

BMED-4962/ECSE-4962 Introduction to Subsurface Imaging Systems

Lecture 11: Pulse-echo Imaging, Fetal Image
Kai E. Thomenius¹ & Badri Roysam²

¹Chief Technologist, Imaging Technologies,
General Electric Global Research Center

²Professor, Rensselaer Polytechnic Institute



Center for Sub-Surface Imaging & Sensing



GE Global Research

de no. 1

Outline of Course Topics

- THE BIG PICTURE
 - What is subsurface sensing & imaging?
 - Why a course on this topic?
- EXAMPLE: THROUGH TRANSMISSION SENSING
 - X-Ray Imaging
 - Computer Tomography
- COMMON FUNDAMENTALS
 - propagation of waves
 - interaction of waves with targets of interest
- PULSE ECHO METHODS
 - Examples
 - Multi-dimensional imaging
- MRI
 - A different sensing modality from the others
 - Basics of MRI
- MOLECULAR IMAGING
 - What is it?
 - PET & Radionuclide Imaging
- IMAGE PROCESSING & CAD

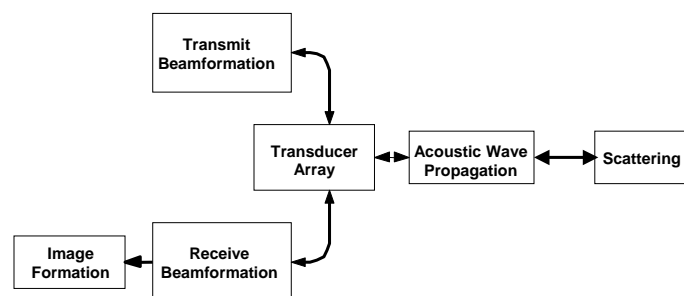
Slide no. 2

Recap of Last Lecture

- We have been introduced to the Field II simulation program and walked through some of the code.
- We have used this Field II example to demonstrate the role of [linear system theory](#) in the understanding of imagers.
- Our goal was to learn about ultrasound scanners as well as the tools we use to understand & design such systems.
- For part 4 of Lecture 10 Homework, download a new version of the m-file ("concave_logo.m") from the course web site.

Slide no. 3

Overall Block Diagram of an Ultrasound Scanner

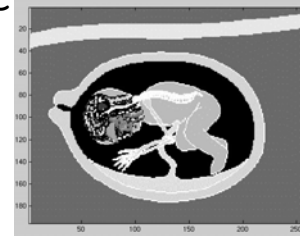
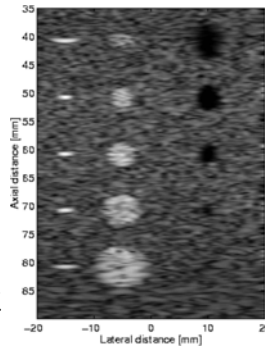


- Beamformation: generation of coordinated timing signals for transmit and delays for receive processes.
- All steps in the above block diagram have well defined transfer functions.
- Our analysis tool, Field II, determines each one of these and determines the image for any desired aperture or scatterer distribution.

Slide no. 4

Acquisition & Processing for a Cyst Phantom Image

- Field II allows one to form an image of a software phantom.
- Jensen *et al.* have generated several such software phantoms.
- We will use the cyst phantom to learn about ultrasound.



Slide no. 5

Code Review: Phantom

- `mk_pht.m`
 - Calls routine `cyst_pht.m`
 - Determines scatterer positions, amplitudes
 - In this case, 100,000 scatterers
- `Cyst_phantom.m`
 - Determines cysts & point scatterers in the phantom
 - Determines scatterer positions

```
% Make the scatterers for a simulation and store
% it in a file for later simulation use

% Joergen Arendt Jensen, Feb. 26, 1997

[phantom_positions, phantom_amplitudes] =
cyst_pht(100000);
save pht_data.mat phantom_positions phantom_amplitudes

function [positions, amp] = cyst_phantom(N)

x_size = 50/1000; % Width of phantom [mm]
y_size = 10/1000; % Transverse width of phantom [mm]
z_size = 60/1000; % Height of phantom [mm]
z_start = 30/1000; % Start of phantom surface [mm];

% Create the general scatterers
x = (rand(N,1)-0.5)*x_size;
y = (rand(N,1)-0.5)*y_size;
z = rand(N,1)*z_size + z_start;

% Generate the amplitudes with a Gaussian distribution
amp=randn(N,1);

Plus a whole bunch of detailed phantom layout type of
stuff ...
```

Slide no. 6

Code Review: Cyst Phantom

- `sim_img.m`
 - Main program
 - Defines:
 - array dimensions, physical constants
 - Field commands for transmit aperture
 - Array impulse response
 - Transmit excitation function



Linear Array Transducer

```
% Generate the transducer apertures for send and receive
f0=5e6; % Transducer center frequency [Hz]
fs=100e6; % Sampling frequency [Hz]
c=1540; % Speed of sound [m/s]
lambda=c/f0; % Wavelength [m]
width = lambda; % Width of element
element_height=5/1000; % Height of element [m]
kerf=0.05/1000; % Kerf [m]
focus={0 0 50}/1000; % Fixed focal point [m]
N_elements=256; % Number of physical elements
N_active=92; % Number of active elements

set_sampling(fs); % Set the sampling frequency

% Generate aperture for emission
xmit_aperture = xdc_linear_array(N_elements, width,
    element_height, kerf, 1, 10, focus);

% Set impulse response and excitation of xmit aperture
impulse_response=sin(2*pi*f0*(0:1/fs:2/E0));
impulse_response=impulse_response.*hanning(max(size(impulse_respo
nse))');
xdc_impulse(xmit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:2/E0));
xdc_excitation(xmit_aperture, excitation);
```

Slide no. 7

Code Review: Cyst Phantom

- `sim_img.m`
 - Main program
 - Defines:
 - Receive aperture characteristics
 - Impulse response of the receive aperture (can be different from transmit)
 - Focal characteristics for dynamic receive focusing
 - Loads the phantom data determined by `mk_pht.m`
 - Defines apodization for transmit & receive operations
 - What is apodization? (HWK)

```
% Generate the transducer apertures for xmt and receive
% Generate aperture for reception
receive_aperture = xdc_linear_array(N_elements, width,
    element_height, kerf, 1, 10, focus);

% Set the impulse response for the receive aperture
xdc_impulse(receive_aperture, impulse_response);

% Load the computer phantom
load /Data/pht_data

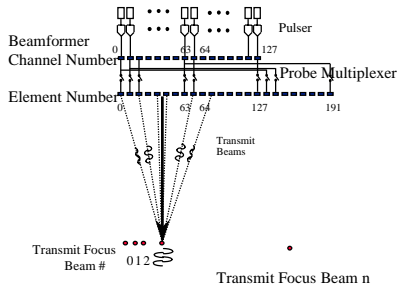
% Set the different focal zones for reception
focal_zones = [30:10:100]'/1000;
NF = max(size(focal_zones));
focus_times = (focal_zones-10/1000)/1540;
z_focus = 60/1000; % Transmit focus

% Set the apodization
apo = hanning(N_active)';
```

Slide no. 8

Code Review: Cyst Phantom

- `sim_img.m`
 - Main loop for each image line



```

% Do linear array imaging
no_lines = 50; % Number of lines in image
image_width = 40/1000; % Size of image sector in mm
d_x = image_width/no_lines; % Increment for image lines

% Do imaging line by line
for i = [1:no_lines]
    x = -image_width/2 + (i-1)*d_x; % The imaging direction

    % Set focus for this direction with reference point
    xdc_center_focus (xmit_aperture, [x 0 0]); % focal refr.
    xdc_focus (xmit_aperture, 0, [x 0 z_focus]);
    xdc_center_focus (receive_aperture, [x 0 0]);
    xdc_focus (receive_aperture, focus_times, [x*ones(Nf,1),
        zeros(Nf,1), focal_zones]);

    % Apodization
    N_pre = round(x/(width+kerf) + N_elements/2 - N_active/2);
    N_post = N_elements - N_pre - N_active;
    apo_vector = [zeros(1,N_pre) apo zeros(1,N_post)];
    xdc_apodization (xmit_aperture, 0, apo_vector);
    xdc_apodization (receive_aperture, 0, apo_vector);

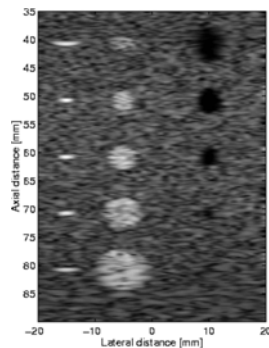
    % Calculate the received response
    [rf_data, tstart] = calc_scatt(xmit_aperture, receive_aperture,
        phantom_positions, phantom_amplitudes);

    % Store the result
    cmd = ['save /Data/sim_cyst/rln_', num2str(i), '.mat rf_data
        tstart'];
    eval(cmd);
end
    
```

Slide no. 9

Code Review: Cyst Phantom

- Image formation
 - Line by line reconstruction



```

for i=1:no_lines
    % Load the result
    cmd = ['load /Data/sim_cyst/rln_', num2str(i), '.mat'];
    eval(cmd);

    % Find the envelope
    rf_env = abs(hilbert([zeros(round(tstart*fs-min_sample),1); rf_data]));
    env(1:max(size(rf_env)), i) = rf_env;
end

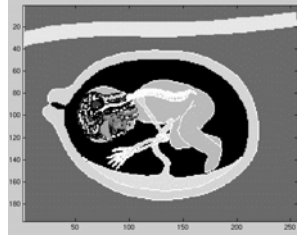
% Do logarithmic compression
D=10; % Sampling frequency decimation factor

log_env = env(1:D:max(size(env)), :)/max(max(env));
log_env = log(log_env+0.01);
log_env = log_env - min(min(log_env));
log_env = 64*log_env/max(max(log_env));
    
```

Slide no. 10

Acquisition & Processing for a Fetal Image

- In addition to the cyst phantom, Jensen *et al.* have generated several software phantoms.
 - Fetus
 - Kidney
 - Blood flow
- We will use the fetal example to learn about ultrasound.



Slide no. 11

Code Review

- We'll walk through some parts of the code.
- The first group defines:
 - Acquisition parameters
 - Simulation parameters

```
% Generate the transducer apertures for send and receive

f0=5e6;           % Transducer center frequency [Hz]
fs=100e6;        % Sampling frequency [Hz]
c=1540;          % Speed of sound [m/s]
lambda=c/f0;     % Wavelength [m]
width=lambda/2;  % Width of element
element_height=7/1000; % Height of element [m]
kerf=0.0025/1000; % Kerf [m]
focus=[0 0 70]/1000; % Fixed focal point [m]
N_elements=64; % Number of physical elements

% Use triangles
set_field('use_triangles',0);

% Set the sampling frequency
set_sampling(fs);
```

Note: The transducer is a 64 element linear array. Code url:
http://www.es.oersted.dtu.dk/staff/jaj/field/examples/ftp_files/fetus

Code Review

- As with previous Field examples, we now define:

- Transmit aperture
- Transducer element impulse response
- The excitation signal, i.e. the signal to be transmitted
- Receive aperture and its impulse response

```
% Define Transmit aperture
xmit_aperture = xdc_linear_array (N_elements, width,
    element_height, kerf, 1, 4, focus);

% Set the impulse response and excitation of the xmit
aperture

impulse_response=sin(2*pi*f0*(0:1/fs:2/f0));
impulse_response=impulse_response.*hanning(max(size(imp
ulse_response)));
xdc_impulse (xmit_aperture, impulse_response);

excitation=sin(2*pi*f0*(0:1/fs:1/f0));
xdc_excitation (xmit_aperture, excitation);

% Generate aperture for reception

receive_aperture = xdc_linear_array (N_elements, width,
    element_height, kerf, 1, 4, focus);

% Set the impulse response for the receive aperture

xdc_impulse (receive_aperture, impulse_response);
```

Slide no. 13

Code Review

- Next, we'll define the phantom.
 - The fetal_data is calculated by an m-file titled 'feu_ph.m'
 - The phantom is composed of 2,000 scatterers
 - Their positions are determined by the feu_ph.m code
- The authors have also developed kidney and blood vessel phantoms.
 - The blood vessels include moving red blood cells

```
% Load the computer phantom

load fetal_data

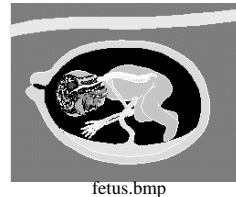
[phantom_positions, phantom_amplitudes] =
    feu_ph(2000);
save fetal_data.mat phantom_positions
    phantom_amplitudes

% Load input baby map (from feu_ph.m code)

[fetus,MAP]=bmpread('fetus.bmp');
% This is cute, we could put any bmp picture in here

% Find the white structures and generate data for
them

Code to associate scatterers with bright areas of
bmp image.
```



Slide no. 14

Code Review

- These steps set up the linear array acquisition
 - No. of acoustic lines
 - Angular increment
- Number of focal zones defined:
 - Their locations
 - When to switch focal zones
- Definition of apodization for both xmit & rcv.
- Transmit focal location

```
% Do linear array imaging
no_lines=128; % Number of lines in image
d_theta=0.7/180*pi; % Angle increment

% Set the different focal zones for
reception
z_rec_zones=[40:10:140]'/1000;
focus_times=(z_rec_zones-5/1000)/c;

% Set Hanning apodization on apertures
apo=hanning(N_elements)';
xdc_apodization(xmit_aperture, 0, apo);
xdc_apodization(receive_aperture, 0, apo);

% Transmit focus
z_focus=70/1000;
```

Slide no. 15

Code Review

- This loop performs the line by line image formation.
 - New xmt & rcv apertures need to be defined for each new acoustic line.
 - calc_scat determines the rf echoes from all the scatterers involved

```
% Do imaging line by line
i_start = 1;
theta = -no_lines/2*d_theta + (i_start-1)*d_theta;

for i=i_start:no_lines
    % Set the focus for this direction
    xdc_focus(xmit_aperture, 0, [z_focus*sin(theta) 0
    z_focus*cos(theta)]);
    xdc_focus(receive_aperture, focus_times,
    [z_rec_zones*sin(theta)
    zeros(max(size(z_rec_zones)), 1)
    z_rec_zones*cos(theta)]);

    % Calculate the received response
    [rf_data, tstart]=calc_scat(xmit_aperture,
    receive_aperture, phantom_positions,
    phantom_amplitudes);

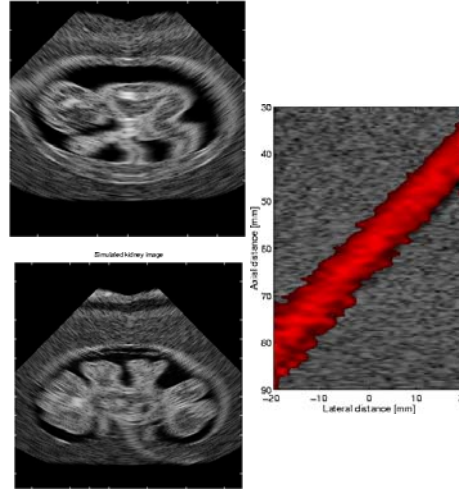
    % Store the result
    cmd=['save /Data/sim_feu/rf_ln',num2str(i),'.mat
    rf_data tstart'];
    eval(cmd)

    % Steer in another direction
    theta = theta + d_theta;
end
```

Slide no. 16

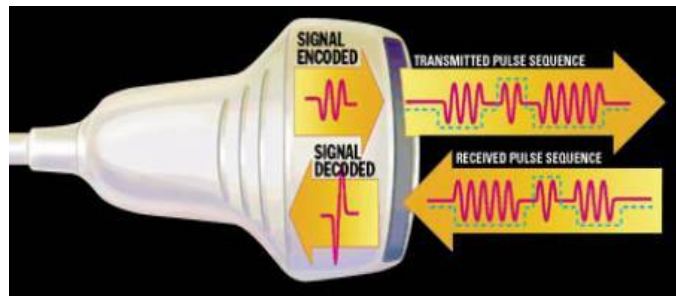
Code Review

- Remaining code involves image formation from the simulated RF lines.
- Other software phantoms include a kidney and a blood vessel with moving scatterers for Doppler processing.



Slide no. 17

Example of Simulation-based Development



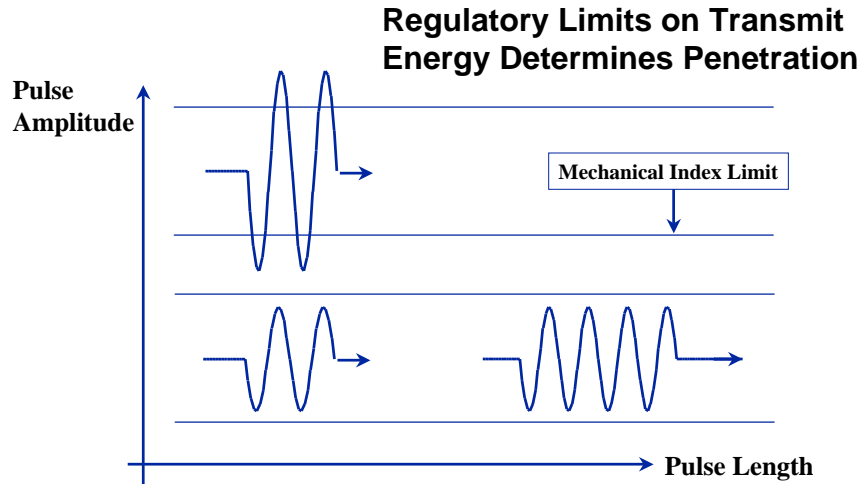
1. Transmit a coded pulse sequence
2. Programmable digital decoder
3. Decode pulse sequence on receive

1. Boost weak signals
2. Suppress unwanted signals or frequencies

Codes boost weak signals or suppress unwanted signals

Slide no. 18

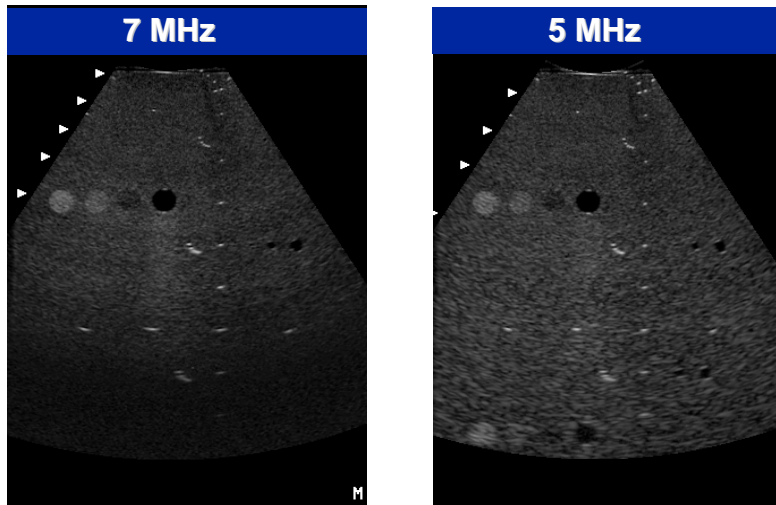
Resolution / Penetration Dilemma



Longer pulse gains penetration but sacrifices resolution

Slide no. 19

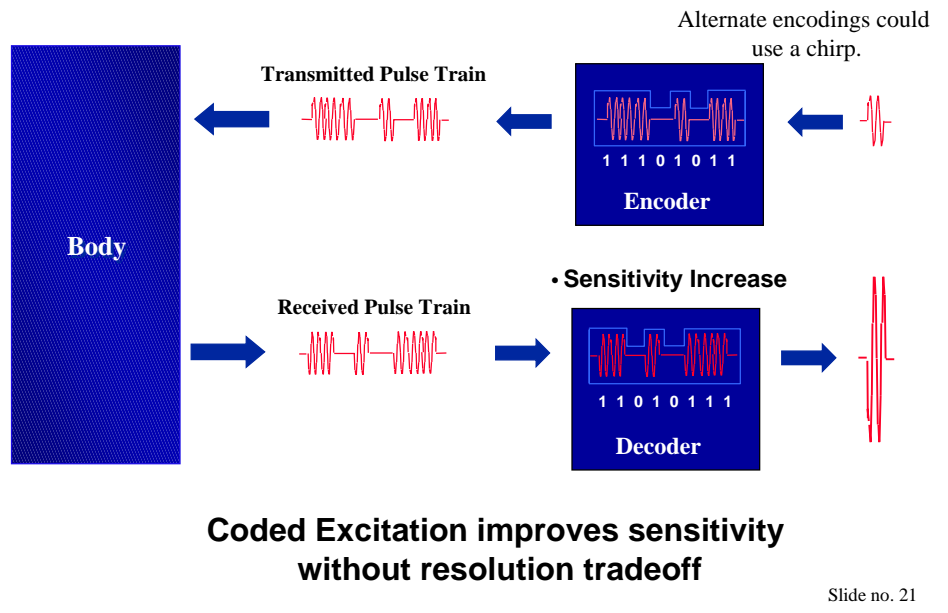
Resolution / Penetration Dilemma



Penetration limited due to Beer's Law

Slide no. 20

Solution: Coded Excitation



Decoding

- Decoding is done with matched or correlation filtering.
- This is often implemented as a convolution.
- Let us assume we transmit a code $p(t)$ and received signal $s(t)$.
- Matched filter can be implemented as the convolution of the received signal by the time-reversed transmit pulse.

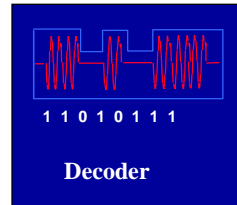
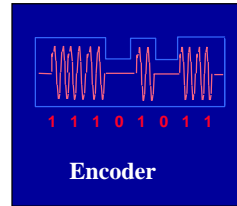
$$R_{sp}(t) = \int_{-\infty}^{\infty} s(\theta)p(\theta+t)d\theta$$

$$s(t) * p(t) = \int_{-\infty}^{\infty} s(\theta)p(t-\theta)d\theta$$

Hardware is readily available to implement convolution based filtering.

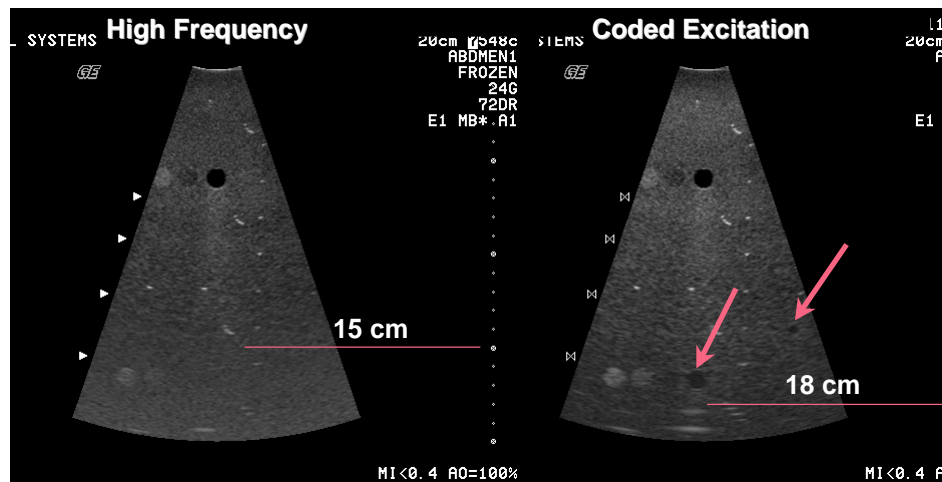
How could we evaluate this w. Field II?

- Encoding step:
 - We need to define an excitation function to represent the coded signal.
 - This can be applied thru the xdc_excitation function.
- Decoding step:
 - Implement the matched filter step in matlab.
 - Apply the resulting signal exactly the same way as any RF echo.
- The rest of the processing is identical to any of the imaging examples given.



Slide no. 23

Coded Excitation - Experiment



Improve penetration by 3 cm with same
resolution to -50 dB

Slide no. 24

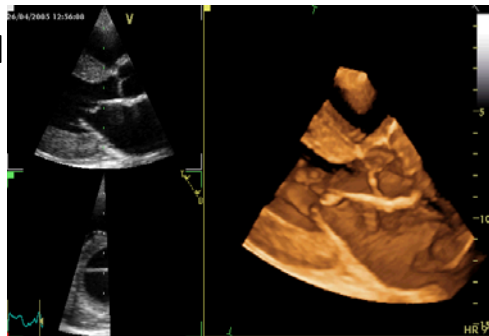
Utility of Tools like Field II

- Here are some applications:
 - Design tools
 - Evaluate an array design without building it.
 - Evaluate signal processing techniques
 - Use the phantoms to compare detectability of use defined lesions
 - Teach image formation techniques
- Field II is rapidly becoming the most commonly used design tool for medical ultrasound.

Slide no. 25



3D/4D Imaging

- In CT, we learned about helical/spiral CT acquisition
 - This meant getting 3D data.
- So what is 4D Imaging?
 - Time is the fourth dimension





Slide no. 26


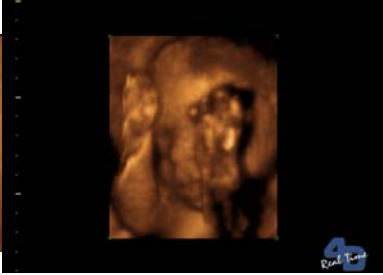
2D

3D

4D

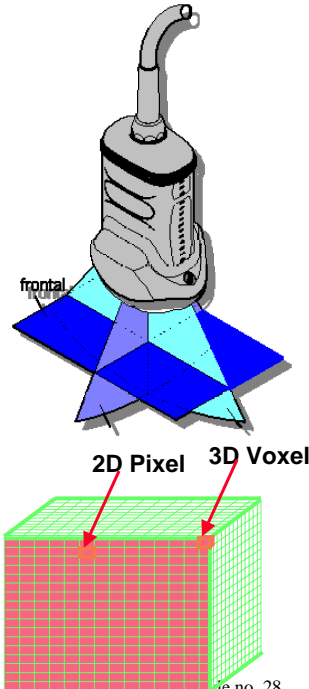
3D imaging in real-time

Slide no. 27

Real-Time 3D

3D imaging:

- Need to move acoustic beams in 3-space
 - Mechanical
 - Electronic
- Multi-planar, real-time display:
 - longitudinal, transversal and horizontal planes
 - Volume or surface rendered 3D image sets
- Volume rate (as opposed to the frame rate) becomes key



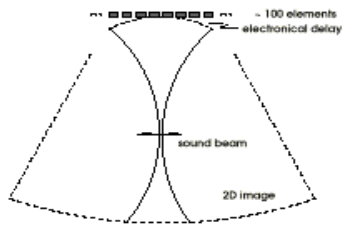
Slide no. 28

Realtime 3D Beamformation

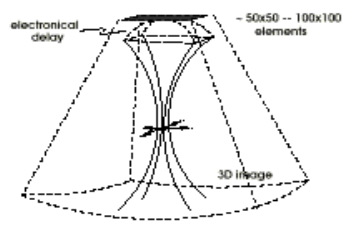


UNIVERSITY OF OSLO

2D image



3D image



DEPARTMENT OF INFORMATICS

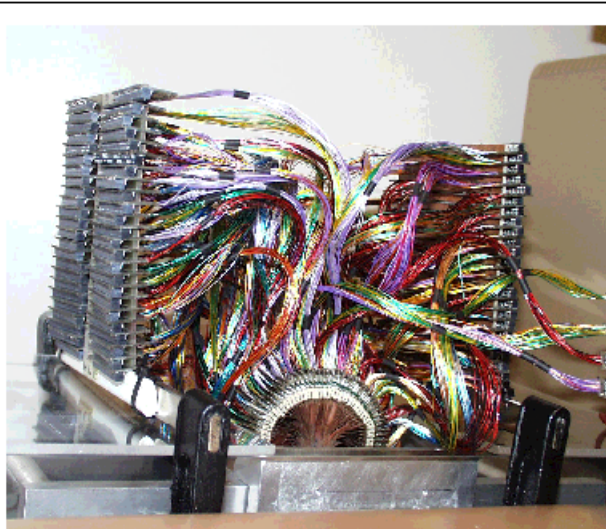
AA: NORSIG-99 2 of 17

Slide no. 29

Fully connected 50 by 50 array



UNIVERSITY OF OSLO

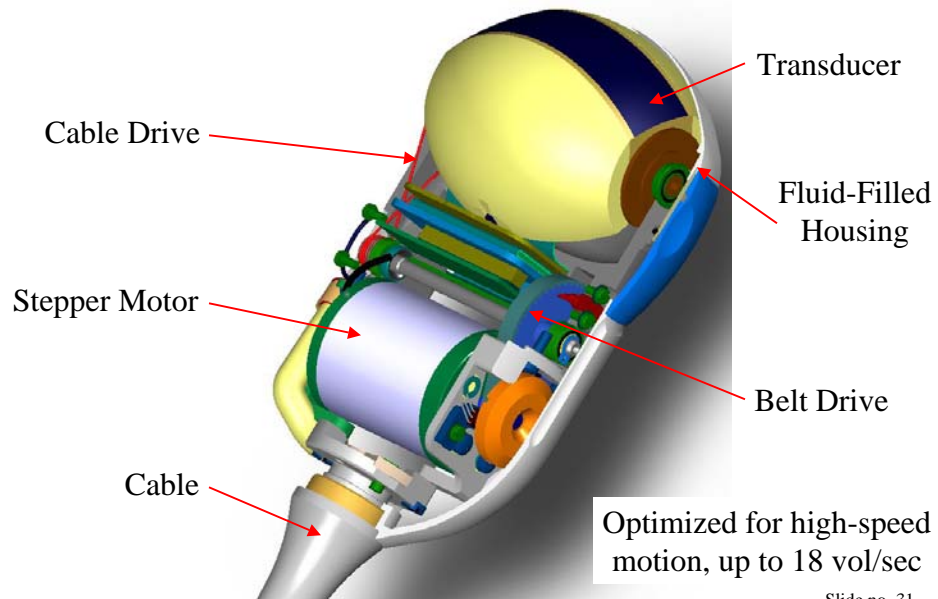


DEPARTMENT OF INFORMATICS

AA: NORSIG-99 16 of 17

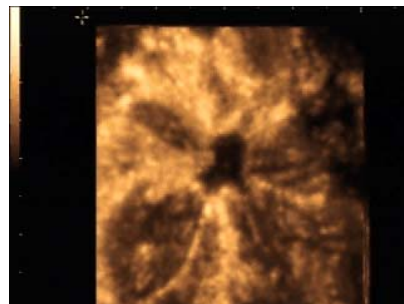
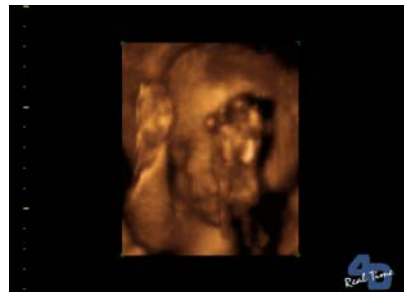
p. 30

Mechanical 4D Solution



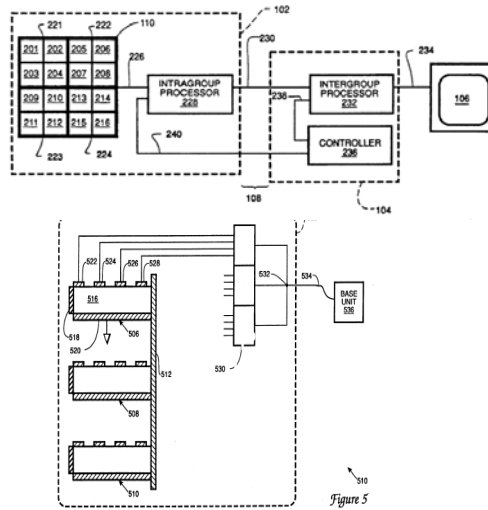
Mechanical Probe Images

- Move a 1D array back and forth rapidly over a sector.
- Volume rates in the order of 15 – 25 vols/sec possible.



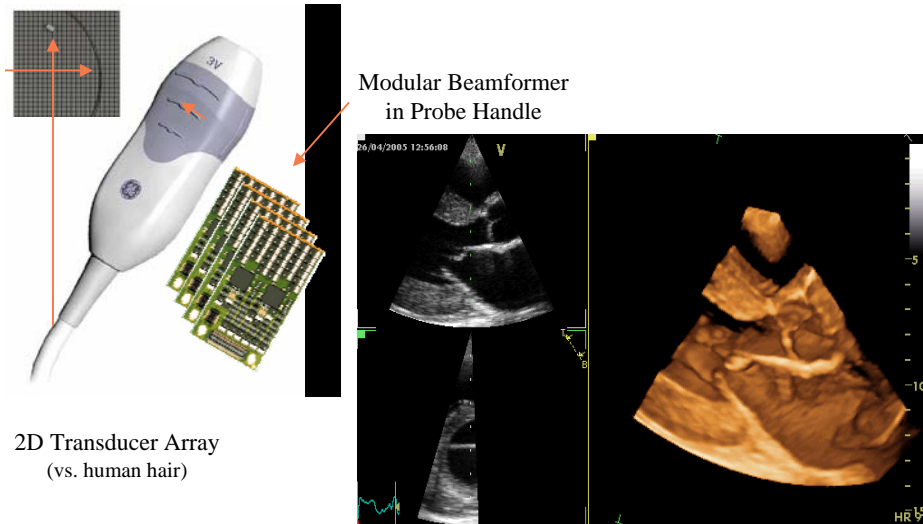
One Possible Solution

- Migrate beamformer components to probe handle.
- With multi-row probes, muxing is in the handle.
- Patent by Larson from 1993
 - group 2D array elements into subarrays
 - combine echoes from subarrays and send summed signals
 - cable count reduced w. reasonable spatial sampling.
- Look for more system changes along these lines



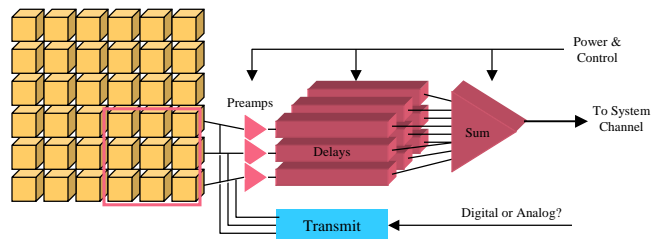
Slide no. 33

Migration of Beamformation to Handle



Slide no. 34

Sub-Array Beamformer in Probe



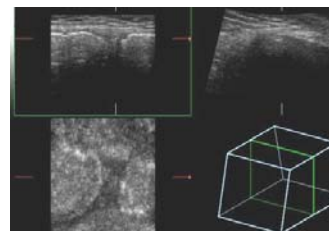
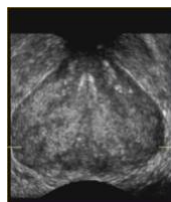
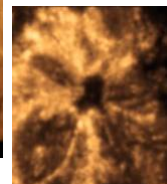
- Connects a group of transducer elements to each system channel
- Low-power analog beamformer: Phase rotation or Delay lines
- Small delays only: static steering of small sub-aperture
- Dynamic focusing & full-aperture delays by system beamformer

Slide no. 35

3D / 4D Applications



- Obstetrics
- Women's Health
 - Virtual Hysteroscopy
 - Coronal pelvic views
 - 3D Breast Imaging
- Musculoskeletal
- Pediatrics
- Urology
- Entire organ scans



New applications evolving w. 3D/4D

Slide no. 36

Homework Lecture 13

- Using identical circular transducers, compare axial responses from O'Neil's expression and a Field II simulation.

- For the Field II case, use:

- Study example for 'Calculation of intensities & peak pressure'
- Matlab code snippet shown to help out.

- How are the results different?

- We discuss apodization both in time domain & spatial domain w. array apertures.

- Why do we use apodization?
- How do we apply it in spatial domain?

```
point=[0 0 0]/1000;  
zvalues=(1:0.1:100)/1000;  
index=1;  
Ppeak=0;  
for z=zvalues  
    point(3)=z;  
    [y,t] =  
    calc_hp(Th,point);  
    Ppeak(index)=max(y);  
    index=index+1;  
end  
% Display of the envelope  
figure(1)  
plot(zvalues, Ppeak)
```

Slide no. 37

Instructor Contact Information



Badri Roysam

Professor of Electrical, Computer, & Systems Engineering

Office: JEC 7010

Rensselaer Polytechnic Institute

110, 8th Street, Troy, New York 12180

Phone: (518) 276-8067

Fax: (518) 276-6261/2433

Email: roysam@ecse.rpi.edu

Website: <http://www.rpi.edu/~roysab>

NetMeeting ID (for off-campus students): 128.113.61.80

Secretary: Laraine Michaelides, JEC 7012, (518) 276 -8525, michal@rpi.edu



Slide no. 38

Instructor Contact Information



Kai E Thomenius

Chief Technologist, Ultrasound & Biomedical

Office: KW-C300A

GE Global Research

Imaging Technologies

Niskayuna, New York 12309

Phone: (518) 387-7233

Fax: (518) 387-6170

Email: thomeniu@crd.ge.com, thomenius@ecse.rpi.edu

Secretary: Laraine Michaelides, JEC 7012, (518) 276 -8525, michal@rpi.edu



GE Global Research



Rensselaer

why not change the world?SM

39