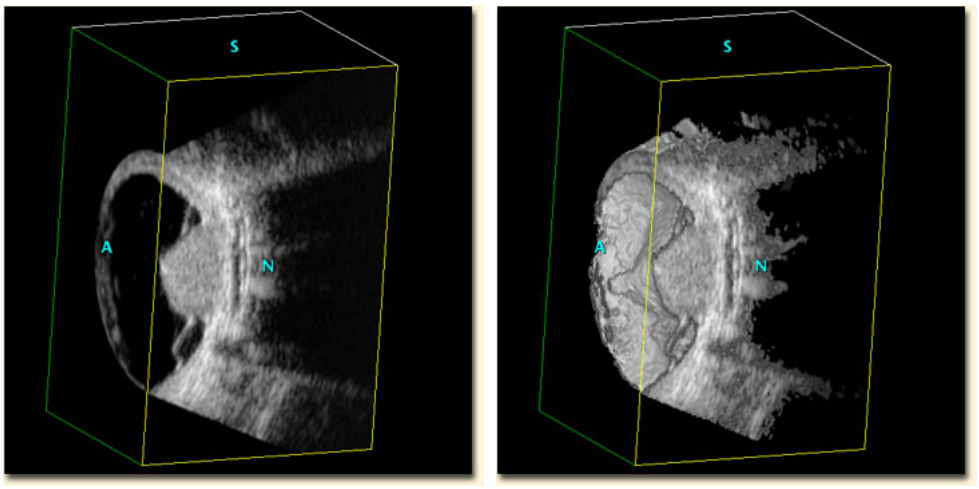
University of Cambridge
Engineering Part IB
Paper 8: Selected Topics

Bioengineering
Imaging the Eye 2: Ultrasound and 3D



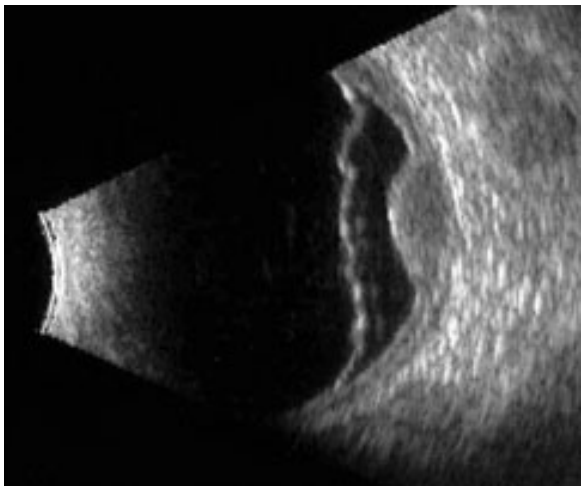
OTI Scan 1000, Advanced Retinal Imaging Laboratory, www.nyee.edu

Graham Treece
April 2024

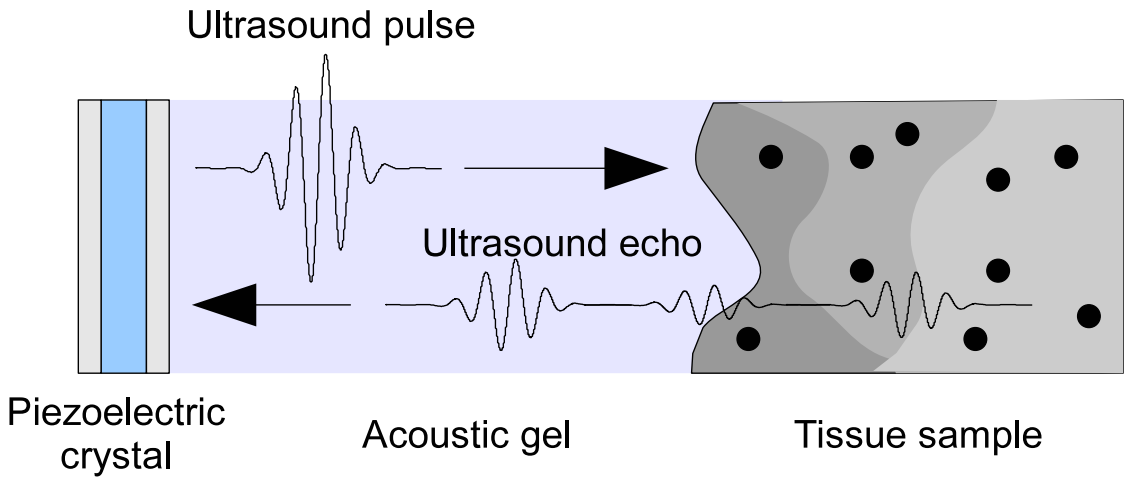
Ophthalmic Ultrasonography

Ultrasound biomicroscopes can be used to scan the anterior eye, but more conventional ultrasound is also used to scan the posterior. As seen above, ultrasound units can be very small.

This is a typical ultrasound image, showing a choroidal melanoma (cancerous growth) and retinal detachment. It is not as good resolution as optical imaging, but it is possible to see deeper into the eye.



Ultrasound imaging



A broadband (high bandwidth) acoustic wave is generated by applying a short duration voltage to a piezoelectric crystal[1], causing it to vibrate. The pulse travels into the tissue, reflects at boundaries and is scattered by very small objects in the tissue.

The ultrasound echo from the tissue in turn causes the piezoelectric crystal to vibrate, which generates an electrical signal. For medical imaging, the signal is in the range 3 MHz to 30 MHz.

[1] Alternatively, it is possible to use Capacitive Micromachined Ultrasound Transducers (CMUTs)

Comparison to optical imaging

So what are the differences between this and optical imaging, which also relies on echos from broadband (low-coherence) pulses?

We will look at various features of ultrasound in comparison to optical imaging, namely:

- What material properties govern how tissues interact with optical and acoustic signals.
- To what extent sound and light is absorbed or attenuated by tissue.
- The relative speed, frequency and bandwidth of the pulses in each case.
- The differences in focusing between ultrasound and optical systems.
- How the respective signals are scanned across the anatomy to generate 2D or 3D data.

Scattering and reflection

Light is:

- Strongly scattered by small objects in tissue.
- Reflected by changes in the refractive index of tissue.

Whereas ultrasound is:

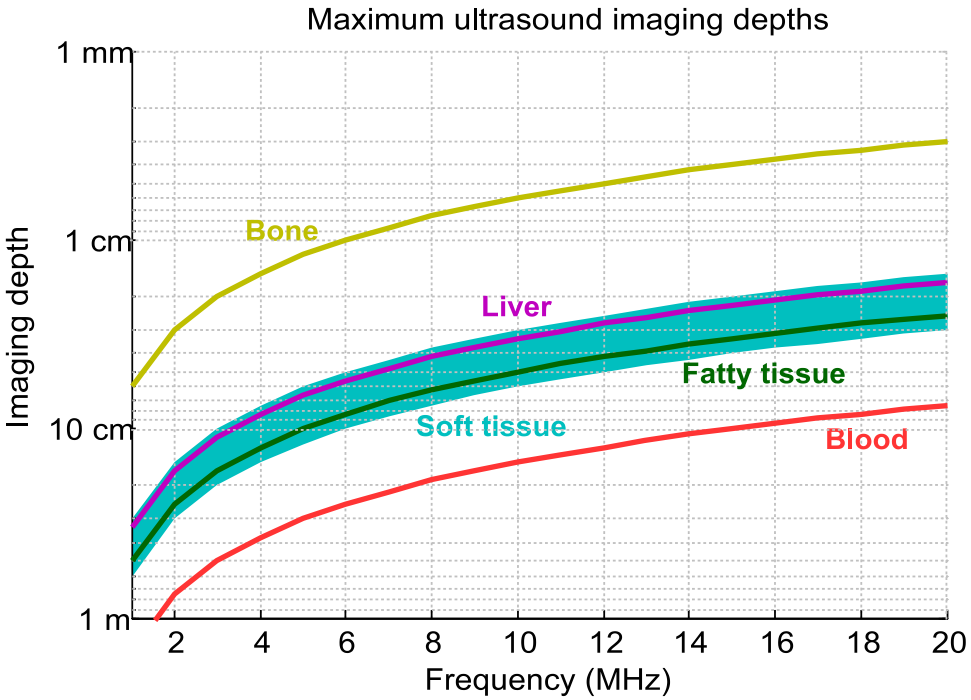
- Weakly scattered by small objects in tissue.
- Reflected by changes in the acoustic impedance.

Both signals have some dependency on material density.

Generally, there is better optical than acoustical *contrast* between different tissue types, hence different properties are clearer in an optical imaging system.

The scattering coefficient is nearly 1000 times less for acoustic than optical signals. This makes the acoustic echo much weaker, but it also minimises problems caused by multiple scattering (when an echo itself scatters off other tissues).

Attenuation



Since ultrasound scattering is so weak, attenuation is mostly due to absorption, and is much less than for optical imaging. Typical ophthalmic ultrasound systems at 10 MHz or higher can image 3 cm into tissue, which equates to at least 1 cm into the retina.

Imaging depth is increased if the ultrasound frequency is reduced. Ultrasound is attenuated very little by water, but dramatically by bone: hence it can 'see' through water but not past bone.

Frequency and bandwidth

Recall that for a Gaussian-modulated broadband pulse, the resolution is given by l_c, the correlation length, as:

$$l_c \approx 0.44 \frac{\lambda_0^2}{\Delta\lambda} \tag{1}$$

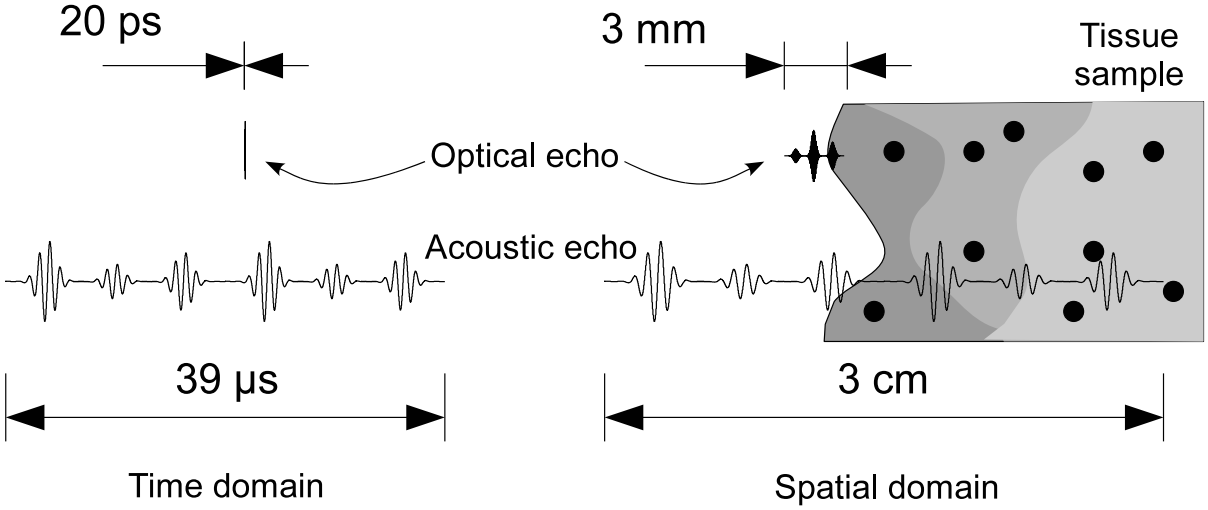
where λ_0 is the centre wavelength, and $\Delta\lambda$ is the bandwidth. Using typical values we have:

Source	Wavelength λ_0 (μm)	Bandwidth $\Delta\lambda$ (μm)	Resolution l_c (μm)
Optical	0.5–1.4	0.02–0.15	1-40
Acoustic	80–500	50-120	100-1000

OCT depth resolution is about 5 μm, whereas ultrasound imaging of the eye can reach 200 μm. This is slightly better than might be expected from the wavelength, since ultrasound transducers can generally produce shorter pulses (relative to the wavelength) than lasers.

Speed

The speed of sound normally assumed for tissue is 1.54×10^3 ms^{-1}, which is far slower than the speed of light in tissue, which is approximately 2.19×10^8 ms^{-1}. This has important consequences for echo measurement.

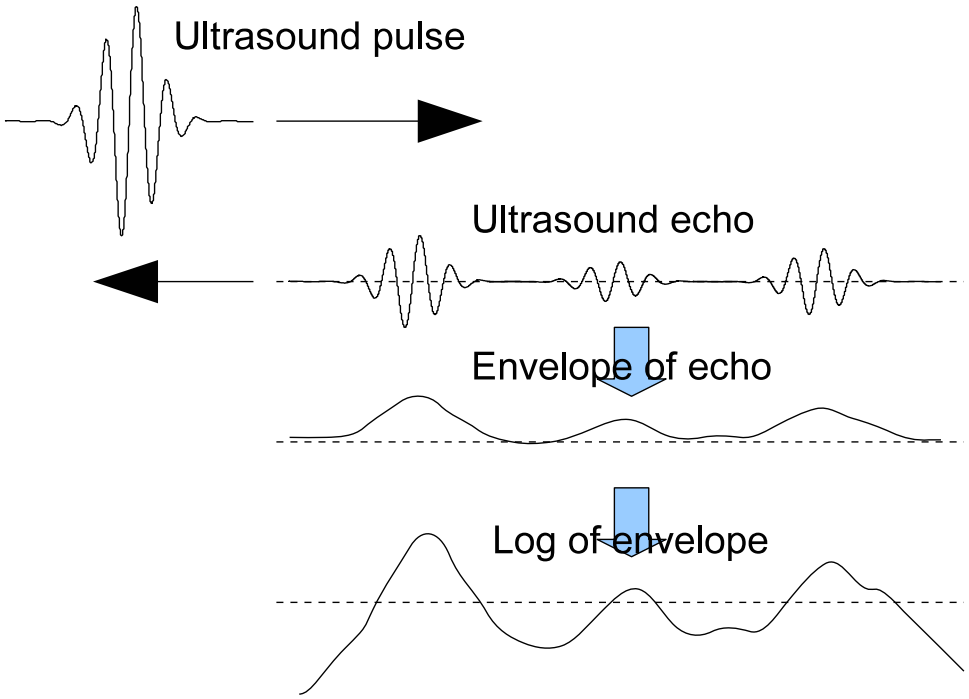


There is only a factor of 10 difference in the spatial extent of optical and acoustic signals in tissue, but there is *a factor of 1000000* difference in the temporal extent of these signals.

Whereas with light we record a single intensity for the echo, sound is slow enough that we can sample the echo *as it arrives*.

Ultrasound acquisition

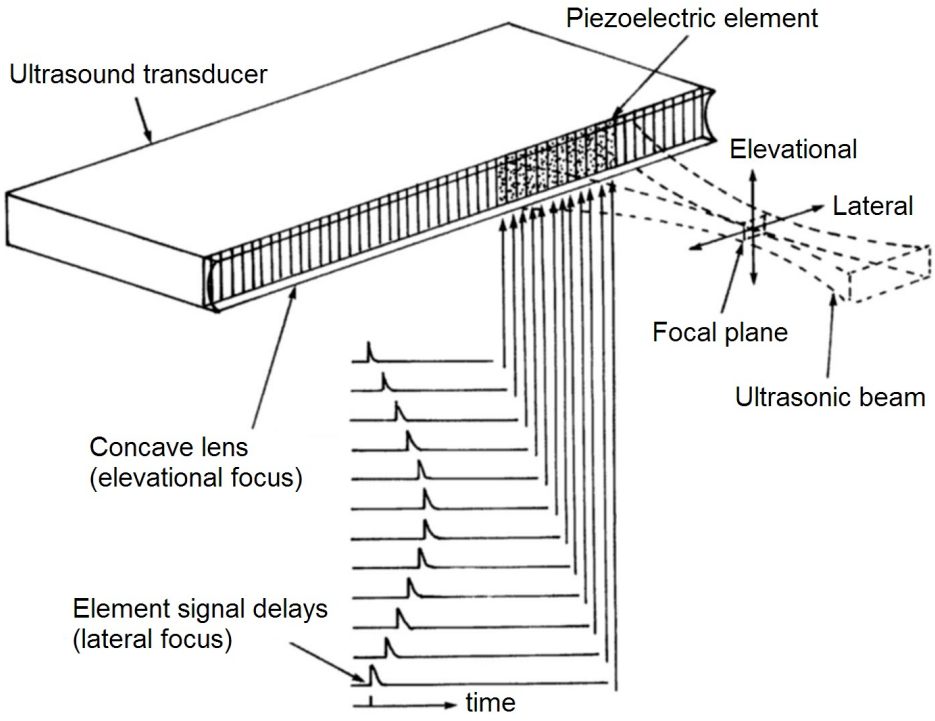
To estimate the acoustic intensity with depth:



Although the signal is actually a function of time, we make it a function of depth by assuming that the speed of sound in tissue is a constant 1540 ms^{-1}. In reality it varies by at least $\pm 5\%$.

For both OCT and acoustic imaging, the *logarithm* of the signal is displayed, since there is useful signal content across quite a wide dynamic range.

Focusing



Adapted from Wells, P.N.T.

Most ultrasound transducers have a linear array of piezoelectric elements. Focusing in the elevational direction is by a lens, just like in optical imaging. Lateral focusing is achieved electronically, by sending acoustic pulses from multiple elements so that they all arrive at the focal point simultaneously.

Lateral and elevational resolution varies a lot with depth, at best it is ≈ 0.5 mm and 1 mm respectively, and never better than the depth resolution.

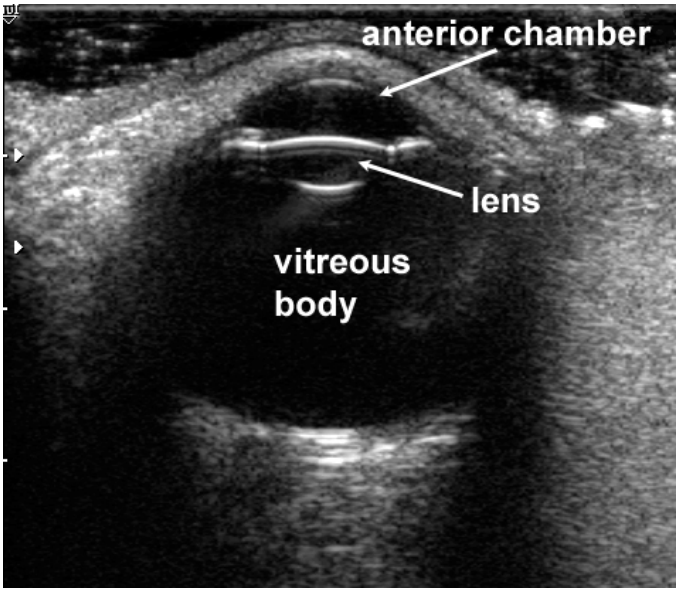
Scanning to create 2D data

For the Scanning Laser Ophthalmoscope and Optical Coherence Tomography there was only one laser source. In order to acquire data from more than one path into the tissue, this source had to be scanned across the subject mechanically.

Most ophthalmic ultrasound probes are linear arrays as on the previous page, and not all the piezoelectric elements are used for the transmission of a single pulse. Hence we can record multiple acoustic paths in the lateral direction by effectively moving the group of elements we are using along by one and transmitting/receiving again.

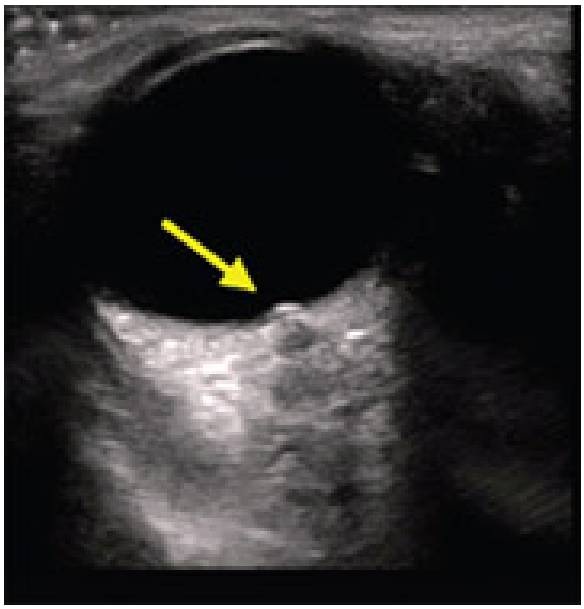
This means that we can create an entire 2D cross-section of the anatomy without actually moving the ultrasound transducer at all. Typically it might take 5 ms to acquire a cross-sectional image of the retina.

Typical ultrasound images of the eye



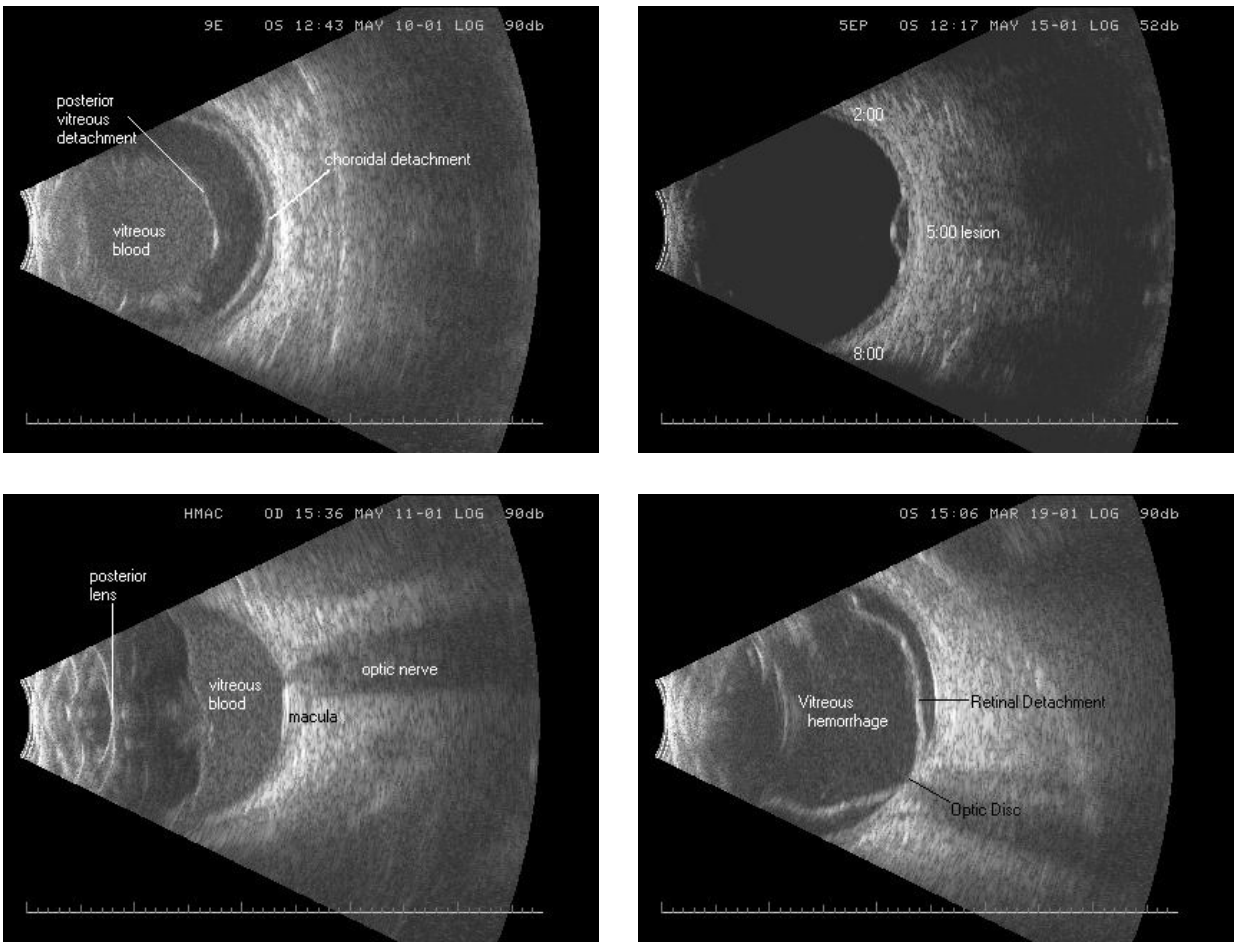
Ultrasound can be used to look at the anterior or posterior segment of the eye. Here is an example showing all of the eye in cross-section. The cornea and lens are particularly visible.

This is an example focusing on the retina, in which the cap of the optic disc can be seen.



Left: by Michael Blaivas, right: from `acep.org`

Typical ultrasound images of the eye



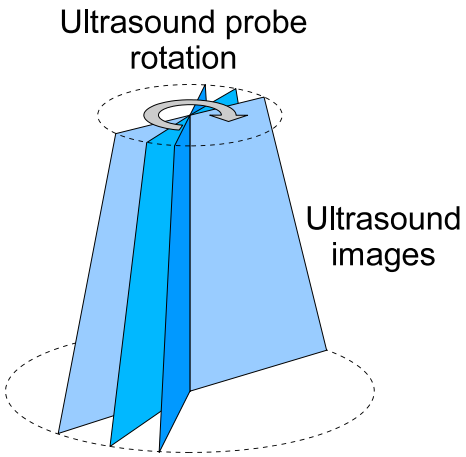
by Rhonda Waldron

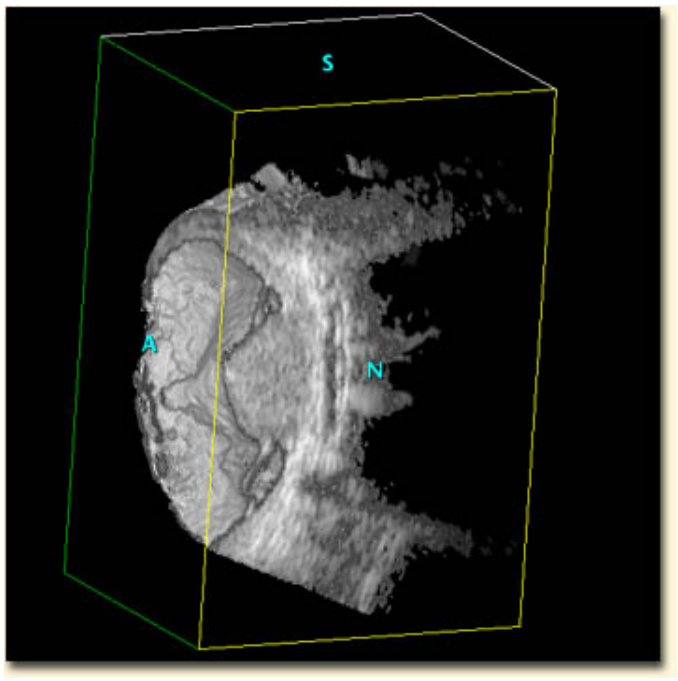
Ultrasound can also produce fan-shaped images by using a curvilinear array, or laterally steering the ultrasound beam at different angles.

Left-top shows a choroidal and vitreous detachment, right-top a lesion, left-bottom œdema and right-bottom retinal detachment.

3D ultrasound of the eye

A 3D ultrasound volume is usually acquired with a special transducer which is actually a linear array rotated by a stepper motor inside a case with an acoustic window.





This is an image from a 3D ultrasound scan of the retina. The large lump is a choroidal melanoma, with surrounding retinal detachment.

www.nyee.edu

It would also be possible to generate 3D data by using a 2D array of piezoelectric or CMUT elements.

Ultrasound and optical summary

In summary, ultrasound can see deeper into the retina, but with less resolution and less tissue contrast than OCT. Despite considerable differences in technology, scanning times are quite similar.

Feature:	OCT	Ultrasound
Imaging depth	3 mm	30 mm
Depth resolution	5 μm	200 μm
Lateral resolution	15 μm	$>$ 500 μm
Time for 2D	10 ms	5 ms
Time for 3D	400 ms	200 ms

Unsurprisingly, ultrasound is more complex than has been presented here. Some of these complications are addressed in 3G4: Medical Imaging and 3D Computer Graphics.

Visualisation of 3D data

Both OCT and ultrasound can generate 3D data of the retina, i.e. volumetric scalar data. In fact it is also possible to use either OCT or Ultrasound to generate vector velocity data for looking at blood flow (Doppler imaging).

There are a variety of difficulties with looking at 3D scalar or vector data:

- Almost all computer displays are rectilinear 2D arrays of pixels.
- Hard copies of the examination are in 2D.
- We generally only 'see' 2D surfaces - i.e. we are not used to looking inside objects, we just see the nearest surface.

We need a technique for taking 3D data and mapping it onto a 2D display in a way which reveals useful information in the data, and is easy for us to interpret correctly.

Visualisation of 3D data

There are a variety of ways of visualising or using 3D data:

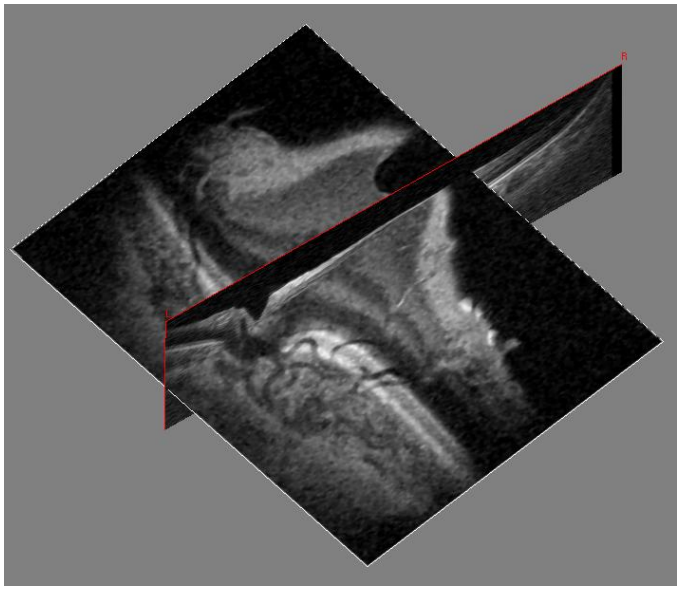
- Display a 2D planar cross-section through the 3D data in any location or orientation (*reslicing*).
- Take measurements of length or volume directly from the 3D data.
- Display an iso-surface within the 3D data (*surface rendering*).
- Display the 3D data as if it was translucent or transparent (*volume rendering*).

Of course it is also possible to generate displays of 3D data which use features from all of the above.

In the next few pages we will look at the advantages and disadvantages of these options without going into technical details regarding implementation.

Reslicing 3D data

Probably the simplest option is to only display the 3D data on a plane cutting through it. This image is an example of such a *reslice* along with one of the actually acquired cross-sections.



Reslicing is very simple, easy to understand, and allows the display of cross-sections which would be difficult to acquire directly. It requires *interpolation* of the data so we know what happens *between* data samples.

Although reslicing is very common, it is not a very efficient visualisation, since it only shows a small fraction of the 3D data at any one time.

Data interpolation

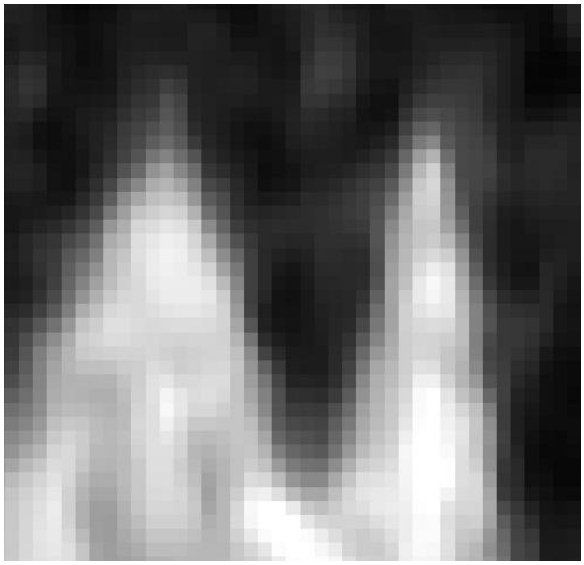
We only have values of the 3D data at each of the points on the 3D grid where we actually made measurements. But these points will not in general coincide with the points on the reslice through the data which we want to display.

Interpolation refers to the process of estimating values for the data *between* the sampled points. There are lots of ways of doing this, depending on what assumptions are made about the actual anatomy:

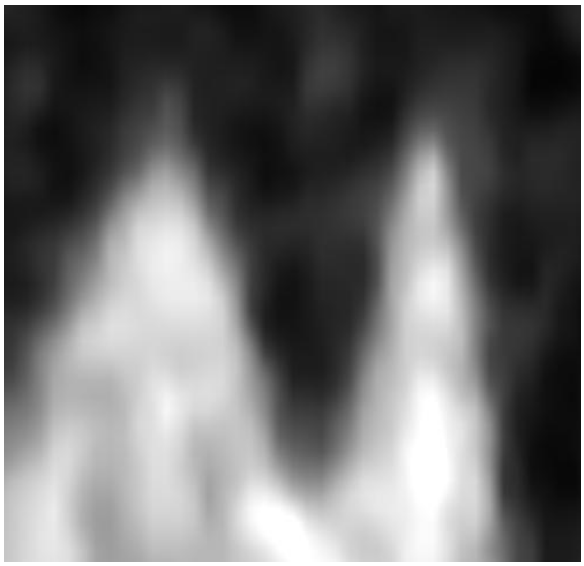
- In *nearest neighbour interpolation* we use the data from the nearest sampled point. Since we never 'make up' data, in a sense this makes no assumptions at all.
- In *linear interpolation* the data varies uniformly between samples. We have hence assumed that anatomical variation is continuous.
- There are a variety of forms of more complex *cubic interpolation*. In this case it is possible to assume that anatomical variation is both continuous and smooth (continuous in the second derivative).

Data interpolation

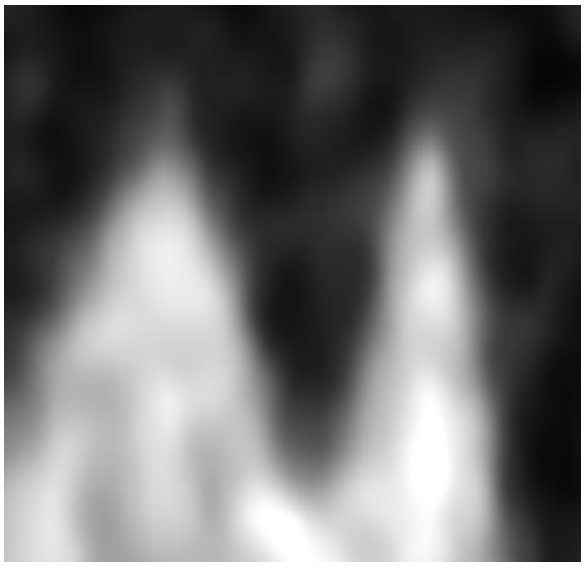
No interpolation



Nearest-neighbour interpolation



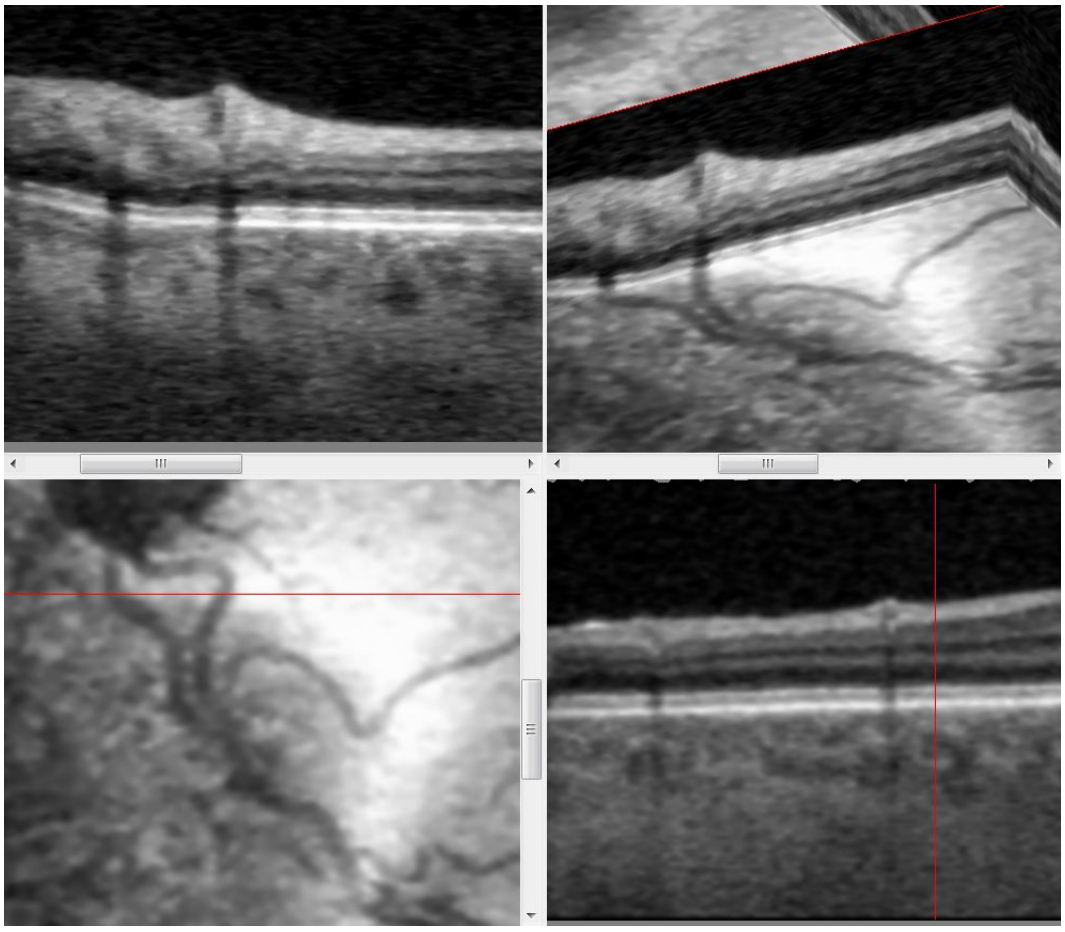
Linear interpolation



Cubic interpolation

These are some examples from the same zoomed section of OCT data.

Reslicing artifacts



Reslices through OCT or ultrasound data contain artifacts which are confusing when the direction of travel of the light or sound is not apparent.

The 'blood vessels' on the bottom-left image are actually 'shadows' cast from a plane closer to the retinal surface. These are visible as dark vertical strips on the top-left and bottom-right images.

Measurements from 3D data

Measurement of the volume of an anatomical structure is far more accurate if the object can be delineated in 3D.

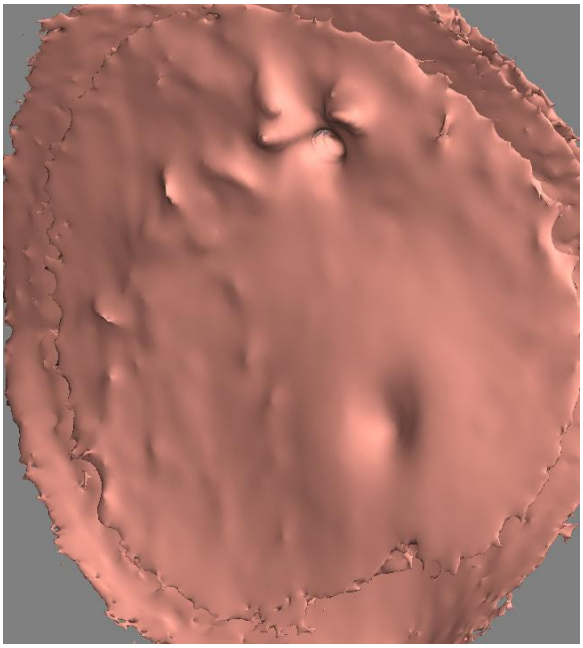
Although cross-sectional area is fundamentally a 2D measurement, this can be achieved more easily from 3D data because of the flexibility of reslicing. We can easily position the reslice plane so that it passes exactly through the feature of interest.

Similarly, distance measurements are possible in 2D but can be more accurate in 3D since the end points can be precisely defined and do not need to lie on a single imaged plane.

More complex measurements are also possible in 3D, for instance the amount of cupping (concavity) of the optic disc.

Surface rendering

It is possible to extract an *iso-surface* within the 3D data set and display this in a way which makes it look three-dimensional. This is a retinal surface which shows the location and geometry of the optic disc (top-centre) and the macula (bottom-right).



This technique gives very good contrast and definition of the surface being visualised, and the display is very efficient since graphics cards are already optimised for this sort of visualisation (since it is ubiquitous in computer games).

However, it is not always easy to define the surface in the first place. If the data is noisy a simple iso-surface will not necessarily display anything useful and other more complex processing may be required.

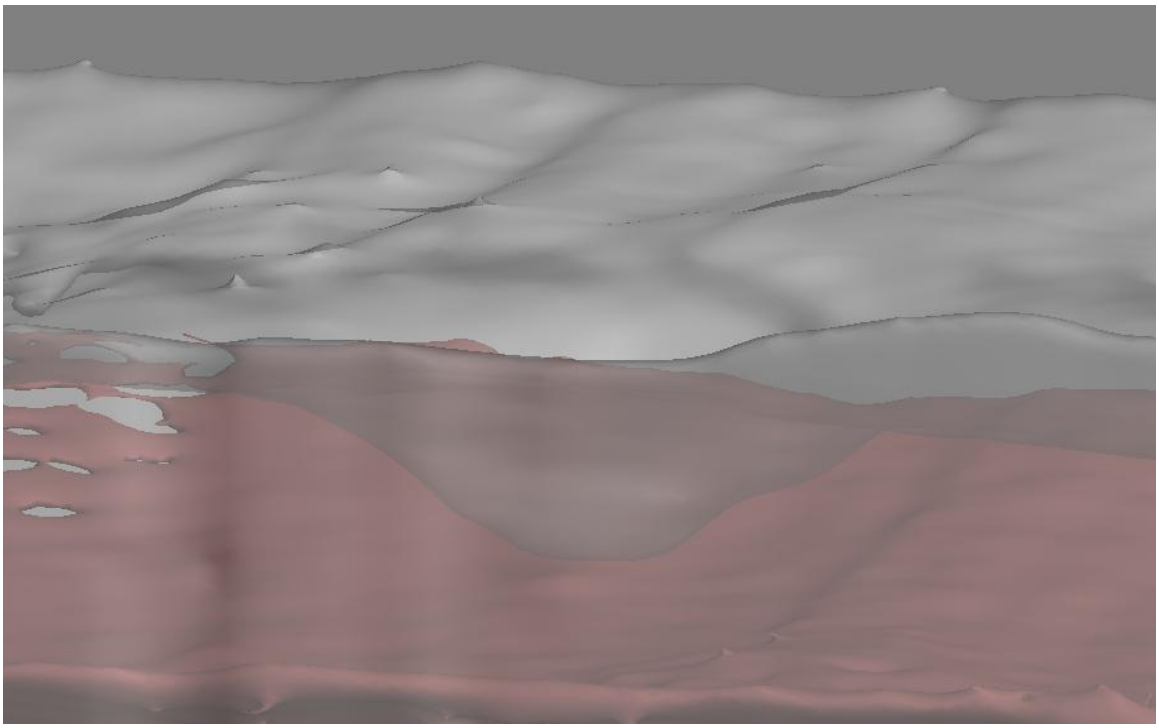
Surface rendering

The main processes involved in creating a surface rendering are:

- Define *thresholds* which determine where the iso-surface is in the data set.
- Possibly smooth the data set to reduce noise.
- Step through the data fitting triangles to each location which the iso-surface passes through, using e.g. *Marching Cubes*.
- Feed these triangles and lots of other information on lighting, material characteristics, viewpoints, camera properties, etc. to the graphics card on the computer.
- The surface can be interactively displayed by changing the viewpoint and camera properties (e.g. zoom).

Surface rendering

This is another example of a retinal surface rendering, showing a close-up of the macula.

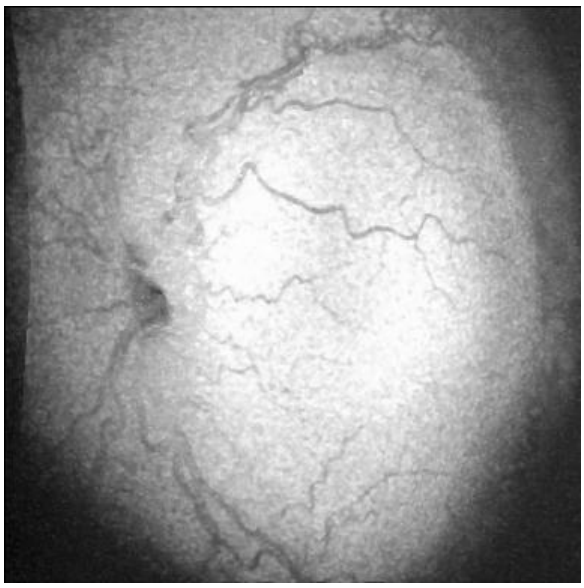


It is possible to display multiple surfaces by making one appear translucent. Here the top of the retinal surface (inner limiting membrane) is rendered in translucent grey, and a deeper surface (the photoreceptor segments) in red.

Volume rendering

In volume rendering, potentially the entire 3D data set contributes to each 2D image. This can generate results which are either extremely impressive or very confusing, depending on the skill of the operator.

For instance in *Maximum Intensity Projection* the maximum value in the 3D data behind each 2D pixel is found and displayed. This is much more informative when the 3D data is rotated.



Volume rendering is extremely flexible and is probably the most common way of visualising medical 3D data, second to reslicing, because, unlike in surface rendering, it is not necessary to define any exact surfaces.

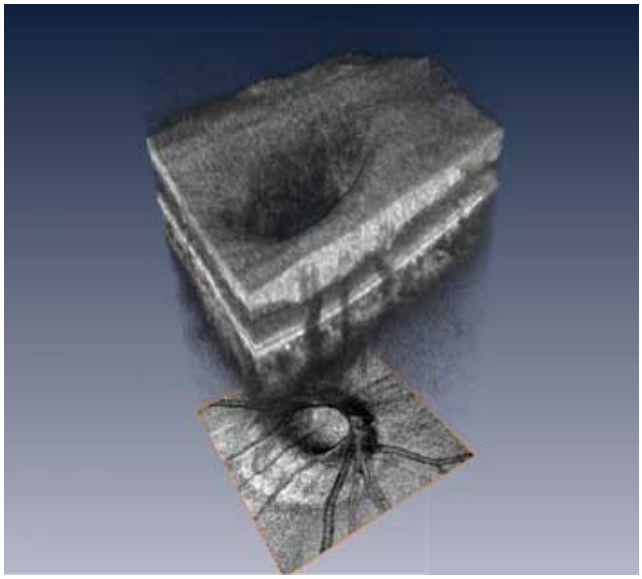
Volume rendering

There are a lot of properties to set for volume rendering:

- Each scalar data value is assigned a colour and also a transparency value.
- It is also usual to define thresholds in the data and use data gradients at these points to give an effect similar to a surface rendering.
- In this case we also need to define lighting parameters, as well as the view direction and camera properties.
- It is usually necessary to restrict the volume rendering to a specific region within the data in order to make the rendering clearer.

It is harder to do volume rendering very fast, since graphics cards have been designed for surface rendering. However, recent changes in card design have made accelerated real-time volume rendering possible for reasonable size data sets ($256 \times 256 \times 256$ data values).

Volume rendering

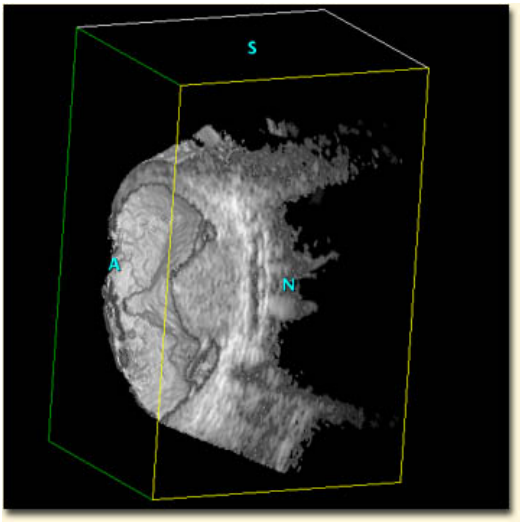


OCT data

www.infotech.ouliu.fi

This is a volume rendering from Ultrasound data of a lesion on the retina, with an emphasis on using data gradients about a particular threshold. This gives a visualisation which looks a little like a surface rendering.

This is a volume rendering of OCT data of the optic disc. Very dark data has been set to transparent to make the various layers of the retina clearer. There is also a projected volume rendering underneath.



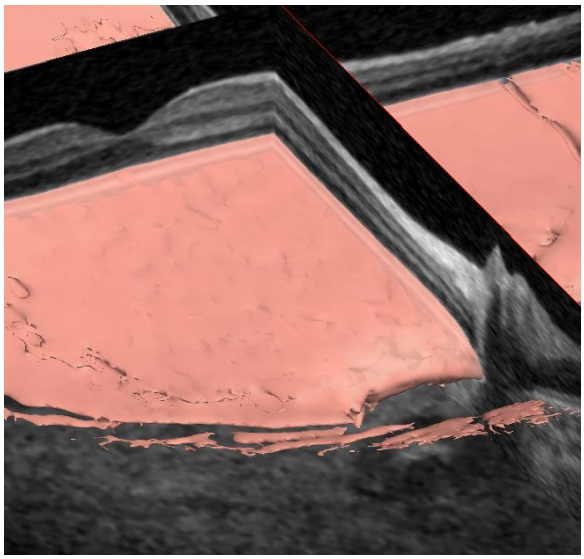
Ultrasound data

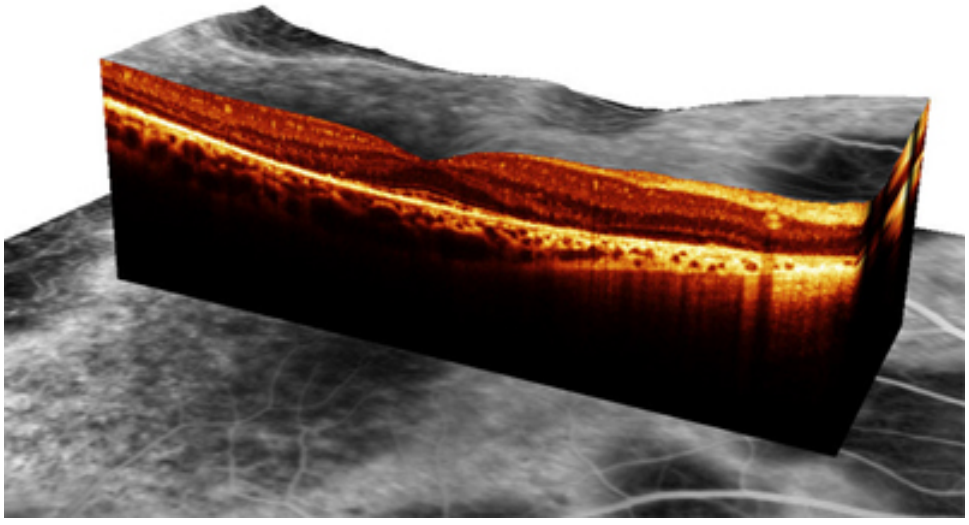
www.nyee.edu

Combinations of techniques

Many visualisations combine several techniques to give the most useful or visually pleasing results.

Here one of the retinal layers has been surface rendered, overlaid on several orthogonal reslices, which show the relative locations and details of the macula (top-left) and optic disc (bottom-right).





Here the top surface has been *texture-mapped* with data from a reslice.