Fresnel Zone based Frequency domain reconstruction of Ultrasonic data- Fresnel SAFT
Aswath Rangarajan

Abstract

An Ultrasound Synthetic Aperture Imaging method based on the Fresnel Zone concept is presented in this paper. Fresnel Zones are frequency specific and vary with the position of the point under consideration. In the approach presented, a method of improving the defect reconstruction using the focusing effect of the Fresnel Zone is shown. The reconstruction is compared to the normal Frequency Domain Synthetic Aperture Focusing (F-SAFT). The experiment was done on an Aluminium block, and the imaging of defects of size 1.5mm diameter has improved using the Fresnel Zone approach in comparison with F-SAFT. The time taken for the image reconstruction is considerably large while using the Fresnel zone approach when compared with the F-SAFT algorithm.
Introduction

The Fresnel SAFT method is based on the Frequency Domain SAFT. Synthetic aperture focusing technique is a method of reconstructing the image from data typically obtained in the pulse echo mode. The SAFT algorithm can be implemented in the Time domain or in the Frequency domain. The time domain reconstruction can be done in the pixel approach (where each pixel is considered to be the defect and reconstruction is effected). This approach is also known as the ‘delay and sum’ method. The Frequency domain SAFT is carried out using the back-propagation of the wave fronts in Fourier Domain. The back-propagation of the Fourier domain data is effect using the relations which satisfy the Helmholtz equation. The basic assumption is that there is no addition of energy into the system during the scattering/propagation process.

In the Fourier domain back-propagation, the spatial resolution of reconstruction (this is different from the resolution of defect with respect to the sizing of defect) is fixed by the spatial sampling of the data obtained, while the resolution of reconstruction along the depth axis can be user defined. However, the resolution and sizing of the defect do not improve beyond a certain level, even with increased resolution of reconstruction.

Description

In a typical Frequency Domain SAFT, the input signals to be Fourier transformed must contain the reflection from a single defect or from defects near each other. Otherwise, the Fourier transform of the entire signal would lead to the frequencies due to the interfaces to be amplified, leading to erroneous results. The data is thus initially gated and outside the Gates, the signals are zero padded to their so as to get a matrix having the size equal to the original data set. The gates can be decided by visual inspection of the input B Scan.

Hence, the Fresnel SAFT we are talking about is a two step process, which involves the gating of the signal around the defective region so as to obtain signals due to certain defects, and then applying the Fresnel SAFT algorithm to it.

The Fresnel SAFT method involves the reconstruction of the data in the Fourier space. The original B Scan is first temporally Fourier Transformed. The Phase and amplitude information of the various components is obtained through this temporal Fourier Transform. Now, the region of interest, i.e., the region where the defect is expected to exist. (As obtained through visual observation of the input B Scan).

The Fresnel SAFT algorithm specified below is then applied to each pixel within the specified region. For each pixel within the region of interest, and for every frequency in the temporal Fourier space, a Fresnel zone can be constructed in space, so as to obtain a constructive interference of the waves of that frequency. Now, for each of the pixels, the data pertaining to the Fresnel Zone obtained using the procedure specified above is Fourier Transformed (making the data at other locations equal to zero, i.e., eliminating those signals which will not constructively add up). The data thus obtained is used in the back propagation, and inverse Fourier Transformed in order to
obtain the reconstructed signals. All such reconstructions (for pixels in the zone specified) are combined to get the final reconstruction.

Construction of the Fresnel Zone:
The Fresnel Zone can be constructed for a given frequency and depth by computing the radii of circles inside which constructive interference occurs (i.e. the path lengths of the waves reaching differs by less than $\lambda/4$, where $\lambda$-wavelength of the wave).

![Fig. Computing the Fresnel Zone. The Fresnel Zone can be computed using the Pythagorean Theorem, with the zones being in-between radii where the path difference to the point differ by less than or equal to $\lambda/4$, where $\lambda$ is the wavelength under consideration.](image)

Frequency Domain SAFT has been documented by Leveresque et al. (1999), and the basic F SAFT procedure has been outlined.

F-SAFT Theory:

Synthetic Aperture Focusing Technique (SAFT) is a method of reconstruction of the defect using signals obtained in the Pulse Echo mode from a series of transducers which are moved in the XY plane. The region over which the data is taken is known as the Synthetic Aperture. The data could be taken along one axis (B Scan- 2D Reconstruction) or along both axis (C Scan- 3D Reconstruction). SAFT can be performed in the time domain as well as in the Frequency domain.

In the time domain, the classic “Delay and Sum” Algorithm is used, wherein the signals are time shifted and added in order to obtain the reconstruction of the defect. The delay and sum algorithm works on the fact that the time taken for the signals to reach the defect would vary for different transducer locations.

In the Frequency Domain, the signals are back propagated in the Fourier Space, using the Angular Spectrum Approach, which is based on the fact that the Wave equation satisfies the Helmholtz equation when there are no external sources of energy. The back-propagation is done using the following equations.

$$\overline{s}(\sigma_x, \sigma_y, z, f) = \overline{s}(\sigma_x, \sigma_y, 0, f) 
\times \exp \left( \pm 2\pi i \sqrt{\left(\beta f_{0}\right)^2 - \sigma_x^2 - \sigma_y^2} \right)$$

Where the $\pm$ signs are for the positive and negative frequencies respectively.
F-SAFT Algorithm:

Values of input Matrix is made zero outside the two specified time gates (as in F SAFT) The Gates can be fixed by inspection of the raw B Scan data.

*For each Depth:*

2D FFT of that matrix is taken and FFT Shifting is performed along both axes.

The back-propagation operator as given in the formula above is then computed for the given depth. The back-propagator values which contain a negative value within the square root (of the above equation), are converted to zero. This is done in order to reduce numerical overflows, as these values are due to the evanescent waves. (However, in the forward propagation models, the removal of the evanescent waves would not be required).

**Back-Propagation Operator:**

\[ \exp \left( \pm 2\pi iz \sqrt{\left(2f/v\right)^2 - \sigma_x^2 - \sigma_y^2} \right) \]

Each frequency component is then convoluted with the back-propagation operator. These values are summed up to obtain a signal in Fourier Space.

\[ \bar{S}(\sigma_x, \sigma_y, 0, -f) \] = \[ \bar{S}(\sigma_x, \sigma_y, 0, f) \]

The above procedure is done for each Frequency and stored in an output matrix, with the data going into the corresponding rows-for each depth.

The data is still in the Fourier Space. However, the deconvolution has occurred in the temporal Fourier Axis through the use of the Back-Propagation operator. Hence the data is Inverse Fourier Transformed with respect to the Spatial Frequency Axis.

End of Algo.
From levesque paper.

Experimental Setup

The experiments were done using the Phased array contact type probe having 32 elements. The sample used is an Aluminium block having a side drilled hole of diameter 1.5 mm. The B Scan was obtained in the Pulse Echo mode. The inter-element spacing between the elements of the phased array is 0.6 mm. The data collation was done using the Omniscan setup, from which the data was extracted. The Central excitation frequency of the transducer elements was 5 MHz (have to check) and the sampling frequency was 25 MHz. The defect was at a depth of 19mm from the surface. The velocity of the wave in the Aluminium block was taken as 6396 m/s in the reconstruction.

Algorithm

Fresnel SAFT Algorithm:

1. Values of input Matrix is made zero outside the two specified time gates (as in FSAFT) The Gates can be fixed by inspection of the raw B Scan data.
2. Depth and spatial positioning of defect is specified.
3. Fresnel ring radii are calculated for each frequency based on defect depth (position along Z), and spatial positioning (position along X).
LOOP FOR FRESNEL SAFT BEGINS

For each Frequency:
4. Input matrix (still a B Scan) is convoluted (multiplied) with the Fresnel matrix for that frequency to get a new input matrix.
5. 2D FFT of that matrix is taken.
6. Row corresponding to frequency is selected and it is stacked up vertically such that the same row repeats from the indices of the GATES (as used in F SAFT), while all other rows are made 0.
7. Now, the resulting matrix is added to the required output matrix (which is initially zeros).

End of Algo.

Results

Fig. Input B Scan obtained using the Phased Array probe.

Fig. Reconstruction using Fresnel idea

Fig. Line Profile at z=0.02035 m
Fig. Reconstruction using F-SAFT

Fig. Line Profile at \( z = 0.02035 \) m

Fig. Comparison of line profiles for reconstructions using various number of Fresnel rings for computation, with image reconstructed using F-SAFT, at a depth of 0.02035 m
The maximum intensity point occurs at 0.0235m depth, and hence is used in the plots.

**Discussion**

The Fresnel SAFT method can be applied taking into consideration various number of rings for the Fresnel Zone. A comparison of the reconstruction using various number of rings is given below.

The reconstruction improves marginally with an increase in the number of Fresnel rings used for the computation.

![Fig. Comparison of line profiles for reconstructions using various number of Fresnel rings for computation, at a depth of 0.0201 m](image1)

![Fig. Comparison of line profiles for reconstructions using various number of Fresnel rings for computation, with image reconstructed using F-SAFT, at a depth of 0.0201 m](image2)
Summary/Conclusion

The image reconstructed using the Fresnel SAFT method has lesser noise in the sides, i.e. a better signal to noise ratio (as can be seen from the figures). The reconstruction takes a considerably longer time (around 3 hrs on an 8 GB, 2 GHz Pentium 4 processor), for a 64X640 data set (input signal). The time required for computation can, however be reduced by optimizing the algorithm. The time taken for F SAFT for the same data set is around 20 secs. The reconstruction efficiency increases marginally when the number of rings are increased upto 3, is the number of Fresnel rings used for calculation. There is no improvement in the reconstruction when the number of Fresnel rings used for the computation is increased from 3 to 4.

The sizing of the defect is done with the 6dB cut-off, i.e. the sizing is done taking the values along the line plot, where the intensity value is half the maximum intensity along the line.

The sizing is done at the depth where the maximum intensity pixel occurs. However, the maximum intensity occurs at a depth of 0.0201m (5.8 % error- defect depth=0.019m) using the Fresnel SAFT algorithm, while it occurs at a depth of 0.02035 m (7.1 % error- defect depth=0.019m) using the F-SAFT algorithm.

The spatial resolution, i.e. the resolution along the axis of scanning is obtained by taking the 6dB limits, and at a depth of 0.0201 m. The defect diameters are found to be 1.2 mm through the Fresnel SAFT method (20% error), while the defect diameter obtained through the F SAFT algorithm is 2.1mm (40% error). Fresnel SAFT gives a better sizing than F-SAFT. The error percentage appears large, but as per previous studies, the SAFT resolution was found to be resolution of $d/2$ where $d$ is the spatial sampling (In the present experiment, $d= 0.6$ mm).

The Fresnel SAFT method would be well suited for materials with high impedance since the focusing effect of the Fresnel Zone would enable a better image reconstruction.

Acknowledgements

We would like to thank Mr. Alavudeen who has helped in acquiring the data from the Phased Array probe.

References


Future Work

The Fresnel SAFT algorithm which has been implemented now is completely in the Fourier domain, i.e. 2D FFT of the data is taken and used for reconstruction. This leads to a dramatic increase in the computation time for improved sampling. However, the same principles can be applied in the quasi Fourier domain, i.e. only the temporal Fourier transform is taken, and the reconstruction for each pixel is done based on the signals from the transducers which fall in the Fresnel zone of that pixel. This approach would lessen the time taken, and may also be as fast as the F SAFT method.

One other Improvement that could be implemented would be to directly compute the waves which would constructively interfere, from the 2D FFT data, and then adding only those waves for any given pixel. This approach also would reduce the time required for computation drastically. (This might not give good reconstruction-Asw)

Code used- Full/Partial

Fresnel SAFT Matlab Code:

```matlab
% Fresnel Saft 09th Dec 2008

clear
c1c
tic

warning off all

% % % % % INPUTS % % % % % %
M=dlmread('1point5mm
sdh_at19mndepth_.04microsecondsampling_point6mmspacestep_PHtool.txt')
;x_sam=.6e-3;
c=6396;
time_step = .04e-6;
% Selecting the GATE for the signal to be F SAFTed
```
\begin{verbatim}
% y_min=100;
y_max=350;
num_fresnel=4; % No of fresnel rings to be considered for the computation

% END OF INPUTS %

%M=M';
n=1:num_fresnel;
x_num=size(M,2);
y_num=size(M,1);
y_sam=c*time_step;
M_input_signal=M;
M_result=zeros(size(M));
M_fresnel_fsaft=zeros(size(M));
M_fresnel2=zeros(size(M));

% Considering the signal only in the required gate
M(1:y_min,:)=0;
M(y_max:siz

N=size(M,1);
fmax=1/time_step;
if mod(N,2)==0
  f=(-N/2:N/2-1)/N;
elseif mod(N,2)==1
  f=(-(N-1)/2:(N-1)/2)/N;
end
temporal_frequency=f*fmax;
temporal_frequency=ifftshift(temporal_frequency);

%M Spatial frequencies: sigma_x
N=size(M,2);
fmax=1/x_sam;
if mod(N,2)==0
  f=(-N/2:N/2-1)/N;
elseif mod(N,2)==1
  f=(-(N-1)/2:(N-1)/2)/N;
end
sigma_x=f*fmax;
sigma_x=ifftshift(sigma_x);

%M Computing the sign for back propagation
sign_backpropagation=temporal_frequency;
sign_backpropagation(sign_backpropagation>=0)=1;
sign_backpropagation(sign_backpropagation<0)=-1;

sigma_x_backprop=repmat(sigma_x,[y_num,1]);
temporal_freq_backprop=repmat(temporal_frequency',[1,x_num]);
k1=sqrt((2*temporal_freq_backprop/c).^2-sigma_x_backprop.^2);
k1(find(imag(k1)))=0;
sign_backpropagation=repmat(sign_backpropagation',[1,x_num]);
back_prop_matrix=sig

y_index=[1:y_num]';
y=y_index.*y_sam/2;
y=repmat(y,[1,x_num])
\end{verbatim}
% LOOPING IN Z VALUES AND X

for x_fresnel_index=25:45
    for y_fresnel_index=y_min:y_max
        tic
        x_fresnel_index=36;
        y_fresnel_index=159;
        x_fresnel=x_fresnel_index*x_sam;
        y_fresnel=y_fresnel_index*y_sam/2;
        zz=y_fresnel_index;
        zz=repmat(zz,[length(temporal_frequency),length(n)]);
        lamda_fresnel=abs(c./temporal_frequency');
        lamda_fresnel=repmat(lamda_fresnel,[1,length(n)]);
        n1=repmat(n,[length(temporal_frequency),1]);
        r1=sqrt((y_fresnel+(4.*n1-3).*lamda_fresnel/4).^2-(y_fresnel).^2);
        r1=round(r1./x_sam);
        r1=[-fliplr(r1),r1]+x_fresnel_index;
        r2=sqrt((y_fresnel+(4.*n1-1).*lamda_fresnel/4).^2-(y_fresnel).^2);
        r2=round(r2./x_sam);
        r2=[-fliplr(r2),r2]+x_fresnel_index;
        r1(find(r1<=0))=1;
        r2(find(r2<=0))=1;
        r1(find(r1>x_num))=x_num;
        r2(find(r2>x_num))=x_num;
        M_fresnel=ones(size(M));
        for cc=1:y_num
            for vv=1:size(r1,2)
                if r1(cc,vv)<=r2(cc,vv)
                    M_fresnel(cc,r1(cc,vv):r2(cc,vv))=0;
                elseif r2(cc,vv)<=r1(cc,vv)
                    M_fresnel(cc,r2(cc,vv):r1(cc,vv))=0;
                end
            end
            M_fresnel(cc,1:min(min(r1(cc,:),r2(cc,:))))=0;
            M_fresnel(cc,max(max(r1(cc,:),r2(cc,:))):x_num)=0;
        end
        for freq_index=1:y_num
            M_fresnel1=repmat(M_fresnel(freq_index,:),[y_num,1]);
            M_fresnel1=M.*M_fresnel1;
            M_fresnel1=fft2(M_fresnel1);
            back_prop_matrix_inloop=back_prop_matrix(freq_index,:)*y_fresnel;
            M_fresnel_dummy=M_fresnel1(freq_index,:).*exp(back_prop_matrix_inloop);
        end
    end
end
M_fresnel2(y_fresnel_index,:) = M_fresnel2(y_fresnel_index,:) + M_fresnel_dummy;

%         y_fresnel_index
%         ttttttttttt=toc

end

end

M_fresnel_fsaft = ifft(M_fresnel2,[],2);

% figure, imagesc(abs(M_fresnel_fsaft))
% figure, plot(abs(M_fresnel_fsaft(159,:)))
% figure, plot(abs(M_fresnel_fsaft(159,:))/max(max(abs(M_fresnel_fsaft))))

save('fresnel_output_multiple_x', 'M_fresnel_fsaft')

ttt=toc;
save('time_taken', 'ttt')

F SAFT Code:

% function saft_algo_asw_mtmr_F_SAFT
% (source_separation, width, x_resolution, num_transducers)
% cd (strcat('width= ',num2str(width),'source_separation= ' ,num2str(source_separation)));
% for n = 0:num_transducers-1
%     number = num2str( n );
%     file = strcat('width= ',num2str(width),'source_separation= ' ,num2str(source_separation),'source_',number,'.mat');
%     load(file);
%     data_total_2(n+1,:) = rcv(n+1).v_sig;
% end
% clear
clc
tic

M = dlmread('1point5mm_sdh_at19mmdepth_.04microsecondsampling_point6mmspacestep_PHtool.txt');
x_sam = .6e-3;
M = data_total_2;
M = M';

% Storing the signal in a separate variable for future reference
M_input_signal = M;

M(size(M,1)+1:2^nextpow2(size(M,1)+1),:) = 0;  % Introducing zeros to make it 2^n points
x_num = size(M,2); y_num = size(M,1);
c = 6396;
time_step = .04e-6;
y_sam = c*time_step;

% beam_div = 20*pi() / 180;

% x_sam_reconstruction = x_sam / x_resolution;
% M1 = zeros(size(M,1), (size(M,2)-1)*x_resolution+1);

% Selecting the GATE for the signal to be F SAFTed
y_min=100;
y_max=350;

% Considering the signal only in the required gate
M(1:y_min,:)=0;
M(y_max:size(M,1),:)=0;

% Taking the first FFT2 - 2 dimensional fft
M_fft2=fft2(M);
M_fft2=fftshift(M_fft2,1);
% frequencies temporal:
num_frequency=size(M,1)/2;
temporal_frequency=(0:((y_num/2-1))/(y_num/2-1))*(1/time_step/2);
temporal_frequency=[-temporal_frequency(length(temporal_frequency):-1:1),temporal_frequency];

% Spatial frequencies: sigma_x
sigma_x=0:((x_num/2-1))/(x_num/2-1)*(1/x_sam)/2;
sigma_x=[sigma_x,sigma_x(length(sigma_x):-1:1)];

% Computing the sign for back propagation
sign_backpropagation=temporal_frequency;
sign_backpropagation(sign_backpropagation>=0)=1;
sign_backpropagation(sign_backpropagation<0)=-1;

M_fsaft=zeros(size(M));
for y_index=y_min:y_max
  y=y_index*y_sam/2;
  for freq_index=1:y_num
    kl=sqrt((2*temporal_frequency(freq_index)/c)^2-sigma_x.^2);
    kl(find(imag(kl)))=0;
    M_fsaft(y_index,:)=M_fsaft(y_index,:)+M_fft2(freq_index,:).*exp(sign_backpropagation(freq_index).*2*pi()*i*y.*k1);
  end
end

M_fsaft=ifft(M_fsaft');M_fsaft=M_fsaft';
M_fsaft=fftshift(M_fsaft);

M_fsaft=ifftshift(M_fsaft,2);
M_fsaft=ifft(M_fsaft,[],2);

figure, imagesc(abs(M_fsaft));colormap(gray(256));
figure,subplot(1,2,1); imagesc(M); colormap(gray(256));
M_fsaft1=M_fsaft/max(max(abs(M_fsaft)));g=figure;
subplot(1,2,2);imagesc(real(M_fsaft1));colormap(gray(256));
toc