Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Adaptive Volume Construction from Ultrasound Images of a Human Heart

Gerd Reis, Martin Bertram, Rolf H. van Lengen, and Hans Hagen

German Research Centre for Artificial Intelligence, Kaiserslautern, Germany

Abstract

We present a volume modelling approach based on sequences of two-dimensional ultrasound images. Though generally applicable to arbitrary freehand ultrasound, our method is designed for the reconstruction of time-varying volumes from ultrasound images of a human heart. Since the reliability of the reconstructed data depends very much on the spatial density of ultrasound images, we apply a hierarchical modelling approach. The volume produced for each time step is represented as adaptive mesh refinement (AMR) data such that regions of low reliability in the reconstructed volume can be recognized by their coarse resolution.

Keywords: Freehand Ultrasound, Echocardiography, Volume Reconstruction, Scattered Data Approximation, Adaptive Mesh Refinement.

1. Introduction

Ultrasound imaging is commonly used in medicine for diagnosis and preventive examinations. It provides an inexpensive, non-invasive way to study vascular and musculoskeletal anatomy in real time. Ultrasound is suitable for detecting boundaries of anatomical structures with different sonic impedance rather than classifying material properties.

Ultrasound waves emitted by a transducer are reflected at tissue boundaries and recorded by a probe. The reflected intensities of multiple waves are translated into greyscale values of certain pixels in a two-dimensional ultrasound image. Since the intensity associated with a certain boundary depends on the sonic impedance difference of intermediate tissues, it does not behave proportional to certain material properties. For example, if the transducer is not placed exactly on a patient's skin, the resulting image will be severely degraded by the high impedance of the air in between. Similar artefacts occur due to air in the lungs. The opposite occurs if the sound waves hit the ribs, where most of the sonic energy is reflected, causing extremely overexposed regions, to use a terminology from photography. Both aspects together extremely limit the paths of unaffected ultrasound propagation. These reasons motivate the usage of *freehand* ultrasound, especially in the field of cardiac ultrasound.

Reconstruction of volumes from sequences of arbitrarily aligned ultrasound images is a problem of recent interest[BAJ*, TGP*02, RN00]. High-resolution ultrasound images known as *B-scans* are recorded by a movable probe of a freehand ultrasound system. The spatial locations of these B-scans are recorded simultanously by an electromagnetic or optical tracking sensor, such that the individual images can be embedded into a volume.

The construction of an interpolating volume is a difficult task, since the individual ultrasound images are composed of Gaussian-like speckles rather than a continuous density function. The distance between adjacent B-scans is typically much greater than their resolution which makes it difficult to track features between these scans using, for example, scattered-data interpolation techniques[RGBT99].

In this paper, we contribute a highly efficient and accurate volume construction method from individual B-scans. This method is designed for realtime echocardiography (ultrasound of a human heart), where a sequence of images is slowly scanned within a fan-shaped region while the heart is beating. Out method does not require any triggering during the scanning process, neither by the electrocardiogram, nor by any external synchronisation impulse like the V-sync signal from a monitor. This way we are able to acquire extremely dense data with respect to time, i.e. we can use an

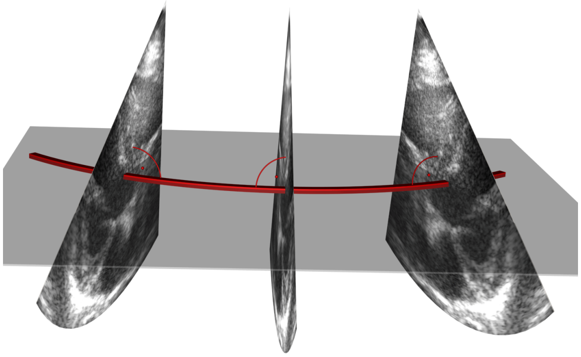

Figure 1: *Interpolation between two nearest scans using a circular arc to locate corresponding points on B-scan slices.*

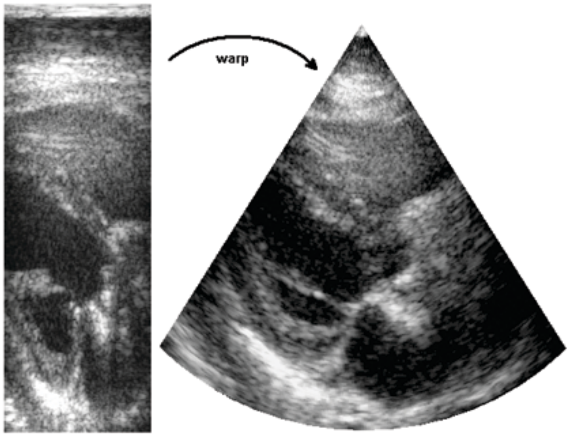

Figure 2: *Scan conversion (warp) of the original data during reconstruction.*

acquisition frequency ranging from 60Hz up to over 100Hz, depending on the actual image size and the probe frequency used. The position sampling is done in a de-synchronized manner for the same reason.

Because of the very short acquisition procedure, the patients are able to hold breath, so that respiration artefacts are almost completely avoided. This way the subsequent registration procedures can, though not being dismissed completely, simplified. This is extremely advantageous, since the otherwise necessary full registration is not able to decide between motions introduced by the patient and motions introduced by the heart itself. The latter would significantly degrade the data quality during the registration process, because the image content has to be altered in contrast to slightly altering the image positioning.

A second favourable aspect of our acquisition procedure is, that the patient and the physician as well undergo almost the same procedure as they are used to conventional 2D ultrasound examinations. The main differences are a second cable attached to the probe and the remote steering of the ultrasound device by means of a virtual foot switch. The acquisition itself does not depend on any external signals as they are known for R-wave triggered acquisition using a frame grabber [MBPE02].

In contrast to many other approaches based on scattered-data approximation using three-dimensional basis functions, our method interpolates for each voxel the data on nearest B-scans. We found that features between two B-scans are best propagated along circular arcs. To accommodate to the varying density of B-scans, we generate adaptive mesh refinement (AMR) data[BO84] based on an octree hierarchy. When sampling the volume at different resolutions, aliasing needs to be avoided. Therefore, we adapt the resolution of interpolated B-scan to the local volume resolution using a mip-mapping method.

The paper is structured as follows: In section 2, we summarize related work. Section 3 provides our volume construction including pre-processing of the raw data, interpolation between B-scans and adaptive mesh refinement based on spatial reliability. In section 4, we present numerical results for the reconstruction of three-dimensional ultrasound of a beating heart. Finally we conclude our work in section 5.

2. Related Work

A variety of approaches for three-dimensional ultrasound have recently emerged. A comprehensive overview of freehand ultrasound systems is provided by Treece et al.[TGP*02].

The interpolation of pixel values on B-Scans mapped into a spatial domain, can be solved by scattered-data interpolation methods[FN80]. Rohling et al.[RGBT99] use radial basis functions (RBF's) for volume reconstruction. RBF's are computationally expensive, despite of recent approaches accelerating their evaluation[BLB00].

Roxborough / Nielson[RN00] use a hierarchy of tetrahedral grids providing locally supported piecewise linear basis functions for scattered-data approximation. Despite of this flexibility, their approach requires the solution of a global least-squares problem slowing down their algorithm. Adaptive scattered-data approximation methods based on hierarchical B-Splines[BTH03] are more efficient and may also be used in ultrasound reconstruction.

However, all these methods have in common that they do not adhere to the structure of given data, which is highly precise on the B-scan slices with large gaps in between.

Therefore, our approach attempts to propagate data from the B-scans into the volume in the most intuitive manner without using prescribed trivariate basis functions. Like Roxborough / Nielson[RN00], we use an adaptive method for volume construction. To avoid a large overhead taken by the meshing structure, we employ AMR [BO84] using an octree with data values corresponding to each leaf voxel. Hierarchical data structures of this type are frequently used in visualization and contouring of large-scale volumes[WKE99, WKL*03].

3. Volume Construction from Ultrasound Images

In this section the new volume construction approach will be described. We start with a sequence of B-scans, each defined by a greyscale image and a transformation into world space. Since all ultrasound rays virtually emanate from a single point, the raw data greyscale images are warped on their corresponding slices in the volume to fit their real world geometry using a scan conversion process, see Figure 2. Greyscale values associated with points on these B-scan slices are computed by a simple parameter transformation and bilinear interpolation using the four closest pixels.

In the case of echocardiography, we have a fan of B-scans for each time interval in the heart cycle (there are between 25 and 50 intervals). Each fan consists of up to 100 B-scans which are selected based on their location in the ECG cycle. The local coordinate system of the ultrasound probe is aligned such that the x- and y-axes span the plane of the current B-scan, where the x-axis corresponds to the ultrasound (axial) direction and the y-axis points to the right (lateral) on the B-scan. During the scan acquisition process, the probe is mainly rotated around the y-axis resulting in a fan of B-scans, but minor rotations around the x- and z-axis can take place.

In the case of freehand ultrasound, the probe can be moved and rotated in arbitrary directions as long as it does not leave the patient's skin. Due to the fact that the probe is held by a human the probe positions alter slightly during a fan acquisition. In addition the angular velocity changes significantly. In consequence there is need for a method which synchronizes the ECG, the probe tracking information and the image data and which is able to reorder the slices within time domain. The result is a sequence of unique B-scans from which a dynamic volume can be constructed by interpolation.

3.1. Pre-processing the raw data

As mentioned, the individual B-scans are not ordered in space-time, but only in time and in space. That is, we only know, that two successive scans were acquired at two successive time steps. Caused by the motion of the heart consecutive images do not belong to the same heart phase in general. By the same reason we are not allowed to take sequent scans to build a 3D model. As a first step we have to synchronize the tracking data with the image data. The only known parametars are the elapsed acquisition time, the frequency of the ultrasound machine and the tracking device frequency. The start time of the acquisition is not available because ultrasound devices use FIFO buffers which loose the earliest information on buffer overflow.

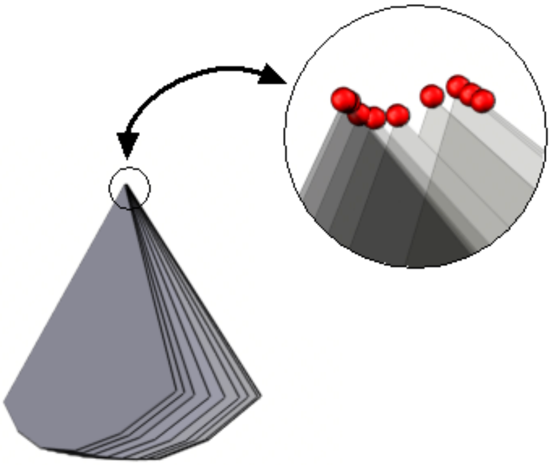

Figure 3: *Geometry of the fan acquired dataset and close-up of the positions (10 out of 1182 samples)*

With this information at hand it is possible to juxtapose image and tracking data. Caused by the in general different acquisition frequencies of the two devices, the timings between the respective data samples clearly cannot fit. In order to get rid of that, new data has to be calculated. It is preferable to change the tracking data only. The main reason is, that we do not want to degrade any image data by interpolation as long as this is not necessary. A minor reason is that the generation of new tracking data is much cheaper. A last point is that the tracking data is severely degraded by noise, which cannot be avoided in a clinical setting, and which has to be compensated for. Lower acquisition frequencies of the tracking device would remedy the situation but on cost of much fewer samples per time step.

To process the tracking data, two different interpolation methods are needed for position and orientation data. The positions are interpolated via a hermite spline fit (1), which is steered by the normals of the B-scan planes and the respective positions.

$$\begin{aligned} \mathbf{P}_n &= \vec{\mathbf{g}} \cdot \mathbf{H} \cdot \vec{\mathbf{t}}, \\ \vec{\mathbf{t}} &= (1, t, t^2, t^3)^T, \\ \vec{\mathbf{g}} &= (\mathbf{P}_i, \mathbf{P}_{i+1}, \vec{\mathbf{z}}_i, \vec{\mathbf{z}}_{i+1})^T. \end{aligned} \tag{1}$$

$\mathbf{P}_n$ denotes the interpolated point based on the points $\mathbf{P}_i$

and $\mathbf{P}_{i+1}$, t is the interpolation parameter which is raised to the respective power in $\vec{\mathbf{t}}$ and $\vec{\mathbf{g}}$ is the geometry vector composed of the points to be interpolated and the corresponding normal vectors $\vec{\mathbf{z}}_*$. Matrix $\mathbf{H}$ is the Hermite basis matrix, which blends the input data to the respective curve point.

The orientations are interpolated via key framing methods. We found that the spherical linear interpolation (2) is appropriate for our needs.

$$\begin{aligned} S(\mathbf{P},\mathbf{Q},t) &= a\cdot\mathbf{P} + b\cdot\mathbf{Q}, \\ a &= \frac{\sin((1-t)\cdot\alpha)}{\sin\alpha}, \\ b &= \frac{\sin(t\cdot\alpha)}{\sin\alpha}, \\ \alpha &= \arccos(\mathbf{P}\cdot\mathbf{Q}). \end{aligned} \tag{2}$$

Here $\mathbf{P}$ and $\mathbf{Q}$ denote the coordinate frames to be interpolated and t denotes the same interpolation parameter as in (1).

After this step we have a single tracking sample for every single image. Due to the fact that the frequencies are fixed for both devices during an acquisition, the synchronization of their output can be calculated reliably.

Having synchronized the image and tracking information, the next step is the synchronization of this data with the ECG signal. The sample frequency is fixed to 250Hz for a usual ECG signal, which is significantly higher than those from the ultrasound or tracking devices. Here an additional advantage arises, when the tracking information is adapted to the image information and not vice versa. Every ECG sample has its own tag, namely the image number. Because of the high ECG sample frequency, there are between two and seven ECG samples per image. This way not only sample numbers can be assigned to image numbers, but also ECG features, i.e. there is enough information to decide whether an image corresponds to an r-wave event or any other heart phase in scope.

Other heart phases are the p-, q-, s-, t- and u-waves. At this place it should be mentioned that the t-wave is split up into two parts due to its temporal duration. Caused by the flatness of the u-wave in normal ECGs from healthy persons and the limited resolution of the electric dissipation, its maximum cannot be localized in a real robust manner. This fact encouraged us to introduce fixed time features, which only depend on surrounding features and a subdivision interval size. So the u-wave is most often replaced by two or three virtual features, evenly distributed between t- and p-wave. The reliable identification of the heart phase features is crucial for reconstruction because the stream of information is now split up into several parts, namely the single heart cycles.

In a last step the heart cycles are normalized and stacked onto each other. The naive stacking shows severe misalignments within the single ECG traces hence the images themselves can not be aligned in an optimal manner, too. Therefore the ECG trace is resampled with variable time steps thereby fixing the trace at important features. Examples of such features can be the r-wave and the t-wave. If the fixation with two features does not result into a close fit of all traces, additional features can be introduced as fixed-points. The resampling results into a much better covering of the traces and hence an better aligning of the associated images too, as can be seen in Figure 4. The process of ECG normalization will be described elsewhere.

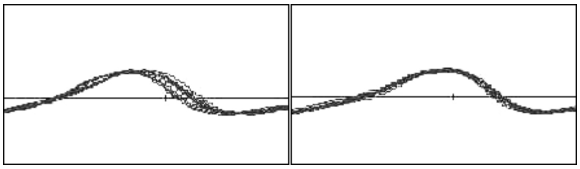

Figure 4: *Normalization of different ECG-traces. The effect of the procedure at the t-wave (left) before (right) after the normalization process.*

After the described procedure a four-dimensional dataset of the heart has been constructed, which is then stored into a special data format we call the ETS format. ETS is an abbreviation for <E>cg <T>agged <S>equence. There are different blocks each representing a 3D data set of the entire heart. The 25 to 50 blocks in total represent the beating heart, i.e. the heart cycle has been split into 25 to 50 parts. The number of parts can be chosen more or less arbitrary but one has to keep in mind, that the more time steps are introduced the less image information is available for every step. The chosen interval respects the common image repetition standard in video processing, which is 25 images per second with respect to a range of 60 to 120 bpm heart frequency during the examination.

However, splitting the stream information into several 3D data sets at different time intervals introduces gaps into each 3D data set. Assuming a time resolution of 25 blocks per cycle and a very smooth acquisition with constant angular velocity, it is obvious that splitting the data stream into 25 different parts results in a reduction per part down to 4% of the original amount of data. Fortunately, the data is spread almost homogenous over the spatial domain, so that a reconstruction is still possible. The complexity of the preprocessing step is $O(n)$ where n is the number of B-scans.

It should be mentioned explicitely that until now the original ultrasound image data is still unaltered. There has been neither scan conversion nor interpolation of the grey values obtained during the acquisition. This fact is important in contrast to other acquisition procedures, which degrade the information first by scan converting them to the cartesian domain and second due to the 24-Bit video grabbing of

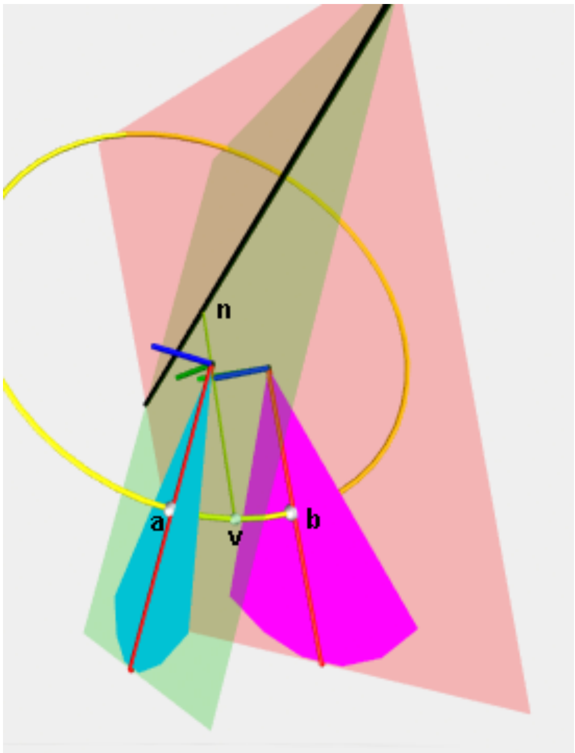

Figure 5: *Parametric situation for the interpolation of a single data point.*

8-Bit data. Applying the scan conversion too early implies a significant loss in resolution especially near the scan tip. The unavoidable quantization of the 8-Bit grey level data over 24 bits of colour data during the frame grabbing on a colour video device again introduces unnecessary data degradation.

3.2. Interpolation of B-scans

From the pre-processed sequence of B-scans, an interpolating trivariate scalar field needs to be constructed within the smallest bounding box of the scans. As a first approach we use a uniform hexahedral grid from which the scalar field can be evaluated by trilinear interpolation. An adaptive approach is described in the next section.

In principle, for every voxel the two adjacent B-scans are determined. If the planes of these scans are not parallel, they can be mapped into each other by a rotation around their intersection line. Under the assumption of rotational movement, a circular arc around the intersection line is the optimal interpolating curve. Even in the case of freehand ultrasound, this circular arc is a good approximation for the normal movement of the scanning plane, since it is a smooth curve orthogonally intersecting both scan planes and the virtual reconstruction plane. Figure 5 gives an impression of the parametric situation for the interpolation of a single data point. We note that only the normal component of an oblique movement is relevant for interpolation purposes. Tangential movement does not change the location of features in the volume (despite of their translation on the B-scan image).

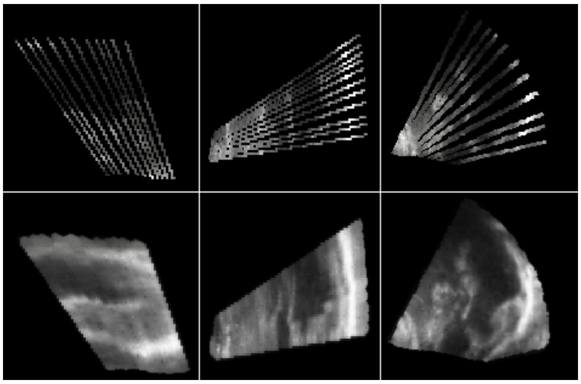

Figure 6: *Filling spaces between slices by interpolation along spherical arcs. (l) small, (m) medium, and (c) large angles between sucessive slices.*

The greyscale value c_v of the voxel's midpoint $\mathbf{v}$ is determined from the data located at the intersection points $\mathbf{a}$ and $\mathbf{b}$ of the circular arc with the two B-scan planes, weighted by their distance (d_a, d_b respectively) from $\mathbf{v}$:

$$\begin{aligned} c_v &= \tfrac{d_b}{d_a+d_b} c_a + \tfrac{d_a}{d_a+d_b} c_b, \\ d_a &= \|\mathbf{a}-\mathbf{v}\|, \\ d_b &= \|\mathbf{b}-\mathbf{v}\|. \end{aligned} \tag{3}$$

In the case of a small rotation angle or parallel scanning planes, we choose $\mathbf{a}$ and $\mathbf{b}$ to be the closest points with respect to $\mathbf{v}$ on the scanning planes. For bigger angles we offer the choice between the approximation as in (3) and the respective arclengths:

$$\begin{aligned} d_a &= \|\mathbf{v}-\mathbf{n}\| \cdot \arccos\left(\tfrac{\mathbf{v}-\mathbf{n}}{\|\mathbf{v}-\mathbf{n}\|} \cdot \vec{\mathbf{x}}_a\right), \\ d_b &= \|\mathbf{v}-\mathbf{n}\| \cdot \arccos\left(\tfrac{\mathbf{v}-\mathbf{n}}{\|\mathbf{v}-\mathbf{n}\|} \cdot \vec{\mathbf{x}}_b\right), \end{aligned} \tag{4}$$

choosing $\vec{\mathbf{x}}_a$ and $\vec{\mathbf{x}}_b$ as the directions of insonification and $\mathbf{n}$ as the nadir point, i.e. the centre of the interpolation circle.

In the case that $\mathbf{a}$ or $\mathbf{b}$ is located outside the B-scan image on the plane, zero (black color) is used for c_a or c_b, respectively. This way, the grey intensity is faded out at the boundaries of the scanned region. An example for this interpolation scheme is depicted in Figure 6.

If the two scans in scope are parallel, the circular arc degenerates to a line that is the radius becomes infinite. In this case the interpolation becomes a simple linear interpolation between the scans, based on the bilinearly sampled scan data.

It should be mentioned, that the points **a** and **b** are defined in cartesian co-ordinates, whereas the image data is given in the polar domain. In order to reference the image data an appropriate transformation is necessary. Aliasing artefacts in this process are minimized by bilinear interpolation of four neighbouring data values in the image data domain. This way the complete amount of data can be used, which is especially advantageous if the mean scan orientation is not parallel to one of the coordinate axes of the reconstruction volume.

The naive algorithm for a uniform grid works as follows: For every consecutive pair of B-scans, all rows of the volume are traversed and the greyscale values of voxel centres located between these scans are computed, as described above. To avoid traversing the entire volume for every pair of slices, we intersect for each row of voxels a horizontal line with the bounding volume of the scanning planes. This way, only the voxels between the two intersection points need to be traversed. Additionally we use a frame cache, which stores the indices of those slices which are currently active. Because we linearly traverse the volume, a direct hit occurs in most cases. If no hit is encountered we can identify the new active slices in $O(1)$ time, since we simply have to test the next left and right frames. Even if we traversed the bounding volume the new slices can be calculated in constant time. If the traversal is perpendicular to the scan planes, the new start index will equal the old one. In case of a parallel traversal, we again have to check the two adjacent slices, only. The time complexity of this algorithm is $O(m+n)$ based on n voxels and m B-scans.

3.3. Adaptive Mesh Refinement

In most cases, the B-scans are not uniformly distributed within the volume. If there are large gaps between B-scans, the reliability of the data computed in these regions is lower than in a close neighbourhood of the scans. Based on this fact it is not necessary to interpolate a grey value for every single voxel, but the resolution of the volume can be locally adapted to save computation time. Additionally, the volume resolution then exposes the local data accuracy to the user, for example when visualizing the data set by volume rendering techniques. These facts lead to AMR methods.

AMR originates from the solution of partial differential equations, representing simulation data on a hierarchy of grids adapted to the geometric complexity. AMR data is defined on a regular grid with dyadic refinement in regions of high complexity.

To embed AMR into the algorithm we use an octree, which is built up as follows: starting with the entire bounding box of the volume as seed voxel, we subdivide every voxel into eight subvoxels, until one of the following conditions is satisfied:

- The fine target resolution is reached.
- There is no B-scan slice located within a distance of r from the voxel centre.

For the radius r we choose the diameter of the current voxel. The target resolution is chosen in advance by minimum voxel size. It should be somewhat finer than the resolution of B-scans to avoid aliasing artefacts.

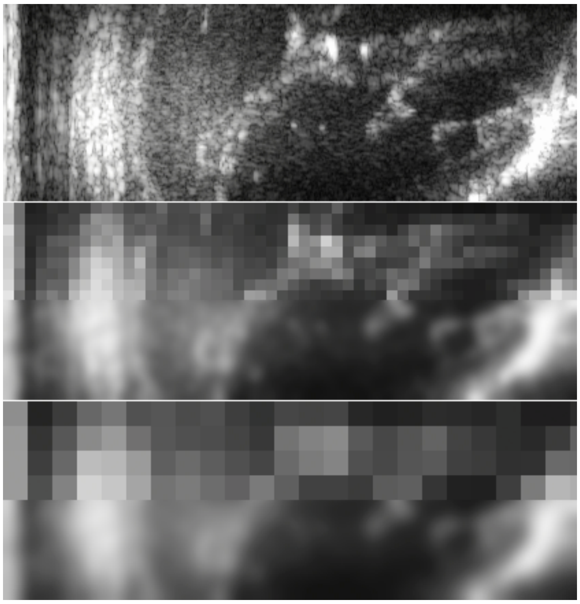

Figure 7: *Excerpt of a MIP stack. Top: original frame, middle: frame at level 4, bottom: frame at level 7*

Aliasing may occur in coarse regions of the mesh, where consecutive voxels are interpolated from a sequence of points on a particular B-scan, whose distance is greater than one pixel. To avoid aliasing, we employ mip-mapping (multum in parvo mapping), a filtering method used for texture mapping. Therefore, we compute a hierarchy of greyscale images for every B-scan by averaging four pixels into one, each time the image is coarsened (this corresponds to the low-pass filtering performed by the Haar wavelet). When sampling the B-scan for interpolation purposes, its resolution is adapted to the height in the octree where the corresponding voxel is located. Figure 7 shows an ultrasound frame at different MIP levels.

Of course there is no need to implement the MIP levels corresponding to the Haar wavelet transform, but also the filtering with appropriate Gaussian kernels is possible. This way the localization of the image content is spread much more uniform in contrast to the blocky appearance of the averaging process. In Figure 7 the upper halves of the middle

# slices	time [sec]	# interpolations
86	13	2322785
10	8	1899731
4	8	999831

Table 1: *Processing times. Shown are the number of slices, the processing time itself and the number of interpolations necessary to reconstruct the volume. The image resolution of the original slices is 420×144 pixels.*

and bottom images were calculated using the simple mean, whereas the lower halves were calculated using a Gaussian as low-pass filter.

In summary, our hierarchical algorithm works as follows: First, the octree is refined until the leaf voxels are reached. For every leaf voxel, the data corresponding to the voxel centre is interpolated as described in 3.2. The output of our method is a grid hierarchy with data associated to every grid cell. If no AMR-based rendering method is available, the octree can be further refined to the target resolution where the sub-voxels inherit the data values of their parent voxels.

Due to the different resolutions in the voxel hierarchy, it is not straightforward to efficiently locate the neighbouring B-scans for every voxel centre. When traversing the tree (for example in pre-order), we observe that large sequences of consecutive voxels are located between the same B-scans. Hence, we first examine the sampling planes used for the previous voxel before starting a linear search determining the B-scans to be used for interpolation.

Caused by the huge amount of interpolations it appears advantageous to use the special configuration of the scans, i.e. their spatial arrangement. Using this information it is possible to reduce the number of scans to be searched by an average of 50% simply by testing the signed distance. Within the tree we store an additional hint during construction, namely the index of the frame which caused the respective splitting.

4. Results

Using the presented method it is possible to construct high resolution four dimensional ultrasound datasets within short time. The method is robust and works for all types of acquisition, i.e. it is not limited to fan acquisition. Especially cone acquisition is supported, where we expect a big impact of the hierarchical method and a perfectly suited interpolation path. Nevertheless the method works for parallel slices, as well. Table 1 gives an impression of the calculation times. The computer used was a standard PC with less than 2 GHz clock frequency.

By applying the adaptive refinement mesh approach the procedure was not only speeded up significantly, but additionally this approach enables to check the quality of the reconstruction based on the resolution of the mesh. A comparison between the completely calculated data and that from the octree shows, that the errors introduced are negligible simultaneously to a significant reduction in calculation time as can be seen from Tables 1 and 2. Figure 8 gives a visual impression of the error.

# slices	standard		octree	
	$\lvert\varepsilon_n\rvert$	$\lvert\varepsilon_n\rvert/N[\%]$	$\lvert\varepsilon_o\rvert$	$\lvert\varepsilon_o\rvert/N[\%]$
44	2567	0.379	13428	2.107
30	3541	0.523	13538	2.124
16	8936	1.320	14923	2.232
10	11007	1.634	14884	2.235
4	17757	2.686	19480	2.980

Table 2: *Errors of reduced data sets. The absolute and relative errors of the standard and the octree based method are summarized. The resolution of the reconstruction was* 512^3 *voxels.*

Table 2 shows the errors produced due to slice reduction and hierarchical implementation. $|\varepsilon_*|$ indicates the absolute error for the whole volume, $|\varepsilon_*|/N$ indicates the mean error per voxel. The subscripts indicate the *normal* and *octree* method respectively. The errors were calculated using

$$|\varepsilon_*| = \sqrt{\sum (B_i - T_i)^2} \tag{5}$$

where B_i are the voxels of the original data set we test against and T_i are the voxels from the test data set. Here we clearly state that only those portions of the volumes are tested, where new data was inserted. We did *not* incorporate any blank voxels, so that $\frac{|\varepsilon_*|}{|\varepsilon_*|/N}$ is the defacto number of interpolated voxels. Of course the octree voxels from lower levels are counted with their respective multiplicity.

The volumes are compared against a volume which was calculated at full cost with all available frames using the interpolation method in scope. Although the octree introduces a relative high error when used with dense data, the advantage becomes obvious when the frame number drops down below 20 frames per volume. Here the time savings are tremendous, whereas the error difference keeps low $(0.3-0.9\%)$.

Table 3 shows the impact of the octree. Even if the slice density is very high, the amount of interpolations is reduced significantly. Summing over all levels shows, that only 20% of all possible leaves are used, including those from higher levels.

It should be mentioned at this place, that the octree organization overhead might outperform the time savings, especially if the ultrasound data is extremely dense. In this case

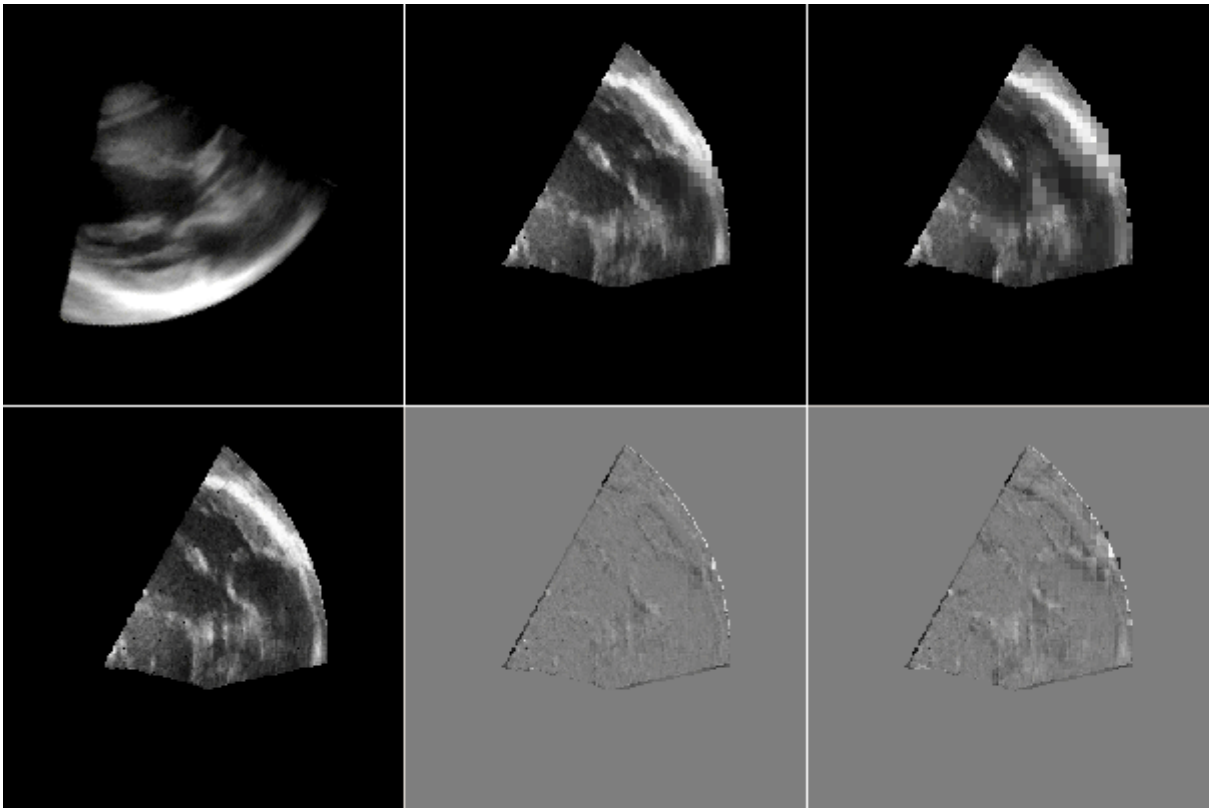

Figure 8: *Signed errors for different basis resolutions. From left to right, top to bottom the images show a volume rendering of the data set based on 14 frames, a slice from the reconstructed data set based on 10 (4) frames, the same slice from the reconstruction based on all 86 frames, and the difference images between the full resolution reconstruction and the 10 (4) slice based reconstruction.*

Level	two slices	all slices
0	100.00%	100.00%
1	100.00%	100.00%
2	100.00%	100.00%
3	80.86%	80.86%
4	56.93%	56.93%
5	37.52%	38.25%
6	22.37%	27.02%
7	12.23%	20.88%
8	6.38%	17.45%

Table 3: *Octree statistics. Percentage of leafs per level. In both cases less than 20% of all nodes are used in total.*

the tree is filled to a high degree resulting in a significant overhead due to the fact that our octree implementation is not optimized.

Figure 9 gives an impression of the reconstructed volume. The artefacts at the front side and the bottom of the volume stem from a slight gradient opacity mapping, which enhances the voxel structure at the outer borders. The interior of the volume does not show any such artefacts. Figure 10 shows the same situation for a reconstruction based on only two of the original slices. Here the octree structure can be seen very clearly due to the rendering method. Nevertheless the reconstruction is of quite high fidelity. The artefacts at the boundaries of the volume again stem from the gradient opacity mapping this time visualizing the different resolutions depending on the distance from the scans.

To get an impression on the temporal resolution Figure 11 shows a sequence of slices extracted from the reconstructed volume. The temporal offset is 20% of a heart cycle, which equals five images out of 25. The unusual form of the extracted scans is due to the fact that the data isn't placed parallel to any of the coordinate axes, forcing a non symmetric cross section.

This method has a slight drawback compared to the

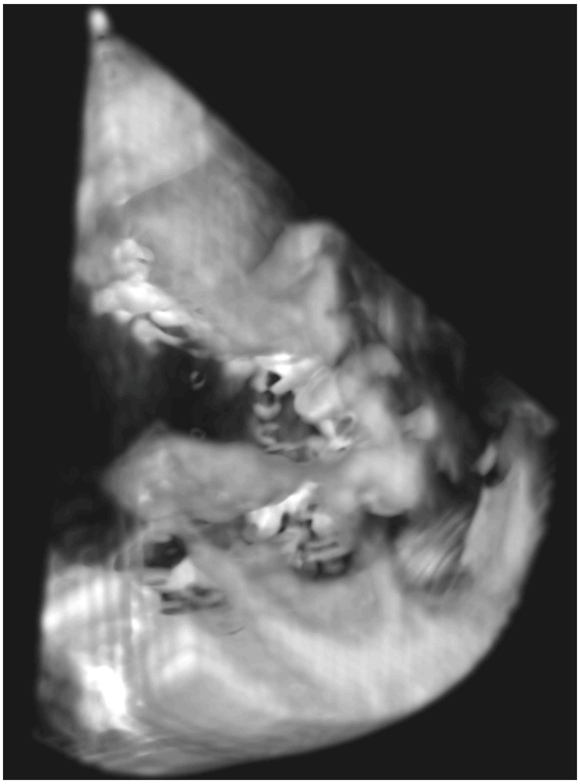

Figure 9: *Volume rendering of a reconstructed ultrasound data set. The image shows the left ventricle in the long axis view. (Artefacts at the front and bottom due to gradient opacity mapping during visualization)*

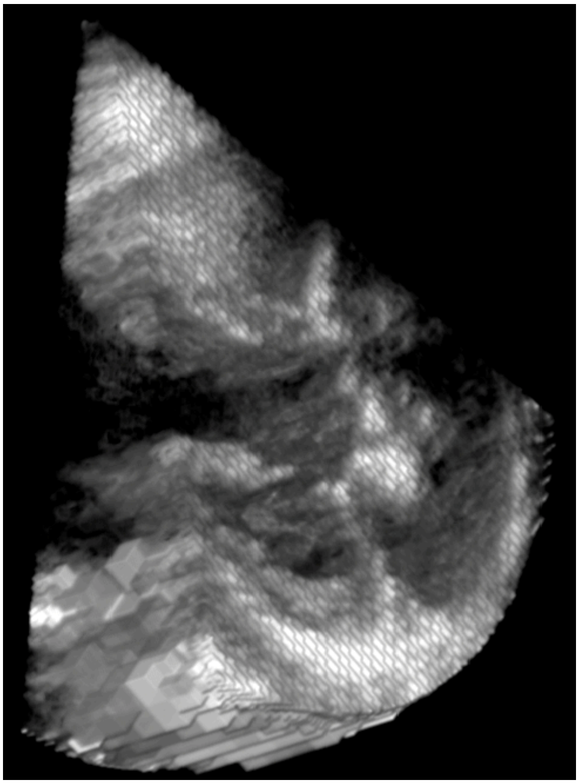

Figure 10: *Volume rendering of a reconstructed ultrasound data set at a very coarse level. The image clearly exhibits the octree structure for a reconstruction based on two slices. (Artefacts at the sides due to gradient opacity mapping during visualization)*

gated acquisition method. With gated acquisition the hart is scanned slice by slice, always acquiring a whole heart cycle for every slice. This way the temporal decomposition does not reduce the spatial density of the data. With our method a continuous sweep is performed, so that there is only a single slice for every position and time step, i.e. there are no two slices, which represent the same plane within the volume. Thus the temporal decomposition results into virtually shifted data sets.

The shift depends on the frame acquisition frequency as well as the angular velocity during the examination. The greater the angular velocity and the smaller the frame acquisition frequency the greater is the virtual shift. The shift between two successive volumes is very small, so the user might recognize a sort of flickering at most, but the situation gets much worse when we look at the jump between the last and the first volume within a loop. Here the small shifts added up into a significant jump. In Figure 11 the virtual shift can be seen, when looking at the topmost point of the data, which creeps with vertical direction. The difference between two successive slices is negligible, but the difference between the last and the first slice is significant.

To solve the above problem we can use another kind of acquisition procedure in addition to a more complex treatment of the slices. The acquisition has to be altered in the way that the probe is swept forward and then backward again. This way the acquired data is of a quasi spatio-periodical form, so that the spatial distance of two volumes is minimal when the temporal distance is maximal. The problem with this kind of data is, that the slices cannot be ordered in linear space-time any more, so that the organization of the scans and e.g. the calculation of the nearest surrounding scans with respect to a given point becomes much more expensive.

5. Conclusions

We presented a highly efficient and robust volume reconstruction method from freehand scanned echocardiographic images. The method is suited for fan acquired data sets, although it can be applied sucessfully to all kinds of data configurations, especially the cone acquired ultrasound data.

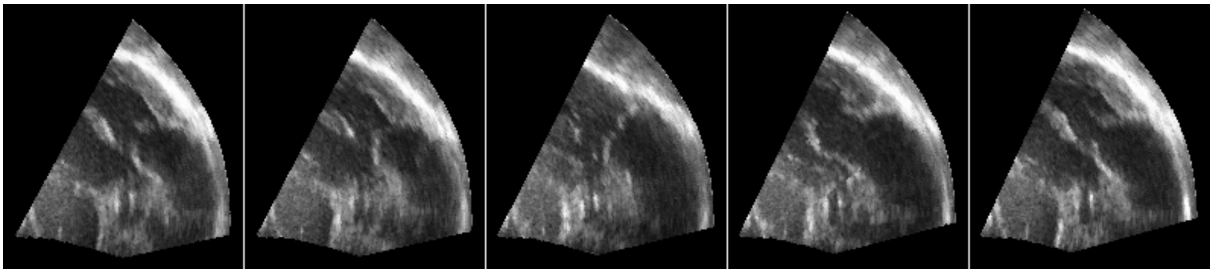

Figure 11: *Slice sequence of reconstructed volume with a temporal offset of 20% heart cycle.*

The original data can be static or time variing. The reconstructed data sets can be three- or fourdimensional, providing an extremely high spatial and temporal density.

The computation time can, in contrast to other techniques, be termed as low thereby not burdening the precision. The hierarchical approach offers a natural method to verify the reliability of the recontructed data sets. Additionally, it offers an appropriate data basis for subsequent algorithms like those for data compression.

We explicitely state at this place that this work is concerned with the reconstruction of the volumetric data sets, only and not with the visualization of this data. In subsequent works the aspect of visualizing can be treated in particular.

Acknowledgements

This work was supported by the German Research Foundation project 4DUS, and in part by the German Federal Ministry of Education and Research (BMBF), under contract number NR 01 1W A02, through project VES. The ultrasound data used in this work was provided by Prof. Dr. W. Voelker, Institut für Kardiologie, Universitätsklinik Würzburg, Germany. We thank the members of the DFKI Intelligent Visualization and Simulation Research Laboratory for stimulating discussion and comments during the preparation of this paper.

References

[BAJ*] Barry C., Allot C., John N., Mellor P., Arundel P., Thomson D., Waterton J.: Three-dimensional freehand ultrasound: Image reconstruction and volume analysis. *Ultrasound in Medcine & Biology*.

[BLB00] Beatson R. K., Light W. A., Billings S.: Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM Journal of Scientific Computing 22*, 5 (2000), 1717–1740.

[BO84] Berger M., Oliger J.: Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics 53* (1984), 484–512.

[BTH03] Bertram M., Tricoche X., Hagen H.: Adaptive smooth scattered-data approximation for large-scale terrain visualization. *Proceedings of VisSym* (2003), 177–184 & 297.

[FN80] Franke R., Nielson G. M.: Smooth interpolation of large sets of scattered data. *International Journal of Numerical Methods in Engineering 15* (1980), 1691–1704.

[MBPE02] Müller S., Bartel T., Pachinger O., Erbel R.: 3d-Echokardiographie: neue Entwicklungen und Zukunftsperspektiven. *Herz 27* (2002), 227–236.

[RGBT99] Rohling R., Gee A., Berman L., Treece G.: Radial basis function interpolation for freehand 3d ultrasound. In *Proceedings of the International Conference on Image Processing in Medical Imaging* (Visegrad, Hungary, 1999), Springer.

[RN00] Roxborough T., Nielson G.: Tetrahedron based, least squares, progressive volume models with applications to freehand ultrasound data. *Proceedings of IEEE Visualization* (2000), 93–100.

[TGP*02] Treece G., Gee A., Prager R., Cash C., Berman L.: High resolution freehand 3d ultrasound. *Technical Report, University of Cambridge* (2002).

[WKE99] Westermann R., Kobbelt L., Ertl T.: Real-time exploration of regular volume data by adaptive reconstruction of isosurfaces. *The Visual Computer 15*, 2 (1999), 100–111.

[WKL*03] Weber G., Kreylos O., Ligocki T., Shalf J., Hagen H., Hamann B., Joy K.: Extraction of crack-free isosurfaces from adaptive mesh refinement data. *Approximation and Geometrical Methods for Scientific Visualization* (2003), 19–40.