2D beamformation

Lasse Thurmann Jørgensen

Center for Fast Ultrasound Imaging, Department of Health Technology, Technical University of Denmark, Lyngby, Denmark

DTU Health Technology

1

Outline

- **Wave propagation model**
  - Focused/De-focused waves
- **Delay-and-sum**
  - Time-of-flight profile
  - Interpolation
- **Apodization**
  - Fixed
  - Dynamic
- **Break**

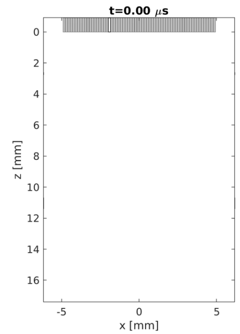
2

Wave propagation model

3

Single element emission

- Transmitted wave is an expanding sphere (circle in 2D)



4

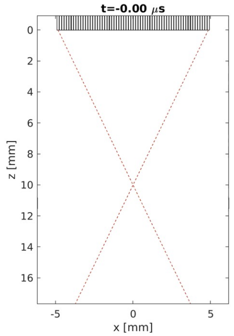
Focused emission

- Waves converge at a focal point
- Focal point behaves as a single transmitting element:

Focal point

=

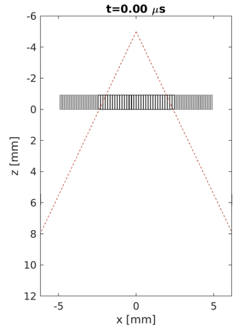
Virtual emission source



5

(De)focused emission

- Diverging wavefront
- Virtual emission source is behind aperture



6

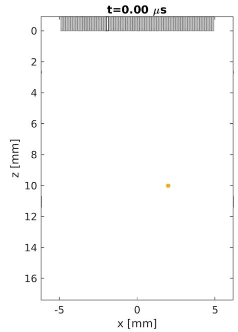
Deriving the "distance-of-flight"

7

Single element emission

Wave propagation path:

Element source
↓
Image point
↓
Receiving element

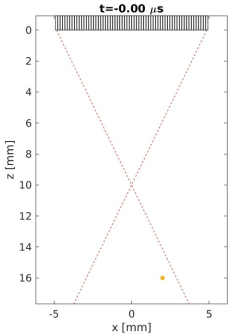


8

Focused emission (Case 1)

Wave propagation path:

Aperture
↓
Virtual source
↓
Image point
↓
Receiving element

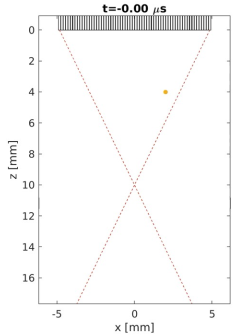


9

Focused emission (Case 2)

Wave propagation path:

Aperture
↓
Image point
↓
Virtual source
↓
Receiving element

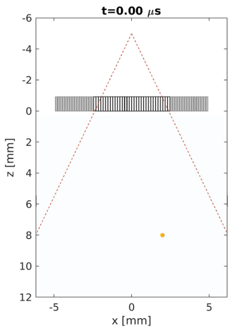


10

(De)focused emission

Wave propagation path:

Virtual source
↓
Aperture
↓
Image point
↓
Receiving element

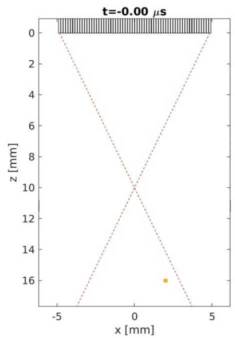


11

When is time equal to zero?

- Can be chosen arbitrarily
- Defined by the zero point of the element's transmit profile
- In this course time is zero when the wavefront hits $(x,z) = (x_v,0)$.

(virt. source's x-position)



12

Time-of-flight

DTU **C**enter for **F**ast **U**ltrasound Imaging

- Generic "Distance-of-flight" equation:

$$D = \left(z_v \pm \sqrt{(x-x_v)^2 + (z-z_v)^2} + \sqrt{(x-x_e)^2 + (z-z_e)^2}\right)$$

- Time-of-flight equation:

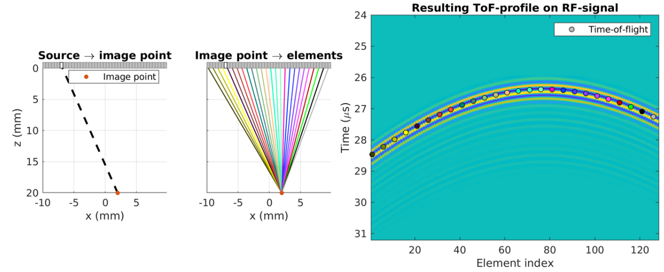
$$t_{ToF} = D/c$$

Virtual source's (x,z)-position: (x_v,z_v)
Receiving element's (x,z)-position: (x_e,z_e)
Speed of sound in the medium: c

Time-of-flight (ToF)

DTU **C**enter for **F**ast **U**ltrasound Imaging

- ToF-profile: ToF to each element



Response delay

DTU **C**enter for **F**ast **U**ltrasound Imaging

- Inherent delay in the RF signal.
- Delay determined using the matched filter:

$$s_h[n] = (s * h)[n], \qquad n = 0,1,\ldots,N-1$$

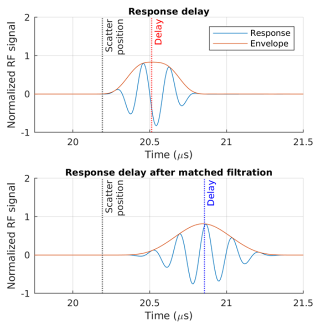
- Before matched filtration:

$$t_{\text{delay}} = \underset{n}{\operatorname{argmax}}(s_h[n])\,\Delta t$$

- After matched filtration:

$$t_{\text{delay}} = (N-1)\Delta t$$

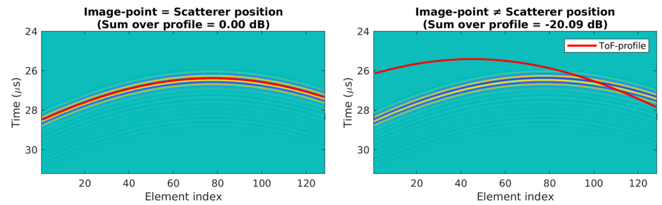
Matched filter: $s_h[n]$
Excitation pulse: $s[n]$
PE impulse response: $h[n]$
Sampling interval: Δt



From ToF-profile to image value

DTU **C**enter for **F**ast **U**ltrasound Imaging

- The sum over the ToF-profile yields the image value



- Repeat for each image-point and emission to obtain the final image

Delay-and-sum (DAS)

DAS equation for each emission:

$$I(x,y) = \sum_{i}^{N} a_i r_i \left(t_{ToF}(x,y,x_{e,i},z_{e,i}) \right)$$

i^{th} apodization value: a_i
i^{th} element position: $(x_{e,i}, z_{e,i})$
RF-signal from i^{th} element: $r_i(t)$
Number of elements: N

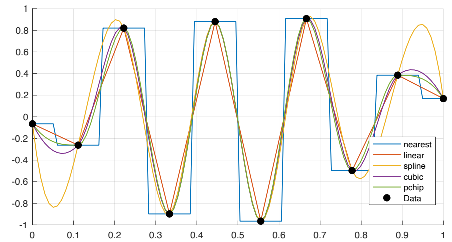
MATLAB code:

```
a = hanning(n_elements); % apodization window
im = 0;                  % image point value
for i = 1:n_elements
    im = im + a(i)*interp1(t, rf_data(:,i), t_tof(i), 'spline', 0);
end
```


18

Interpolation

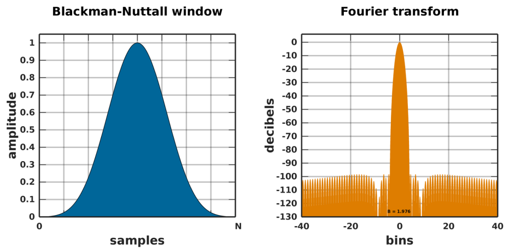
- Good interpolation required for low side-lobe levels
- Avoid:
  - Nearest neighbor
  - Linear
- Preferably use:
  - 3rd order polynomial
  - Spline



19

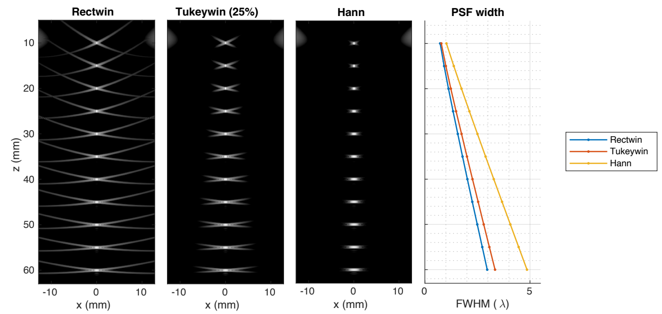
Apodization – Fourier relation

- Side-lobe suppression vs main-lobe width



20

Apodization – Fixed



21

Apodization – Dynamic

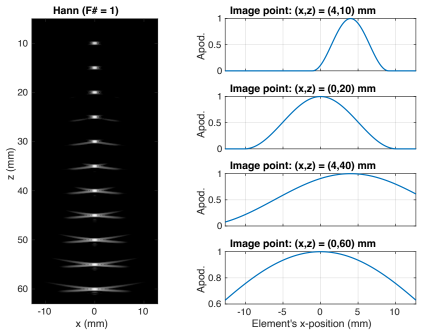
DTU Center for Fast Ultrasound Imaging

- Apodization window changes based on image point position
- Aims to keep F# constant

F# = depth/width

- Makes image quality more consistent

FWHM = F#λ



Apodization – Dynamic

DTU Center for Fast Ultrasound Imaging

Equation (Hann):

$$a_i(x,z) = \cos\left(F\# \frac{\pi(x_{e,i}-x)}{|(z_{e,i}-z)|}\right)^2, \quad \text{if} \quad F\# \frac{|x_{e,i}-x|}{|(z_{e,i}-z)|} < \frac{1}{2}$$

i^{th} apodization value: a_i
i^{th} element position: $(x_{e,i}, z_{e,i})$
F-number: F#

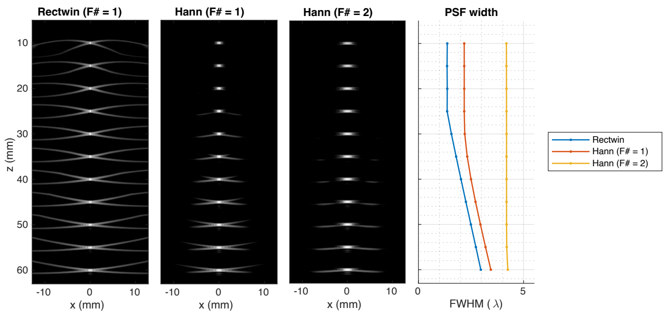
MATLAB code:

```
% Element's (x,z)-position: [xe, ze]
% Image-point's (x,z)-position: [x, z]

xn = linspace(-0.5, 0.5, 64)              % norm. element x-position
xnq = fnum*(xe - x) ./ abs(ze - z);       % scaled and shifted x-position
a = interp1(xn, hanning(64) , xnq, 'spline', 0); % dynamic window
```


Apodization – Dynamic

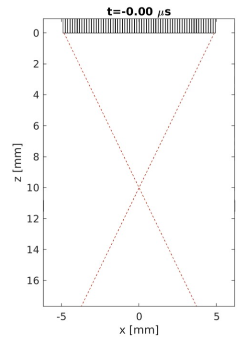
DTU Center for Fast Ultrasound Imaging



Summary

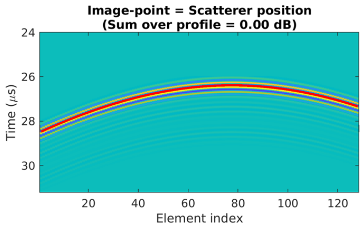
DTU Center for Fast Ultrasound Imaging

- **Wave propagation model**
  - Always one virtual emission source
- **Delay-and-sum**
  - Sum over time-of-flight profile
- **Apodization**
  - Compromise between side-lobes and FHWM



Summary

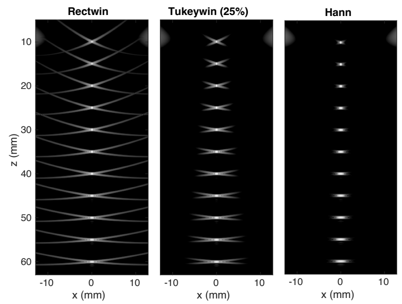
- **Wave propagation model**
  - Always one virtual emission source
- **Delay-and-sum**
  - Sum over time-of-flight profile
- **Apodization**
  - Compromise between side-lobes and FHWM



26

Summary

- **Wave propagation model**
  - Always one virtual emission source
- **Delay-and-sum**
  - Sum over time-of-flight profile
- **Apodization**
  - Compromise between side-lobes and FHWM



27

Thank you for your attention

28