Ultrasound in Med. & Biol., Vol. 33, No. 7, pp. 991–1009, 2007

0301-5629/07/$–see front matter

ELSEVIER

doi:10.1016/j.ultrasmedbio.2007.02.015

● *Review*

FREEHAND 3D ULTRASOUND RECONSTRUCTION ALGORITHMS—A REVIEW

Ole Vegard Solberg,*†‡ Frank Lindseth,*† Hans Torp,‡ Richard E. Blake,§ and Toril A. Nagelhus Hernes*†‡

*SINTEF Health Research; Medical Technology, Trondheim, Norway; †National Center for 3D Ultrasound in Surgery, Trondheim, Norway; ‡Norwegian University of Science and Technology (NTNU), Faculty of Medicine; Department of Circulation and Medical Imaging, Trondheim, Norway; and §NTNU, Faculty of Information Technology, Mathematics and Electrical Engineering; Department of Computer and Information Science, Trondheim, Norway

(*Received* 21 *July* 2006; *revised* 31 *January* 2007; *in final form* 25 *February* 2007)

Abstract—Three-dimensional (3D) ultrasound (US) is increasingly being introduced in the clinic, both for diagnostics and image guidance. Although dedicated 3D US probes exist, 3D US can also be acquired with the still frequently used two-dimensional (2D) US probes. Obtaining 3D volumes with 2D US probes is a two-step process. First, a positioning sensor must be attached to the probe; second, a reconstruction of a 3D volume can be performed into a regular voxel grid. Various algorithms have been used for performing 3D reconstruction based on 2D images. Up till now, a complete overview of the algorithms, the way they work and their benefits and drawbacks due to various applications has been missing. The lack of an overview is made clear by confusions about algorithm and group names in the existing literature. This article is a review aimed at explaining and categorizing the various algorithms into groups, according to algorithm implementation. The algorithms are compared based on published data and our own laboratory results. Positive and practical uses of the various algorithms for different applications are discussed, with a focus on image guidance. (E-mail: ole.v.solberg@sintef.no)

Key Words: **3D ultrasound, Reconstruction methods, Positioning, Freehand acquisition, 3D imaging, Ultrasound compounding.**

INTRODUCTION

Minimally invasive surgery or image-guided surgery is an important field in therapy, becoming more and more widespread. To minimize the intervention, high demands must be made to the imaging modalities used due to image quality and accuracy. During surgery, intraoperative imaging is needed in addition to preoperative images, since changes occur during surgery. Even though the more commonly used imaging technologies such as magnetic resonance imaging (MRI) and computed tomography (CT) are possible to use during surgery, there are still some significant practical limitations, due to costs, equipment adaptation in the magnetic field, user friendliness, image quality and radiation doses. Three-dimensional (3D) ultrasound (US) is already being introduced alone or together with preoperational images for guidance of surgical applications.

Two-dimensional (2D) US is being extensively used for a variety of clinical applications and 3D US is now also more frequently demonstrated in the clinic. The main advantage of 3D US is that arbitrary 2D images through the volume may be visualized and not only images in the same plane as the US acquisition is performed, which is the only option with 2D US. 3D allows views not possible with 2D. In addition, 3D US also allows a 3D volume rendered view and 3D segmentations of objects. Two different main approaches for 3D US creation exist: using a dedicated 3D US probe or using a regular 2D US probe for acquiring the images and combining these 2D slices to a 3D volume. A 3D US probe may be a 2D array acquiring 3D volumes directly or a mechanical 3D probe consisting of a regular one-dimensional (1D) array acquiring multiple 2D images with a motor that sweeps the 1D array over the scanned area in a certain manner: linear, tilt or rotational (Fenster

Address correspondence to: Ole Vegard Solberg, SINTEF Health Research, Medical Technology, 7465 Trondheim, Norway. E-mail: ole.v.solberg@sintef.no

et al. 2001). The 3D reconstructions with motorized probes are very similar to a freehand 3D reconstruction, although the 2D image positions have a more regular pattern than that for freehand 3D US. A 2D array US probe may also be used to acquire real-time 3D volumes (also called 4D).

3D volume reconstruction from free-hand acquired 2D images usually needs position data of the 2D slices. The most common method for obtaining positioning data is to attach a position sensor to the probe: electromagnetic, optical, mechanical arm or acoustic. [See Cinquin et al. (1995) for a description of these positioning systems.] However, some systems use alternative methods such as the I-beam probe (Hossack et al. 2000), where the image positions are tracked with respect to each other with a special probe configuration and the US data. Other methods do not use any external positioning measurements at al.: the predefined operator probe movement (Downey and Fenster 1995), speckle decorrelation (Tuthill et al. 1998), frame distance estimation (Lee et al. 2001) or linear regression (Prager et al. 2003). In the literature so far, the sensorless methods have not been shown to give the same accuracy as tracking systems.

In addition to the reconstruction algorithm itself, several factors affect the 3D volume reconstruction accuracy. High quality 3D reconstructions depend on both the quality of the input 2D images and the accuracy of the position data. Tracking system inaccuracy, the ultrasound probe calibration process, sound of speed variation and tissue movement are all important error sources that are handled elsewhere (Lindseth et al. 2002; Treece et al. 2002; Mercier et al. 2005).

Earlier work has been done to explain 3D US in general (Nelson and Pretorius 1998; Fenster et al. 2001), 3D US in neurosurgery specifically (Unsgaard et al. 2006) and some of the different reconstruction algorithms and a grouping of these (Rohling et al. 1999). The article by Rohling et al. (1999) is often referred to by others for demonstrating examples of different algorithms and is sometimes also given as reference to specific algorithms as well. Some confusion exists in the literature about algorithm and group names along with some unclear algorithm origins. We believe that these problems arise from lack of a clear overview of freehand 3D US reconstruction algorithms.

The present article will, therefore, provide a thorough description and grouping of the various freehand 3D reconstruction algorithms with focus on the recently published. Although choice of positioning system and probe calibration also affects reconstruction accuracy, this will not be the focus of this article. For a comprehensive review, see the article by Mercier et al. (2005). Benefits and drawbacks of the various 3D reconstruction algorithms will be discussed with emphasize on time usage, quality, practical implementation and usefulness in image-guided surgery applications.

DESCRIPTIONS OF RECONSTRUCTION ALGORITHMS

In the following, the different reconstruction algorithms have been sorted into three groups based on implementation: Voxel-Based Methods (VBM), Pixel-Based Methods (PBM) and Function-Based Methods (FBM). VBMs traverse all voxels in a target volume and inserts corresponding pixels from the input images. PBMs traverse the input pixels and insert them into the corresponding target volume voxels. FBMs estimate functions of the input data that are used for creating the voxel grid.

Some of the algorithms in this section are explained with illustrations, shown as 2D simplification of the 3D cases. In these illustrations, the 3D voxel grid is shown as a 2D grid marking the centers of the voxels and the 2D input images are illustrated as lines where the pixel centers are marked. An overview of the different algorithms discussed in the article can be found in Table 1.

Voxel-Based Methods (VBM)

VBMs traverse each voxel in the target voxel grid and gather information from the input 2D images to be placed in the voxel. In the different algorithms, one or several pixels may contribute to the value of each voxel.

Each voxel is assigned the value of one pixel. Some methods use only one input pixel for deciding the value of a voxel. An algorithm usually implemented this way is the Voxel Nearest Neighbor (VNN) (Sherebrin et al. 1996). VNN traverses each volume voxel and assigns the value of the nearest image pixel (Fig. 1). Stradx (Prager et al. 1999) is a real-time acquisition and visualization system that uses a method that in practice works as a VNN interpolation. Stradx achieves a fast reconstruction of arbitrarily oriented 2D slices through the set of 2D images without creating a 3D volume. Stradx traverses each pixel in the target 2D slice and inserts the value of the nearest pixel in the input images. The Stradx system also supports nonplanar reslices, resampling data along a curved surface which is unrolled for display on a flat screen (Gee et al. 1999). Even though Stradx is placed in the same group as VNN, it is quite different from other reconstruction algorithms by not using a voxel grid at all.

Voxel-Based Methods With Interpolation (VBMI). VBMI use an interpolation between several input pixel values for deciding a voxel value.

1D interpolation (Berg et al. 1999) is the reconstruction implementation used in the EchoPAC-3D software (GE Vingmed Ultrasound, Horten, Norway). This

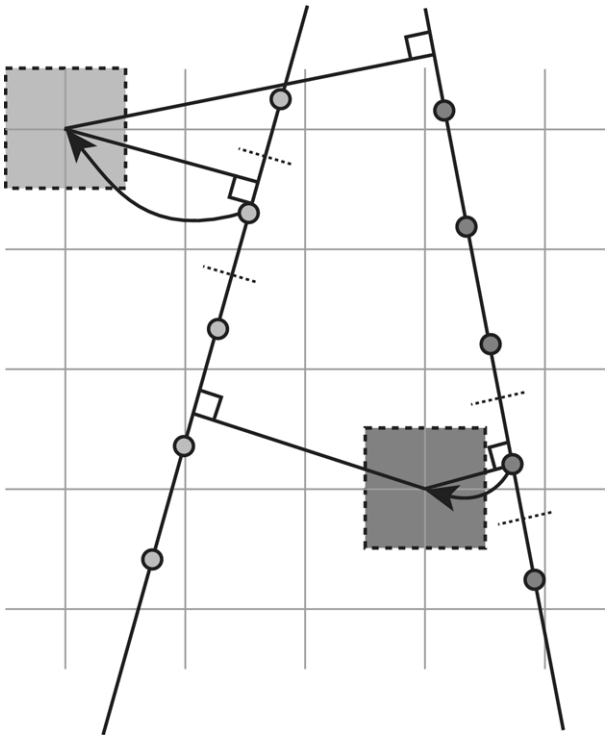

Fig. 1. VNN. A VBM that traverse each voxel and assign the value of the nearest input pixel. A normal from the voxel to the image is calculated and the nearest pixel value is inserted into the voxel. (The input images are illustrated by lines and the points in the lines illustrate the centers of the pixels. The 2D grid marks the centers of the voxels in the 3D voxel grid.)

method is based on acquiring the raw data from the US scanner, but the method can also be used with video grabbed images, since the reconstruction is done along one dimension; this may be done based on either a 2D image or a 1D scan line. Berg et al. (1999) use a tilt scan and present two interpolation methods based on the maximum angle between two enclosing scan planes. If the maximum angle is less than 20°, a linear interpolation orthogonal to a virtual scan plane in the middle of the scan planes are used (Fig. 2a). If the maximum angle is more than 20°, each 2D scan plane is defined with a certain thickness defining the region where it contributes to the interpolation (Fig. 2b). The interpolation in both cases uses a linear weight decreasing with the distance to the scan planes. The algorithm is used to create 4D volumes in a region of the human heart by also storing ECG signals. Another description of the EchoPAC-3D software (Martens and Gilja 2005) indicates that only the second method (Fig. 2b) is currently being used and that compounding may also be achieved with the same algorithm.

An interpolation from the two nearest surrounding slices is used by Trobaugh et al. (1994). This algorithm traverses the voxels and finds the two nearest 2D slices on each side of the voxel (Fig. 3). A normal is calculated to each of these slices and the four surrounding pixels are bilinearly interpolated in each plane. The final voxel value is calculated as a weighted sum with contributions from the two planes based on the distances from the voxel to the planes.

The Probe Trajectory (PT) method (Coupé et al. 2005) builds on the algorithm by Trobaugh et al. (1994). Instead of using an orthogonal projection of the closest 2D planes to a point, a probe trajectory is estimated and used for finding the corresponding pixels in the nearest 2D planes (Fig. 4). This is done by computing a "virtual" plane through the point to be calculated. First, the probe distance to this "virtual" plane is estimated by interpolating the time stamp of the two closest 2D planes. This estimate assumes a constant probe speed between these planes. Then a cubic interpolation of position parameters at the estimated time stamp is done. Four B-scans are used for this cubic interpolation. The values to be used in the interpolation can be obtained directly from the nearby planes; because the "virtual" plane will have the same in-plane 2D coordinates.

Voxel-based tri-linear interpolation (Thune et al. 1996) is based on acquiring the raw data from the US scanner. The 3D volume is then reconstructed from the 1D scan lines instead of the 2D scan converted planes. Each voxel is interpolated from the eight closest points in the scan lines.

Pixel-Based Methods (PBM)

PBMs traverse each pixel in the input images and assign the pixel value to one or several voxels. A PBM may consist of two steps: a Distribution Step (DS) and a Hole-Filling Step (HFS). In the DS, the input pixels are traversed and the pixel value applied to one or several voxels, often stored together with a weight value. In the HFS, the voxels are traversed and empty voxels are being filled. Most hole-filling methods have a limit on how far from away from known values the holes are filled, so if the input images are too far apart or the hole-filling-limits are too small there will still be holes in the constructed volume.

One pixel contributes to one voxel. In this method, each input pixel value is only applied to one voxel. Several pixels may in this way contribute to the same voxel if the algorithm supports this.

In the bin-filling stage of Pixel Nearest Neighbor (PNN), the algorithm runs through each pixel in all the 2D US images. Each pixel value is filled into the nearest voxel (Fig. 5a). Multiple contributions to the same voxel are usually averaged (Nelson and Pretorius 1997; Gobbi and Peters 2002), but different variants are possible, like

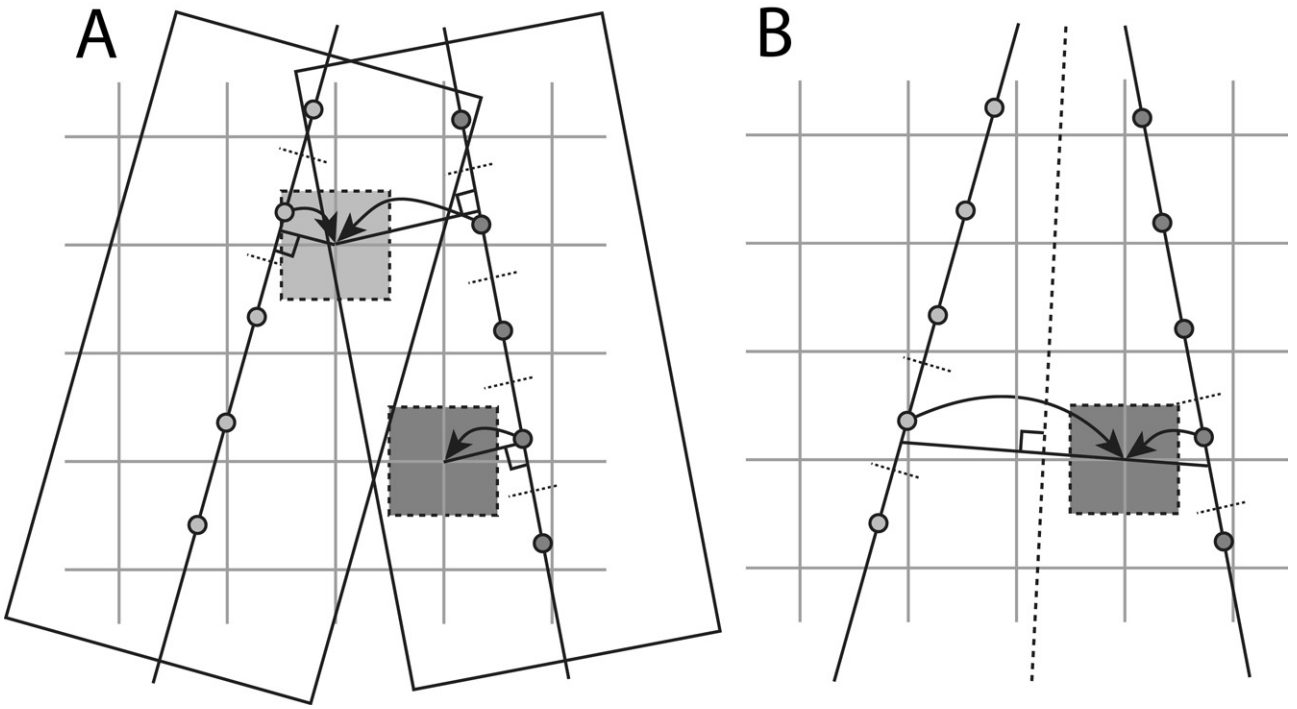

Fig. 2. 1D interpolation. (a) A VBM with 1D interpolation orthogonal to each of the nearby scan planes. Each voxel value is assigned a value orthogonal to the nearest scan planes. If more scan planes lie within a certain range an average is calculated, weighted with the inverse of the distance to the planes. (b) VBM with 1D interpolation orthogonal to a virtual middle plane. A virtual middle plane is created through the center of each voxel. The voxel value is calculated as a linear interpolation orthogonal to this plane, with use of each nearest pixel in each of the nearest enclosing planes. (In both illustrations, the input images are illustrated by lines and the points in the lines illustrate the centers of the pixels. The 2D grids mark the centers of the voxles in the 3D voxel grids.)

keeping the maximum value (Nelson and Pretorius 1997), the most recent value (Ohbuchi et al. 1992) or the first value (Trobaugh et al. 1994). Most PBMs use a PNN bin-filling as the Distribution Step (DS) and, sometimes, it may not be necessary with a Hole-Filling Step (HFS) afterwards if the input images are close enough (Gobbi and Peters 2002).

The Solus system (Carr et al. 2000) is based on the Stradx system (Prager et al. 1999). Instead of using a global VNN, the Solus system implements a localized approach to a VNN interpolation. In the DS, each 2D image is inserted into a 3D voxel grid with a predefined image thickness corresponding to the thickness of the US image (Fig. 6). The voxels intersecting the thick input image are assigned the value of the nearest pixel and the distance to the value in the voxel is stored. If the voxel already has a value only the value from the nearest pixel is used. A HFS is not used/needed because of the image thickness in the DS. The result is similar to a VNN interpolation with holes where the input 2D images are too far away from the voxels.

PNN with hole-filling from a local neighborhood. After the DS, there are usually holes in the voxel array, especially if the volume has not been scanned with a dense sampling. In the HFS, the volume from the DS is traversed and each empty voxel is attempted filled with the information of the nearby already filled voxels (Fig. 5b). Various methods have been presented for this purpose: an average of nonzero pixels from an intersecting 2D plane (McCann et al. 1988), an average or maximum (Nelson and Pretorius 1997; San José-Estépar et al. 2003a) or a median (San José-Estépar et al. 2003a) of nonzero voxels in a 3D local neighborhood, or interpolation between the two closest nonzero voxels (Hottier and Billon 1990). Nelson and Pretorius (1997) propose a method of reducing the resolution of the voxel grid in steps by a factor of two in each step (in each direction). It is unclear if they insert the 2D slices again in this reduced grid or use the already existing grid for generating the reduced version. After the reconstruction, Nelson and Pretorius (1997) propose to use a $3 \times 3 \times 3$ Gaussian or median filter to improve the volume data.

PNN with a 3D kernel around filled voxels. Instead of basing the hole-filling on the holes, it is possible to base it on the already filled voxels by applying this value with a weighting to the nearby voxels. In the HFS, these

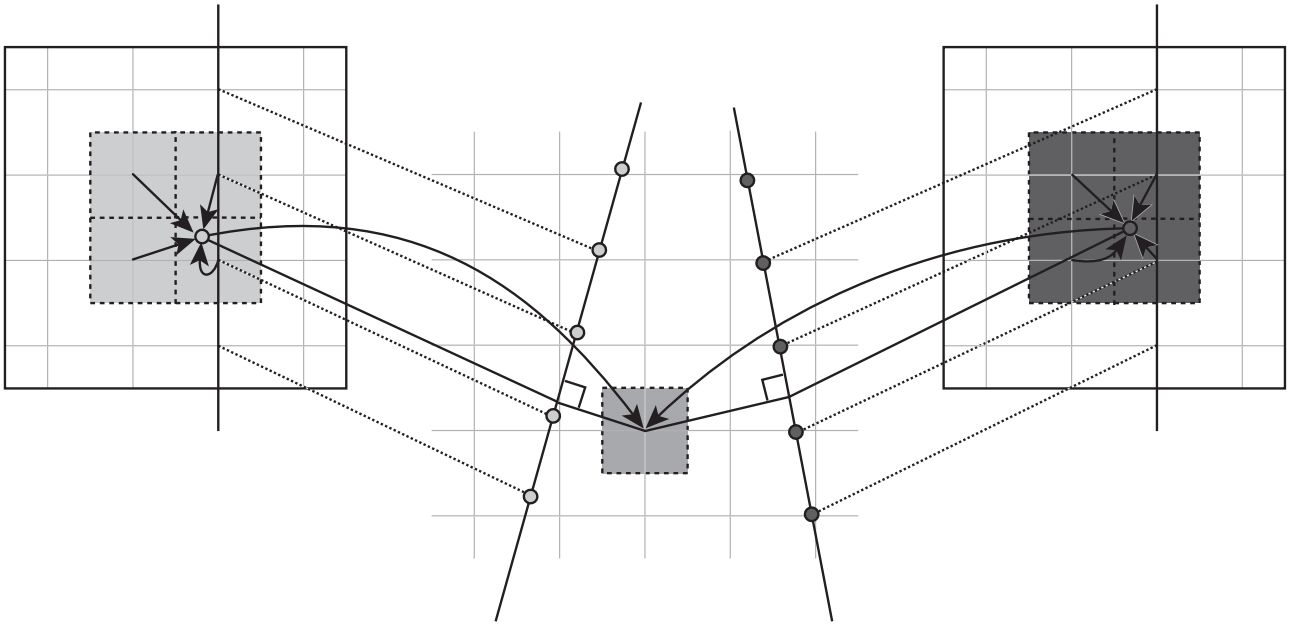

Fig. 3. VBM with interpolation from the two nearest surrounding images. A normal to each of the surrounding images is determined. A contribution from each image is determined by bilinear interpolation of the four nearest pixels. Each voxel is assigned a value based on a distance-weighted interpolation of the contributions from the two surrounding images. (The 2D grid marks the centers of the voxels in the 3D voxel grid. The input images are illustrated by lines in the voxel grid illustration, and separately by 2D grids on each side. The points in the lines illustrate the centers of the pixels.)

algorithms traverse the voxels and apply the voxel value to a local neighborhood (called kernel) around the voxels filled in the DS step.

An adaptive spherical Gaussian kernel is presented by San José-Estépar et al. (2003b). Each filled voxel is applied to the neighboring voxels with the help of a spherical Gaussian kernel (Fig. 7a). The variance of the kernel is dependent of the variance of the intensity of the nearby pixels. This pixel variance is calculated based on the PNN bin-filling step. The result is kernels of various sizes according to the density of the input data. A bank of precomputed kernels is used to speed up the interpolation process. The result is then normalized to avoid an increase in intensity values.

Normalized Convolution (Knutsson and Westin 1993) is used by San José-Estépar et al. (2003a) as a method of interpolating the data after the PNN bin-filling stage (Fig. 5a) has been performed. The Point Spread Function of the US system is used as an input to the applicability function used in the Normalized Convolution. This applicability function will determine the extent of both the kernel and the weighting that the input values are weighted with in the neighboring voxels. See Fig. 7b for illustration. The Normalized Convolution method also uses a certainty function, which is used for weighting the accuracy of the data. San José-Estépar et al. (2003a) uses this to apply a certain weight to the data locations from the bin-filling stage, while the empty voxels are weighted as empty.

3D kernel around input pixels. Instead of running a two-step algorithm like the PNN, some algorithms do both the insertion and hole-filling in the DS. These algorithms are traversing the input pixels and assigning the pixels value into the voxels in a local neighborhood (called kernel) around the pixel position. In addition to the pixel values, a weight value is associated with each voxel. The weight value is used for calculating the final voxel values when several pixels supply values to the voxel. In this process, the actual pixel positions are used, since the pixel and voxel positions are not exactly the same. This is in contrast to the PNN algorithms with a 3D kernel around the filled voxels.

Barry et al. (1997) use a spherical kernel with an inverse distance weighting. This is the most commonly used example of this kind of kernel-based algorithms. This algorithm can be seen as a simplified variant of the algorithm presented by Ohbuchi et al. (1992). The algorithm uses a spherical kernel around each pixel (Fig. 8a). This is then used to calculate a normalized distance weighted average of all pixels that lie in the neighborhood of the voxels. Barry et al. (1997) also store an ECG signal together with the input images for elimination of motion during the cardiac cycle. The user may then

Table 1. Grouping of different reconstruction algorithms

References	Implementation variant
Voxel-Based Methods (VBM): Traverse target volume and gather information	
Each voxel is assigned the value of one pixel	
Sherebrin et al. (1996)	Standard VNN (Fig. 1)
Prager et al. (1999; 2002)	VNN. Stradx (no 3D voxel volume)
Voxel-based methods with Interpolation (VBMI). Each voxel is assigned a value based on several pixels	
Berg et al. (1999), Martens and Gilja (2005)	lD interpolation (Fig. 2)
Trobaugh et al. (1994)	Two surrounding slices interpolation (Fig. 3)
Coupé et al. (2005)	Probe Trajectory (PT) (Fig. 4)
Thune et al. (1996)	Voxel-based tri-linear interpolation
Pixel-Based Methods (PBM): Traverse the 2D input images and distribute the information to the target volume; may consist of a Distribution step (DS) and an additional hole-filling step (HFS)	
DS: One pixel contributes to one voxel, no HFS	
Trobaugh et al. (1994)	DS: PNN bin-filling (Fig. 5a)
Carr et al. (2000)	VNN with image thickness. Solus (Fig. 6)
DS: PNN bin-filling, HFS: traverse empty voxels and fill these from a local neighborhood (Fig. 5)	
McCann et al. (1988)	HFS: Average from an intersecting 2D plane
Hottier and Billon (1990)	HFS: Interpolation between two closest voxels
Nelson and Pretorius (1997)	HFS: Interpolation by reduction of voxel grid
DS: PNN bin-filling, HFS: apply a 3D kernel to already filled voxels to determine impact on nearby voxels	
San José-Estépar et al. (2003b)	Adaptive Gaussian kernel (Fig. 7a)
San José-Estépar et al. (2003a)	Normalized Convolution (Fig. 7b)
DS: Apply a 3D kernel to input pixels to determine impact on nearby voxels	
Barry et al. (1997)	Spherical kernel, linear weighting (Fig. 8a)
Gobbi et al. (2001), Gobbi and Peters (2002)	Pixel trilinear interpolation (PTL)
Ohbuchi et al. (1992)	Ellipsoid Gaussian kernel, Gaussian weighting (Fig. 8b)
Meairs et al. (2000)	Ellipsoid Gaussian kernel, exponential weighting (Fig. 8b)
Function-Based Methods (FBM), see Fig. 9	
Rohling et al. (1999)	Radial basis function interpolation (RBF)
Sanches and Marques (2000; 2002)	Rayleigh reconstruction/interpolation with a Bayesian framework

The columns list references and implementation variant. Pseudo code implementations of some of the algorithms can be found in the Appendix.

specify a range within the cardiac cycle that contains the images to be used in the reconstruction.

Pixel trilinear interpolation (PTL) (Gobbi et al. 2001; Gobbi and Peters 2002) interpolation uses a $2 \times 2 \times 2$ kernel with linear weighting, when applying each pixel in the 2D images to the resulting volume. This is similar to the method presented by Barry et al. (1997)), but with a fixed cubic kernel instead of a configurable spherical kernel. Gobbi and Peters (2002) have implemented both a compounding method [similar to Ohbuchi et al. (1992), Barry et al. (1997) and Meairs et al. (2000)] and an alpha blending method for increased computation speed. The alpha blending provides interpolation without using an accumulation buffer. Only the kernel coefficient is used for determining the compounding weight of the newly inserted pixel compared with the existing voxel value. Compared with the compounding method, the alpha blending approach gives much higher weight to the last inserted pixels, than to those inserted previously.

An ellipsoid truncated Gaussian kernel with Gaussian weighting is presented by Ohbuchi et al. (1992). In this algorithm, each pixel in the input slices is convolved with a 3D ellipsoid truncated Gaussian kernel (Fig. 8b). The 3D Gaussian kernel is used as an emulation of the Point Spread Function (PSF) of the US scanner. Three values are stored for each voxel: the reconstruction value, the weight value and the age. The reconstruction value is a sum of all convolutions between the pixels and the 3D kernel that intersects the voxels. The weight value is a sum of the values of the 3D kernels that intersect the voxels. The reconstruction values are divided by the weight values to obtain a normalized value, often called a compound value. The age values of the pixels are used to calculate a decay factor that may be used for giving

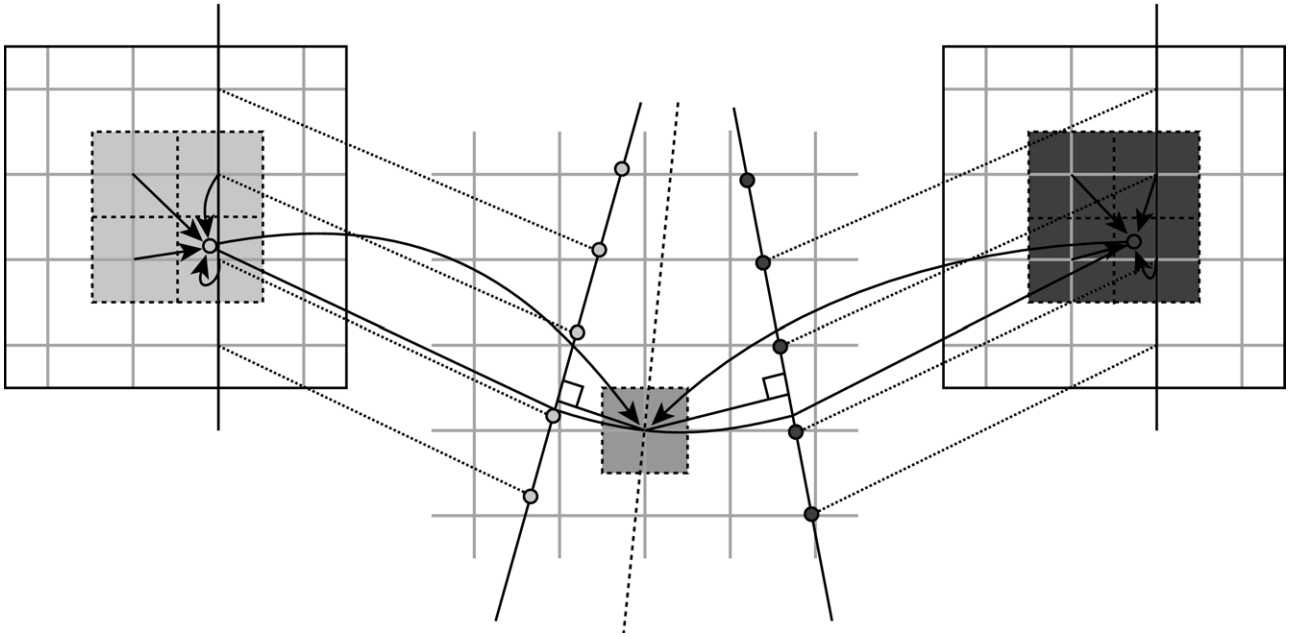

Fig. 4. VBM with interpolation from two nearest images with the PT algorithm. A virtual plane is calculated through the voxel center. The probe trajectory between the two surrounding nearest slices is estimated, assuming a constant probe speed. This probe trajectory is used to find the point on the two planes corresponding to the voxel in the virtual plane. Each of these points gets assigned a value with a bilinear interpolation of the four enclosing pixels. The voxel is assigned a value by a distance-weighted interpolation of the contributions from the two planes. (The 2D grid marks the centers of the voxels in the 3D voxel grid. The input images are illustrated by lines in the voxel grid illustration and separately by 2D grids on each side. The points in the lines illustrate the centers of the pixels.)

more weight to newer pixels than to the older ones. Ohbuchi et al. (1992) describe a method for reconstruction where the input images have both 3 degrees-of-freedom (DOF) and 6 DOF. To speed up the calculations in the 6 DOF case, the in-plane interpolation is linear instead of Gaussian.

The ellipsoid Gaussian kernel with exponential weighting by Meairs et al. (2000) is similar to the algorithm by Ohbuchi et al. (1992) with an ellipsoid kernel around the input values (Fig. 8b). An exponential weighting is used according to the distance from the estimated voxel to weight the pixel value contribution on the nearby voxels. A ECG signal allows Meairs et al. (2000) to create a 4D presentation for imaging the movement during the cardiac cycle.

Function-Based Methods (FBM)

FBMs choose a particular function (like a polynomial) and determine coefficients to make one or more functions pass through the input pixels. Afterwards, the function(s) are used to create a regular voxel array by evaluating the function(s) at regular intervals (Fig. 9).

The Radial Basis Function (RBF) interpolation (Rohling et al. 1999) is an approximation with splines that tries to use the underlying shape of the data in the volume reconstruction. A smooth function is chosen because of the smooth intensity profile of the US beam. To allow for measurement errors, the interpolation function is changed into an approximation function. An RBF is used to create the spline approximation of the volume. Instead of creating one function from all the data points, the input data are divided into small segments. This implementation is done to reduce the very large number of computations otherwise required. Overlapping windows enclose the segments to get smooth connections among the neighboring segments. Rohling et al. (1999) also propose an intelligent window-growing method that expands the window sufficiently in all directions to ensure that the RBFs for each segment closely match the RBFs of neighboring segments. All data points in the window are used to calculate the RBF for that segment. The result of the computations is a localized trivariate spline.

The Rayleigh reconstruction/interpolation with a Bayesian framework (Sanches and Marques 2000) estimates a function for the tissue by statistical methods. This is done by Bayesian estimation methods, where a Rayleigh distribution is used to describe the US data. The computational speed of the algorithm is improved by

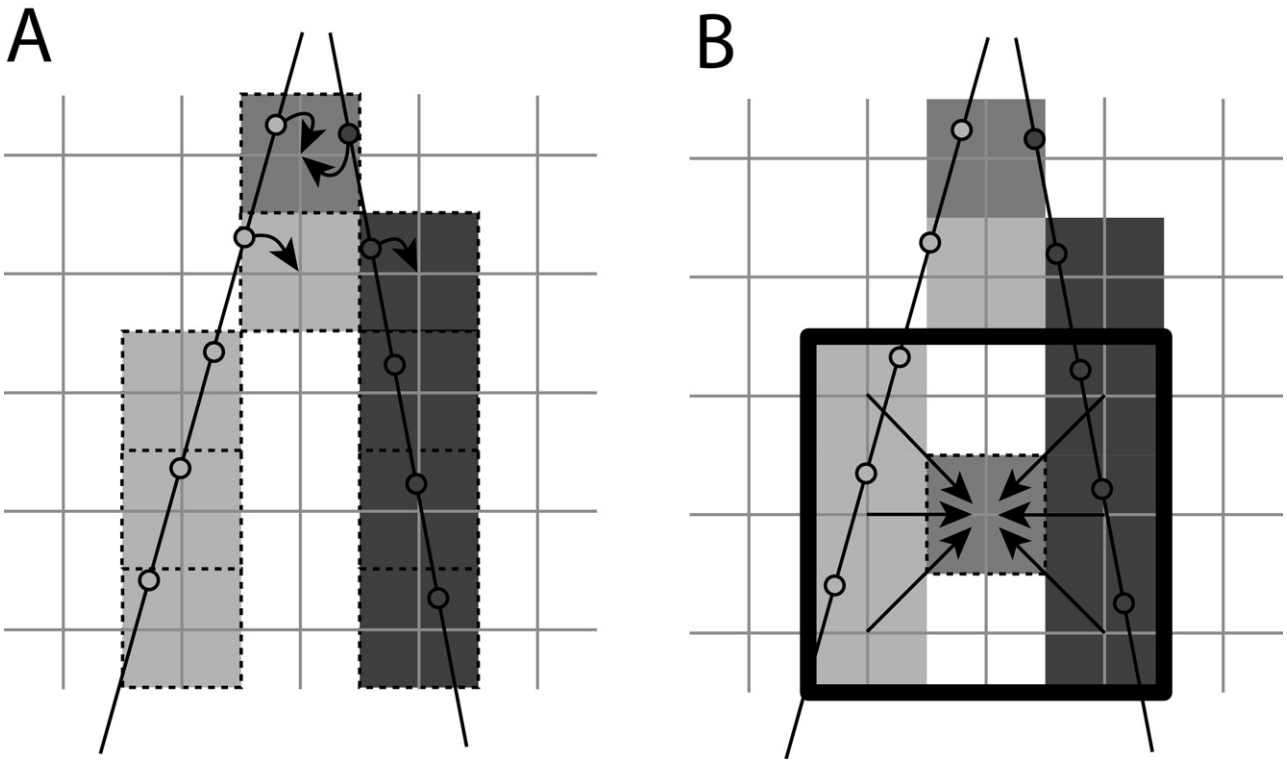

Fig. 5. (a) PBM DS: PNN bin-filling stage, each voxel is assigned the value of the intersecting pixel(s). This is done by traversing each pixel and assigning the pixel value to the voxel occupying the pixel position. Multiple contributions to the same voxel are handled either by using an average of all values or by storing the maximum or the latest value. (b) PBM HFS: PNN hole-filling from a neighborhood around the hole, each empty voxel is assigned a value based on the nearest already filled voxels. This value may either be an average or median, or an interpolation between the two closest nonzero voxels. (In both illustrations, the input images are illustrated by lines and the points in the lines illustrate the centers of the pixels. The 2D grids mark the centers of the voxles in the 3D voxel grids.)

running the first iterations on low resolution versions of the voxel volume (Sanches and Marques 2002).

COMPARISON AND DISCUSSION

In this article, the algorithms are grouped based on how they are implemented. This is an alternative to the grouping used in the literature (Rohling et al. 1999) where algorithms are sorted into groups according to how they work: Voxel Nearest Neighbor (VNN) interpolation, Pixel Nearest Neighbor (PNN) interpolation and Distance Weighted (DW) interpolation. The group name Distance Weighted (DW) interpolation (Rohling et al. 1999) is sometimes confused with the method inverse Distance Weighted interpolation (Barry et al. 1997), which is also often referred to as DW. Some articles are not clear on which algorithms they refer to, or if they refer to a group or an algorithm. To avoid misunderstandings in the literature, the actual algorithm names with references should be used instead of group names, since the different algorithms inside a group may have different properties.

When choosing and implementing the algorithms for clinical use, it is important to take certain aspects into account, especially due to the practical application in which the algorithm is to be used. In the following, the various algorithms are compared and discussed according to: Analog versus digital US image import, VBM versus PBM, deciding Region-of-Interest (ROI), algorithm computation time, reconstruction quality, real-time versus high-quality implementations, postprocessing supporting algorithms and practical considerations. Some essential data (computation time, size of input data and hardware used) disclosed from published material are described in Table 2. Table 3 contains data obtained from our own laboratory experiments where we have tested some reconstruction algorithms due to performance with current computers. The most important parameters, as speed and quality are discussed related to the usage in applications for image-guided surgery.

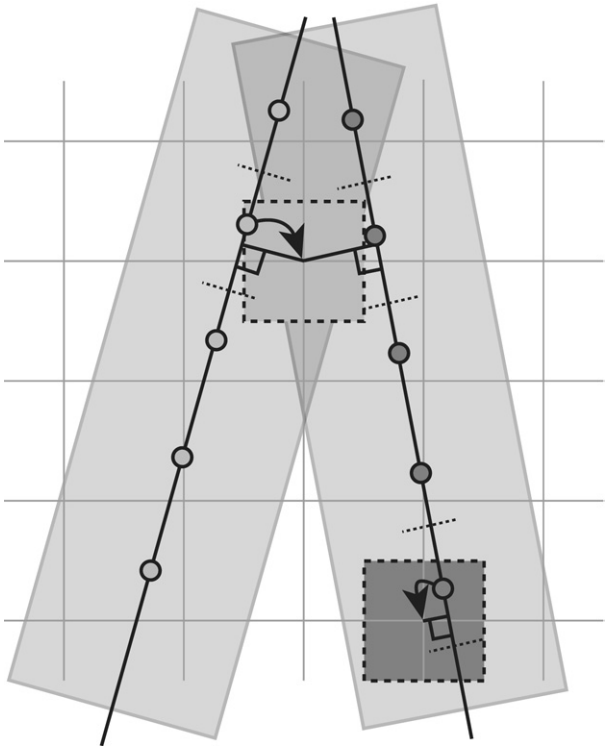

Fig. 6. A PBM implementation of VNN with image thickness. Each 2D image is inserted into a 3D voxel grid with certain thickness determining how large a part of the volume should be updated when inserting the 2D images. The gray area is updated with the pixel values of the nearest image while the white area is left empty. If several image areas intersect a voxel, the pixel value from the nearest image is used. (The input images are illustrated by lines, and the points in the lines illustrate the centers of the pixels. The 2D grid marks the centers of the voxles in the 3D voxel grid.)

Analog versus Digital US Image Import

Most reconstruction algorithms are based on analog video grabbed data. The frame rate of the video stream and video grabber is usually fixed while the frame rate of the US scanner depends on various parameters like application, probe, frequency, visualization depth and number of focus points. This difference in frame rate results in additional inaccuracies between positions and images, and frames are lost if the frame rate of the US scanner is higher than that of the video stream and video grabber. If the frame rate on the scanner is lower than that of the video stream, then the input video stream will contain duplicates of certain frames. In addition, the resolutions of the video grabbed data are usually higher than necessary, resulting in more input data to process and thus more processing time. Meairs et al. (2000) have used coded information in their analog video images for identifying duplicates that may be removed. An even more accurate approach is the one used by Thune et al. (1996) and Berg et al. (1999), where the original data from the US scanner is used. The advantage of this approach is obvious: it allows any frame rate to be used on the US scanner and use of the correct image resolution. Digital data in the form of reconstructed 2D images or 1D scan lines might lead to higher image quality. 1D scan lines will in addition lead to reduction of input data, possibly resulting in reduced processing time. Use of 1D scan lines is interesting for use in image-guided surgery, since both image quality and fast processing time are important parameters. However, most ultrasound scanners use several layers of image processing on the finished 2D images. Some of these processing methods are unpublished algorithms and will, therefore, be lost when working directly with 1D scan lines instead of working on processed digital or analog 2D images. Another disadvantage is that, in contrast to other algorithms based on video grabbed input data, these solutions require access to the digital data on the US scanner. This is the strategy of the SonoWand system (Gronningsaeter et al. 2000).

Voxel-Based Methods (VBM) Versus Pixel-Based Methods (PBM)

As seen in Table 1, most algorithms are either voxel-based or pixel-based. A PBM could be constructed as an iterative method, building the volume along with the images being collected and, thus, be made into a real-time reconstruction (given a fast enough implementation). One possible drawback is that this method will have more problems determining holes in the volume and expanding the neighborhood to fill these. The voxel-based approach can easily determine if there are holes to be filled since these methods traverse through all the voxels. The disadvantage of doing this is that the algorithm must wait for all the data to be collected before the reconstruction may start, thus, making a real-time reconstruction harder. This could be partly overcome by limiting the user to perform the sweep in a specified manner, then after a specified time of delay start the reconstruction of the area recently scanned. If the user sweeps over the reconstructed area again, some calculations could be redone. Still, there will be more delay than with a PBM. A disadvantage of filling all the voxels, like VBMs often do, is that very sparsely sampled areas are shown to contain data, while the nearest real data values might lie relatively far away.

Deciding Region-of-Interest (ROI)

Most reconstruction algorithms create the volume in a regular voxel grid. The properties of this grid must somehow be defined, like: orientation, size of the grid and size of the voxels. Sometimes the user has to define a key US frame that will help position the regular grid (Barry et al. 1997; Rohling et al. 1999). This key frame is typically centrally located and should contain a complete cross-section of the object to be visualized. The

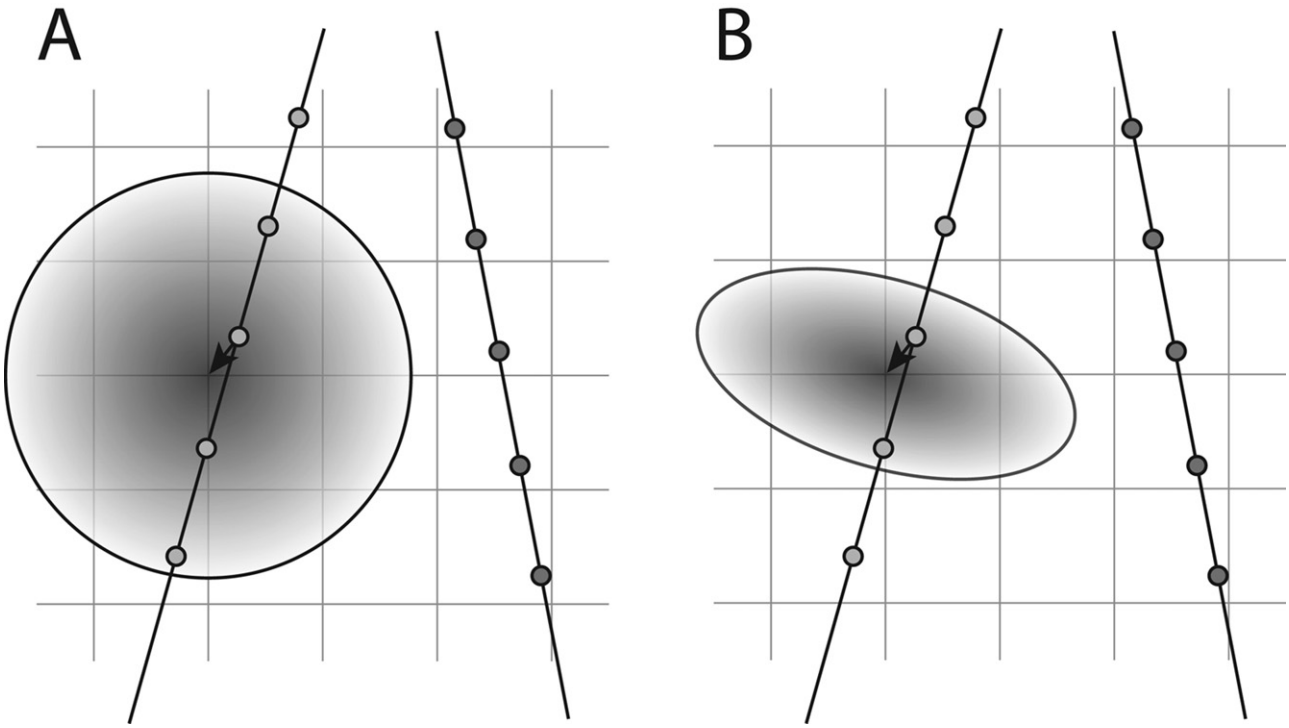

Fig. 7. (a) PBM HFS: PNN hole-filling with a spherical kernel around a voxel. First the pixel values are inserted into the nearest voxel. Then the voxel values are added to the neighboring voxels with a spherical kernel function with a Gaussian weighting. The weighting determines how much the voxels should affect nearby voxels. The variance of the kernel is dependent on the voxel-variance calculated in the bin-filling step, meaning the extent of the kernel should vary according to the uncertainty of the data. (b) PBM HFS: PNN hole-filling with an ellipsoid kernel around a voxel. First the pixel values are inserted into the nearest voxels. Then a normalized convolution with an ellipsoid kernel is performed on each voxel. The PSF of the US system have been used to define the kernel shape and weighting. (In both illustrations, the input images are illustrated by lines and the points in the lines illustrate the centers of the pixels. The 2D grids mark the centers of the voxles in the 3D voxel grids.)

axes of the grid are set equal to the axes of the key frame and the grid origin is typically set to the key frame center. Voxel size may be determined by the theoretical US resolution in focus. Another approach is to use an already defined voxel grid, like the grid of an already acquired MRI volume (Gobbi et al. 2001). Instead of creating a voxel grid large enough to contain all data from all imported images, the grid could be restricted to only contain a small ROI. San José-Estépar et al. (2003b) have proposed a method for greatly reducing the ROI, giving the reconstruction algorithm a much smaller voxel grid to reconstruct. This approach allows a computational increase since it may reduce the amount of input data considerably. A disadvantage is that the ROI estimation has to wait for the data collection to be completed. Also the reconstruction has to wait until the ROI estimation step is complete. This makes this particular technique inappropriate for real-time reconstruction, unless the ROI estimation has been run in an initialization procedure in advance. However, for computationally demanding algorithms, the waiting time is short compared with the potentially large saving in processing time. For real-time reconstructions some problems may arise determining the properties of the voxel grid. Gobbi et al. (2001) uses a pre set voxel grid decided by an already acquired MRI volume. If no previous voxel grid exists, it might be possible to have an algorithm guess the extent, by knowledge of the starting position, movement direction and type of probe movement (translation, tilt, rotation). The direction and the type of movement could be decided in advance. A dynamic expansion of the voxel grid may also be possible, but may also be too slow for a real-time implementation.

Algorithm computation time

Computation time is critical in practice when a result is wanted in or close to real-time, as is in the operation theatre. Several articles contain data about speed and quality of reconstruction algorithms. Data for comparisons of some of these can be found in Table 2.

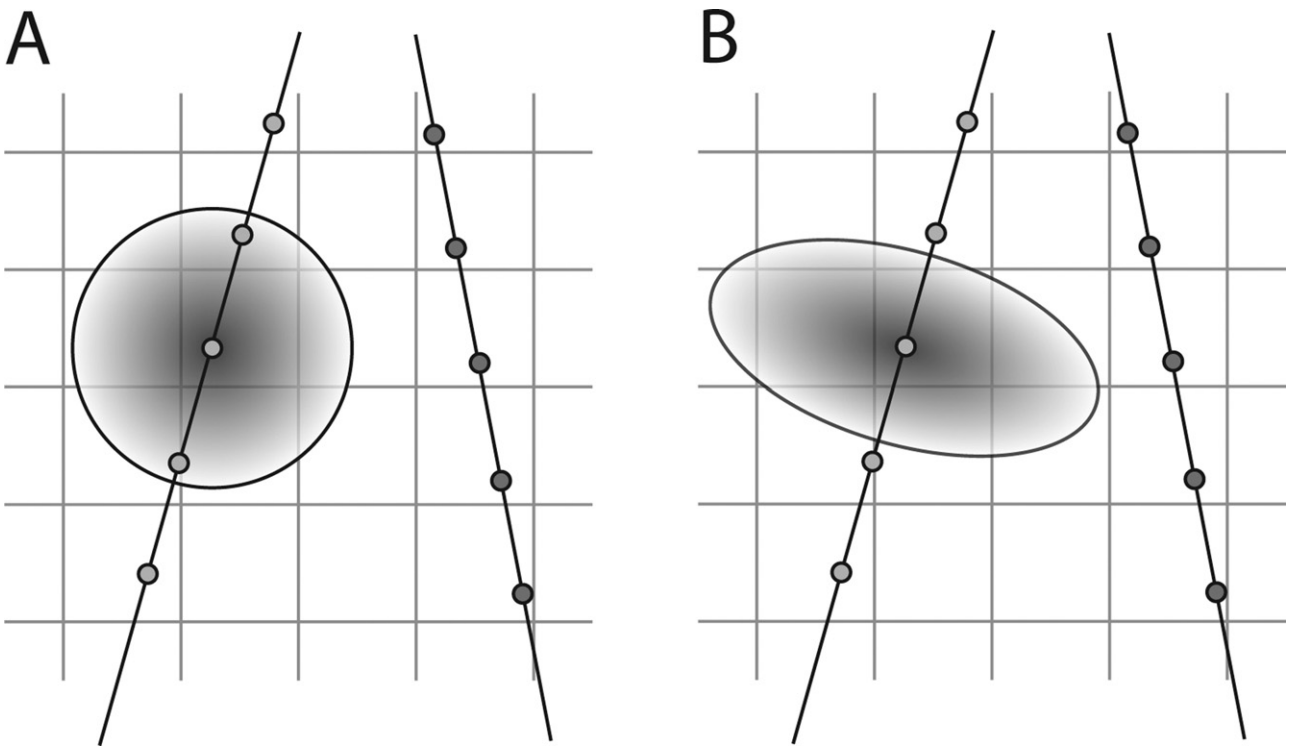

Fig. 8. (a) PBM DS with a 3D spherical kernel around each pixel. A contribution from each pixel is added to a set of nearby voxels, determined by a spherical kernel around the pixel. The contribution is weighted with the inverse of the distance. (b) PBM DS with a 3D ellipsoid Gaussian kernel around a pixel. Each pixel is added to a set of nearby voxels, with extent and weighting determined by an ellipsoid Gaussian kernel. (In both illustrations: The input images are illustrated by lines, and the points in the lines illustrate the centers of the pixels. The 2D grids mark the centers of the voxles in the 3D voxel grids.)

Only algorithms with accessible values have been included and some of the table values are estimated based on the available material and descriptions in the referred articles. The year of the article and the hardware used for the algorithm implementation has been included and must be taken into account. Some of the algorithms might be optimized for speed, while others may not. These factors are based on both programming performance and choice of programming language, so the comparisons in the table are not absolutes. Some algorithms may have a performance that allow real-time or close to real-time implementation variants (Gobbi et al. 2001; Gobbi and Peters 2002), while others may use hours on the reconstructions. To indicate computation times relevant to current computers, results from own laboratory implementations of algorithms similar to some of these reconstruction algorithms are included in Table 3. Simple algorithms like VNN and PNN are relatively fast, while the algorithms using input from a neighborhood (like a 3D kernel) around each voxel use more time. One way of decreasing the computation time is to reduce the size of this kernel. Berg et al. (1999) have taken this a step further by interpolating only along 1D in the out-of-plane direction. This may be a reasonable simplification, since the data in the in-plane directions is already reconstructed in the 2D US images. The advantage is a large decrease in computation time, at the cost of a more inaccurate reconstruction compared with using 3D interpolation. Gobbi and Peters (2002) achieves a real-time implementation by using a small kernel ($2 \times 2 \times 2$) and also manage to reduce computation time further by using alpha blending instead of compounding, with the side effect of a much higher weight on the last pixel contributing to each voxel. If the algorithm or input data are inaccurate, a suitable method of reducing computation time is to simply increase the voxel size in the output volume. This will also reduce the need of hole filling (compare Fig. 10 f to i with Fig. 10 r to u).

Reconstruction Quality

Ultrasound quality is usually defined by probe resolution, an aspect that also should be used when comparing 3D US. The most commonly used 3D reconstruction quality or accuracy estimator is to remove various amounts of input data and test the algorithms ability to recreate the removed data. A form of Mean Square Error

Table 2. Algorithm speed comparison, based on published data

Algorithm	Reconstruction computation time	Specified input image size	Hardware used	References
Voxel-Based Methods (VBM)				
Stradx	0.1 sec per any-plane slice		Regular PC (in 2002)	Prager et al. (1999; 2002)
Two surrounding slices interpolation	54 sec per image*	256 × 256	SUN SPARC-station 1	(Trobaugh et al. 1994)
Probe Trajectory (PT)	0.8–2.6 sec per image*	510 × 441	3.2 GHz Pentium 4	Coupé et al. (2005)
Pixel-Based Methods (PBM)				
Real-time PNN	0.033 sec per image	Cropped from 320 × 240	2 CPU 933 MHZ Pentium workstation	Gobbi and Peters (2002)
1D interpolation	0.06 sec per image†	Up to 512 samples with up to 128 beams	300 MHz Pentium PC	Berg et al. (1999)
Real-time PTL (Kernel: 2 × 2 × 2)	0.05 sec per image (alpha blending) 0.08 sec per image (compounding)	Cropped from 320 × 240	2 CPU 933 MHZ Pentium workstation	Gobbi and Peters (2002)
Spherical kernel, linear weighting (Kernel varying between 3 × 3 × 3 and 4 × 4 × 4)	0.6 sec per image*	Half the size of S-VHS PAL (S-VHS PAL = max 400 analog lines)	100 MHz silicon graphics Indy R4600	Barry et al. (1997)
Ellipsoid kernel, off-plane: Gaussian interpolation, in-plane: linear interpolation	2.6 sec per image*	512 × 480 (Reconstructed into a 128 × 128 × 128 volume)	HP9000/700 workstation	Ohbuchi et al. (1992)
Ellipsoid kernel, Gaussian weighting	1.5 sec per image*	128 × 128 (Reconstructed into a 128 × 128 × 128 volume)	IBM RS6000 model 560 workstation	Ohbuchi et al. (1992)
Function-Based Methods (FBM)				
RBF	Several hours for 219 images	328 × 409 and 480 × 413 masked	Silicon Graphics Indy workstation	Rohling et al. (1999)

* Estimate based on article

† Estimated value, based on article. Real rate is probably lower due to image combinations prior to the reconstruction. The resolution with this frame rate is a bit unclear.

(MSE) is used for describing the accuracy (Rohling et al. 1999; Sanches and Marques 2000; San José-Estépar et al. 2003a, 2003b; Coupé et al. 2005). Barry et al. (1997) compare number of pixels identified by a segmentation process. Sanches and Marques (2000) calculate signal to noise ratio (only on synthetic data) and log-likelihood in addition to MSE.

The basic VNN algorithms take only the nearest pixel into account regardless of distance. This may display volume areas as sharp, even if they contain no real values (Rohling et al. 1999). The Stradx system (Prager et al. 1999) tries to increase the accuracy of the VNN algorithm by skipping one step of interpolation by reconstructing 2D slices through the volume directly, without using a voxel grid. The disadvantage is that only 2D slices may be visualized since no 3D volume exists. The Solus system (Carr et al. 2000) avoids the generation of values in areas far away from the real data by allowing pixels to update voxels at only a certain distance away. This leaves holes in the volume as an indication of a too sparsely sampled volume. VNN algorithms are shown as the most inaccurate method in most comparisons (Rohling et al. 1999; San José-Estépar et al. 2003a, 2003b; Coupé et al. 2005). PNN algorithms are often more accurate by assigning the pixel values to the nearest voxels. The holes may be filled by combining values from a local neighborhood (as demonstrated by laboratory experiments: compare Fig. 10 f to i with Fig. 10 j

Table 3. Own laboratory results of algorithm computation times of some Pixel-Based Methods (PBM)

Algorithm implementation	Reconstruction computation time	Related algorithms
PNN with only bin-filling	0.04 sec per image	Trobaugh et al. (1994), Gobbi and Peters (2002)
PNN with bin-filling and 3 × 3 × 3 hole-filling (averaging)	0.067 sec per image	McCann et al. (1988), Hottier and Billon (1990), Nelson and Pretorius (1997)
3D kernel around voxels. Spherical kernel, Gaussian weighting	0.15 sec per image with 3 × 3 × 3 kernel	San Jos-Estépar et al. (2003b)
3D kernel around voxels. Ellipsoid kernel, Gaussian weighting	0.66 sec per image with 3 × 5 × 9 kernel	San José-Estépar et al. (2003a), San José-Estépar et al. (2003b)
3D kernel around pixels. Spherical kernel, Gaussian weighting	1 sec per image with 3 × 3 × 3 kernel	Ohbuchi et al. (1992), Barry et al. (1997), Meairs et al. (2000), Gobbi and Peters (2002)
3D kernel around pixels. Ellipsoid kernel, Gaussian weighting	2.5 sec per image with 3 × 5 × 9 kernel	Ohbuchi et al. (1992), Meairs et al. (2000)

All reconstructions are implemented in C++ and run on an AMD Athlon 64 Processor 3000+, 1.81 GHz. Input consist of 413 video grabbed US images with a resolution of 720x537, stored on the computer hard drive. Only a masked sector is used of each image; the sector size is 46% of the total image size. The volumes are reconstructed into a 288x247x253 voxel grid with a voxel size of 1.0 mm. Related algorithms that can be expected to have a similar computation time are listed.

to m) or by applying 3D kernels to the already filled voxels. This allows several pixels to contribute to the values of each voxel, which is an improvement over the VNN algorithms. The disadvantage of the PNN algorithms is that actual pixel positions are not used directly, but first resampled by inserting the pixels into the nearest voxels in a regular voxel grid. This error can be made small by using voxels that are smaller than the resolution of the US probe resolution, but this will result in an increased computation time and more holes that need to be filled. VBMI and PBMs with a 3D kernel around pixels (for an example see Fig. 10 n to q) use the pixel positions directly for determining the impact each pixel will have on the nearby voxels. This may improve the accuracy, but may also result in slower computation time (Table 3). The input pixels impacts on the voxels are calculated with different weighting functions based in one way or another on the pixels distance from the voxel. Functional algorithms (Rohling et al. 1999; Sanches and Marques 2000, 2002) may provide even more accurate volumes, but at the cost of highly increased computation time.

Future 3D US quality comparisons should also include a comparison analogous to the probe resolution, a measure of the quality of regular 2D US probe. A resolution-phantom could be scanned and reconstructed with 3D reconstruction algorithms. The algorithms ability to create volumes that distinguish between close point-spread sources is a good measure of resolution. The difference in the algorithms will probably be highly visible when the point-spread sources are lying between image planes. Algorithms that create compounded volumes (like VBMI, PBM with kernels and FBM) will probably show increased resolution, especially if the volumes consist of 2D images from different directions. However this compounding may reduce in-plane resolution in the final volume. Huang et al. (2005) propose a method that will preserve the resolution in all directions. They divide the US into different volumes for different directions and merge the volumes in the Fourier domain

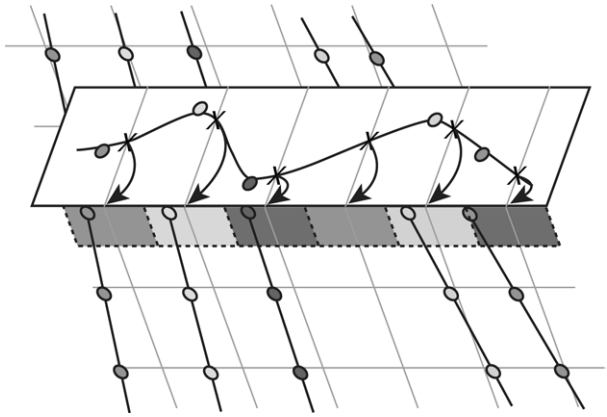

Fig. 9. Functional interpolation, visualized along one dimension. A function through the input points (pixel values) is estimated. To obtain the final voxel values the function is evaluated at regular intervals. (The input images are illustrated by lines, and the points in the lines illustrate the centers of the pixels. The 2D grid marks the centers of the voxles in the 3D voxel grid. The vertical lines represent the regular grid, in which the result is to be placed. The curve illustrates the function made on basis of the input data. The Xs on the curve show the resulting data values gained by evaluating the constructed function at regular intervals corresponding to the target grid.)

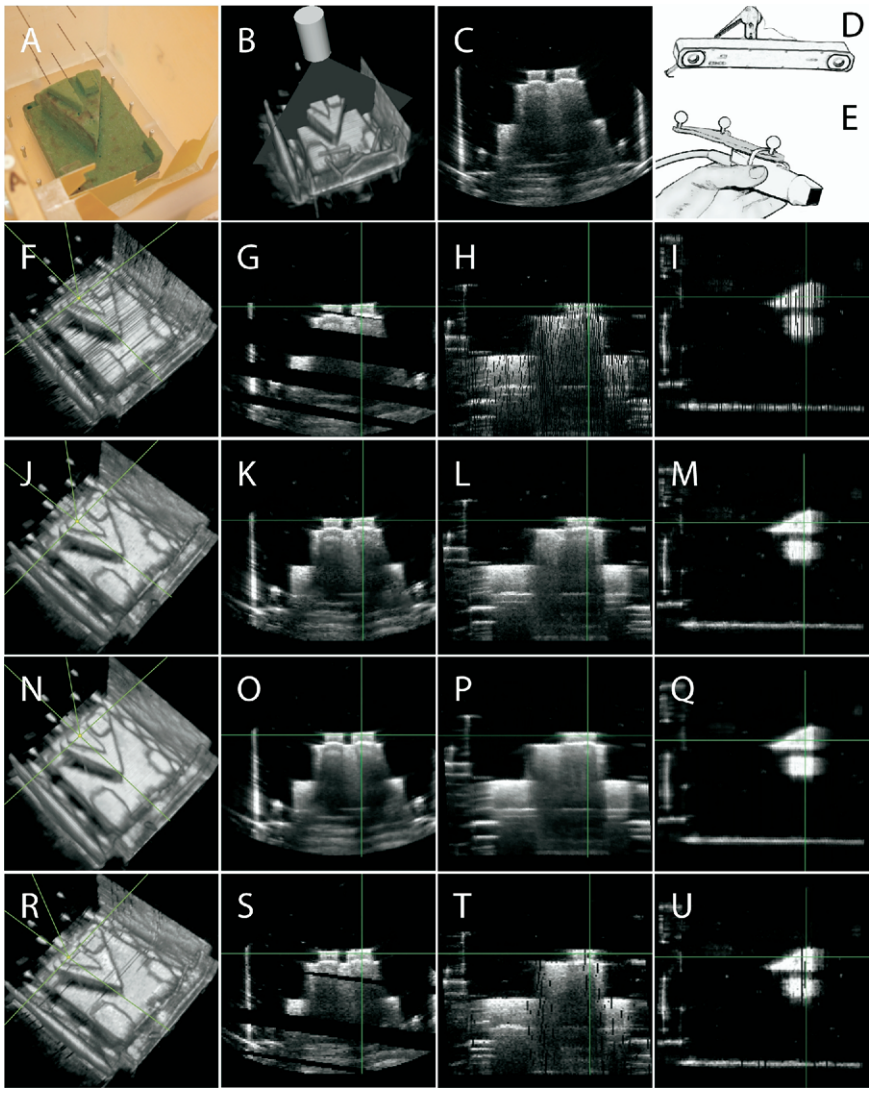

Fig. 10. Laboratory results showing the quality of 3D reconstructed volumes using various reconstruction algorithms and by scanning a US laboratory-phantom. The phantom is made of oasis and was submerged in water and scanned with a freehand translation with a 2D US probe over the area of interest. The scanner used was SystemFiVe GE (Vingmed Ultrasound, Horten, Norway) with a 5 MHz Phased Array (FPA) US probe. Frequency was set to 5.7 MHz, focus positions: 4.9, 6.7 and 11.0 (cm), frames per s: 10.6. The data were imported with a video grabber with a rate of 25 images per s running on a computer. The reconstructed US volumes and slices through the volumes were visualized with own visualization software (SINTEF, Trondheim, Norway). **(a)** Photograph of laboratory-phantom used in the tests. **(b)** Illustration of the US probe position and US 2D scan plane position in relation to the phantom. **(c)** A input 2D image scan (source data from US scanner). **(d)** Illustration of the Polaris optical position tracking used (NDI, Waterloo, Ontario, Canada) **(e)** Illustration of the US probe with an attached frame with reflecting spheres. **(f)** Volume reconstructed with only a Distribution Step (DS) using the PNN bin-filling and a voxel size of 0.6 mm, resulting in several holes in the volume. The 0.6 mm voxel size corresponds to the theoretical azimuth US resolution in the closest focus point. Cross hairs show orthogonal slicing through the volume displayed in (g) to (i). **(g)** to **(i)** Three orthogonal image slices through the volume in (f). The holes in the volume are clearly visible. **(j)** Volume reconstructed as in (f) (PNN bin-filling in the DS), but with an additional Hole Filling Step using an average over a $3 \times 3 \times 3$ neighborhood to fill the empty voxels. Voxel size is 0.6 mm as in (f). An increase in time consumption can be seen compared with a reconstruction without hole-filling (f) to (i) (see Table 3). **(k)-(m)** Orthogonal image slices through the volume in (j). The holes in (f) are now filled. By comparing (k) with (g) it is possible to see that the voxels filled in the HFS are more blurred than the voxels from the DS [also observed by Rohling et al. (1999)]. **(n)** Volume reconstructed with an ellipsoid kernel ($3 \times 3 \times 7$) around each pixel with Gaussian weighting and voxel size 0.6 mm. **(o)** to **(q)** Orthogonal image slices through the volume in (n). Comparing (n) to (q) with the other images it is possible to see that both the volume and the slices are smoother than the others, because the algorithm allows the input samples to influence nearby samples. **(r)** Volume reconstructed with only a DS using the PNN bin-filling, as in (f), but with a larger voxel size of 1.0 mm,

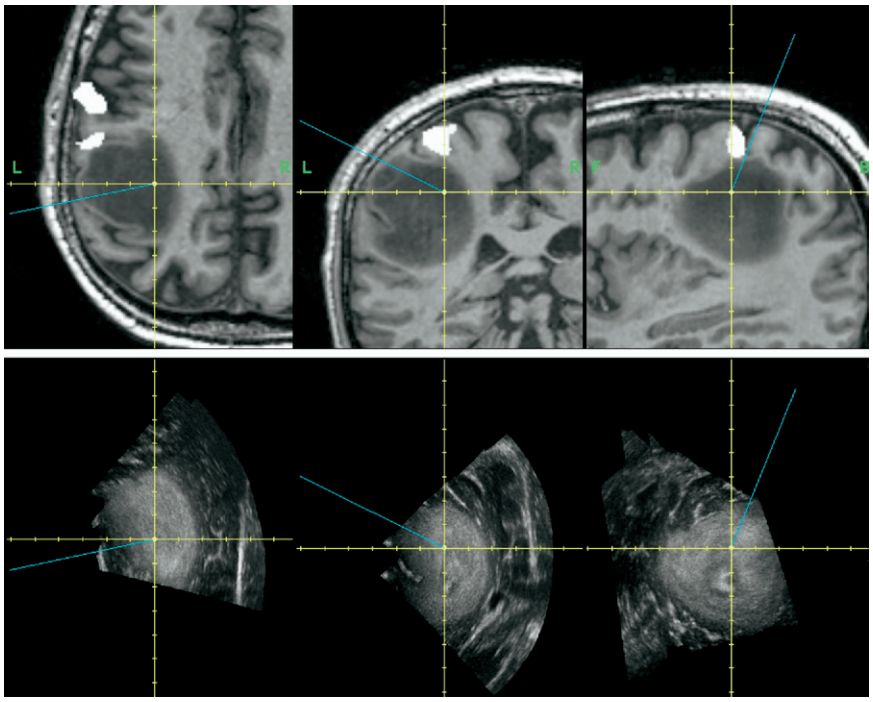

Fig. 11. 3D Ultrasound acquired with the commercial system SonoWand (MISON AS, Trondheim, Norway) (Gronningsaeter et al. 2000). (Top) Orthogonal 2D slices through a MR volume of a patient with brain tumor. The slices are selected due to the position of the surgical instrument. (Bottom) Orthogonal slices through an ultrasound volume corresponding to the MR slices. None of the slices correspond exactly to the US scan plane. The US volume is recorded in about 20 to 30 s and another 30 s were used for reconstruction. The image quality of ultrasound may in many cases be comparable to that of MRI.

by weighting the frequency components according to the signal noise ratio.

Other highly important factors for reconstruction quality (especially for compounded volumes) are position sensor accuracy, probe calibration (Mercier et al. 2005) and tissue movement (Treece et al. 2002). Probe calibration defines the transformation between the 2D ultrasound images and the position sensor attached to the probe and inaccuracies in this probe calibration may have a significant impact on overall accuracy (Lindseth et al. 2002).

For positioning systems, spatial accuracy in the reconstructed volume, could be used as a measure of quality in addition to sensor accuracy measurements. Measurements could be done by performing distance or volume measurements (Barry et al. 1997) of objects with known distances or volumes, and the spatial accuracy would illustrate inaccuracies and drift. This could also define accuracy of sensorless reconstruction like predefined operator probe movement (Downey and Fenster 1995) and speckle decorrelation (Tuthill et al. 1998), which probably would give much lower spatial accuracy than using positioning systems. Drift and noise on magnetic positioning systems would most likely result in poorer spatial accuracy than for optical positioning systems.

Even if compounded volumes may be more accurate and visually pleasing, tissue movement and position inaccuracy may reduce the volume quality greatly, resulting in blurred volumes. In these cases, a simpler algorithm will give better results, so in many cases the more advanced algorithms may not result in better overall volume quality. To achieve highest possible quality in the volumes, the algorithm type must be selected based on position accuracy and the data acquisition situation.

opposed to the images (f) to (q), using 0.6 mm voxel size. The larger voxel size allows this faster algorithm to fill most of the volume leaving only a few holes. This reconstruction algorithm may be implemented in real-time [see Table 3 and Gobbi and Peters (2002)]. **(s)** to **(u)** Orthogonal image slices through the volume in (r) clearly demonstrate that most of the holes are filled with the larger voxel size.

Real-time versus high-quality implementations

Although some of the advanced algorithms may not yet be implemented in real-time with current hardware, some less computationally intensive algorithms have already been implemented and others may be possible to implement. One possibility is to provide the advantages of both real-time and high-quality implementations by using several algorithms. Applications may provide a real-time view straight away and at the same time calculate a higher quality volume in the background, to be shown when ready. Gobbi et al. (2001) uses this method to some degree by first presenting a real-time reconstruction consisting of every third input image and presenting the final result afterwards. This method can be taken further by combining more methods and using even more computationally demanding algorithms. A possible combination for image-guided surgery could consist of three algorithms: a real-time presentation of a PNN volume without hole-filling during data acquisition. After the data acquisition is finished a PNN volume with hole-filling is presented. After a few minutes, a more accurate volume with a 3D kernel around input pixels is presented If the high quality volumes are not needed right away, this may be a possible solution. The lower quality volumes will probably be sufficient for determining the quality of the final volume in advance. Such applications could also be able to select or tune algorithms based on the available hardware.

In the case of sparsely located input 2D frames, PBMs might be a fast alternative but will leave a lot of holes in the volume. The hole-filling part could consume some time if the whole volume is to be filled. In this case the VBMs might be a better choice providing a fast reconstruction. VNN implementations will provide a fast reconstruction, but the more accurate VBMI may also be reasonably fast for image-guided surgery. VBMI range from fast 1D interpolation to an interpolation from the nearest frames (Trobaugh et al. 1994; Coupé et al. 2005) or scan line points (Thune et al. 1996). The algorithm by Coupé et al. (2005) may offer an improvement by calculating the probe trajectory. In ultrasound, different image depths are acquired at different times, dependent on number of focus points. If these acquisition parameters are known this could be used to improve the algorithm further by taking data acquisition times into account in addition to probe trajectory. This could result in an even better accuracy, also for more densely sampled data.

Algorithms supporting postprocessing

Some algorithms support certain postprocessing steps better than others. The construction of a regular voxel grid allows the use of multiple volume processing algorithms after the reconstruction. Segmentation is often performed based on 3D volumes. It is much easier to segment from volumes that are smoothened and have an increased signal to noise ratio as for example function based methods, VBMIs or PBMs with 3D kernels. Most algorithms taking several pixels into account for each voxel have the effect of smoothing the volume: the higher number of pixels used, the smoother the resulting volume. The functional algorithms create very smooth volumes by taking large volume areas into account for the local functions. The effect of using several pixels for deciding a voxel value is that this produces compounded volumes, which are more pleasing to the eye, have reduced speckle, improved spatial coherence and improved signal to noise ratio. The disadvantage is that small structures may be lost. Barry et al. (1997) have shown that segmentations from heavily compounded volumes (using a spherical kernel) give much better results than segmentations from 2D US slices. Another approach is to smooth the volume after creation, for example like the filtering approach presented by Nelson and Pretorius (1997). This approach and the methods that use 3D kernels around voxels (instead of pixels) may produce the same kind of smooth volumes, but with a lower accuracy since pixel positions are not equal to voxel positions.

Practical Considerations and Future Work

To be used in the clinic, freehand US 3D reconstruction algorithms must be integrated with existing clinical equipment or otherwise made easily accessible. The most important parameters for practical implementation are speed and quality. Currently, the Solus system (Carr et al. 2000) offers a clinical interface to the Stradx system (Prager et al. 1999, 2002). The Stadx system (latest version called Stradwin) is currently being used for targeting of radiotherapy for breast cancer treatment through the radiotherapy planning system Orpheus from Qados, Berkshire, England. The Norwegian company, MISON, has the commercially available SonoWand system (Gronningsaeter et al. 2000) for use of freehand 3D US in neuronavigation. The reconstruction method in this system uses raw data from the US scanner and reconstructs the volume from the 1D scan lines (with an unpublished method) within 30 s (Unsgaard et al. 2002). Examples of 3D US volumes acquired with the SonoWand system (MISON) are shown in Fig. 11. Both due to reconstruction time and image quality promising results for use in image-guided surgery are presented. Corresponding planes of preoperative MRI shows that the image quality of ultrasound may improve toward that of MR. This may improve future diagnostics and may also result in improvements when the US volume is to be

used for guidance and control of surgical procedures (Kaspersen et al. 2003). Such image fusion between intraoperative 3D US and preoperative modalities may improve the quality of the surgical procedure (Lindseth et al. 2003).

CONCLUSION

With basis in the literature of 3D reconstruction algorithms, this article has described various algorithms and sorted them into three groups based on implementation method: Voxel-Based Methods (VBM), Pixel-Based Methods (PBM) and Function-Based Methods (FBM). Different practical applications will require different solutions, leading to the conclusion that future 3D ultrasound applications should probably consist of several reconstruction algorithms. These reconstruction algorithms should be either real-time or other kinds of fast reconstruction algorithms in collaboration with slower high-quality algorithms to ensure high quality image data. Even though high quality volumes are presented in image-guided surgery, speed is almost equally important. Therefore, algorithms capable of providing results in real-time or close to real-time, such as implementations of VNN, PNN, a small kernel around input pixels or VBMI with interpolation over a limited area for each voxel will be the most useful ones. Slower algorithms might be added as support to the fast ones. Faster computers as well as algorithms optimized for speed will result in faster versions of the higher quality algorithms optimizing the usage of 3D ultrasound reconstruction in image-guided surgery.

Acknowledgments—This work was supported by the Research Council of Norway, through the FIFOS Programme Project 152831/530; the Ministry of Health and Social Affairs of Norway, through the National Centre of 3D Ultrasound in Surgery; and by SINTEF Health Research. We also want to thank Geir Arne Tangen, Thomas Langø, Steinar Ommedal, Jon Bang, scientists at SINTEF and Veerle de Smedt (NTNU) for valuable contributions to the laboratory set-up, Atle Kleven (MISON) for information about the SonoWand system, and Nancy Eik-Nes for the text review of the initial version of the article.

REFERENCES

Barry CD, Allott CP, John NW, et al. Three-dimensional freehand ultrasound: Image reconstruction and volume analysis. Ultrasound Med Biol 1997;23:1209–1224.

Berg S, Torp H, Martens D, et al. Dynamic three-dimensional freehand echocardiography using raw digital ultrasound data. Ultrasound Med Biol 1999;25:745–753.

Carr JC, Stallkamp JL, Fynes MM, et al. Design of a clinical free-hand 3D ultrasound system. Proceedings of Medical Imaging 2000: Ultrasonic Imaging and Signal Processing, San Diego, CA, USA: SPIE 2000;3982:14–25.

Cinquin P, Bainville E, Barbe C, et al. Computer assisted medical interventions. IEEE Engineering in Medicine and Biology Magazine 1995;14:254–263.

Coupé P, Hellier P, Azzabou N, Barillot C. 3D freehand ultrasound reconstruction based on probe trajectory. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Palm Springs, CA, USA: Springer, 2005;3749:597–604.

Downey DB, Fenster A. Vascular imaging with a three-dimensional power Doppler system. Am J Roentgenol 1995;165:665–668.

Fenster A, Downey DB, Cardinal HN. Three-dimensional ultrasound imaging. Phys Med Biol 2001;46:R67–R99.

Gee A, Prager R, Berman L. Non-planar reslicing for freehand 3D ultrasound. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Cambridge, UK: Springer-Verlag, 1999, 1679:716–725.

Gobbi DG, Lee BKH, Peters TM. Correlation of preoperative MRI and intraoperative 3D ultrasound to measure brain tissue shift. Proceedings of Medical Imaging 2001: Visualization, Display, and Image-Guided Procedures, San Diego, CA, USA: SPIE, 2001;4319:264–271.

Gobbi DG, Peters TM. Interactive intra-operative 3D ultrasound reconstruction and visualization. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Tokyo, Japan: Springer, 2002;2489:156–163.

Gronningsaeter A, Kleven A, Ommedal S, et al. SonoWand, an ultrasound-based neuronavigation system. Neurosurgery 2000;47:1373–1379,discussion 1379–1380.

Hossack J, Sumanaweera TS, Napel S. Quantitative 3D ultrasound imaging using an automated image tracking technique. Proc IEEE Ultrason Symp 2000;2:1593–1596.

Hottier F, Billon AC. 3D echography: Status and perspective. In: Hohne KH, Fuchs H, Pizer SM, eds. 3D imaging in medicine. Berlin Heidelberg: Springer-Verlag 1990;F 60:21–41.

Huang W, Zheng Y, Molloy JA. 3D Ultrasound Image Reconstruction from Non-Uniform Resolution Freehand Slices. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA: IEEE, 2005;2:125–128.

Kaspersen JH, Sjølie E, Wesche J, et al. Three-dimensional ultrasound-based navigation combined with preoperative CT during abdominal interventions: A feasibility study. Cardiovasc Intervent Radiol 2003;26:347–356.

Knutsson H, Westin CF. Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA: IEEE, 1993;CVPR 93:515–523.

Lee JH, Kim SH, Kim NC. Reconstruction of 3D ultrasound image from freehand scanned frames using frame-by-frame isotropic correlation function. Proceedings of Three-Dimensional Image Capture and Applications IV, San Jose, CA, USA: SPIE, 2001;4298:195–202.

Lindseth F, Kaspersen JH, Ommedal S, et al. Multimodal image fusion in ultrasound-based neuronavigation: Improving overview and interpretation by integrating preoperative MRI with intraoperative 3D ultrasound. Comput Aided Surg 2003;8:49–69.

Lindseth F, Langø T, Bang J, Nagelhus Hernes TA. Accuracy evaluation of a 3D ultrasound-based neuronavigation system. Comput Aided Surg 2002;7:197–222.

Martens D, Gilja OH. The EchoPAC-3D software for image analysis. In: Ødegård S, Gilja OH, Gregersen H, eds. Basic and new aspects of gastrointestinal ultrasonography. Advanced series in biomechanics. Singapore: World Scientific Publishing, 2005:305–329.

McCann HA, Sharp JC, Kinter TM, et al. Multidimensional ultrasonic imaging for cardiology. Proceedings of IEEE 1988;76:1063–1073.

Meairs S, Beyer J, Hennerici M. Reconstruction and visualization of irregularly sampled three- and four-dimensional ultrasound data for cerebrovascular applications. Ultrasound Med Biol 2000;26:263–272.

Mercier L, Langø; T, Lindseth F, Collins LD. A review of calibration techniques for freehand 3-D ultrasound systems. Ultrasound Med Biol 2005;31:449–471.

Nelson TR, Pretorius DH. Interactive acquisition, analysis and visualization of sonographic volume data. Int J Imag Systems Technol 1997;8:26–37.

Nelson TR, Pretorius DH. Three-dimensional ultrasound imaging. Ultrasound Med Biol 1998;24:1243–1270.

Ohbuchi R, Chen D, Fuchs H. Incremental volume reconstruction and rendering for 3-D ultrasound imaging. Proceedings of Visualization

in Biomedical Computing '92, Chapel Hill, NC, USA: SPIE, 1992;1808:312–323.
Prager RW, Gee A, Berman L. Stradx: Real-time acquisition and visualization of freehand three-dimensional ultrasound. Med Image Anal 1999;3:129–140.
Prager RW, Gee A, Treece G, Berman L. Freehand 3D ultrasound without voxels: volume measurement and visualisation using the Stradx system. Ultrasonics 2002;40:109–115.
Prager RW, Gee AH, Treece GM, Cash CJC, Berman LH. Sensorless freehand 3-D ultrasound using regression of the echo intensity. Ultrasound Med Biol 2003;29:437–446.
Rohling R, Gee A, Berman L. A comparison of freehand three-dimensional ultrasound reconstruction techniques. Med Image Anal 1999; 3:339–359.
San José-Estépar R, Martín-Fernández M, Alberola-López C, et al. Freehand ultrasound reconstruction based on ROI prior modeling and normalized convolution. Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI), Montréal, Canada: Springer, 2003a;2879:382–390.
San José-Estépar R, Martín-Fernández M, Caballero-Martínez PP, Alberola-López C, Ruiz-Alzola J. A theoretical framework to three-dimensional ultrasound reconstruction from irregularly sampled data. Ultrasound Med Biol 2003b;29:255–269.
Sanches JM, Marques JS. A Rayleigh reconstruction/interpolation algorithm for 3D ultrasound. Pattern Recog Lett 2000;21:917–926.
Sanches JM, Marques JS. A multiscale algorithm for three-dimensional free-hand ultrasound. Ultrasound Med Biol 2002;28:1029–1040.
Sherebrin S, Fenster A, Rankin RN, Spence D. Freehand three-dimensional ultrasound: Implementation and applications. Proceedings of Medical Imaging 1996: Physics of Medical Imaging, Newport Beach, CA, USA: SPIE, 1996;2708:296–303.
Thune N, Gilja OH, Hausken T, Matre K. A practical method for estimating enclosed volumes using 3D ultrasound. Eur J Ultrasound 1996;3:83–92.
Treece GM, Prager RW, Gee AH, Berman L. Correction of probe pressure artifacts in freehand 3D ultrasound. Med Image Anal 2002;6:199–214.
Trobaugh JW, Trobaugh DJ, Richard WD. Three-dimensional imaging with stereotactic ultrasonography. Comput Med Imaging Graph 1994;18:315–323.
Tuthill TA, Krucker JF, Fowlkes JB, Carson PL. Automated three-dimensional US frame positioning computed from elevational speckle decorrelation. Radiology 1998;209:575–582.
Unsgaard G, Ommedal S, Muller T, Gronningsaeter A, Nagelhus Hernes TA. Neuronavigation by intraoperative three-dimensional ultrasound: Initial experience during brain tumor resection. Neurosurgery 2002;50:804–812,discussion 812.
Unsgaard G, Rygh OM, Selbekk T, et al. Intra-operative 3D ultrasound in neurosurgery. Acta Neurochir (Wien) 2006;148:235–253.

APPENDIX: ALGORITHM IMPLEMENTATIONS

The Appendix contains pseudo code implementations to some of the algorithms presented in the paper. Most papers do not supply detailed information about the algorithms, so the algorithms in this appendix are reconstructed based in the published information and may not be exactly similar to the actual algorithms used. Furthermore, the pseudo code implementations only show a possible implementation, and not all possible improvements. Especially when it comes to algorithm computation time, a lot of different improvements are possible.

VBM: VNN

Algorithm described by McCann et al. (1988), Sherebrin et al. (1996) and Rohling et al. (1999). See Fig. 1 for illustration.

VNN simple implementation

- For each voxel
  - Find nearest pixel:
    - Calculate a normal to each input image
    - The shortest normal gives the closest image
    - The pixel closest to the normal is the pixel nearest the voxel
  - Insert value of nearest pixel into the voxel

VNN faster implementation

Assumes that the ultrasound scan is a translation (or tilt) scan performed in only one direction and not back and forth.

- For each voxel
  - Find nearest pixel:
    - Calculate a normal only to the input image before and after the voxel
    - The shortest normal gives the closest image
    - The pixel closest to the normal is the pixel nearest the voxel
  - Insert value of nearest pixel into the voxel

VBM: VBMI

Two surrounding slices interpolation

Algorithm described by Trobaugh et al. (1994). See Fig. 3 for illustration.

- For each voxel
  - Find two nearest surrounding 2D slices
    - Calculate normal to each slice
    - Interpolate between the four pixels at the contact point of the normal
  - Interpolate between the results from the two slices based on the distances

Two surrounding slices with probe trajectory estimation

Algorithm described by Coupé et al.(2005). See Fig. 4 for illustration.

- For each voxel
  - Find two nearest surrounding 2D slices
  - Compute a "virtual" plane through the voxel by estimating a probe trajectory
  - For each of the two (or four) surrounding slices
    - Interpolate between the four pixels at the contact point of the trajectory
  - Interpolate between the results from the two (four) slices based on the distances

Voxel-based tri-linear interpolation

Algorithm described by Thune et al. (1996).

- For each voxel
  - Interpolate between the eight surrounding points in the input scan lines

PBM: PNN

PNN bin-filling stage with averaging

Algorithm described by Hottier and Billon (1990), Nelson and Pretorius (1997) and Rohling et al. (1999). See Fig. 5a for illustration.

- For each pixel
  - Find the voxel the pixel belongs to
  - If voxel have no value
    - voxelValue = pixelValue
    - pixelCounter = 1
  - Else if voxel have a value
    - Assign an average of all pixels:

$$\text{voxelValue} = \frac{\text{voxelValue} * \text{pixelCounter} + \text{pixelValue}}{\text{pixelCounter} + 1} \quad (1)$$

- pixelCounter = pixelCounter + 1

PNN hole-filling with averaging

Algorithm described by Rohling et al. (1999) and Gobbi and Peters (2002). See Fig. 5b. PNN hole-filling is performed after the bin-filling stage.

- For each voxel
  - value = 0
  - counter = 0
  - If voxel is empty
    - For each nearby voxel (a $3 \times 3 \times 3$, $5 \times 5 \times 5$ grid or larger)
    - If voxel is not empty
    - value = value + voxelValue
    - counter = counter + 1
  - voxelValue = value/counter

PBM: Apply a 3D kernel on input pixels with linear weighting

Algorithms described by Ohbuchi et al. (1992), Barry et al. (1997), Rohling et al. (1999), Meairs et al. (2000), Gobbi et al. (2001) and Gobbi and Peters (2002). See Fig. 8 for illustration.

- Set all voxelWeights and voxelValues = 0
- For each pixel
  - For each nearby voxel (area determined by a 3D kernel)
    - pixelWeight = Inverse distance between the voxel and the pixel

$$\text{voxelWeight} = \frac{\text{pixelValue} * \text{pixelWeight} + \text{voxelValue} * \text{voxelWeight}}{\text{pixelWeight} + \text{voxelWeight}} \tag{2}$$

    - voxelWeight = pixelWeight + voxelWeight

FBM: RADIAL BASIS FUNCTION INTERPOLATION

Algorithm described by Rohling et al. (1999).

- Divide the voxel array into small rectangular segments (not necessarily of the same size but with sufficient number of data points inside)
- Define a window around each segment
  - For each window
    - Grow the window in all directions separately, so that each grown direction holds a sufficient number of neighboring data points
    - If the segment containing too many data points
    - Divide it into smaller segments
    - Use all data points within the window to calculate the Radial Basis Function for the segment
    - Evaluate the function at regular intervals to gain the voxel data