasing the capability of RAM in the PC. Moreover, the proposed median filters required more computation time for sorting the intensities in the pixel array, as demonstrated in Table 2. In this study, Quicksort sorting algorithm [19] was used to improve the computation speed of image reconstruction. Nevertheless, the computation time for the median filters was still about 3-5 times longer than that for the DW interpolation for the data set used in our practical tests (with a dimension of 150x150x128 voxels). The time cost of median filters will further increase when the image data set is larger. Therefore, the implementation of the median filters requires a high-performanced PC with enough memory. With the recent advancement in multi-core and multi-processor PCs, we believe that the computational time for realizing the proposed median filters can be greatly reduced in the near future. In addition, to develop sorting algorithms with higher efficiency will be another approach to reduce the computation time.

In conclusion, we have reported to use median filters for the 3D image reconstruction in freehand 3D US system. Four median filters with different forms were employed for this purpose. In comparison with the VNN and DW methods, the reconstruction results of the median filters demonstrated their good

performances for reducing registration errors and preserving edges, particularly for raw image sets with large slice gaps. Future studies should be followed to improve the computation speed of the median filters using more efficient algorithms, and to investigate new techniques for the economical storage of the pixel intensity and distance arrays. The performances of the median filters should also be verified with more image sets with different conditions in the future.

Acknowledgements

This work was partially supported by The Hong Kong Polytechnic University (G-YE22) and the Research Grants Council of Hong Kong (PolyU 5245/03E).

Appendix A

The weighted median (WM) filter was derived as a generation of the standard median (SM) filter, where a nonnegative weight is assigned to each position in the filter window [15]. We present here a definition and an example for the WM filters.

Definition: The output of a WM filter is the value x^* that minimizes the following equation

$$L(x^*) = \sum_{i=1}^{n} w_i |x_i - x^*| \qquad (6)$$

where x^* is guaranteed to be one of the elements in an input vector X, because $L(x^*)$ is piecewise linear and convex if $w_i \geq 0$, i=1, 2, …, n. The computation procedures can be summarized as follows: sort the elements in the filter window; then add up the corresponding weights from the upper end of the sorted sequence until the sum equals or just exceeds half of the total sum of the weights, i.e., the sum $\geq \frac{1}{2}\sum_{i=1}^{n} w_i$; the element corresponding to the last added weight is the final output of the WM filter.

Example: Given an input vector [0.5, -0.5, -1, 1, 0] and a real positive weight vector [0.1, 0.2, 0.3, 0.2, 0.1], sort the elements in the input vector and then we get the new element sequence with corresponding weights:

0.2	0.1	0.1	0.2	0.3
1	0.5	0	-0.5	-1

The total sum of weights is 0.9 so that we add the weights from the left until the sum equals or just exceeds 0.45. The sum is 0.4 after adding the weight (0.1) corresponding to the element 0, and 0.6 after adding the weight (0.2) corresponding to the element -0.5. Therefore, the output of the WM filter is -0.5, which is different from the output of an SM filter.

References

[1] T. R. Nelson, D. H. Pretorius, Three-dimensional ultrasound imaging, Ultrasound Med. Biol. 24 (9) (1998) 1243-1270.

[2] A. Fenster, D. B. Downey, H. N. Cardinal, Three-dimensional ultrasound imaging, Phys. Med. Biol. 46 (5) (2001) R67-R99.

[3] A. H. Gee, R. W. Prager, G. M. Treece, L. Berman, Engineering a freehand 3D ultrasound system, Pattern Recogn. Lett. 24 (4-5) (2003) 757-777.

[4] P. W. Prager, A. Gee, L. Berman, Stradx: real-time acquisition and visualization of freehand three-dimensional ultrasound, Med. Image. Anal. 3 (2) (1999) 129-140.

[5] R. N. Rohling, A. H. Gee, L. Berman, A comparison of freehand three-dimensional ultrasound reconstruction techniques, Med. Image Anal. 3 (4) (1999) 339-359.

[6] W. Y. Zhang, R. N. Rohling, D. K. Pai, Surface extraction with a three-dimensional freehand ultrasound system, Ultrasound Med. Biol. 30 (11) (2004) 1461-1473.

[7] C. D. Barry, C. P. Allot, N. W. John, P. M. Mellor, P. A. Arundel, D. S. Thomson, J. C. Waterton, Three-dimensional freehand ultrasound: Image reconstruction and volume analysis, Ultrasound Med. Biol. 23 (8) (1997) 1209-1224.

[8] J. M. Sanches, J. S. Marques, A Rayleigh reconstruction/interpolation algorithm for 3D ultrasound, Pattern Recogn. Lett. 21 (10) (2000) 917-926.

[9] S. Meairs, J. Beyer, M. Hennerici, Reconstruction and visualization of irregularly sampled three- and four-dimensional ultrasound data for cerebrovascular applications, Ultrasound Med. Biol. 26 (2) (2000) 263-272.

[10] Q.H. Huang, Y.P. Zheng, M.H. Lu, Z.R. Chi, Development of a portable 3D ultrasound imaging system for musculoskeletal tissues, Ultrasonics 43 (3) (2005) 153-163.

[11] T. Loupas, W. McDicken, P. Allan, An adaptive weighted median filter for speckle suppression in medical ultrasonic images, IEEE Trans. Circuits Syst. 36 (1) (1989) 129-135.

[12] M. Karaman, M. A. Kutay, G. Bozdagi, An adaptive speckle suppression filter for medical ultrasonic imaging, IEEE Trans. Med. Imag. 14 (2) (1995) 283-292.

[13] G. M. Treece, A. H. Gee, R. W. Prager, C. J. C. Cash, L. H. Berman, High-definition freehand 3D ultrasound, Ultrasound Med. Biol. 29 (4) (2003) 529-546.

[14] R. W. Prager, R. N. Rohling, A. H. Gee, L. Berman, Rapid calibration of 3D freehand ultrasound, Ultrasound Med. Biol. 24 (6) (1998) 855-869.

[15] L. Yin, R. K. Yang, M. Gabbouj, Y. Neuvo, Weighted median filters: A tutorial, IEEE Trans. Circuits Syst. II 43 (3) (1996) 157-192.

[16] R. N. Rohling, A. H. Gee, L. Berman, Automatic registration of 3D ultrasound images, Ultrasound. Med. Biol. 24 (6) (1998) 841-854.

[17] A. H. Gee, G. M. Treece, R. W. Prager, C. J. C. Cash, L. Berman, Rapid registration for wide field of view freehand three-dimensional ultrasound, IEEE Trans. Med. Imag. 22 (11) (2003) 1344-1357.

[18] Q. H. Huang, Y. P. Zheng, A new scanning approach for limb extremities using a water bag in freehand 3D ultrasound, Ultrasound. Med. Biol. 31 (4) (2005) 575-583.

[19] C. A. R. Hoare, Quicksort, Comp. J. 5 (1) (1962) 10-15.

Figure captions

Fig. 1. The B-scans transformed to the volume coordinate system in the volume reconstruction of 3D freehand US.

Fig. 2. The calculation of voxel intensity for the DW interpolation and the reconstructed volume data with filled gaps. (a) A spherical region centred about a voxel is predefined before 3D image reconstruction. Pixels falling into this region are weighed regarding to their inverse distances to the voxel centre and then averaged to obtain the voxel intensity. (b) A typical sequence of original B-scan images with certain gaps. (c) A voxel array (volume) was reconstructed using the B-scan images shown in (b). The gaps between raw B-scans can be filled if the radius of the spherical region is large enough.

Fig. 3. The proposed data structure for storing the intensities of pixels falling into the spherical region and their corresponding distances to the voxel centre. Each voxel was associated with such a data structure.

Fig. 4. Averaged interpolation error in grey level. (a) The SM, DWM-1 and DWM-2 were compared with the VNN and DW methods; and (b) the GWM methods with σ=0.01 and 0.1, respectively, were compared with the VNN and DW methods.

Fig. 5. A typical interpolated slice using the VNN, DW and the median filters at the 50% test. Images (a)-(f) were reconstructed using methods: (a) VNN, (b) DW, (c) SM, (d) DWM-1, (e) DWM-2, and (f) GWM (σ=0.01). Images (g)-(l) show the magnified views of the selected regions in images (a)-(f).

Fig. 6. The same slice as shown in Fig. 5 using the VNN, DW and the median filters at the 100% test. Images (a)-(f) were reconstructed using methods: (a) VNN, (b) DW, (c) SM, (d) DWM-1, (e) DWM-2, and (f) GWM (σ=0.1). Images (g)-(l) show the magnified views of the selected regions in images (a)-(f).

Fig. 7. The same slice as shown in Fig. 5 using the VNN, DW and the median filters at the 300% test. Images (a)-(f) were reconstructed using methods: (a) VNN, (b) DW, (c) SM, (d) DWM-1, (e) DWM-2, and (f) GWM (σ=0.1). Images (g)-(l) show the magnified views of the selected regions in images (a)-(f).

Fig.8. Examples for interpreting why a homogeneous phantom with regular shape was not used in the evaluation tests. (a) A phantom example with homogeneous materials. An original slice and a reconstructed slice are not matched well. The interpolation error can only be calculated from mismatched edge regions, because the internal intensities are approximately the same. (b) A phantom with more complicated internal structures. When the original slice and the reconstructed slice are not matched well, the interpolation error can also be contributed by internal regions.

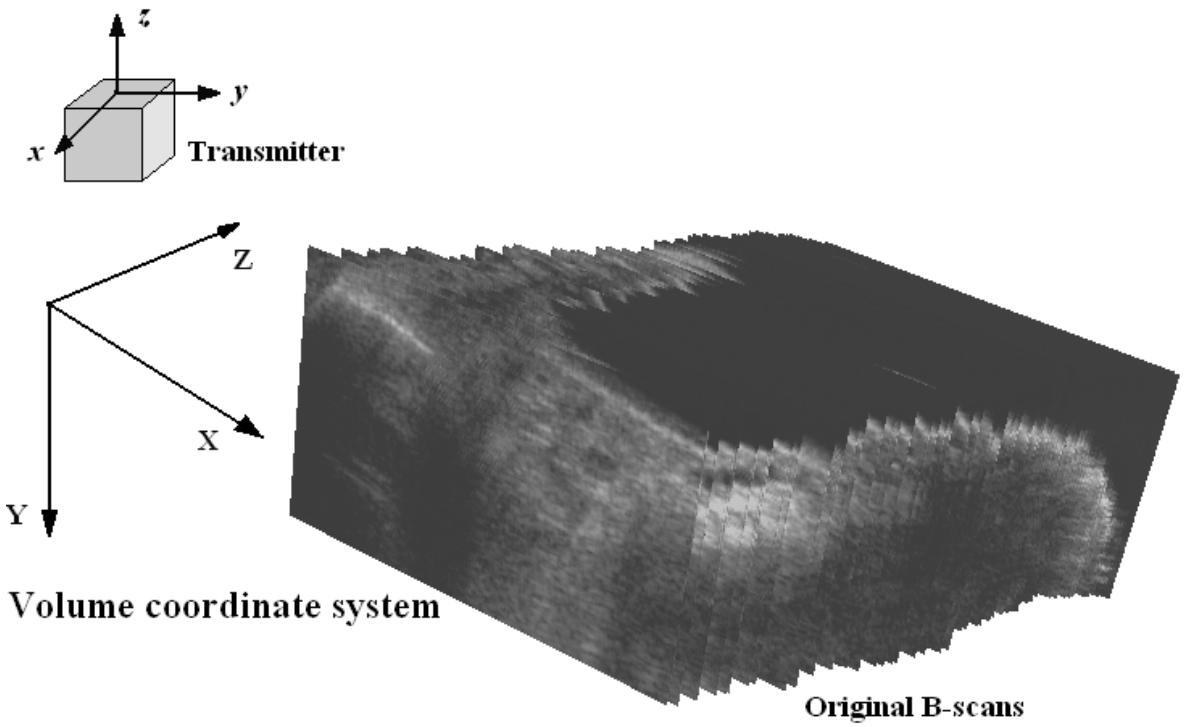

z
y
x
Transmitter
Z
X
Y
Volume coordinate system
Original B-scans

Fig. 1

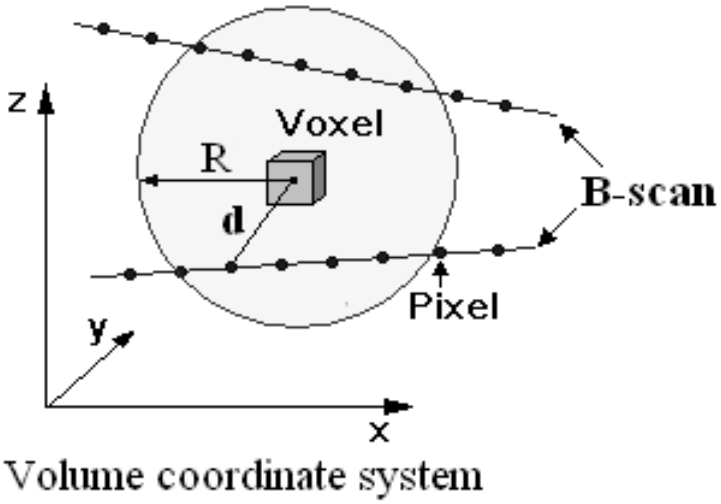
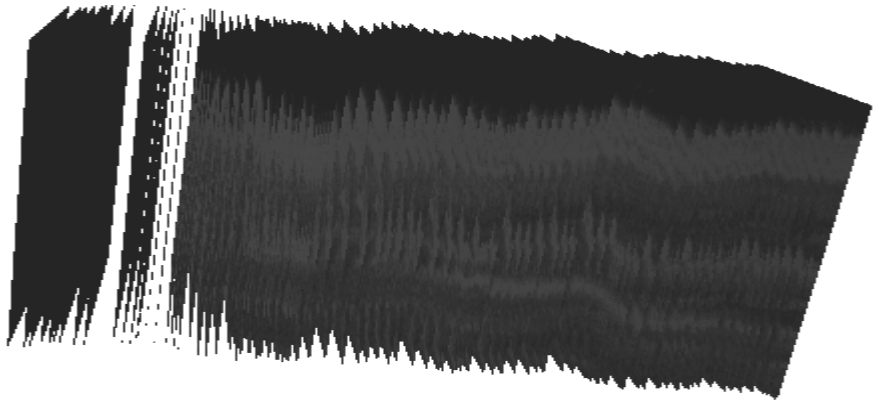
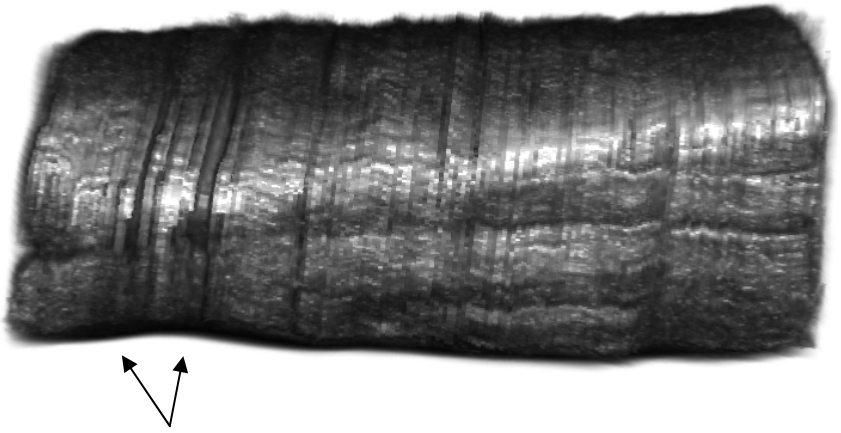

z
Voxel
R
d
B-scan
Pixel
y
x
Volume coordinate system

Fig. 2(a)


Gaps that existed in
raw B-scan data set

(b)


Gaps that have
been filled in

(c)

Fig. 2

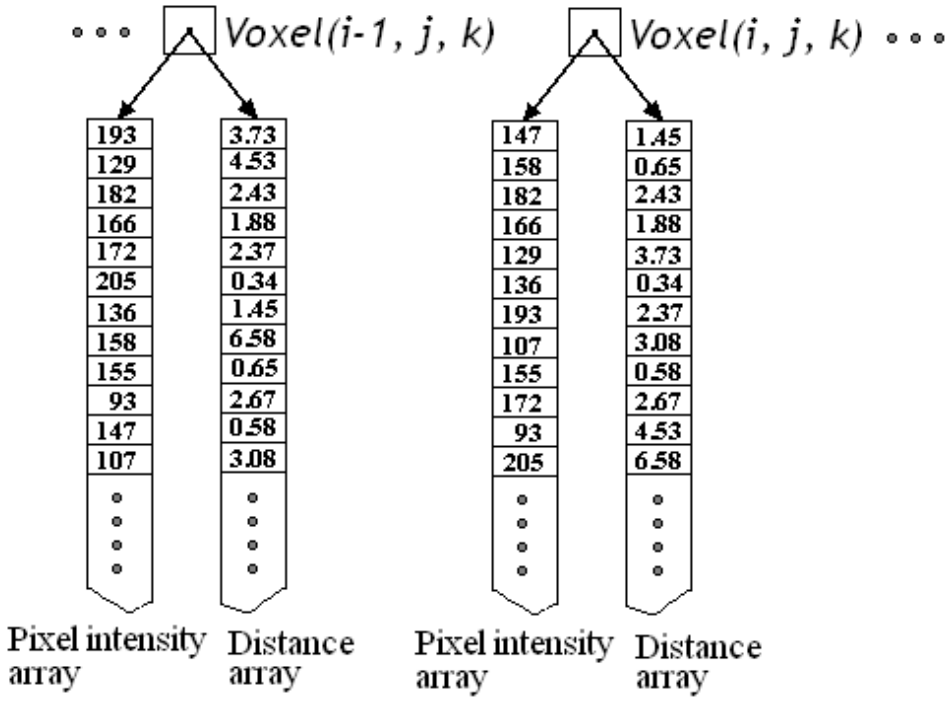
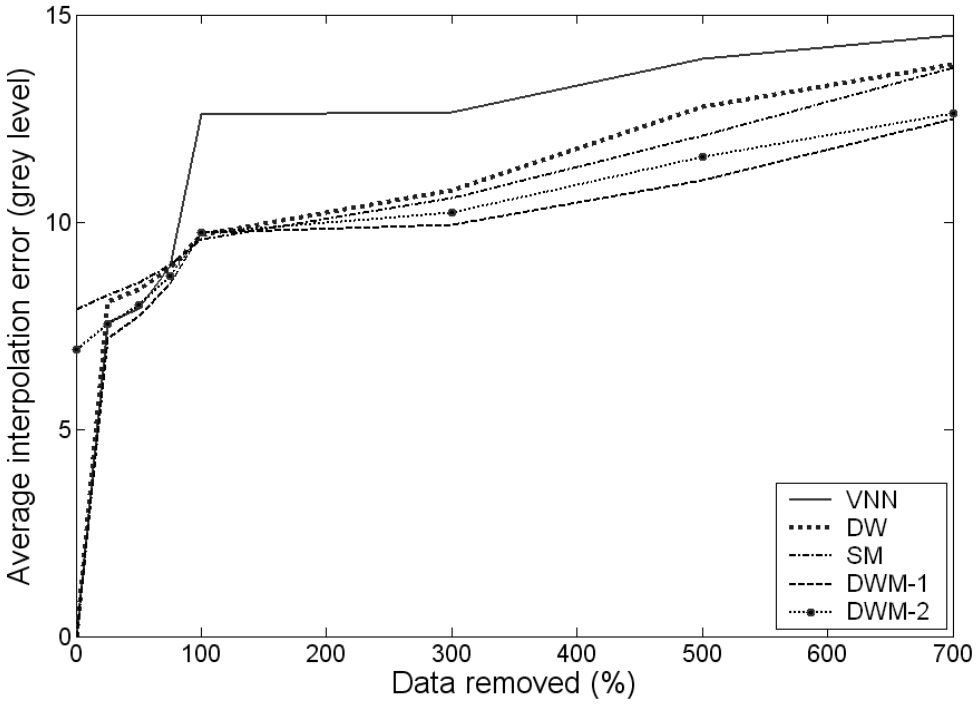
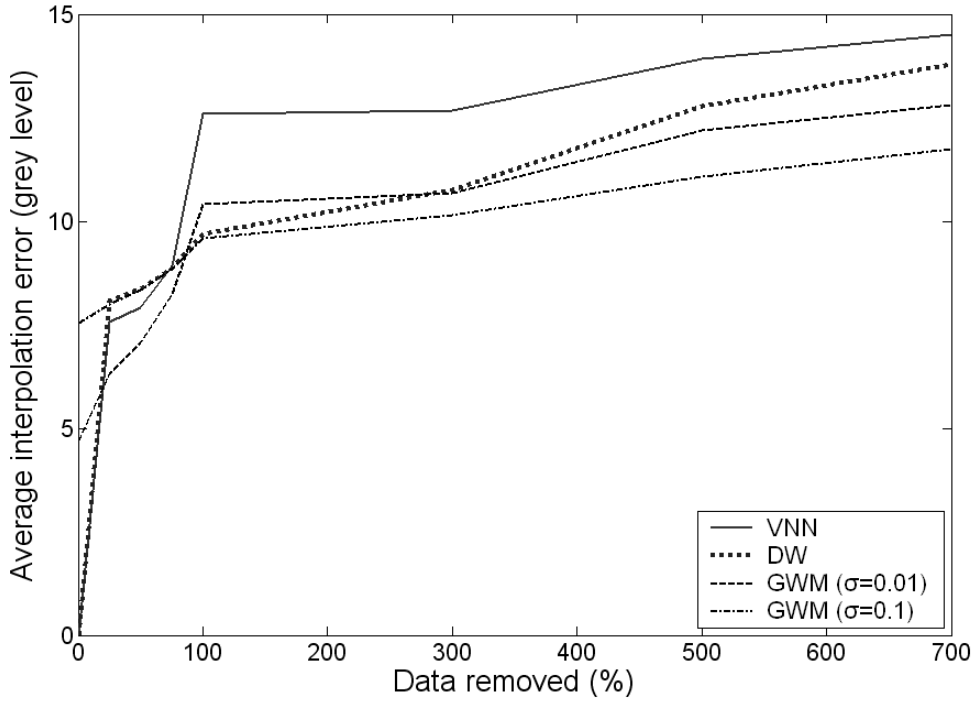

Voxel(i-1, j, k)
Voxel(i, j, k)
193
129
182
166
172
205
136
158
155
93
147
107
3.73
4.53
2.43
1.88
2.37
0.34
1.45
6.58
0.65
2.67
0.58
3.08
147
158
182
166
129
136
193
107
155
172
93
205
1.45
0.65
2.43
1.88
3.73
0.34
2.37
3.08
0.58
2.67
4.53
6.58
Pixel intensity array
Distance array
Pixel intensity array
Distance array

Fig. 3

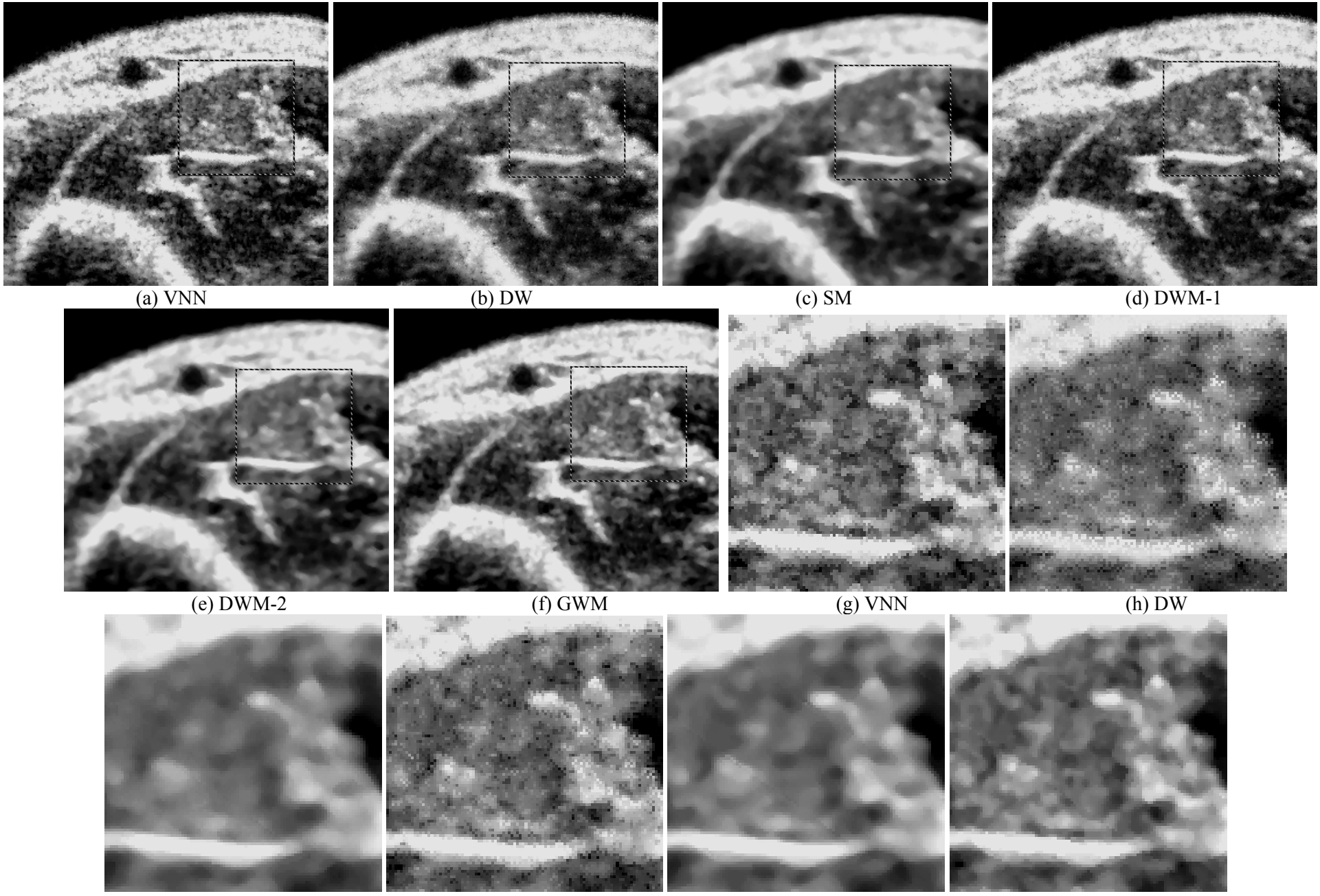

Average interpolation error (grey level)
Data removed (%)
VNN
DW
SM
DWM-1
DWM-2
0
5
10
15
0
100
200
300
400
500
600
700

(a)


Average interpolation error (grey level)
Data removed (%)
VNN
DW
GWM (σ=0.01)
GWM (σ=0.1)
0
5
10
15
0
100
200
300
400
500
600
700

(b)

Fig. 4

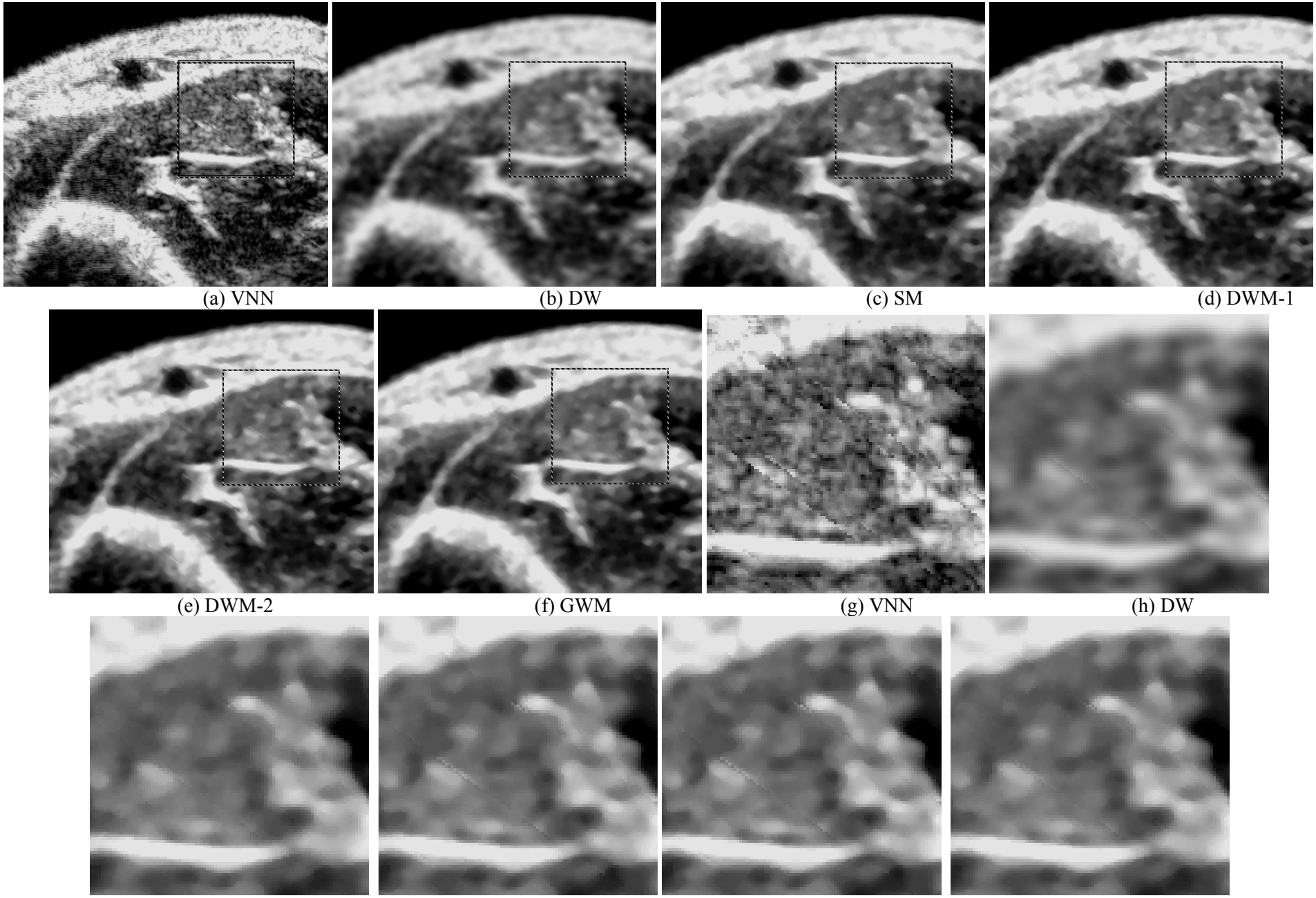


(a) VNN (b) DW (c) SM (d) DWM-1

(e) DWM-2 (f) GWM (g) VNN (h) DW

(i) SM (j) DWM-1 (k) DWM-2 (l) GWM

Fig. 5

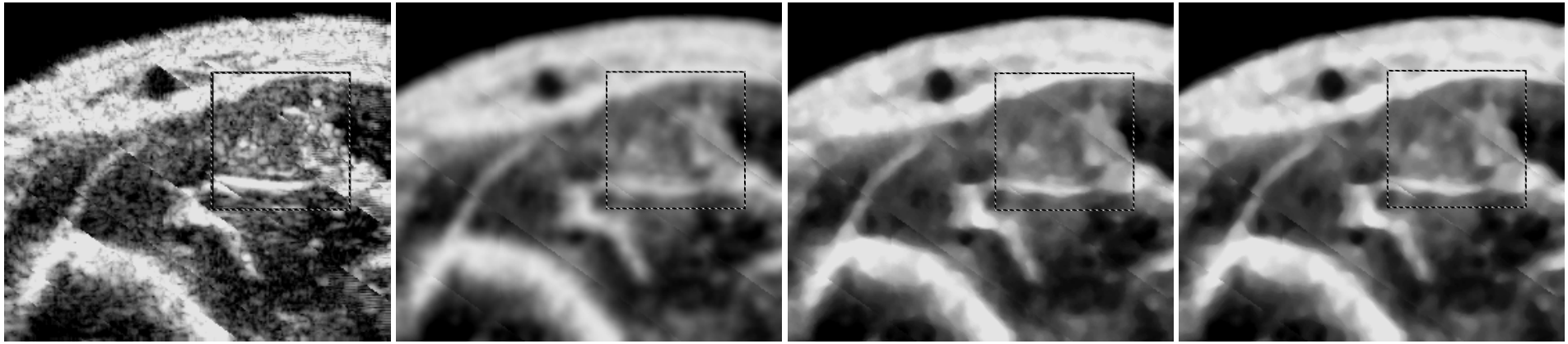


(a) VNN (b) DW (c) SM (d) DWM-1

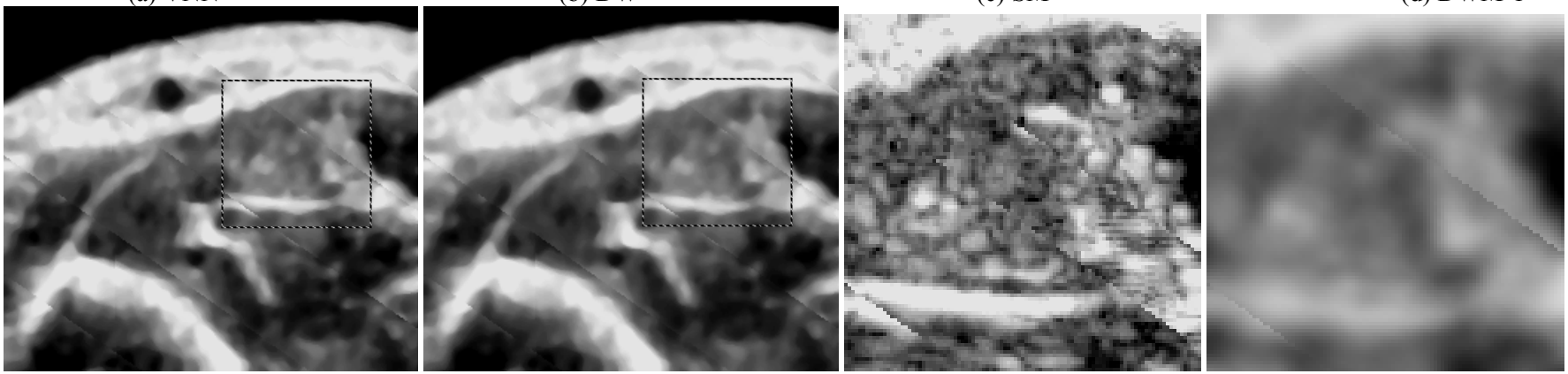
(e) DWM-2 (f) GWM (g) VNN (h) DW

(i) SM (j) DWM-1 (k) DWM-2 (l) GWM

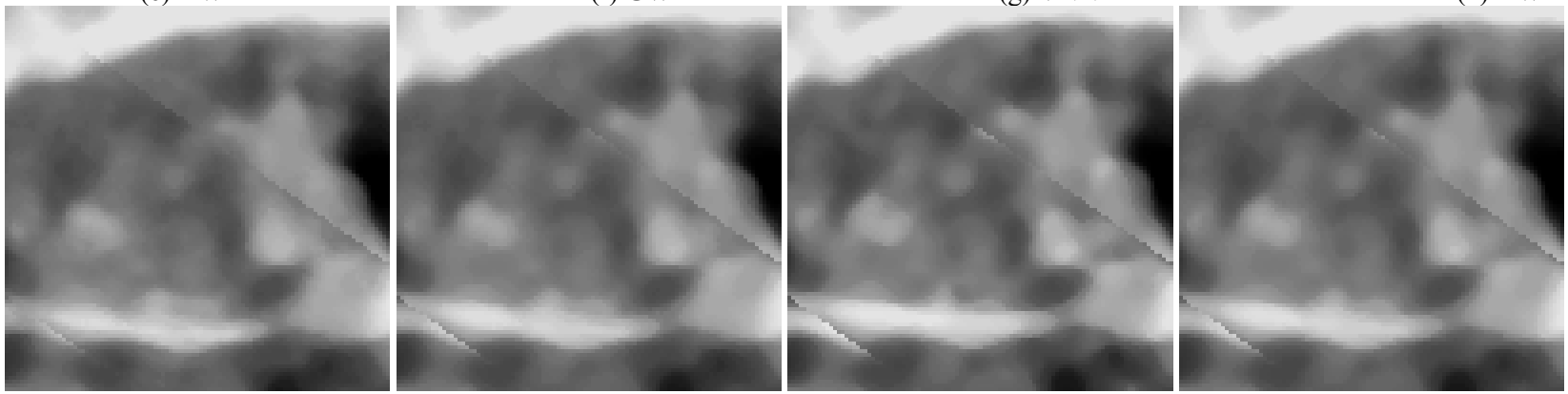
Fig. 6



(a) VNN (b) DW (c) SM (d) DWM-1

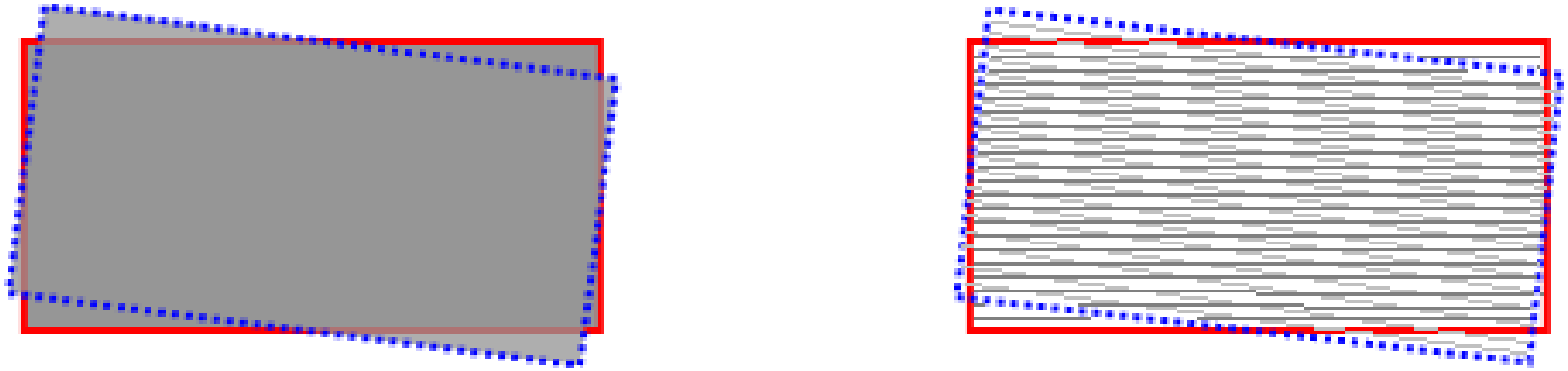


(e) DWM-2 (f) GWM (g) VNN (h) DW



(i) SM (j) DWM-1 (k) DWM-2 (l) GWM

Fig. 7



(a) (b)

Fig. 8

Table 1. Averaged interpolation error *V* using the VNN, DW, SM, DWM-1, DWM-2 and GWM methods. *SD* indicates standard deviation. The unit of the data is grey level (with a range of 0 to 255).

	VNN		DW		SM		DWM-1		DWM-2		GWM (σ=0.01)		GWM (σ=0.025)		GWM (σ=0.05)		GWM (σ=0.075)		GWM (σ=0.1)	
	Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
0%	**0.00**	0.00	**0.00**	0.00	7.88	0.59	**0.00**	0.00	6.91	0.47	4.69	0.77	6.25	0.66	7.10	0.71	7.40	0.73	7.55	0.53
25%	7.56	0.80	8.08	0.46	8.23	0.58	7.19	0.50	7.54	0.52	**6.32**	0.81	7.16	0.78	7.69	0.81	7.89	0.83	7.99	0.54
50%	7.91	0.86	8.37	0.51	8.52	0.68	7.72	0.58	8.01	0.60	**7.05**	0.91	7.73	0.94	8.12	0.96	8.26	0.97	8.33	0.63
75%	8.93	0.94	8.88	0.65	8.95	0.88	8.50	0.76	8.69	0.81	**8.22**	1.04	8.56	1.10	8.75	1.10	8.81	1.11	8.85	0.83
100%	12.59	1.33	9.69	0.85	**9.57**	1.03	9.75	1.01	9.74	1.06	10.41	1.24	10.29	1.45	10.09	1.33	9.78	1.40	9.59	1.01
300%	12.66	1.37	10.76	0.90	10.57	0.70	**9.94**	0.84	10.24	0.71	10.68	1.34	10.64	1.32	10.14	1.44	10.10	1.36	10.14	0.74
500%	13.93	2.28	12.77	1.33	12.06	0.84	**11.01**	1.17	11.58	0.94	12.20	2.29	11.49	1.41	11.13	1.34	11.05	0.96	11.08	1.36
700%	14.51	2.06	13.80	0.92	13.70	0.74	12.48	1.03	12.54	0.70	12.81	2.11	12.07	1.81	11.73	1.55	**11.67**	1.16	11.75	1.17

Table 2. Averaged computation times for the VNN, DW, SM, DWM-1, DWM-2 and GWM methods at the 100% test. *SD* indicates standard deviation. The unit of the data is minute. The software was programmed in VC++ and run in a PC with a 2.8 GHz Pentium IV processor and 1 GB of RAM.

VNN		DW		SM		DWM-1		DWM-2		GWM	
Mean	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*	*Mean*	*SD*
27.6	2.2	8.8	0.5	42.7	2.3	44.5	2.8	43.8	2.7	48.7	2.7