Application of different spatial sampling patterns for sparse array transducer design.

Authors: Svetoslav Ivanov Nikolov and Jørgen Arendt Jensen


Abstract

In the last years the efforts of many researchers have been focused on developing 3D real-time scanners.

The use of 2D phased-array transducers makes it possible to steer the ultrasonic beam in all directions in the scanned volume. An unacceptably large amount of transducer channels (more than 4000) must be used, if the conventional phased array transducers are extrapolated to the two-dimensional case. To decrease the number of channels, sparse arrays with different aperture apodization functions in transmit and receive have to be designed.

The design usually is carried out in 1D, and then transferred to a 2D rectangular grid. In this paper 5 different 2D array transducers have been considered and their performance was compared with respect to spatial and contrast resolution. An optimization of the element placement along the diagonals is suggested. The simulation results of the ultrasound fields show a decrease of the grating-lobe level of 10 dB for the diagonally optimized 2D array transducers.

1 Introduction

To obtain a three-dimensional scan of the body, an ultrasound scanner must be able to focus in any direction of the interrogated volume. This can be obtained either by mechanically rocking a focused transducer or by electronically steering the ultrasound beam as shown in Fig. B.1. The latter makes it possible also to implement parallel receive beamforming and to do the scanning...
Appendix B. Application of different spatial sampling patterns for sparse array transducer design.

Figure B.1: Volumetric scanning. The center of the coordinate system is in the middle of the transducer surface.

in real time [92]. The system is considered to be linear, and can thus be characterized by its point-spread-function (PSF). Ideally the PSF is a spatial δ function. Side-lobes are present in the radiation pattern due to the finite size of the transducers. The periodic nature of the linear arrays introduces grating lobes. The grating lobes are outside the scanned region if the spacing between the elements is λ/2. To obtain high resolution with a small number of channels, arrays with a pitch of p > λ/2 must be used and the grating lobes enter the viewed field.

Randomly sparsed arrays do not have a periodical sampling pattern and thus they do not have prominent grating lobes in the radiation pattern. However, a pedestal of side-lobe energy at level of approximately -30 dB from the peak value is present. Although some optimization can be applied on the weighting coefficients [105], the performance can not be increased to a level comparable to the dense arrays.

The ultrasound system must be evaluated in terms of the two-way (transmit and receive) radiation pattern. A formal approach is introduced by the use of the coarray [52] also known as effective aperture [50].

The effective aperture is briefly introduced in Section 2 and Section 3 shows how it can be used to design linear transmit and receive apertures. In Section 4 the design is extended to 2D. Section 5 gives the radiation patterns of these apertures obtained by computer simulations of ultrasound fields.

2 Effective aperture concept

The effective aperture of an array is the aperture that has a radiation pattern identical to the two-way (transmit and receive) radiation pattern of the array. The connection between the array aperture function \( a(x/\lambda) \) and the radiation pattern in the far field and in the focal region \( P(s) \) is given by the Fourier transform [92]:

\[
P(s) = \int_{-\infty}^{\infty} a \left( \frac{x}{\lambda} \right) e^{j2\pi(x/\lambda)} d \left( \frac{x}{\lambda} \right)
\]

where the aperture function describes the element weighting as a function of the element position, \( s = \sin\phi \), \( \phi \) is the angle measured from the perpendicular to the array, and \( x/\lambda \) is the
3. Aperture design strategies in 1D

Figure B.2: Transmit, receive and effective apertures. The resulting effective aperture, from top to bottom, has rectangular, triangular and cosine² apodizations.

The two way radiation pattern is

\[ P_{TR}(s) = P_T(s)P_R(s) \]  \hspace{1cm} (B.2)

The radiation pattern of the effective aperture can be expressed as a spatial convolution of the transmit and receive apertures.

\[ E(x/\lambda) = a_T(x/\lambda) * a_R(x/\lambda) \]  \hspace{1cm} (B.3)

The design of the transmit and receive apertures is thus reduced to the problem of finding a suitable effective aperture with a desired Fourier transform. The elements in the effective aperture must be spaced at \( \lambda/2 \), and the weighting function shouldn’t have discontinuities to avoid the side and grating-lobes. Since the radiation pattern is the Fourier transform of effective aperture, it is convenient to exploit the properties of the classical windowing functions: rectangular, triangular and hamming. These functions are separable, and the design can be carried out in 1D and then extended to 2D.

3 Aperture design strategies in 1D

Fig. B.2 shows three different design strategies leading to an effective aperture with \( \lambda/2 \) spaced elements.

A simple approach, shown in Fig. B.2 a) is to select a dense transmit aperture with \( N_{xmt} \) elements. Its width is \( D_{xmt} = N_{xmt}\lambda/2 \). The receive aperture must then have spacing between its elements \( d_{rcv} = (N_{xmt} - 1)\lambda/2 \). Hereby a fully populated effective aperture is obtained with the minimum number of elements in the transmit and receive apertures. Because of the rectangular shape of the apodization function, of the effective aperture, this design will be further referred to as ”rectangular approximation”.

Fig. B.2 b) shows how an apodized effective aperture can be obtained by introducing some redundancy in the number of transmit and receive elements. From the properties of the Fourier transform it is expected that this design has lower side-lobes than the design depicted in Fig. B.2 a). Further in the paper this kind of effective aperture will be referred to as “triangular”, and the design as ”triangular approximation”.

217
Appendix B. Application of different spatial sampling patterns for sparse array transducer design.

The transmit aperture has two active sub-apertures. The spacing between two active elements is $\lambda/2$. The number of active elements is $2N_{\text{act}}$. The width of the active sub-aperture is $D_{\text{act}} = N_{\text{act}} \lambda/2$. The spacing between the elements in the receive aperture is $d_{\text{rcv}} = D_{\text{act}}/2$. If it has $N_{\text{rcv}}$ active elements, its size is $D_{\text{rcv}} = (N_{\text{rcv}} - 1)D_{\text{act}}/2$. The spacing between the two active sub-apertures is $d_{\text{sub}} = D_{\text{rcv}}/2$.

Fig. B.2 c) shows how to obtain effective aperture, apodized with the coefficients of a Hamming window. In [50] these arrays are called “vernier arrays” and therefore this term will be used further in the article. The spacing of the arrays is chosen to be $n\lambda/2$ and $(n - 1)\lambda/2$, where $n$ is an integer number. This guarantees that the spacing between the samples of the effective aperture is $\lambda/2$. Additional apodization gives control over the radiation pattern. For the apertures in Fig. B.2 a value of $n = 3$ was used. From the properties of the hamming window it can be expected that this configuration would yield the lowest side and grating lobes, at the expense of decreased resolution.
4 Transition to 2D

After the apertures are created in the one-dimensional case, their design must be extended to the two-dimensional space. Usually a rectangular grid is used for sampling of the transducer surface. The distance between two elements along the same row or column is $\lambda/2$.

The extension of the rectangular approximation to 2D is straightforward. Let the transmit aperture be a rectangular grid of size 11x11 elements, spaced at $\lambda/2$ distance. The horizontal and vertical spacing between the elements of the receive grid, in this case is 5$\lambda$.

Fig. B.3 a) and b) show two examples of triangular approximations. The resulting effective aperture is shown in Fig. B.4. Configuration a) has more transmit elements in a single active sub-aperture than configuration b). To maintain the same number of elements aperture a) has less active sub-apertures.

The vernier approximation can also be extended to the two-dimensional case by selecting the element spacing independently for the $x$ and $y$ axes, as it was previously done in [50]. Such configuration is shown in Fig.B.3 c) and will be further referred to as “rectangular vernier approximation”. The spacing between the elements in the receive aperture is $4\lambda/2$ and in the transmit aperture it is $3\lambda/2$. The vernier nature of the sampling grid is preserved along the rows and the columns but is broken along the diagonals. This results in high grating lobes in the $(x-z)$ and $(y-z)$ planes. In Fig. B.4 d) a new, diagonally optimized pattern is suggested. A new element is inserted in the diagonal direction between every two receive elements. In this way the element spacing along the diagonals in the receive aperture becomes $2\lambda/\sqrt{2}$. The diagonal spacing in the transmit aperture is $\lambda/2$, thus keeping the vernier nature of the sampling pattern along the diagonals. According to the Fourier transform, this design decreases the grating-lobe energy with more than 5 dB.

Figure B.4: Effective aperture obtained by a triangular approximation.
Appendix B. Application of different spatial sampling patterns for sparse array transducer design.

5 Simulation results

All simulations are made by the program Field II [19]. A 60°x60° volume containing a single point scatterer is scanned. The point-spread function is obtained by taking the maximum in a given direction. The simulation parameters are listed in Table B.1.

The size of the transmit and receive apertures are dependent on the design method. The apertures were designed to contain a total of 256 transmit and receive channels. Fig. B.5 shows two point-spread functions: a) a Vernier approximation, extended to 2D on a rectangular grid [50], and b) a Vernier approximation optimized along the diagonals. The diagonal optimization results in almost 10 dB decrease of the highest grating lobe level. From Table B.2 it can be seen that the loss of resolution at -3 dB is 9%.

Figure B.5: Point spread functions. The arrays are designed using: a) Vernier interpolation b) Diagonally optimized Vernier approximation.
### 6. Conclusion

The periodic arrays have lower side-lobe energy than the sparse arrays and they have the potential of giving images with higher contrast. The \textit{effective aperture concept} is a fast and easy way
Appendix B. Application of different spatial sampling patterns for sparse array transducer design.

to design sparse arrays with desired properties. The design can be first implemented in 1D and then extended to 2D. The results from simulations on Vernier arrays show that radiation patterns with higher MSR are obtained when the design includes the diagonals of the rectangular grid.

7 Acknowledgements

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation, and by a grant from B-K Medical A/S.
Recursive ultrasound imaging

Authors: Svetoslav Ivanov Nikolov, Kim Gammelmark and Jørgen Arendt Jensen

Published: Proceedings of the IEEE Ultrasonics Symposium, Lake Tahoe, 1999.

Abstract

This paper presents a new imaging method, applicable for both 2D and 3D imaging. It is based on Synthetic Transmit Aperture Focusing, but unlike previous approaches a new frame is created after every pulse emission. The elements from a linear transducer array emit pulses one after another. The same transducer element is used after $N_{\text{xmt}}$ emissions. For each emission the signals from the individual elements are beam-formed in parallel for all directions in the image. A new frame is created by adding the new RF lines to the RF lines from the previous frame. The RF data recorded at the previous emission with the same element are subtracted. This yields a new image after each pulse emission and can give a frame rate of e.g. 5000 images/sec.

The paper gives a derivation of the recursive imaging technique and compares simulations for fast B-mode imaging with measurements. A low value of $N_{\text{xmt}}$ is necessary to decrease the motion artifacts and to make flow estimation possible. The simulations show that for $N_{\text{xmt}} = 13$ the level of grating lobes is less than -50 dB from the peak, which is sufficient for B-mode imaging and flow estimation.

The measurements made with an off-line experimental system having 64 transmitting channels and 1 receiving channel, confirmed the simulation results. A linear array with a pitch of 208.5 $\mu$m, central frequency $f_{\text{0tr}} = 7.5$ MHz and bandwidth $BW = 70\%$ was used. The signals from 64 elements were recorded, beam-formed and displayed as a sequence of B-mode frames, using the recursive algorithm. An excitation with a central frequency $f_{\text{0}} = 5$ MHz ($\lambda = 297 \mu$m in water) was used to obtain the point spread function of the system. The $-6$ dB width of the PSF is 1.056 mm at axial distance of 39 mm. For a sparse synthetic transmit array with $N_{\text{xmt}} = 22$ the expected grating lobes from the simulations are $-53$ dB down from the peak value at, positioned at $\pm 28^\circ$. The measured level was $-51$ dB at $\pm 27^\circ$ from the peak.

Images obtained with the experimental system are compared to the simulation results for different sparse arrays. The application of the method for 3D real-time imaging and blood-velocity estimations is discussed.
1 Introduction

Advances in DSP technology [36] make the real-time 3-D volumetric scanning a feasible imaging modality in medical ultrasound. Extending the traditional 2-D cross-sectional scanning to 3-D does not yield real-time imaging and new imaging methods have to be developed. The synthetic aperture techniques are attractive alternatives. They make it possible to increase the frame rate of B-mode ultrasound imaging, and obtain a dynamically focused image in both transmit and receive. The synthetic beam-formation approaches can be divided into three classes [36]:

- synthetic receive aperture [70]
- synthetic receive and transmit apertures [65], [61]
- synthetic transmit aperture [1], [94]

In synthetic aperture imaging the time needed to acquire a single high-resolution image (HRI) $T_{HRI}$ is proportional to the number of emissions $N_{xmt}$, the time necessary to record the reflected ultrasound wave from a single emission $T_{rec}$, and the number of scan-lines $N_l$. It is inversely proportional to the number of the parallel receive beam-formers $N_{prb}$:

$$T_{HRI} = T_{rec} \cdot N_{xmt} \cdot N_l / N_{prb}$$  \hspace{1cm} (C.1)

Synthetic receive processing involves transmitting with a full aperture and receiving with two or more sub-arrays. Several emissions are needed for every scan-line, thus, increasing $T_{HRI}$ [36].

In contrast to synthetic receive aperture processing, synthetic transmit aperture imaging involves transmission from two or more sub-apertures and receiving with a full array. In receive the RF signals from the transducer elements can be beam-formed simultaneously for all directions in the image, i.e. $N_{prb} = N_l$ [1], [94]. For every emission a low-resolution image (LRI) is created. After acquiring $N_{xmt}$ low-resolution images, the RF lines of these images are added together to form a single high-resolution image. The acquisition time for a LRI with a typical depth of 15 cm, assuming a speed of sound $c = 1540$ m/s, is $T_{rec} = 200 \mu s$. If $N_{xmt} = 64$, then $T_{HRI} = 12.8$ ms, and the frame rate is 78 frames/sec. This paper suggests further development of the transmit synthetic processing. Instead of discarding the already created high-resolution image and starting the imaging procedure all over again, the next high-resolution image is recursively built from the previous one by adding the beam-formed RF-lines from the next low-resolution image to the existing high-resolution RF-lines. The RF-lines from the low-resolution image obtained at the previous emission of the current transmit sub-aperture are subtracted from the result.

The suggested recursive calculation procedure makes it possible to create a new high-resolution image at every pulse emission, i.e. $N_{xmt} = 1$ and, thus, increase the frame rate up to 5000 frames/sec.

2 Recursive ultrasound imaging

Phased linear arrays are used to create sector B-mode images. The image consists of $N_l$ scan-lines with common origin. Each scan-line $l$ is defined by the angle with the normal vector to the transducer surface $\theta_l$. 
Figure C.1: Recursive ultrasound imaging. In transmit only one element is excited. Multiple receive beams are formed simultaneously for each transmit pulse. Each element is excited again after $N_{xmt}$ emissions ($N_{xmt} = N_{xdc} = 10$ in this example).

A pulse emitted by only one transducer element propagates as a spherical wave, when the element is small, and the received echo signal carries information from the whole region of interest. By applying different delays in receive, any of the scan-lines $l \in [1 \ldots N_l]$ can be formed. The data from one emission is used to beam-form all of the scan-lines creating one image as shown in Fig. C.1. The created image has a low resolution, since only one element is used for emission. A high-resolution image is created by summing the RF lines from $N_{xmt}$ low resolution images, each of them created after emitting with a different transducer element. Let the number of the current emission be $n$, the number of the transducer elements be $N_{xdc}$, the recorded signal by the element $j$ after emitting with element $i$ be $r_{ij}^{(n)}$, and let the necessary delay and the weighting coefficient for beam-forming scan-line $l$ be respectively $d_{lij}$ and $a_{lij}$.
Appendix C. Recursive ultrasound imaging

The beam-forming of a scan-line for a low-resolution image can then be expressed as (see Fig. C.1):

\[ s_{li}^{(n)}(t) = \sum_{j=1}^{N_{xc}} a_{lij} \cdot r_{ij}^{(n)}(t - d_{lij}), \]  

(C.2)

where \( t \) is the time relative to the start of pulse emission. The number of skipped elements between two consecutive transmissions \( n - 1 \) and \( n \) is:

\[ N_{skip} = \text{floor}[(N_{xc} - N_{xmt})/(N_{xmt} - 1)] \]  

(C.3)

If \( N_{xc} = 64 \) and \( N_{xmt} = 4 \) then \( N_{skip} = 20 \). The values for \( N_{xc} \) should be a multiple of \( N_{xmt} \), so that \( N_{skip} \) is an integer number.

The relation between the index \( i \) of the emitting element and the number \( n \) of the emission is given by:

\[ i = [(n - 1) \cdot (N_{skip} + 1)) \mod N_{xc} + 1 \]  

(C.4)

If \( N_{skip} = 20 \) then \( i = 1, 22, 43, 64, 1, \ldots \). It can be seen that emissions \( n \) and \( n \pm N_{xmt} \), are done by the same transducer element.

The forming of the final scan-lines for the high-resolution image can be expressed as:

\[ S_{li}^{(n)}(t) = \sum_{k=n-N_{xmt}+1}^{n} s_{li}^{(k)}(t) \]  

(C.5)

Equation (C.5) implies that a high-resolution image can be formed at any emission \( n \), provided that \( N_{xmt} \) low-resolution images already exist. The images that will be formed at emissions \( n \) and \( n - 1 \) can be expressed as:

\[ S_{li}^{(n)}(t) = \sum_{k=n-N_{xmt}+1}^{n} s_{li}^{(k)}(t) \]  

(C.6)

\[ S_{li}^{(n-1)}(t) = \sum_{k=n-N_{xmt}}^{n-1} s_{li}^{(k)}(t) \]  

(C.7)

Subtracting \( S_{li}^{(n-1)}(t) \) from \( S_{li}^{(n)}(t) \) gives:

\[ S_{li}^{(n)}(t) = S_{li}^{(n-1)}(t) + s_{li}^{(n)}(t) - s_{li}^{(n-N_{xmt})}(t) \]  

(C.8)

In (E.2) the new high-resolution scan-line depends on the low-resolution scan-lines at emissions \( n \) and \( n - N_{xmt} \), and on the high-resolution scan-line at emission \( n - 1 \). This dependence can be extended over a number of low- and high-resolution scan-lines obtained at previous emissions, and Equation (E.2) can be generalized as:

\[ S_{li}^{(n)}(t) = \sum_{k=1}^{B} c_k \cdot S_{li}^{(n-k)}(t) + \sum_{q=0}^{Q} b_q \cdot s_{li}^{(n-q)}(t), \]  

(C.9)

where \( B \) and \( Q \) are respectively the number of high- and low-resolution scan-lines on which \( S_{li}^{(n)} \) depends, and \( c_k \) and \( b_q \) are weighting coefficients. The recursive imaging procedure uses information from previous emissions and therefore suffers from motion artifacts. They can be reduced by decreasing \( Q \) or/and \( B \) and in this way trading resolution for motion artifacts. If \( B = 1 \) and \( Q = 0 \), the imaging procedure becomes *add-only recursive imaging*. 

226
3 Add-only recursive imaging

Let $B$ and $Q$ in (C.9) be respectively 1 and 0. The calculation procedure becomes:

$$S_l^{(n)}(t) = c_1 \cdot S_l^{(n-1)}(t) + b_0 \cdot s_{li}^{(n)}(t)$$

(C.10)

The difference between equations (E.2) and (C.10) is that instead of being subtracted, the information obtained after the emission with element $i$ decays exponentially with time. In this way the information from the past is less prone to introduce motion artifacts in the image. The other benefit is that less memory is needed, since only two frames are stored. The high-resolution image is created by only adding weighted low-resolution images. This process starts at emission $n = 1$. Let all of the transducer elements participate in creating the synthetic transmit aperture ($N_{xmt} = N_{adc}$). At emission $n$ the high-resolution image is a weighted sum of all the low-resolution images obtained at the emissions with the single elements. Consider only the low-resolution images obtained after emissions with element $i$. The first emission with element $i$ is $n = i$. The second emission with the same element is $n = i + N_{xmt}$. Element $i$ is used after every $N_{xmt}$ emissions. The sum $C_{li}$ of the low-resolution scan-lines $s_{li}^{(n)}$ obtained at these emissions will be called partially beam-formed scan-line $C_{li}$. The high-resolution scan-lines are a sum of the partially beam-formed scan lines:

$$S_l(t) = \sum_{i=1}^{N_{xmt}} C_{li}(t)$$

(C.11)

If $b_0 = 1$, then the partially beam-formed scan-line for element $i$, $C_{li}^{(n)}$ at emission $n$ is:

$$C_{li}^{(n)}(t) = s_{li}^{(n)}(t) + c_1^{N_{xmt}} \cdot s_{li}^{(n-N_{xmt})}(t) + c_1^{2N_{xmt}} \cdot s_{li}^{(n-2N_{xmt})}(t) + \ldots$$

(C.12)

This is a geometric series. If the tissue is motionless then:

$$s_{li}^{(n)}(t) = s_{li}^{(n-N_{xmt})}(t) = \ldots = s_{li}(t)$$

(C.13)

$$C_{li}^{(n)}(t) = [1 + c_1^{N_{xmt}} + c_1^{2N_{xmt}} + \ldots]s_{li}(t)$$

(C.14)

$$C_{li}^{(n)}(t) = s_{li}(t) \cdot \frac{1}{1 - c_1^{N_{xmt}}}$$

(C.15)

If $c_1 = 0.9$ and $N_{xmt} = 64$ then $1/(1 - c^{N_{xmt}}) \approx 1$ and $C_{li}^{(n)}(t) \approx s_{li}^{(n)}(t)$. Substituting the result in (C.11) gives the following result for the high-resolution scan-line:

$$S_l^{(n)}(t) \approx \sum_{i=0}^{N_{xmt}-1} c_1^i \cdot s_{li}^{(n-i)}(t)$$

(C.16)

Using (C.16) for imaging, instead of (E.2) gives images with lower resolution due to the weighting in the sum. In this case the resolution is traded for motion artifacts and less memory storage requirements, which is beneficial for flow estimation.
# Appendix C. Recursive ultrasound imaging

<table>
<thead>
<tr>
<th>System parameter</th>
<th>Notation</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of sound</td>
<td>$c$</td>
<td>1485</td>
<td>m/s</td>
</tr>
<tr>
<td>Central frequency</td>
<td>$f_0$</td>
<td>5</td>
<td>MHz</td>
</tr>
<tr>
<td>Sampling frequency</td>
<td>$f_s$</td>
<td>40</td>
<td>MHz</td>
</tr>
<tr>
<td>Oscillation periods</td>
<td>$N_{osc}$</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Pitch</td>
<td>$pitch$</td>
<td>0.2085</td>
<td>mm</td>
</tr>
<tr>
<td>Number of elements</td>
<td>$N_{xdc}$</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Relative two-sided -6dB bandwidth</td>
<td>$B$</td>
<td>70</td>
<td>%</td>
</tr>
</tbody>
</table>

Table C.1: Simulation parameters for a 3 MHz phased array system.

<table>
<thead>
<tr>
<th>$N_{xmt}$</th>
<th>$N_{act} = 1$</th>
<th>$N_{act} = 11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Level</td>
<td>Position</td>
</tr>
<tr>
<td>64</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>22</td>
<td>$\pm 40^\circ$</td>
<td>$-58$ dB</td>
</tr>
<tr>
<td>13</td>
<td>$\pm 21^\circ$</td>
<td>$-54$ dB</td>
</tr>
<tr>
<td>8</td>
<td>$\pm 13^\circ$</td>
<td>$-48$ dB</td>
</tr>
</tbody>
</table>

Table C.2: The position and level of the first grating lobe as a function of the number of emissions $N_{xmt}$.

## 4 Simulation results

Simulations were done to evaluate the performance of the imaging system as a function of the number of emissions $N_{xmt}$. Equation (E.2) was used to create high-resolution images of the point spread function.

All the simulations were done with the program Field II [19]. The parameters are listed in Table C.1.

The beam-formed signal was decimated 10 times and then envelope detected by a Hilbert trans-
4. Simulation results

Table C.4: Measured versus expected grating lobe position and level for $N_{act} = 11$.

<table>
<thead>
<tr>
<th>$N_{xmt}$</th>
<th>Expected Position</th>
<th>Level</th>
<th>Measured Position</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>13</td>
<td>$\pm 15^\circ$</td>
<td>$-53$ dB</td>
<td>$\pm 17$</td>
<td>$-51$ dB</td>
</tr>
<tr>
<td>8</td>
<td>$\pm 8^\circ$</td>
<td>$-47$ dB</td>
<td>$\pm 10$</td>
<td>$-44.5$ dB</td>
</tr>
</tbody>
</table>

Table C.4: Measured versus expected grating lobe position and level for $N_{act} = 11$.

Figure C.2: The development of a single high-resolution scan-line as a function of the number of emissions $n$ for normal recursive imaging (top), and for add-only recursive imaging (bottom).

The envelope of the signal was logarithmically compressed with a dynamic range of 60 dB. Since the B-mode image is a sector image, the point spread function was obtained by taking the maximum of the reflected signal from every direction. In the first simulation only one element ($N_{act} = 1$) was used for a single emission. The $-6$ dB width of the acquired point-spread-function was 1.01° and the $-40$ dB width was 5.03°. The levels and positions of the grating lobes as a function of the number of emissions $N_{xmt}$ are shown in Table C.2. The simulations, however, did not account for the attenuation of the signal and the presence of noise. For real applications the energy sent into the body at one emission must be increased. One way is to use multiple elements whose delays are set to create a spherical wave [1]. To verify the method, simulations with 11 active elements forming a spherical wave at every transmission, were done. The width of the point-spread-function was identical to the one obtained with $N_{act} = 1$. The levels and positions of the grating lobes are given in Table C.2. These results show that the radiation pattern of a single element can be successfully approximated by using several elements to increase the signal-to-noise ratio.

One of the problems, accompanying all synthetic aperture techniques are motion artifacts, and simulations with moving scatterers were therefore done. The signal from a single point scatterer moving at a constant speed $v = 0.1$ m/s away from the transducer was simulated. The simulation parameters were the same as those in Table C.1 except for the number of oscillation which in this case were $N_{osc} = 5$. The pulse repetition frequency was $f_{prf} = 5000$ Hz. Figure C.2 shows one RF-line of the high-resolution image as a function of the number of emissions $n$. From Fig. C.2, top it can be seen that the recursive imaging procedure suffers from motion artifacts as the other synthetic focusing algorithms. However, it can be seen from Fig. C.2, bottom that these artifacts are reduced for the add-only recursive imaging and it can be used for velocity...
Appendix C. Recursive ultrasound imaging

5 Experimental results

The measurements were done with the off-line experimental system EXTRA [161] in a water tank and on a wire phantom with attenuation. The parameters of the system are listed in Table C.3. The transducer was a linear array with a pitch $p = \lambda$. These parameters differ from the desired ($p = \lambda/2$), and new simulations were made in order to determine the expected point-spread-function, and to compare it to the measured one. The expected $-6$ dB width point-spread-function was $1.38^\circ$ and the $-40$ dB width was $-4.4^\circ$. The expected positions and levels of the grating lobes are given in Table C.4. The results of the measurements are in good agreement with the simulation results. The result of scanning a wire phantom is given in Fig. C.3. This image was obtained using 11 elements at every emission. The phantom has a frequency dependent attenuation of $0.25$ dB · (cm · MHz)$^{-1}$. The focusing delays are calculated for every scan-line sample, and the two-way propagation time of the acoustic wave is taken into consideration. Therefore the image has the quality of a dynamically focused in transmit and receive image.

6 Conclusions

A new fast imaging method has been presented. The created images have the quality of dynamically focused image in transmit and receive. The time necessary to create one frame is equal to the time of acquisition of a single scan line in the conventional scanners. The signal-to-noise ratio can be increased by using multiple elements in transmit. The motion artifacts are decreased by add-only recursive imaging and the acquired information can be used for velocity estimations.
Some of the possible applications of the method are in the real-time three-dimensional imaging and blood velocity vector estimation.

The method can be further optimized by the use of coded excitations to increase the penetration depth and the signal-to-noise ratio.

7 Acknowledgement

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation and by B-K Medical A/S.

The XTRA system was developed by S. K. Jespersen and provided by CADUS, Center for Arteriosclerosis Detection with Ultrasound, Technical University of Denmark.
3D synthetic aperture imaging using a virtual source element in the elevation plane

Authors: Svetoslav Ivanov Nikolov and Jørgen Arendt Jensen


Abstract

The conventional scanning techniques are not directly extendable for 3D real-time imaging because of the time necessary to acquire one volume. Using a linear array and synthetic transmit aperture, the volume can be scanned plane by plane. Up to 1000 planes per second can be scanned for a typical scan depth of 15 cm and speed of sound of 1540 m/s. Only 70 to 90 planes must be acquired per volume, making this method suitable for real-time 3D imaging without compromising the image quality. The resolution in the azimuthal plane has the quality of a dynamically focused image in transmit and receive. However, the resolution in the elevation plane is determined by the fixed mechanical elevation focus.

This paper suggests to post-focus the RF lines from several adjacent planes in the elevation direction using the elevation focal point of the transducer as a virtual source element, in order to obtain dynamic focusing in the elevation plane.

A 0.1 mm point scatterer was mounted in an agar block and scanned in a water bath. The transducer is a 64 elements linear array with a pitch of 209 µm. The transducer height is 4 mm in the elevation plane and it is focused at 20 mm giving a F-number of 5. The point scatterer was positioned 96 mm from the transducer surface. The transducer was translated in the elevation direction from -13 to +13 mm over the scatterer at steps of 0.375 mm. Each of the 70 planes is scanned using synthetic transmit aperture with 8 emissions. The beamformed RF lines from the planes are passed through a second beamformer, in which the fixed focal points in the elevation plane are treated as virtual sources of spherical waves. Synthetic aperture focusing is applied on them. The -6 dB resolution in the elevation plane is increased from 7 mm to 2 mm. This gives a uniform point spread function, since the resolution in the azimuthal plane is also 2 mm.
1 Introduction

In the last years the interest in 3-D ultrasound imaging has been constantly increasing. However, due to technological limitations, there is only one real-time 3-D scanner [6], which uses 2-D matrix transducer arrays. Most other scanners employ conventional linear arrays to scan the volume of interest plane-by-plane, and then the information is reconstructed in a workstation. For a typical scan-depth of 15 cm and speed of sound 1500 m/s, the time for scanning a single plane consisting of 100 scan lines is 20 ms. Because of the long acquisition time for a single plane, this method has a low frame rate. Another draw-back is the non-uniform resolution in the elevation and azimuth planes. The latter can be solved by using 1.5-D arrays, but the frame rate remains low.

The frame rate can be increased by employing a sparse transmit synthetic aperture as suggested in [1]. In this approach only a few emissions are used per plane. If only 5 emissions were used, the time for scanning the plane is reduced from 20 ms to 1 ms, increasing the frame rate 20 times.

Previously a method for increasing the resolution of ultrasound images obtained by a fixed-focus transducer was suggested in [84]. In this approach the fixed focal point is treated as a virtual source of ultrasound, and the recorded RF lines are post focused to increase the resolution.

This paper suggests the combination of the two methods to improve both the frame rate and the resolution, since the linear array transducers are usually focused in the elevation plane. The planes are scanned one-by-one using synthetic transmit aperture focusing, and then the beamformed scan lines from the planes are refocused in the elevation plane to increase the resolution.

The paper is organized as follows. Section 2 gives the theory behind the methods and how the two methods are combined. The results from simulations and measurements are given in Sections 3 and 4, respectively. Finally the conclusions are drawn in Section 4.

2 Theoretical background

The following sections give the theoretical background for obtaining images using a synthetic aperture imaging and for performing post focusing.

2.1 Synthetic transmit aperture

When a single element of a linear array ultrasound transducer is excited, a spherical acoustic wave is created, provided that the element is small enough. The back scattered signal carries information from the whole region of investigation. In receive the RF lines in all directions are beamformed in parallel. Then another transmit element is excited and the process is repeated. The beamformed RF lines are summed after $N_{\text{em}}$ elements have been used in transmit. The beamforming process is shown in Fig. D.1 and can be described as follows:

$$s_l(t) = \sum_{i=1}^{N_{\text{em}}} \sum_{j=1}^{N_{\text{det}}} a_{lkj}(t)r_{kj}(t - \tau_{lkj}(t)), \quad l \in [1...N_l],$$

$$k = f(i)$$

(D.1)
2. Theoretical background

where \( l \) is the number of the scan line, \( t \) is time relative to the trigger of the current transmit, \( r_{kj}(t) \) is the signal received by the \( j \)th element after transmitting with the \( k \)th element. \( a_{kj}(t) \) and \( \tau_{kj}(t) \) are the applied apodization factor and delay, respectively. \( N_{adc} \) is the number of transducer elements, and \( N_{xmt} \leq N_{adc} \) is the number of emissions. The index of the transmitting element \( k \) is related to the number of the current emission \( i \) by a simple relation \( f \). \( k \) is equal to \( i \), when \( N_{xmt} = N_{adc} \). Only some of the transducer elements are used in transmit if \( N_{xmt} < N_{adc} \), and a sparse transmit aperture is formed [1]. Because the delay \( \tau_{kj}(t) \) is applied only in receive, and is a function of time, the image is dynamically focused in transmit and receive assuming a linear and stationary propagation medium.
2.2 Focusing using virtual source element

In the elevation plane the transducers are either unfocused or have a fixed focus, and thereby the scanned image has a poor resolution in this plane.

Figure D.2 shows a transducer in the elevation plane at several successive positions. The wavefront below the focal point can be considered as a spherical wave within a certain angle of divergence \[ \delta \], and the focal point can be treated as a virtual source of ultrasound energy. The 3D volume is scanned by translating the transducer in the elevation direction in steps of \( \Delta y \). The focal points lie on a line parallel to the transducer surface. The data can be considered as acquired by using one virtual element in transmit and receive.

Thus, synthetic aperture focusing can be applied on the beamformed RF lines from several emissions in order to increase the resolution in the elevation direction.

Let \( n \), \( 1 \leq n \leq N_p \), denote the position and \( N_p \) be the number of several successive positions used for refocusing the data. The scan line \( s_i(t) \) beamformed at position \( n \) will be denoted as \( s_i^{(n)}(t) \). The final lines in the volume \( S_i(t) \) are beamformed according to:

\[
S_i(t) = \sum_{n=1}^{N_p} w_n(t) s_i^{(n)}(t - d_n(t)), \tag{D.2}
\]

where \( w_n(t) \) is a weighting coefficient and \( d_n(t) \) is the applied delay. The delay necessary to focus at a given distance \( z \) \((z \geq f_{ez})\) is given by:

\[
d_n(t) = \frac{2}{c} \left( z - f_{ez} - \sqrt{(z - f_{ez})^2 + \left( (n - 1 - \frac{N_p - 1}{2})\Delta y \right)^2} \right) \tag{D.3}
\]

where \( c \) is the speed of sound and \( f_{ez} \) is the distance to the elevation focus.

The best obtainable resolution in the elevation plane after post focusing is expected to be [83]:

\[
\delta y_{6dB} \approx k \frac{0.41\lambda}{\tan \frac{\theta}{2}}, \tag{D.4}
\]

where \( \lambda \) is the wavelength, and \( \theta \) is the angle of divergence after the focal point. The variable \( k \) \((k \geq 1)\), is a coefficient depending on the apodization. For a rectangular apodization \( k \) equals 1 and for Hanning apodization it equals 1.64.

The angle of divergence can be approximated by [84]:

\[
\frac{\theta}{2} \approx \tan^{-1} \frac{h}{2f_{ez}}, \tag{D.5}
\]

where \( h \) is the size of the transducer in the elevation plane, and \( f_{ez} \) is the distance to the fixed focus in the elevation plane. Substituting (D.5) in (D.4) gives:

\[
\delta y_{6dB} \approx 0.82\lambda k \frac{f_{ez}}{h}, \tag{D.6}
\]

Equation (D.6) shows that the resolution is depth independent. However, this is true only if the number of the transducer positions is large enough to maintain the same F-number for the virtual array as a function of depth. For real-life applications the achievable resolution can be substantially smaller.
2.3 Combining the two methods

The whole process can be divided into two stages and is summarized in Fig. D.3. In the first stage a high resolution image is created using only a few emissions, say $N_{\text{emt}} = 5$ as given in Fig. D.1. This is repeated for several positions. Then the beamformed RF lines from these images are delayed, weighted, and summed a second time using (D.2) to form the final 3D volume. The considerations so far have been only for transducers translated in the $y$ direction. The synthetic aperture focusing is applicable for any kind of transducer motion (translation, rotation, or a free-hand scan), as long as the exact positions of the focal points are known.

3 Simulations

The simulations were done using the program Field II [19]. The simulation parameters, given in Table D.1, were chosen to match the parameters of the system used for the measurements.

Seven point scatterers lying at depths from 70 to 100 mm were simulated at 70 positions. The distance between every two positions in the elevation direction was 0.7 mm. Figure D.4 on the left shows the -10 dB isosurfaces of the measured point-spread-functions (PSF). Then the beamformed scan lines were post-focused using $N_p = 30$ planes to create one new plane. If a dynamic apodization is to be used, then the number of usable positions $N_p$ for depth $z$ from the
Appendix D. 3D synthetic aperture imaging using a virtual source element in the elevation plane

Figure D.4: The 3-D point-spread function outlined at -10 dB.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Notation</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of sound</td>
<td>$c$</td>
<td>1480</td>
<td>m/s</td>
</tr>
<tr>
<td>Sampling freq.</td>
<td>$f_s$</td>
<td>40</td>
<td>MHz</td>
</tr>
<tr>
<td>Excitation freq.</td>
<td>$f_0$</td>
<td>5</td>
<td>MHz</td>
</tr>
<tr>
<td>Wavelength</td>
<td>$\lambda$</td>
<td>296</td>
<td>µm</td>
</tr>
<tr>
<td>-6 dB band-width</td>
<td>$BW$</td>
<td>4.875 - 10.125</td>
<td>MHz</td>
</tr>
<tr>
<td>Transducer pitch</td>
<td>$p$</td>
<td>209</td>
<td>µm</td>
</tr>
<tr>
<td>Transducer kerf</td>
<td>$kerf$</td>
<td>30</td>
<td>µm</td>
</tr>
<tr>
<td>Number of elements</td>
<td>$N_{xdc}$</td>
<td>64</td>
<td>-</td>
</tr>
<tr>
<td>Transducer height</td>
<td>$h$</td>
<td>4</td>
<td>mm</td>
</tr>
<tr>
<td>Elevation focus</td>
<td>$f_{ez}$</td>
<td>20</td>
<td>mm</td>
</tr>
</tbody>
</table>

Table D.1: Simulation parameters

The real transducer can be determined by:

$$N_p = \left\lfloor \frac{2h z - f_{ez}}{f_{ez} \Delta y} \right\rfloor,$$

(D.7)

Figure D.4 on the right shows the PSF after the post focusing was applied. Table D.2 shows the -6 dB resolution in the azimuth and the elevation planes. The lateral size of the PSF in the azimuth plane increases linearly with depth:

$$\delta_{x_{6dB}} = z \sin \phi_{6dB},$$

(D.8)

where $\phi_{6dB}$ is the angular size of the PSF in polar coordinates.

The $\delta_{y_{6dB}}$ prior to the synthetic aperture focusing also increases almost linearly with depth, which shows that the beam is diverging with a certain angle as shown in Fig. D.2. After applying the synthetic aperture focusing $\delta_{y_{6dB}}$ becomes almost constant as predicted by (D.6).
4. Measurements

Before SAF After SAF
Depth [mm] \( \delta_x [\text{dB}] \) \( \delta_y [\text{dB}] \) \( \delta_y [\text{dB}] \) 
70 1.44 4.78 1.72 
75 1.54 5.16 1.72 
80 1.65 5.48 1.72 
85 1.75 5.80 1.85 
90 1.85 6.18 1.85 
95 1.96 6.56 1.85 
100 2.06 6.75 1.97 

Table D.2: The resolution at -6 dB as a function of depth.

A Hann window was used for \( w_n \), and this gives \( k \approx 1.6 \). Substituting \( h = 4 \text{ mm} \), \( f_{ez} = 20 \text{ mm} \), and \( \lambda = 0.296 \text{ mm} \), gives \( \delta_{y6dB} \approx 1.87 \).

4 Measurements

Figure D.5: PSF in the elevation plane: (top) before and (bottom) after synthetic aperture focusing. The innermost contour is at level of -6 dB, and the difference between the contours is also 6 dB.

The measurements were done using the department’s off-line experimental system XTRA [161]. The parameters of the system are the same as the ones used in the simulations and are given in Table D.1.

In [84] it is argued that due to the narrow angle of divergence after the focal point, the grating lobes are greatly suppressed. Therefore it is possible to traverse the elevation direction at steps \( \Delta y \) bigger than one wavelength \( \lambda \). Two experiments were conducted:

1. A point scatterer mounted in an agar block 96 mm away from the transducer was scanned, at step \( \Delta y = 375 \mu \text{m} \). The diameter of the point scatterer was 100 \( \mu \text{m} \).
2. A wire phantom was scanned at steps of $\Delta y = 700 \mu m$. The wires were positioned at depths from 45 to 105 mm, 20 mm apart. At every depth there were two wires, perpendicular to each other. The diameter of the wires was 0.5 mm.

The first experiment was conducted to verify the resolution achieved in the simulations. The goal of the second experiment was to verify that the resolutions in the elevation and azimuthal planes are comparable in size.

Using only a few emissions per plane corresponds to using a sparse transmit aperture. The use of wires as a phantom gives a good signal-to-noise ratio (compared to the 0.1 mm point scatterer) necessary to evaluate the level of the associated grating lobes. The SNR was further increased by using 11 elements in transmit to create a spherical wave instead of 1 as described in [1].

Figure D.5 shows the PSF of the point scatterer in the elevation plane. The contours are drawn at levels 6 dB apart. Sixty planes ($N_p = 60$) were used in the post-focusing. This maintains the same size of the virtual array as the one in the simulations. The achieved resolution at -6 dB is 2 mm and is comparable with the resolution obtained in the simulations.

![Figure D.5: PSF of the point scatterer in the elevation plane.](image)

Figure D.6: Outline at -10 dB of the wire phantom: (top) before, and (bottom) after post focusing.

Figure D.6 shows a -10 dB outline of the wire phantom. The image was beamformed using $N_{xmt} = 8$ emissions per plane. The post focusing was performed using $N_p = 21$ planes for each
5. Conclusions

new. The image shows that the resolution in the elevation and azimuth planes are of comparable size.

One of the problems of the approach are the grating lobes. They can be caused by two factors: (a) using only a few emissions per plane and (b) using a large step $\Delta y$ between two planes.

The step $\Delta y$ must not exceed the size of the PSF at the elevation focus, in order not to get image with discontinuities. For a larger step $\Delta y$, a transducer with a higher F-number must be used. Such transducers have a smaller angle of divergence $\theta$ and therefore the level of the grating lobes in the elevation direction is greatly suppressed. However, this is not the case in the azimuth plane. Table D.3 shows the compromise between the level of the grating lobes and

<table>
<thead>
<tr>
<th>$N_{xmt}$</th>
<th>Position</th>
<th>Level</th>
<th>$\max(N_p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>NA</td>
<td>NA</td>
<td>7</td>
</tr>
<tr>
<td>13</td>
<td>$\pm 17$</td>
<td>$-51.0$ dB</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>$\pm 10$</td>
<td>$-44.5$ dB</td>
<td>62</td>
</tr>
<tr>
<td>5</td>
<td>$\pm 7$</td>
<td>$-41.3$ dB</td>
<td>100</td>
</tr>
</tbody>
</table>

Table D.3: Grating lobes in the azimuth plane.

the maximum available number of planes for the post focusing. The figures are derived for speed of sound $c = 1540$ m/s and scan depth $\max(z) = 15$ cm. The table shows, that the larger the number of emissions per position $N_{xmt}$ is, the larger the step $\delta y$ must be, in order to maintain the number of volumes per second constant. This requires the use of a transducer which is not strongly focused in the elevation plane.

5 Conclusions

An approach for 3-D scanning using synthetic transmit aperture in the azimuth plane followed by synthetic aperture post focusing in the elevation plane was presented. The acquisition is done plane by plane using a conventional linear array transducer. The obtained resolution in the elevation plane is $\approx 2$ mm, and is comparable to the one in the azimuth plane. Up to 10 volumes per second can be scanned if only 8 emissions per plane are used, and each volume contains 62 planes. This makes the approach a feasible alternative for real-time 3-D scanning.

6 Acknowledgements

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation and by B-K Medical A/S.

Some of the measured data were supplied by Søren Pihl Rybro, Peter Foged Christensen, and Madalina Breten.

The measurement system was built by Søren Kragh Jespersen as part of his Ph.D. study.
Velocity estimation using recursive ultrasound imaging and spatially encoded signals

Authors: Svetoslav Nikolov, Kim Gammelmark and Jørgen Jensen

Published: Proceedings of the IEEE Ultrasonics Symposium, Puerto Rico, October 2000.

Abstract

Previously we have presented a recursive beamforming algorithm for synthetic transmit aperture focusing. At every emission a beamformed low-resolution image is added to an existing high-resolution one, and the low-resolution image from the previous emission with the current active element is subtracted yielding a new frame at every pulse emission.

In this paper the method is extended to blood velocity estimation, where a new Color Flow Mapping (CFM) image is created after every pulse emission. The underlying assumption is that the velocity is constant between two pulse emissions and the current estimates can therefore be used for compensation of the motion artifacts in the data acquired in the next emission.

Two different transmit strategies are investigated in this paper: (a) using a single defocused active aperture in transmit, and (b) emitting with all active transmit sub-apertures at the same time using orthogonal spatial encoding signals.

The method was applied on data recorded by an experimental system. The estimates of the blood velocity for both methods had a bias less than 3 % and a standard deviation around 2 % making them a feasible approach for blood velocity estimations.

1 Introduction

Modern scanners estimate the blood velocity by sending ultrasound pulses in the same direction and processing the signal returned from a given depth. To create a map of the velocity distribution in the area of investigation, the signal must be sent several consecutive times in each of several different directions. The precision of the estimates increases, if the estimates are based on a larger number of acquisitions in one direction. This, however, decreases the frame rate and the choice is based on a compromise between frame rate and precision.
This compromise can be avoided if a new frame is created after every emission and its data used for velocity estimation. The continuous flow of data allows the use of stationary echo canceling filters with longer impulse responses, and estimates based on a larger number of emissions, which both improve the estimates’ precision.

One approach to create a new frame at every pulse emission is to use Recursive Ultrasound Imaging [80]. The beamformed data as proposed in [80] is suitable for B-mode imaging but not for blood velocity estimation, because of the present motion artifacts.

In this article the CFM is calculated after every emission, and the velocity estimates from the current frame are used for correcting the motion artifacts in the next one. Since the estimates are based on longer sample sequences, they have a high precision, and the motion artifacts can thereby be compensated fully.

Since each emission is performed only by one element, and the blood is moving, the performance of the above mentioned procedure depends on the shot sequence. This dependency can be avoided by using the same elements at every emission with a spatial encoding scheme as suggested in [3].

2 Theoretical background

The following sections give the theoretical background for recursive ultrasound imaging and the use of spatially encoded transmits to increase the signal-to-noise ratio.

2.1 Recursive imaging

A pulse emitted by only one transducer element propagates as a spherical wave, when the element is small, and the received echo signal carries information from the whole region of interest. By applying different delays in receive, any of the scan-lines \( m \in [1 \ldots N_m] \) can be formed. The data from one emission is used to beam-form all of the scan-lines creating one image as shown in Fig. E.1. The created image has a low resolution, since only one element is used for emission. A high-resolution image is created by summing the RF lines from \( N_{xmt} \) low resolution images, each of them created after emitting with a different transducer element. Let the number of the current emission be \( k \), the number of the transducer elements be \( N_{xdc} \), the recorded signal by the element \( j \) after emitting with element \( i \) be \( r_{ij}^{(k)} \), and let the necessary delay and the weighting coefficient for beam-forming of scan-line \( m \) be \( d_{mij} \) and \( a_{mij} \), respectively. The beam-forming of a scan-line for a low-resolution image can then be expressed as (see Fig. E.1):

\[
 s_{mi}^{(k)}(t) = \sum_{j=1}^{N_{xdc}} a_{mij} \cdot r_{ij}^{(k)}(t - d_{mij}), \quad (E.1)
\]

where \( t \) is time relative to start of pulse emission. Provided that the tissue below the transducer is motionless, the forming of the final scan-lines for the high-resolution image can be expressed as [80]:

\[
 S_{m}^{(k)}(t) = S_{m}^{(k-1)}(t) + s_{mi}^{(k)}(t) - s_{mi}^{(m-N_{xmt})}(t) \quad (E.2)
\]

This method, however, suffers from a low signal-to-noise (SNR) ratio and from motion artifacts. Using multiple elements in transmit to send “defocused” ultrasound wave improves [1] the
Figure E.1: Recursive ultrasound imaging. In transmit only one element is excited. Multiple receive beams are formed simultaneously for each transmit pulse. Each element is excited again after $N_{xmt}$ emissions ($N_{xmt} = N_{xdc} = 10$ in this example).

situation. Further, the SNR can be increased by using encoded signals. The encoding can be temporal (for example using linear frequency modulated excitation) or spatial as described in the next section.

### 2.2 Spatial encoding

The idea behind the spatial encoding is to send with all of the $N_{xmt}$ elements as shown in Fig. E.2, instead of sending with only one element $i$, $1 \leq i \leq N_{xmt}$ at a time [3]. The signal sent into
Appendix E. Velocity estimation using recursive ultrasound imaging and spatially encoded signals

Figure E.2: Spatially encoded transmits using 4 transmit elements.

the tissue by each of the transmit elements \( i \) is

\[
e_i(t) = q_i \cdot e(t), \quad 1 \leq i \leq N_{xmt},
\]

where \( e(t) \) is a basic waveform and \( q_i \) is an encoding coefficient.

Assuming a linear propagation medium, the signal \( r_j(t) \) received by the \( j \)th element can be expressed as:

\[
r_j(t) = \sum_{i=1}^{N_{xmt}} q_i \cdot r_{ij}(t),
\]

where \( r_{ij}(t) \) would be the signal received by element \( j \), if the emission was done only by element \( i \).

From Eq.(E.1) it can be seen that the components \( r_{ij}(t) \) must be found in order to beamform the signal. The received signals can be expressed in a matrix form:

\[
\begin{bmatrix}
  r_{j}^{(1)} \\
  r_{j}^{(2)} \\
  \vdots \\
  r_{j}^{(N_{xmt})}
\end{bmatrix} =
\begin{bmatrix}
  q_1^{(1)} & q_2^{(1)} & \cdots & q_{N_{xmt}}^{(1)} \\
  q_1^{(2)} & q_2^{(2)} & \cdots & q_{N_{xmt}}^{(2)} \\
  \vdots & \vdots & \ddots & \vdots \\
  q_1^{(N_{xmt})} & q_2^{(N_{xmt})} & \cdots & q_{N_{xmt}}^{(N_{xmt})}
\end{bmatrix}
\begin{bmatrix}
  r_{1j} \\
  r_{2j} \\
  \vdots \\
  r_{N_{xmt}j}
\end{bmatrix}
\]

\[
(E.5)
\]

where the superscript \( (k) \), \( 1 \leq k \leq N_{xmt} \) is the number of the emission, \( q_i^{(k)} \) is the encoding coefficient applied in transmit on the transmitting element \( i \), and \( r_j^{(k)} \) is the signal received by the \( j \)th element. In the above system of equations the time is skipped for notational simplicity. Also stationary tissue is assumed so that:

\[
r_{ij}^{(1)} = r_{ij}^{(2)} = \cdots = r_{ij}^{(N_{xmt})} = r_{ij}
\]

\[
(E.6)
\]

More compactly, the equation can be written as:

\[
\vec{r}_j = Q \vec{r}_{ij},
\]

\[
(E.7)
\]

where \( Q \) is the encoding matrix. Obviously the responses \( r_{ij}(t) \) are:

\[
\vec{r}_{ij} = Q^{-1} \vec{r}_j
\]

\[
(E.8)
\]
2. Theoretical background

A suitable encoding matrix $Q$ is the Hadamard matrix $H$ [3]. The inverse Hadamard matrix is a scaled version of itself, i.e. for a matrix $H_{N_{xmt}}$ with $N_{xmt} \times N_{xmt}$ elements, the inverse is $H_{N_{xmt}}^{-1} = 1/N_{xmt}H_{N_{xmt}}$.

The above derivation strongly relies on the assumption in (E.6). In the case of abdominal scanning, and for low values of $N_{xmt}$, this assumption is “almost fulfilled”. However, in cardiac imaging and blood velocity estimation, this assumption is severely violated and the movement of the blood and heart must be compensated for.

2.3 Motion compensation

The motion compensation is considered for two cases: (a) recursive imaging without spatial encoding and (b) recursive imaging with spatial encoding.

Without spatial encoding

![Figure E.3: Motion compensation for recursive imaging without spatial encoding.](image)

During the first stage of the beamforming process, low-resolution images are created, using dynamic receive focusing. The assumption is that within one scan line $s_{mi}(t)$, the wavefront propagates as a plane wave, as shown in Fig. E.3. Figure E.3 shows the movement of one point scatterer within the limits of one scan line. The scatterer moves with velocity $\vec{v}$, from position $p_0(\vec{x}_0, kT)$ to a new position $p_1(\vec{x}_1, (k + 1)T)$ for the time $T$ between two pulse emissions. The movement across the beam (perpendicular to the direction $\vec{n}$) determines the strength of the backscattered energy, while the movement along the beam determines the time instance when the backscattering occurs.

For the case depicted in Fig. E.3 the difference in time when the backscattering occurs for the positions $p_0$ and $p_1$ is:

$$\tau = \frac{2 \cdot \Delta l}{c}, \quad (E.9)$$

where $\Delta l$ is the distance traveled from one pulse emission to the next:

$$\Delta l = \langle \vec{v}, \vec{n} \rangle T, \quad (E.10)$$

where $\langle \vec{v}, \vec{n} \rangle$ is the inner product between the velocity vector $\vec{v}$ and the directional vector $\vec{n}$.

The velocity at emission $k$ as a function of time $t$ from the emission of the pulse along the line $m$ is $v^{(k)}_m(t)$. The delay $\tau$ is also a function of $t$, $\tau^{(k)}_m(t)$. The beamformation process with
the velocity incorporated in it becomes:

\[
\text{for } m = 1 \text{ to } N_m
\]

\[
s_{mi}^{(k)}(t) = \sum_{j=1}^{N_{adc}} a_{mi j} \cdot r_{ij}^{(k)}(t - d_{mij})
\]

\[
M_m^{(k)}(t) = S_m^{(k-1)}(t) - s_{mi}^{(k-N_{xmt})}(t)
\]

\[
\tau_m^{(k+1)}(t) = \frac{2|v_m(t)|T}{c}
\]

\[
\Delta_m^{(k)}(t) = \Delta_m^{(k-1)}(t) + \tau_m^{(k)}(t) - \tau_m^{(k-N_{xmt}+1)}(t)
\]

\[
S_m^{(k)}(t) = M_m^{(k)}[t + \tau_m^{(k-N_{xmt}+1)}(t)] + s_{mi}^{(k)}[t - \Delta_m^{(k)}(t)]
\]

(E.11)

where \(\Delta\) is the delay between the first and the last of the low-resolution images, currently comprised in the high-resolution one.

**With spatial encoding**

Figure E.4 shows the model adopted for motion compensation in the presence of spatially encoded signals. Several transducer elements across the whole span of the transducer aperture are used in transmit. The sum of the emitted waves creates a planar wave propagating in a direction perpendicular to the transducer surface. A point scatterer moves for one pulse repetition period from positions \(p_0(\vec{x}_0, kT)\) to a new position \(p_1(\vec{x}_1, (k+1)T)\). The difference between the time instances, when the scattering occurs, is:

\[
\tau = \frac{2\Delta l}{c}, \quad (E.12)
\]

where \(\Delta l\) is the distance traveled by the point scatterer:

\[
\Delta l = v_z T \quad (E.13)
\]

In the above equation \(v_z\) is the component of the velocity normal to the transducer surface. The delay \(\tau\) is a function of the time \(t\), and the emission number \(k\), \(\tau = \tau^{(k)}(t)\)

Thus, the signals received by element \(j\) for emission number \(k \in [1, N_{xmt}]\) are:

\[
r_j^{(2)}(t) = r_j^{(1)}(t - \tau^{(1)}(t))
\]

\[
r_j^{(3)}(t) = r_j^{(2)}(t - \tau^{(2)}(t))
\]

... ...

\[
r_j^{(N_{xmt})}(t) = r_j^{(N_{xmt}-1)}(t - \tau^{(N_{xmt}-1)}(t))
\]

(E.14)
The reconstruction must be performed prior to beamforming the signal at a given point. First the received signals \( r^{(k)}(t) \) are appropriately delayed, and then the system of equations (E.5) is solved.

3. Experimental results

3.1 Measurement setup

The measurements were done, using the department’s off-line experimental system XTRA [161]. The most important parameters are listed in Table E.1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Notation</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed of sound</td>
<td>( c )</td>
<td>1540</td>
<td>m/s</td>
</tr>
<tr>
<td>Sampling freq.</td>
<td>( f_s )</td>
<td>40</td>
<td>MHz</td>
</tr>
<tr>
<td>Excitation freq.</td>
<td>( f_0 )</td>
<td>5</td>
<td>MHz</td>
</tr>
<tr>
<td>Pulse duration</td>
<td>( T_p )</td>
<td>1.5</td>
<td>cycles</td>
</tr>
<tr>
<td>-6 dB band-width</td>
<td>( BW )</td>
<td>4.875 - 10.125</td>
<td>MHz</td>
</tr>
<tr>
<td>Transducer pitch</td>
<td>( p )</td>
<td>209</td>
<td>( \mu )m</td>
</tr>
<tr>
<td>Transducer kerf</td>
<td>( kerf )</td>
<td>30</td>
<td>( \mu )m</td>
</tr>
<tr>
<td>Number of elements</td>
<td>( N_{xdc} )</td>
<td>64</td>
<td>-</td>
</tr>
<tr>
<td>Transducer height</td>
<td>( h )</td>
<td>4</td>
<td>mm</td>
</tr>
<tr>
<td>Elevation focus</td>
<td>( f_{ez} )</td>
<td>20</td>
<td>mm</td>
</tr>
</tbody>
</table>

Table E.1: Measurement parameters

A tissue mimicking phantom with frequency dependent attenuation of 0.25 dB/[cm.MHz] and speed of sound \( c = 1540 \) m/s was scanned at 65 positions in a water bath. From position to position the phantom was moved 70 \( \mu \)m at an angle of 45° to the transducer surface. Assuming a pulse repetition frequency \( f_{pr f} = 1/T = 7000 \), this movement corresponds to a plug-flow with velocity \( |\vec{v}| = 49.5 \) cm/s.

A precision translation system was used for the movement of the phantom. The precision of the motion in the axial and lateral directions were: \( \Delta z = 1/200 \) mm, and \( \Delta x = 1/80 \) mm, respectively.

3.2 Velocity estimation

In the theoretical considerations, it was assumed that the blood velocity was estimated, without any considerations about the velocity estimator.

The cross-correlation estimator suggested in [133] is suitable for the broad band pulses used by this method. In the implementation it is assumed, that the two consecutive high-resolution lines \( S^{(k)}(t_2) \) and \( S^{(k-1)}(t_1) \) are related by:

\[
S^{(k)}(t_2) = S^{(k-1)}(t_1 - t_s), \tag{E.15}
\]

where \( t_s \) is a time lag due to the movement of the scatterers and is related to the axial component
Appendix E. Velocity estimation using recursive ultrasound imaging and spatially encoded signals

<table>
<thead>
<tr>
<th>Reference</th>
<th>Spatially encoded</th>
<th>Non encoded</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\vec{v}</td>
<td>$ [m/s]</td>
</tr>
<tr>
<td>$\sigma/</td>
<td>\vec{v}</td>
<td>$ %</td>
</tr>
</tbody>
</table>

Table E.2: Results from the velocity estimation at angle $(\vec{v}, \vec{n}) = 45^\circ$

of the velocity $v_z$ by:

$$t_s = \frac{2v_z T}{c}$$  \hspace{1cm} (E.16)

The peak of the cross-correlation between segments of $S^{(k)}(t)$ and $S^{(k-1)}(t)$ would be found at time $t_s$. Estimating $t_s$ leads to the estimation of $\vec{v}$.

### 3.3 Reference velocity estimation

In order to obtain a reference estimate of the velocity, at each position of the phantom a high resolution image was created using 13 emissions per image. The velocity was estimated using a cross-correlation estimator. The correlation length was equal to the length of the transmitted pulse. The number of lines, over which the calculated correlation function was averaged was 8. The search length was $\pm \lambda/4$ to avoid aliasing problems. Figure E.5 shows the mean velocity $|\vec{v}|$ for the central line as a function of depth. The mean was calculated over 55 estimates.

In the axial direction the translation system has a precision of $\Delta z = 5 \mu m$, which is 10 % of the desired step. The results are, thus, within the precision of the system.

### 3.4 Recursive velocity estimation

The mean velocity $|\vec{v}|$ and the normalized standard deviation $\sigma/|\vec{v}|$ estimated using recursive ultrasound imaging are shown in Table E.2. The angle between the velocity vector $\vec{v}$ and the directional vector $\vec{n}$ of the scan line is $\angle(\vec{v}, \vec{n}) = 45^\circ$. The number of frames is 36. In this table $|\vec{v}|$ is the average of the mean velocity in the range from 30 to 80 mm. $\sigma$ is also averaged in the same range. The angle dependence of the estimates is shown in Figure E.6. The dashed lines show the velocity at $\pm \sigma$. 

Figure E.5: Mean reference velocity.
4. Conclusions

It can be seen that the reference velocity estimation exhibits a smaller bias than the velocity estimations using recursive imaging.

The recursive imaging using spatially encoded transmits exhibits angular dependence. At angles of $42^\circ - 45^\circ$ it has a low bias and standard deviation, comparable to that of the reference velocity estimates. One of the possible reasons for the angular dependency is the low number of emissions ($N_{xmt} = 4$), resulting in higher side and grating lobes in the image.

4 Conclusions

In this paper a method for motion compensation and velocity estimation using recursive ultrasound imaging was presented. The method provides the blood velocity estimator with as much as several thousand measurements per second for every sample in the investigated region.

It has been experimentally verified that the method works for a speckle generating phantom with frequency dependent attenuation. One limitation is that no noise was present in the experiment and the velocity was constant.

Future work will include velocity profiles and mixture of moving and stationary tissue.

5 Acknowledgements

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation and by B-K Medical A/S.

The measurement system XTRA was built by Søren Kragh Jespersen, as part of his Ph.D. study.

Figure E.6: The mean velocity and the velocity at $\pm \sigma$ as a function of angle.
Fast simulation of ultrasound images

Authors: Jørgen Arendt Jensen and Svetoslav Ivanov Nikolov


Abstract

Realistic B-mode and flow images can be simulated with scattering maps based on optical, CT, or MR images or parametric flow models. The image simulation often includes using 200,000 to 1 million point scatterers. One image line typically takes 1800 seconds to compute on a state-of-the-art PC, and a whole image can take a full day. Simulating 3D images and 3D flow takes even more time. A 3D image of 64 by 64 lines can take 21 days, which is not practical for iterative work. This paper presents a new fast simulation method based on the Field II program. In imaging the same spatial impulse response is calculated for each of the image lines, and making 100 lines, thus, gives 100 calculations of the same impulse response delayed differently for the different lines. Doing the focusing after this point in the simulation can make the calculation faster. This corresponds to full synthetic aperture imaging. The received response from each element is calculated, when emitting with each of the elements in the aperture, and then the responses are subsequently focused. This is the approach taken in this paper using a modified version of the Field II program. A 64 element array, thus, gives 4096 responses. For a 7 MHz 64 element linear array the simulation time for one image line is 471 seconds for 200,000 scatterers on a 800 MHz AMD Athlon PC, corresponding to 17 hours for one image with 128 lines. Using the new approach, the computation time is 10,963 seconds, and the beamforming time is 9 seconds, which makes the approach 5.5 times faster. For 3D images with 64 by 64 lines, the total conventional simulation time for one volume is 517 hours, whereas the new approach makes the simulation in 6,810 seconds. The time for beamforming is 288 seconds, and the new approach is, thus, 262 times faster. The simulation can also be split among a number of PCs for speeding up the simulation. A full 3D one second volume simulation then takes 7,500 seconds on a 32 CPU 600 MHz Pentium III PC cluster.

1 Introduction

The simulation of ultrasound imaging using linear acoustics has been extensively used for studying focusing, image formation, and flow estimation, and it has become a standard tool in ultrasound research. Simulation, however, still takes a considerable amount of time, when
realistic imaging, flow, or 3D imaging are studied. New techniques for reducing the simulation time are, thus, desirable.

The main part of an ultrasound image consists of a speckle pattern, which emanates from the signal generated by tissue cells, connective tissue, and in general all small perturbations in speed of sound, density, and attenuation. The generation of this can be modeled as the signal from a large collection of randomly placed point scatterers with a Gaussian amplitude. Larger structures as vessel or organ boundaries can be modeled as a deterministically placed set of point scatterers with a deterministic amplitude. The relative amplitude between the different scatterers is then determined by a scatterer map of the structures to be scanned. Such maps can be based on either optical, CT or MR images, or on parametric models of the organs. Currently the most realistic images are based on optical images of the anatomy [162]. Blood flow can also be modeled by this method. The red blood cells, mainly responsible for the scattering, can be modeled as point scatterers and the flow of the blood can be simulated using either a parametric flow model [163] or through finite element modeling [164]. The received signal is then calculated, and the scatterers are propagated between flow emissions. The simulation of all linear ultrasound systems can, thus, be done by finding the summed signal from a collection of point scatterers as shown in Fig. F.1. The random selection of point scatterers should consist of at least 10 scatterers per resolution cell to generate fully developed speckle, and for a normal ultrasound image this results in 200,000 to 1 million scatterers. The simulation of the responses from these scatterers must then be done for each line in the resulting image, and the simulation for the whole collection is typically done 100 times with different delay focusing and apodization. This makes the simulation take several days even on a fast workstation.

A second possibility is to do fully synthetic aperture imaging in which the received response by all elements are found, when transmitting with each of the elements in the array. The response of each element is then only calculated once, and the simulation time can be significantly reduced. This is the approach suggested in this paper. A second advantage of such an approach is that the image is focused after the field simulation. The same data can, thus, be used for testing a number of focusing strategies without redoing the simulation. This makes it easier to find optimized focusing strategies.

Figure F.1: Set-up for simulation of ultrasound imaging.
2. Theory

The field simulation must find the received signal from a collection of point scatterers. Using linear acoustics the received voltage signal is [30]:

\[ v_r(t) = v_{pe}(t) \star f_m(\vec{r}_1) \star h_{pe}(\vec{r}_1, t), \]  

where \( \star \) denotes spatial convolution, \( \star_t \) temporal convolution, and \( \vec{r}_1 \) the position of the point scatterer. \( v_{pe}(t) \) is the pulse-echo wavelet, which includes both the transducer excitation and the electro-mechanical impulse response during emission and reception of the pulse. \( f_m \) accounts for the inhomogeneities in the tissue due to density and speed of sound perturbations that generates the scattering, and \( h_{pe} \) is the pulse-echo spatial impulse response that relates the transducer geometry to the spatial extent of the scattered field. Explicitly written out the latter term is:

\[ h_{pe}(\vec{r}_1, t) = h_t(\vec{r}_1, t) \star h_r(\vec{r}_1, t) \]  

where \( h_t(\vec{r}_1, t) \) is the spatial impulse response for the transmitting aperture and \( h_r(\vec{r}_1, t) \) is the spatial impulse response for the receiving aperture. Both impulse responses are a superposition of spatial impulse responses from the individual elements of a multi-element aperture properly delayed and apodized. Each impulse response is:

\[ h(\vec{r}, t) = \sum_{i=1}^{N_e} a_i(t)h_i(\vec{r}_1, t - \Delta_i(t)), \]

where \( a_i(t) \) denotes the apodization and \( \Delta_i(t) \) focusing delay, which both are a function of position in tissue and thereby time. \( N_e \) is the number of transducer elements.

The received signal from each scatterer must be calculated for each new focusing scheme corresponding to the different lines in an image. The resulting rf signal is then found by summing the responses from the individual scatterers using (F.1). The number of evaluations of spatial impulse responses for individual transducer elements is:

\[ N_h = 2N_eN_sN_i, \]  

where \( N_s \) is the number of point scatterers and \( N_i \) is the number of imaging directions. It is assumed that the number of elements in both transmitting and receiving aperture are the same, and that the apodization and focusing are included in the calculation. A convolution between \( h_t(\vec{r}_1, t), h_r(\vec{r}_1, t) \) and \( v_{pe}(t) \) must be done for each scatterer and each imaging direction. This amounts to

\[ N_c = 2N_sN_i \]  

convolutions for simulating one image.

The same spatial impulse response for the individual elements are, thus, being evaluated \( N_i \) times for making an image, and an obvious reduction in calculation time can be gained by just evaluating the response once. This can be done by making a synthetic aperture simulation approach. Here the response on each of the receiving elements from excitation of each of the transmitting elements are calculated. The received responses from the individual elements are beamformed afterwards. Hereby the number of evaluations of the spatial impulse responses is

\[ N_{hs} = N_cN_s. \]
Appendix F. Fast simulation of ultrasound images

The number of convolutions is increased to

\[ N_{cs} = N_s N_e^2 + N_i N_e, \tag{F.7} \]

since all emissions must be convolved with the response from all receiving elements and \( v_{pe}(t) \) must be convolved with the responses. This can be reduced to

\[ N_{cs} = N_s (N_e + \sum_{i=1}^{N_e} i) = 0.5 N_s (N_e^2 + 3N_e), \tag{F.8} \]

if the transmitting and receiving elements are the same, whereby the signal received is the same due to acoustic reciprocity \([165]\), when the transmitting and receiving elements are interchanged. The beamforming is done after the calculation, but this can be done very efficiently as demonstrated in Section 3. The improvement in calculation of responses is given by

\[ I_h = \frac{2N_e N_i N_r}{N_e N_s} = 2N_i \tag{F.9} \]

and for the convolutions

\[ I_c = \frac{2N_e N_i}{0.5 N_s (N_e^2 + 3N_e)} = \frac{4N_i}{(N_e^2 + 3N_e)} \tag{F.10} \]

For a 64 element array and an image with 100 directions, the theoretical improvements are \( I_h = 200 \) and \( I_c = 0.0933 = 1/10.7 \). The efficiency of the approach is, thus, very dependent on the actual balance between evaluating the spatial impulse responses and performing convolutions. A significant speed-up is attained for few elements and many imaging directions, since few convolutions are performed. The balance is affected by the method for calculating the spatial impulse responses. The simulation program Field II \([166, 19]\) offers three different possibilities, which are all based on dividing the transducer into smaller elements. The program uses a far-field rectangle solution \([166]\), a full solution for triangles \([26]\), or a bounding line solution \([28]\). The first solution is very fast, whereas the last two solutions are highly accurate but significantly slower. The choice of method will, thus, affect the balance.

A second aspect, in the implementation of the approach, is the use of memory. The number of bytes, when using double precision data, is

\[ B_y = 8N_e^2 N_r \tag{F.11} \]

where \( N_e \) is the number of samples in the response. For at 64 elements array covering a depth of 15 cm, this gives 625 MBytes at a sampling frequency of 100 MHz. This is clearly too much for current standard PCs and even for some workstations. The simulation must be made at a high sampling frequency to yield precise results, but the data can, however, be reduced by decimating the signals after simulation of individual responses. A factor of 4 can e.g. be used for a 3 or 5 MHz transducer. The memory requirement is then 156 MBytes, which is more acceptable. It is, however, still large, and much larger than the cache in the computer. It is therefore necessary to reduce the number of cache misses. This is sought achieved in the program by sorting the scatterers according to the distance to the array, which gives results that are placed close in the memory. The memory interface of the computer is, however, very important in obtaining a fast simulation.

A significant reduction can in general be attained with the approach as will be shown later, and the method makes it very easy and fast to try out new focusing schemes once the basic data has been simulated. This would demand a full recalculation in the old approach.
3. Examples

All the examples in the following section have been made by a modified version of the Field II simulation system. The parameters used in the simulation are shown in Table F.1.

An artificial kidney phantom based on data from the Visible Human Project\(^1\) has been used as the simulation object. The phantom consists of 200,000 point scatterers within a box of 100 × 100 × 35 mm (lateral, axial, elevation dimension), which gives a realistic size for a full computer simulation of a clinical image. The optical image in Fig. F.2 is used for scaling the standard deviation of the Gaussian random amplitudes of the scatterers. The relation between the gray level value in the image and the scatterer amplitude scaling is given by:

\[
a = 10 \cdot \exp(\text{img}(\vec{r}_k)/100)
\]

where img is the gray-level image data with values from 0 to 127, and \(\vec{r}_k\) is the discrete position

---

Table F.1: Simulation parameters for phased array imaging.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_0)</td>
<td>7 MHz</td>
</tr>
<tr>
<td>(f_s)</td>
<td>120 MHz</td>
</tr>
<tr>
<td>(D)</td>
<td>3</td>
</tr>
<tr>
<td>(h_e)</td>
<td>5 mm</td>
</tr>
<tr>
<td>pitch</td>
<td>(\lambda/2)</td>
</tr>
<tr>
<td>(w)</td>
<td>0.9(\lambda/2)</td>
</tr>
<tr>
<td>(k_e)</td>
<td>0.1(\lambda/2)</td>
</tr>
<tr>
<td>(N_e)</td>
<td>64</td>
</tr>
</tbody>
</table>

Figure F.2: Optical image from the visual human project of a right kidney and a liver lobe.

\(^1\)Optical, CT and MR images from this project can be found at: http://www.nlm.nih.gov/research/visible/visible_human.html
Appendix F. Fast simulation of ultrasound images

Figure F.3: Synthetic ultrasound image of right kidney and liver based on an optical image from the visual human project.

<table>
<thead>
<tr>
<th>$N_s$</th>
<th>$N_e$</th>
<th>Method</th>
<th>Time [s]</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>20,000</td>
<td>32</td>
<td>Line</td>
<td>2944</td>
<td>-</td>
</tr>
<tr>
<td>20,000</td>
<td>32</td>
<td>Synthetic</td>
<td>494</td>
<td>5.96</td>
</tr>
<tr>
<td>20,000</td>
<td>64</td>
<td>Line</td>
<td>6528</td>
<td>-</td>
</tr>
<tr>
<td>20,000</td>
<td>64</td>
<td>Synthetic</td>
<td>1108</td>
<td>5.65</td>
</tr>
<tr>
<td>200,000</td>
<td>64</td>
<td>Line</td>
<td>60288</td>
<td>-</td>
</tr>
<tr>
<td>200,000</td>
<td>64</td>
<td>Synthetic</td>
<td>10972</td>
<td>5.49</td>
</tr>
</tbody>
</table>

Table F.2: Simulation times for scanning the human-liver phantom.

of the scatterer. This scaling ensures a proper dynamic range in the scatterering from the various structures. The resulting image using a synthetic aperture simulation is shown in Fig. F.3.

The simulations have been carried out using a standard PC with 512 MBytes RAM and an Athlon 800 MHz CPU running Linux RedHat 6.2. The various simulation times are shown in Table F.2. These data also include the beam focusing for the synthetic aperture simulation, which took 9 seconds in all cases. It can be seen that the simulation times increase nearly linearly with the number of elements, and linearly with the number of scatterers. The improvement for a phased array image with 128 lines is roughly a factor 5.5 to 6, which lies between the two boundaries given earlier. The actual improvement is dependent on the object size, transducer, sampling frequency, CPU, and memory interface, and the numbers will be different for other scan situations.

A three-dimensional scanning has also been implemented. A two-dimensional sparse array ultrasound transducer was used, and a volume consisting of $64 \times 64$ lines with 200,000 scatterers was made. Simulating one line takes 455 seconds, which gives a full simulation time of 1,863,680 seconds (21 days and 13 hours). Using the new approach the whole volume can be simulated in one pass. This takes 6,810 seconds and the beamforming 288 seconds, which in total gives an improvement in simulation time by a factor of 262. A further benefit is that different focusing strategies also can be tested without a new simulation, and a new volume...
3. Examples

image can then be made in 288 seconds.

A parallel simulation has also been performed using a Linux cluster consisting of 16 PCs with dual 600 Pentium III processors and 256 MBytes of RAM for every 2 CPUs. The scatterers are then divided into 32 files and the simulation is performed in parallel on all machines. The total simulation time is 935 seconds for the 2D simulation and beamforming takes 9 seconds, when using 200,000 scatterers for the phantom. A full simulation of a clinical image, thus, takes 15 minutes and 44 seconds, which is acceptable for iterative work. This should be compared with 16 hours and 45 minutes on one CPU using the line based simulation. This approach can also be employed for the three-dimensional scanning and can reduce the time for one volume to roughly $212+288 = 500$ seconds. Simulating 15 volumes of data corresponding to one second of volumes for a 3D scanner can then be done in 7,500 seconds or roughly 2 hours.

Acknowledgment

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation and by B-K Medical A/S, Gentofte, Denmark.
Experimental ultrasound system for real-time synthetic imaging

Authors: Jørgen Arendt Jensen (1), Ole Holm (2), Lars Joost Jensen (2), Henrik Bendsen (2), Henrik Møller Pedersen (1,2), Kent Salomonsen (2), Johnny Hansen (2) and Svetoslav Nikolov (1)

Published: Proceedings of the IEEE Ultrasonics Symposium, Lake Tahoe, 1999.

Abstract

Digital signal processing is being employed more and more in modern ultrasound scanners. This has made it possible to do dynamic receive focusing for each sample and implement other advanced imaging methods. The processing, however, has to be very fast and cost-effective at the same time. Dedicated chips are used in order to do real time processing. This often makes it difficult to implement radically different imaging strategies on one platform and makes the scanners less accessible for research purposes. Here flexibility is the prime concern, and the storage of data from all transducer elements over 5 to 10 seconds is needed to perform clinical evaluation of synthetic and 3D imaging. This paper describes a real-time system specifically designed for research purposes.

The purpose of the system is to make it possible to acquire multi-channel data in real-time from clinical multi-element ultrasound transducers, and to enable real-time or near real-time processing of the acquired data. The system will be capable of performing the processing for the currently available imaging methods, and will make it possible to perform initial trials in a clinical environment with new imaging modalities for synthetic aperture imaging, 2D and 3D B-mode and velocity imaging.

The system can be used with 128 element transducers and can excite 128 channels and receive and sample data from 64 channels simultaneously at 40 MHz with 12 bits precision. Data can be processed in real time using the system’s 80 signal processing units or it can be stored directly in RAM. The system has 24 GBytes RAM and can thus store 8 seconds of multi-channel data. It is fully software programmable and its signal processing units can also be reconfigured under software control. The control of the system is done over an Ethernet using C and Matlab. Programs for doing e.g. B-mode imaging can directly be written in Matlab and executed on the system over the net from any workstation running Matlab. The overall system concept is presented and an example of a 20 lines script for doing phased array B-mode...
imaging is presented.

1 Introduction

New imaging techniques based on synthetic imaging are currently being suggested and investigated [61, 75]. The methods can potentially increase both resolution and frame rate, since the images are reconstructed from RF data from the individual transducer elements. Hereby a perfectly focused image in both transmit and receive can be made. Research in real time 3D imaging is also underway [92, 1]. The purpose is to make systems that in real time can display a pyramidal volume of the heart, where different slices hereafter can be visualized. These images have a poor signal-to-noise ratio, and several groups are working on employing coded signals to enhance the signal-to-noise ratio [167].

All of the above techniques require digital signal processing on the signals from the individual transducer elements, and in some instances it is also necessary to send out coded signals on the individual elements. For research purposes this can be difficult to attain with commercial scanners, since they are often highly integrated and it is difficult to access individual signals. Programming commercial scanners for new imaging techniques is often also either cumbersome or impossible. It is, thus, beneficial to develop a dedicated research system, that can acquire, store, process, and display ultrasound images from multi-element transducers.

2 System specification

The purpose of the system is to make possible the acquisition of multi-channel data in real-time from clinical multi-element ultrasound transducers, and to enable real-time or near real-time processing of the acquired data. The system will be capable of performing the processing for all currently available imaging methods, and will make it possible to carry out initial trials with new imaging modalities for synthetic aperture imaging, 3D imaging, and 2D and 3D velocity estimation. It is capable of working in a clinical environment to evaluate the performance of various algorithms. The system is specifically intended for research purposes, and is not intended for commercial use.

The function of the system is defined by the different imaging methods for which it can be used. Each of the imaging types will be described and the consequence for the system then given.

**Linear array imaging:** A linear array image is generated by a multi-element transducer with 128 to 256 elements. The beam is moved by selecting e.g. 64 adjacent elements and emitting a focused beam from these. The focusing in receive is also done by a number of elements, and multiple foci are used. Apodization in both transmit and receive are often applied. The focusing delay in both transmit and receive are both less than 40 μs. The number of active elements is usually 32 to 64. The transducer frequency is from 2 MHz to 10 MHz. Imaging is done down to a depth of 30 cm.

The demands on the system is, thus, for 64 channels simultaneous sampling at 40 MHz. The maximum delay in both transmit and receive is 40 μs. The maximum time to sample one line is $2 \times 0.3/1540 + 40 \cdot 10^{-6} = 430 \mu s$ corresponding to 17,200 samples at 40 MHz.
3. System realization

Phased array imaging: The beam is here electronically swept over the imaging area by using a 128 to 256 element array. All the elements are used at the same time, and focusing time delays used are less than 50 µs. The transducer frequency is from 2 MHz to 10 MHz. Investigations are done to a depth of 20 cm.

The demands on the system is, thus, for 128 channels sampling at 40 MHz. The demands on delay, sampling time and storage are the same as for linear array imaging.

Flow estimation, spectrum: Beamforming is done in one direction with either a linear or phased array. The flow signal from blood has 40 dB less power than that from stationary tissue. The dynamic range of the flow signal is 30 dB. The effective number of bits must be 12 or more, when the signals from all channels have been combined. The pulse emitted can have from 4 to 16 periods of the center frequency of the transducer or a coded signal can be employed.

Flow imaging: Imaging is done by pulsing repeatedly in one direction and then change the direction to generate an image. An image can therefore be assembled from up to a 1000 pulse emissions.

Three-dimensional imaging: A matrix element transducer is used with up to 40 × 40 elements. Only some of the elements are used for transmit and receive. The area of the elements is small and pulsing should be done with 100 to 300 volts. Coded signals should be used. Coded pulses with up to 64 cycle periods must be possible with a high amplitude accuracy. This corresponds to emission over a period of 32 µs with a sampling frequency of 40 MHz and an accuracy of 12 bits.

Phasing is done with a delay up to 50 µs, and parallel lines are generated by using parallel beam formers and reusing data from one pulse emission. The system must be capable of reading the data sampled from one elements a number of times, and use different phasing schemes for each cycle through the data.

Synthetic aperture imaging: A standard linear or phased array multi-element transducer is used. Pulsing is done on a few elements and the received response is acquired for all elements. The image is then reconstructed from only a small number of pulse emissions by using the data from all the elements.

This type of imaging needs large amounts of storage and the ability to reuse the data for the different imaging directions. This should be solved by having a multi-processor system connected to the sampling system for storage and image reconstruction.

It must be possible to acquire several seconds of data. Assuming sampling in 80 % of the time at 40 MHz, and 8 seconds of sampling gives a storage need of 256 Mbytes per channel.

3 System realization

The multi-channel sampling and processing system consists of four distinct modules: The transmit unit, the receive/transmit (Rx/Tx) amplifiers, receiver sampling unit, and the sync/master unit. The main blocks are depicted on the drawing in Fig. G.1. The connection to the transducer is through a 128 wire coaxial cable through the Rx/Tx amplifiers. The transmitter sends the signals through the transmit amplifier, and the receiver unit samples the
amplified and buffered signals from the Rx/Tx amplifiers. The sync/master unit holds a crystal oscillator and controls the central timing of the scanning process. The overall operation of the system is controlled through a number of single board PCs in the individual units interconnected through a standard 100 Mbit Ethernet. The waveforms and phasing data are transmitted from the controlling PC to the transmitters and receiver boards. The data from the sampling is processed by FPGAs (field programmable gate arrays), that can be configured for specific signal processing tasks over the net. One Focus FPGA is used for each element and a Sum FPGA is placed for each eight elements. The processed and summed signal can then be routed from Sum FPGA to Sum FPGA. The resulting signal is read by one or more signal processors, that can be connected through serial interfaces capable of transmitting 40 Mbytes per second. Each processor has 6 such links. Data and programs are transferred through these links. The beamformed and envelope detected signal is send via the link channels to the PC for display.

The following paragraphs detail the overall design of the individual boards.

### 3.1 Transmitter

The transmitter is capable of generating an arbitrary transmitted pulse with a bandwidth below 20 MHz. The transmitter consists of 8 transmitter boards each equipped with 16 channels. In total the transmitter controls 128 channels.

A transmitter board consists of two control FPGA's, 16 pulse RAM's, two delay RAM's and sixteen 12 bit digital to analog converters (DAC).

Each of the 16 channels has a pulse RAM memory implemented as a 128 k × 12 bit SRAM, which allows the user to store for instance 32 different pulse emissions of 100 \( \mu \)s duration.

The delay RAM holds the start address of the pulse emission in the pulse RAM and the corre-
sponding delay for each line. The delay RAM is implemented as 32 k × 32 bit SRAM. At the start of each line the pulse emission is delayed according to the delay value for each channel.

3.2 Receiver

The Receiver board is illustrated in Fig. G.2. The board samples and processes 8 analog signals selected from 16 inputs.

The receiver, transmitter and sync/master boards are accessible from a general purpose LINUX based compact PCI single board PC, which controls the different boards.

A 12 bit analog to digital converter (ADC) samples one input channel and the data is temporarily stored in a SRAM buffer. The data in the SRAM buffers are processed by the Focus FPGA using the Focus RAM. The Sum FPGA processes the focused data from the 8 Focus FPGA’s and transfers it to the Analog Devices signal processor (ADSP) or stores it in the storage RAM. The functionality of the individual blocks of the receiver board is explained in further detail below.

Focus FPGA

The Focus FPGA controls the initial storing and processing of the sampled data. The Focus FPGA fetches the sampled data from the SRAM and the corresponding focusing parameters from the Focusing RAM and processes the data before transferring the result to the Sum FPGA.

Two independent memory burst SRAM banks are used to bank switch between the sampled data and processed data. While the sampled data is being written to one of the two banks, the other bank can be read by the Focus FPGA. Each SRAM is implemented as 256 kbytes, which is equivalent to a line length of 3.3 ms sampled at 40 MHz.

The basic focusing algorithm uses a combination of coarse and fine delays. The coarse delay is in steps of the 25 ns sampling interval, and it is implemented as a general table look up address generator. For each sample a 16 bit address index is read from the SDRAM. In this way a random sorting algorithm can be implemented. The fine delay is implemented as a linear interpolation with two signed 10 bit apodization coefficients, which are read from the focusing RAM for each sample.

The Focus FPGA is implemented using a XILINX device from the Virtex family: XCV300 in a 352 pin BGA package speed grade -4. The simple B-mode beamformer described above uses less than 10% of the logical resources of the chip. This makes it possible to investigate hardware implementations of more advanced beamformers including pulse compression, synthetic aperture, and parallel beamforming.

Sum FPGA

The Sum FPGA is used to perform digital signal processing on the 8 channels. The most basic operation is to sum the 8 focused channels. Further, it is used as the gateway between the eight independent sampling channels and the ADSP. The Sum FPGA controls the 2 Gbyte storage SDRAM.

When the focusing is done in the Focus FPGA, the 8 channels are added to the accumulated
Appendix G. Experimental ultrasound system for real-time synthetic imaging

Figure G.2: Main diagram of Receiver board.

sum that is passed to the next Receiver board using a high speed cascade bus connecting the Sum FPGA’s directly with each other. The last Sum FPGA in the chain uses the ADSP link ports to transfer the final result to the multiprocessor cluster.

The Sum FPGA is implemented using a XILINX device from the Virtex family: XCV1000 in
a 560 pin BGA package speed grade -4. The simple design described above uses less than 5% of the logical resources of the chip.

3.3 Sync/master unit

The Sync/master unit controls the timing of the system. A highly stable oven controlled crystal oscillator generates the 40 MHz clock frequency. The clock jitter is below 5 ps. The clock is distributed using coax cables and emitter coupled logic (ECL) in order to minimize jitter. The timing source also transmits a synchronization signal. The receiver and transmitter uses the SYNC signal to start and stop the sampling cycles. An image consists of a number of lines, each with a transmission of a pulse and reception of the echoes. The transmitter and receiver generates an internal LINE signal from the SYNC signal to control the sampling process for the received signal.

4 Programming of the system

From the software point of view, the system consists of several computers that are directly connected to a number of transmitter and receiver boards. The computers are linked by a LAN, and uses Linux as operating system and TCP/IP as the underlying communication protocol.

The software was designed to meet the following requirements:

- Flexibility and ease of use. It is of prime importance that new imaging methods can be quickly implemented with a minimal amount of programming also for new users.

- Data encapsulation. All data for the imaging algorithm are stored in the boards of the scanner.

- Distributed computations. The computers work independently one from another, and each calculates only those imaging parameters concerning the boards plugged in it.

- Portability. The software is written in ANSI C and is platform independent.

The client/server communication model was adopted for the software. The computers controlling the boards run a server program. The server waits for requests coming from the LAN and processes them. The requests can be sent by any client program running on a computer connected to the LAN using the TCP/IP communication protocol.

Figure G.3 shows the client-server model of the software. At start-up the server detects which boards are in the PCI enclosures. The computers can handle any combination of transmitter and receiver boards, plugged into the same PCI back-plane. The server is in idle mode until an event occurs. In the case of a hardware malfunction, the server sends an emergency message to a program, called monitor daemon. Another event is a request by the client. The request can be for transferring parameters to the boards, for performing some calculations, or for retrieving data from the boards to the computer with the user.

The interface to the client program is implemented as a MATLAB tool-box. The function calls are implemented to be as close to the functions in the simulation program Field II [19] as possible. Algorithms created using Field II can thereby easily be tested on the scanner with
only minor changes in the Matlab program. An example for setting the system to perform phased array B-mode imaging is shown below:

```matlab
% Auto-detect and initialize the system
sys_init('auto');
% Set the pulse repetition frequency
sys_set_fprf(fprf);
% Set the sampling range gate in receive
sys_set_sampling_interval(start_depth, end_depth);
% Set the number of scan-lines per frame
sys_set_no_lines(no_lines);
% Define the transducer. Necessary for the delay calculations
tr_linear_array(no_elements, width, height, kerf);
% Do for all lines in the image:
for line_no = 1 : no_lines
    % Set the pulse and the apodization for the current line
    xmt_excitation(waveform(line_no));
    xmt_apodization(line_no,xmt_apo(line_no, : : ));
    rcv_apodization(line_no,times, rcv_apo(line_no, : : : ));
    % Set the focus, defined in 3D coordinates
    xmt_focus(line_no,focus(line_no));
    rcv_dynamic_focus(line_no, theta(line_no), fi(line_no));
end
% Set the time-gain compensation curve
tmg_tgc(tgc_vector);
% Start the continuous imaging process
 tmg_start
```

In order to make the system perform linear array imaging only one line needs to be added, which changes the origin of the individual scan-lines.
5 Conclusion

A system for doing research into new imaging methods in medical ultrasound has been described. The system is capable of emitting complex, arbitrary waveforms, and can sample and store data for all transducer channels in real time. Sufficient storage is included in the system for 8 seconds of data at a sampling frequency of 40 MHz for a 64 element transducer. The system is easily programmable through Matlab and a network interface. It can perform real time processing and image display for all commercially available ultrasound systems and can function in a clinical environment.

Acknowledgment

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation, by B-K Medical A/S, and by grant EF-782 from the Danish Academy of Technical Sciences.
Real time 3D visualization of ultrasonic data using a standard PC

Authors: Svetoslav Ivanov Nikolov, Juan Pablo Gómez Gonzaléz, Jørgen Arendt Jensen

Abstract

This paper describes a flexible, software-based scan converter capable of rendering 3D volumetric data in real-time on a standard PC. The display system is part of a Remotely Accessible and Software-Configurable Multichannel Ultrasound Sampling system (RASMUS system) developed at the Center for Fast Ultrasound Imaging.

The RASMUS system is connected to a PC via the link channels of an ADSP 21060 signal processor. A DMA channel transfers the data from the ADSP to a memory buffer. A software library based on OpenGL uses this memory buffer as a texture map that is passed to the graphics board.

The scan conversion, image interpolation, and logarithmic compression are performed by the graphics board, thus reducing the load on the main processor to a minimum.

The scan conversion is done by mapping the ultrasonic data to polygons. The format of the image is determined only by the coordinates of the polygons allowing for any kind of geometry to be displayed on the screen. Data from color flow mapping is added by alpha-blending. The 3D data are displayed either as cross-sectional planes, or as a fully rendered 3D volume displayed as a pyramid. All sides of the pyramid can be changed to reveal B-mode or C-mode scans, and the pyramid can be rotated in all directions in real time.

The PC used in RASMUS has a 800 MHz AMD Athlon processor and an NVIDIA GeForce2 video card. The resolution is 1280x1024 pixels, 32 bits per pixel. The system can display a B-mode video at 360 frames per second (fps), or it can simultaneously display up to 4 fps. A 3D volume is rendered at 41 fps.
Appendix H. Real time 3D visualization of ultrasonic data using a standard PC

1 Introduction

Modern ultrasound scanners are able to display B-mode (brightness mode) data, combined B-mode data and a sonogram, and/or B-mode data with a superimposed color flow map. The visualization of ultrasonic data involves the following operations:

1. Logarithmic compression.
2. Scan conversion.
3. Display and user interface.

The most computationally extensive operation is the scan conversion. The number of lines is not sufficient and interpolation of the data must be done [13, 14]. Usually specialized hardware for the scanner is designed to speed up the process. Because the implementation is hardware dependent, only a fixed number of imaging modalities are supported. Every new modality requires the introduction of changes in the implementation.

The purpose of this work was to develop a flexible software-based display of ultrasound data. The advantages of using a software solution on a standard PC platform are:

1. Low cost. The ever growing PC market provides inexpensive components. The development is confined only to the parts specific to the visualization of ultrasound data, not to the problem of developing video controllers.
2. Scalability. The actual performance of the system can be scaled according to the needs of the scanner.
3. Reusability. The use of open standards ensures the continuance and reusability of the work already done. Using higher-level, platform-independent solutions ensures that the manufacturer is not bound to specific hardware.
4. Flexibility. A software solution can be easily reconfigured to accommodate different image geometries. The user interface can be based on a standard graphical user interface of a PC.

The idea of using a software solution and standard microprocessor equipment is not new [15]. A pure software solution is computationally extensive and requires a lot of processing power and memory bandwidth. The previous work in scan conversion has been focused on the algorithmic optimization [14, 16, 168]. No matter how efficient one algorithm is, it still requires a lot of processing power of the central processor. This problem has been addressed by the vendors of video cards for the needs of the PC gaming and computer aided design (CAD) industries. Specialized graphics processing units with 256 bit cores and 128 bit wide memory interface easily outperform most of the general-purpose processors in displaying graphics. Hence, a software display using hardware acceleration is capable of not only displaying all of the imaging modalities, but it can render a 3D volume in real time. This can be done by defining the mapping of the coordinates not as look-up tables but as geometric transformation of polygons. A number of libraries and application programmer interfaces (API) are readily available for the purpose. We have chosen to implement the display using OpenGL because many of the video cards have
OpenGL hardware acceleration, and it is available for most operating systems and computer languages.

In the next section the requirements to the program are given. Then the implementation is described in Section 3, and finally the performance is measured and discussed in Section 5.

## 2 System specification

This section describes the requirements on the software display so it can be used with the RASMUS system [41] for all image types. The demands are:

1. The display must be integrated with RASMUS and show images in real time at more than 30 frames per second. This imposes certain requirements on the transfer rate of RF data to the program. Assuming a speed of sound of $c = 1540$ m/s, a pulse repetition frequency of $f_{prf} = 5000$ Hz, a sampling frequency $f_s = 40$ MHz, and an image depth from 0.005 to 0.15 m, the maximum amount of data transferred is 71.8 MB/sec.

2. It should be capable of displaying images saved on a disk for off-line examination. Hence the display part should be separate from the data transferring part. Thus the source of data can be supplied either from the scanner, from a hard disk, or any other source. The implementation must therefore include inter-process communications.

3. Ability to display more than one images simultaneously. RASMUS is an experimental system and the result of different algorithms can then be compared.

4. Both B-mode images and color flow maps must be displayed. Both types of images can be of any geometry - phased array sector images, linear array recti-linear images, linear array tilted images, convex array sector images, 3D images with a pyramidal scan, or 3D images obtained as a rectilinear scan, etc.

5. The implementation must reduce the load on the main processor to a minimum. Thus all the processing related to the image display must be carried out by the video controller.

6. The program must be platform independent and portable. The development must be based on open standards and standard software tools.

## 3 Realization

The whole implementation is software based. The programs are written in ANSI C using the OpenGL library. The reason for using OpenGL is that the interface is a de facto standard, that is stable and fast. A number of implementations exist such as those by SGI and Microsoft as well as the Open Source version represented by the library ”Mesa” (http://www.mesa3d.org). The library is supported under most of the operating systems such as Linux, the various flavors of UNIX and MS Windows. The sources are compiled for most of the hardware platforms and OpenGL oriented drivers are available from the vendors of video controllers. The whole development of the image display was done under Linux, using open source tools.

There are three implementation issues:
1. How to feed data in real-time to the display system PC.

2. How to define the geometry of the image and how to map the data.

3. How to display 3D data in a way that is both familiar to the clinician, and makes it intuitive to manipulate in 3D and relate it to the anatomy.

3.1 Data transfer

The data is fed to the PC display system through a Blacktip PCI board ver 2.0 by BittWare Research Systems from the RASMUS system. The PCI board features an Analog Devices ADSP 21060 processor with 2 MB external RAM. The RASMUS system \[41\] consists of a number of custom made transmitter and receiver boards. Each of the receiver boards contains 8 receive channels. The sampled data from the individual channels is appropriately delayed, summed and passed to the next board for cascade summing. The last of the receiver boards hosts an ADSP 21060 which is connected via two of link channels to the ADSP on the Blacktip board in the PC display system. The transfer rate is 80 MB/sec. The PLX PCI 9060 PCI controller of the board transfers the data from the RAM memory on the board to the memory of the computer. Working as a bus master it is capable of a theoretical peak transfer rate of 128 MB/sec. The control of the Blacktip board is performed by custom-made software and drivers. The driver reserves the last 4 MB of the physical memory of the computer and uses them for two buffers for the DMA transfer. The buffers are swapped - while one is being filled in with new data, the other is being used as a texture map by the visualization program, which maps the buffers in its memory space.

3.2 Display of phased array images

Fig. H.1 illustrates the scan conversion geometry. The phased-array data are acquired as a function of angle \(\theta\) and depth \(r\). It is mapped to the screen Cartesian coordinates \(x, y\), along the horizontal and vertical directions respectively as shown in Fig. H.1 (a). Traditionally this is done by using either look-up tables or some modified algorithm for line drawing. Modern video controllers, such as the graphical processing unit GeForce 256 by NVIDIA (http://www.nvidia.com) used in the display system, can render up to 15 million polygons per second. The gaps in the formed image are filled by interpolation. It has been found that bilinear interpolation \[14\] is a good trade-off between quality and computational cost. Modern accelerated video-cards support bilinear and trilinear interpolation as a standard hardware accelerated feature. Thus, the problems associated with the interpolation are taken care of by the video controller without any software overhead. Having this in mind it is quite natural to let the specialized hardware take care of the video display. The only problem that the developer must consider is how to express the mapping of the data.

The approach used in this work is to split the the input data and the output image into convex polygons. The envelope detected lines are stacked together as shown in Fig. H.1 (a) with dots. Each column corresponds to one scan line in the image. The arcs of the output image are approximated by a number of segments, and the whole image is represented by a number of polygons shown in the figure with dashed lines. The segments approximating the arcs have length equal to \(L \approx r \cdot (\theta_{\text{max}} - \theta_{\text{min}})/N\), where \(N\) is the number of segments and \(r\) is the radius of the respective arc. The bigger the number of polygons the better the approximation of the
3. Realization

Figure H.1: Mapping of ultrasonic data from polar to Cartesian coordinate system.

Figure H.2: Phased array image.

output image is. Such division corresponds to equal in size rectangles in the \((r, \theta)\) space of the acquired data (The lateral size of these rectangles, \(\Delta \theta\), can be different than the actual angle step used for the scanning). The only task for the software is to specify the mapping from the normalized coordinates in the data space to the coordinates of the polygons comprising the scan-converted image as shown in Fig. H.1(b) and (c). The normalized coordinates of the four vertexes \(A, B, C, D\) of each of the rectangles from the data space are mapped to the coordinates of the four vertexes \(A', B', C', D'\) of the corresponding tetragon in the image space. Fig. H.1(b) and (c) show the cases when the start scan depth is 0 and bigger than 0, respectively. In OpenGL
Appendix H. Real time 3D visualization of ultrasonic data using a standard PC

Figure H.3: Displaying data parallel to the transducer surface: c-scan (left) and a cross-section with a plane (right).

The data are treated as a textures mapped onto the polygons. The choice of the number of tetragons forming the image has a major impact on the appearance of the image. As a rule of thumb the number of polygons should be at least twice the number of acquired scan lines. This ensures that a sample from one scan line is used in several polygons resulting in a gradual transition. A video card like the one used in the display system can render more than 15 million triangles per second so the number of used polygons can be rather high. The size of the image shown in Fig. H.2 is $512 \times 512$ pixels (this is the size of the active area - without the user interface), the width of the sector ($\theta_{\text{max}} - \theta_{\text{min}}$) is $\pi/3$, the number of scan lines is 64, and the number of polygons $N$ used to create it, is $N = 256$.

The user has the ability to zoom in the image. For every new zoom, the coordinates of the displayed polygons are recalculated in order to maintain the quality of the display. The number of displayed polygons is kept constant.

For linear scan images there is no need for going from one coordinate system to another (for example from polar to Cartesian coordinates). In this case only a single rectangle should be displayed on the screen. The only concern of the developer is to ensure a fixed aspect ratio.

The color flow mapping is done by defining a second set of polygons, corresponding to the coordinates of the velocity estimates to be displayed. The display is done by using alpha blending, which is setting a transparency coefficient.

3.3 Display of 3D images

The display software is targeted at real-time 3D display for volumetric data acquired at a rate higher than 10 volumes/sec [6]. The acquisition of volumetric data in real-time can be done using a 2D matrix arrays. The data in this case is in polar coordinates and must be converted to Cartesian ones. The display must at the same time be both familiar to the medical doctor and giving orientation in the 3D volume. It was decided that the best display for the purpose would be the visualization of a solid with the geometric shape of the scanned volume as the one shown in Fig. H.3. The faces of the solid are rendered with the envelope detected data.
The clinician is able to view the data inside the volume by moving in and out the bounding planes. If the top and the bottom (the faces parallel to the transducer surface) are chosen to be planar, then the speckle exhibits radial symmetry as seen from the right image in Fig. H.3. This radial symmetry is an artifact from the interpolation. Instead the program allows the clinician to change the start and end depths of the displayed volume. The browsing of data in the elevation and azimuth directions is done by changing the start and end angles. These operations are illustrated in Fig. H.4. An example of the rendered volume is given in Fig. H.5. The same display method as presented in Section 3.2 used with a 3D orientation of the polygons. The zooming in the volume is implemented by changing the position of the view camera. Because not all of the sides of the volume are facing the user, only those that are actually visible are passed for rendering, thus reducing the overall data stream.

4 Performance

Two aspects of the system are of interest: (1) the ability of interfacing it to a real-time acquisition hardware, (2) the ability to display data in real time. The tests are based on measured performance rather than estimates of the computational cost.

The PC on which the development and tests were done comprises an ASUS K7V motherboard with VIA KX133 chip set and 800 MHz AMD Athlon processor. The video card is an ASUS V6800 with an NVIDIA GeForce 256 GPU and 32 MB DDR. The software was tested under a
standard Red Hat 6.2 distribution of Linux. The X server is XFree86 (http://www.xfree86.org), version 4.0.1 with drivers from NVIDIA.

The sustained transfer rate of data from the memory of the Blacktip board to a shared memory buffer in the PC (including the latencies of interrupt handling, task switching, display of information, etc.) is 76 MB/s, sufficient for the real-time connectivity of the system.

The performance of the visualization software was tested for the display of a phased array sector image and a linear array rectangular image. The size of the input data set was the same \((1024 \times 64)\). The display mode was set to \(1280 \times 1024\) pixels, 32 bits per pixel. Table ?? shows the speed of display for a window with size \(800 \times 600\) and \(1280 \times 1024\) pixels (full screen). The size of the displayed image is \(512 \times 512\) and \(860 \times 860\) pixels, respectively. For the cases of more than one views, the images had the same type of geometry either sector (pie) or rectangle, since these exhibit the extremities in performance. The cases of mixed geometries have performance ranging between these extreme cases. The speed of display scales with the size of the image that must be rendered and the size of the data that must be transfered to the memory of the graphics board. This is quite clearly manifested in the speed for the 3D display, in which the change in the number of displayed polygons almost does not affect the display.

5 Conclusion

This paper showed that it is possible to use a software display program based on standard software tools and standard PC hardware to scan convert, interpolate and display 2D and 3D ultrasound images in real time. Taking advantage of the hardware acceleration of OpenGL up to 360 phased array B-mode images and 41 3D volumes per second can be displayed which is sufficient for all clinical applications in medical ultrasound. All the processing is done by the video controller, thus, reducing the load on the main processor, which is used for other tasks such as envelope detection, user interface and control.

6 Acknowledgements

This work was supported by grant 9700883 and 9700563 from the Danish Science Foundation, and by B-K Medical A/S, Gentofte, Denmark.
Velocity estimation using synthetic aperture imaging

Authors: Svetoslav Ivanov Nikolov and Jørgen Arendt Jensen

Published: This paper will be presented at the IEEE Ultrasonics Symposium Atlanta, 2001, and published in the proceedings of the conference. The work on the article is still in progress. Only the abstract is given.

Abstract

In a previous paper we have demonstrated that the velocity can be estimated for a plug flow using recursive ultrasound imaging. The approach involved the estimation of the velocity at every emission and using the estimates for motion compensation. An error in the estimates, however, would lead to an error in the compensation further increasing the error in the estimates.

In this paper the approach is further developed such that no motion compensation is necessary. In recursive ultrasound imaging a new high resolution image is created after every emission. The velocity was estimated by cross correlating RF lines from two successive emissions n and n+1 and then average over a number of lines. In the new approach images n and n+N, n+1 and n+N+1 are cross correlated, where N is the number of emissions for one image. These images experience the same phase distortion due to motion and therefore have a high correlation without motion compensation. The advantage of the approach is that a color flow map can be created for all directions in the image simultaneously at every emission, which makes it possible to average over a large number of lines. This makes stationary echo canceling easier and significantly improves the velocity estimates. Simulations using the Field II program yielded a bias of the estimate of -3.5 a mean standard deviation less than 2.0 velocity profile.

The method was investigated using our real-time experimental system RASMUS using a B-K Medical 8804, 7.5 MHz linear array. The pulse repetition frequency was 7 kHz. The method was applied on a flow phantom made using 100 um polymer spheres dissolved in water, driven by a SMADEGARD Type EcoWatt 1 pump generating a peak velocity of 0.42 m/s. The mean velocity was determined by a MAG 1100 flow meter by Danfoss. The length of the tube was 1 m, allowing for a parabolic profile to develop. The number of color flow lines per frame was 90, the number of color flow maps per second was 7000, the bias of the estimate was -7 % and the mean standard deviation was 1.9 %.

The system is capable of acquiring in-vivo images and these will be presented.