

Chapter 4

Volume Rendering

The basic idea of volume rendering is that a viewer should be able to perceive a data volume from a rendered projection on the view plane. There are two main types of volume rendering:

- **Direct volume rendering** - involves generating a 2D representation of a 3D volumetric data set by projecting the voxels of the data set on to a 2D view plane.
- **Indirect volume rendering** - involves extracting a polygonal mesh representation of an isosurface of interest from a volumetric data set and rendering it to a 2D view plane using conventional 3D surface rendering techniques.

Volume rendering involves assigning transparency and colour to voxels within the data set. The data set can then be viewed from any angle by projecting the voxels values on to a suitably orientated view plane.

In medical applications this process is sometimes referred to as a computed X-ray as it is analogous to a conventional X-ray. It is possible to generate a computed X-ray from any viewing angle, including angles that may be physically impossible with conventional X-ray equipment.

Currently, the main application of volume rendering is in medical imaging due to the volumetric nature of data acquired using 3D medical imaging modalities. A 2D representation of the volume rendering process is illustrated in Figure 4.1.

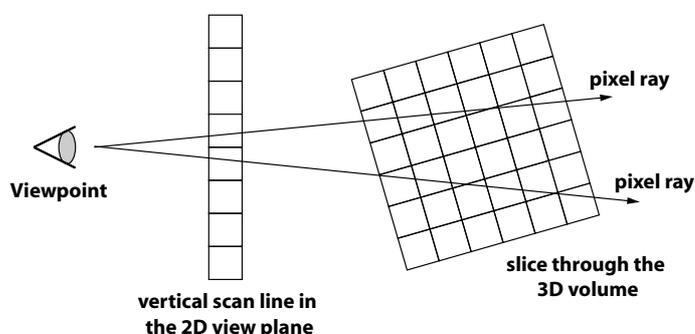


Figure 4.1: A 2D representation of the volume rendering process. Pixel rays are fired from the viewpoint through the volume. The colour of the a pixel in the 2D view plane is defined by the path of its associated pixel ray.

The illustration of the volume rendering process in Figure 4.1 uses perspective projection where all the rays emanate from a single viewpoint. An alternative would be to use parallel projection where a parallel rays is generated from each pixel in the view plane. In either case the volume rendering process is referred to as ray casting.

Note: The term ray casting is used to distinguish the method from ray tracing, which models the path taken by light rays as they interact with optical surfaces.

References:

- Levoy M. (1988) “Display of surfaces from volume data” IEEE Computer Graphics Applications, **8**(3), 29-37.
- Levoy M. (1990) “Efficient ray tracing of volume data” ACM Transactions on Graphics, **9**(3), 263-270.

The following sections describe different approaches to direct volume rendering that can be used to create a 2D representation of a 3D volumetric data set.

4.1 Maximum Intensity Projection

A maximum intensity projection (MIP) is a simple technique for transforming a 3D data sets to a 2D image. It involves identifying the maximum valued pixel along each ray projected from the pixels of the 2D view plane. The resulting maximum value is ultimately assigned to the associated pixel. An illustration of this process is illustrated in Figure 4.2.

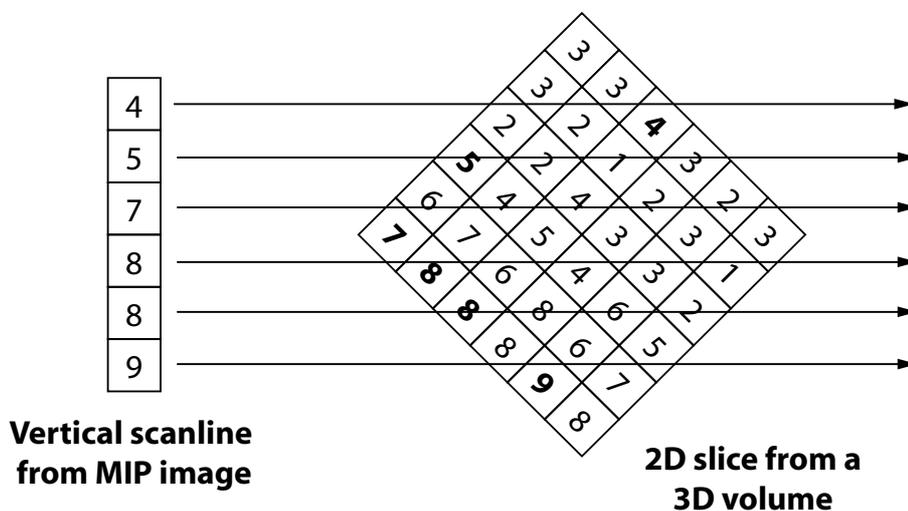


Figure 4.2: An illustration of the maximum intensity projection process. Maximum valued voxels are identified along each pixel ray. The resulting value is assigned to the associated pixel in the MIP image.

Maximum intensity projection is particularly useful in medical imaging applications where contrast enhanced material has been used to highlight a abnormal region of anatomy. One example of this is in a medical imaging technique known as magnetic resonance cholangiopancreatography. This involves the use of magnetic resonance imaging to visualise the biliary tree and pancreatic ducts in a non-invasive manner.

This procedure can be used to determine if gallstones are lodged in any of the ducts surrounding the gallbladder. An illustration of the types of images obtained in an MRCP examination and the resulting MIP representation are illustrated in Figure 4.3.

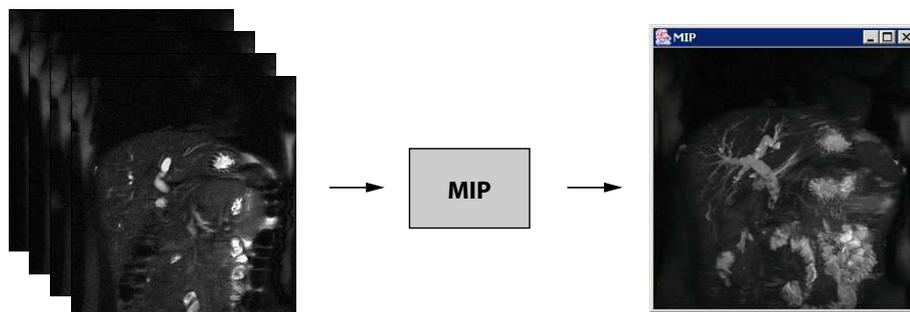


Figure 4.3: An illustration of the MIP process used in conjunction with MRCP data. The maximum value voxels projected in the the MIP image give a much better representation of the imaged anatomy than the individual slices.

It is clear that the structure of the anatomy is much more apparent from the MIP image. The structure of the biliary tree and liver can easily be seen in this image.

4.2 Average Intensity Projection

An average intensity projection is the same as a computed X-ray (mentioned earlier). The average value of the voxels encountered by each pixel ray is assigned to the associated pixel in the 2D view plane. The theory behind this process is illustrated in Figure 4.4. It is clear that the approach for average intensity projection is a variation of the approach outlined for maximum intensity projection.

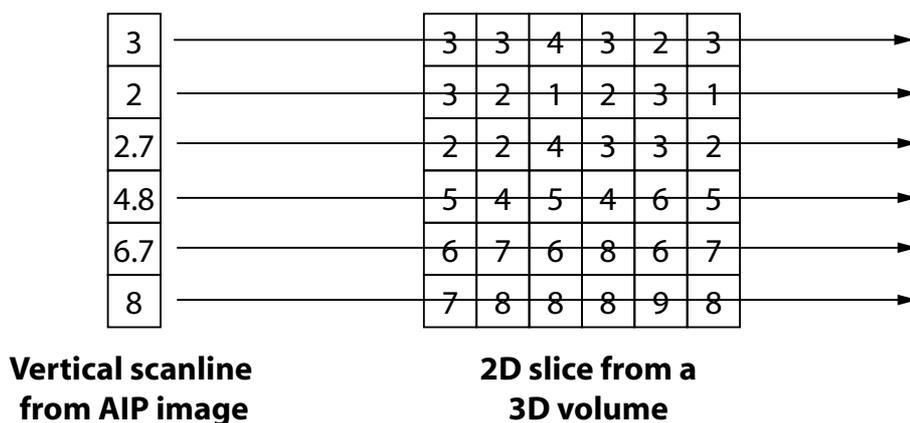


Figure 4.4: An illustration of the average intensity projection process. The average voxel value is identified along each pixel ray. The resulting value is assigned to the associated pixel in the AIP image.

An example of how the average intensity projection process can be applied to a CT data set is illustrated in Figure 4.5. It is clear that the output of this process is very similar to an X-ray image.

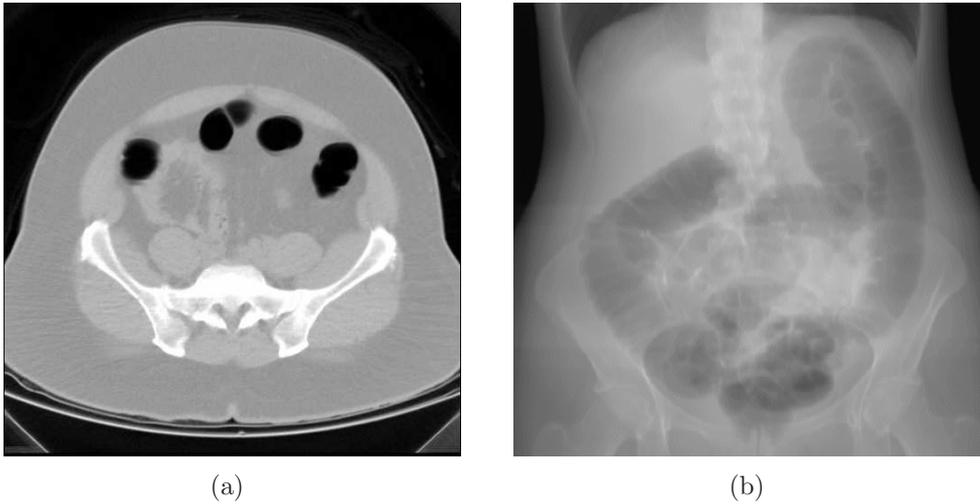


Figure 4.5: An example of the average intensity projection process. (a) A single slice from the data set being projected. (b) The resulting average intensity projection. Note that the AIP image is very similar to an X-ray image.

4.3 Transparent Voxel Rendering

The previous examples dealt with voxels that have a single property, i.e. the voxel colour. A more comprehensive approach to volume rendering involves giving the viewer the facility to see all the data. This involves assigning an additional opacity property to each voxel. The physical analogue to this model is that each voxel is considered to be made of a coloured transparent gel.

Each voxel is assigned a colour, C , and a transparency, α . The colour associated with the material type can be chosen aesthetically. For example, in the case of CT:

- White could be selected for the colour of bone voxels and an opacity could be selected that makes bone appear completely opaque.
- Mid-grey could be selected for soft tissue voxels and an opacity could be selected that makes soft tissue appear semi transparent.
- Black could be selected for air voxels and an opacity could be selected that makes air appear completely transparent.

In addition, it would be possible to use RGB colours to provide a greater degree of realism that can be achieved when limited to using grey scale colours.

Rendering volumes consisting of transparent voxels involves casting a ray into a volume which has been rotated into the desired viewing orientation. Compositing is performed along the ray path to generate the relevant pixel value in the 2D view plane. The compositing process involves accumulating the colours and opacities along the pixel ray.

This approach to volume visualisation involves the following three steps:

1. Classify each voxel in the original data set and assign the desired colour and opacity values.

2. Transform the classified volume data into the viewing direction.
3. For each pixel, cast a ray and find, by composition along the ray, a colour for that pixel.

4.3.1 Compositing Pixels Along a Ray

The most straightforward compositing operation involves the recursive application of the following formula:

$$C_{out} = C_{in}(1 - \alpha) + C\alpha \quad (4.1)$$

where:

- C_{out} is the accumulated colour emerging from the current voxel
- C_{in} is the accumulated colour going into the current voxel
- α is the opacity at the current voxel.
- C is the colour if the current voxel.

This process is illustrated in Figure 4.6

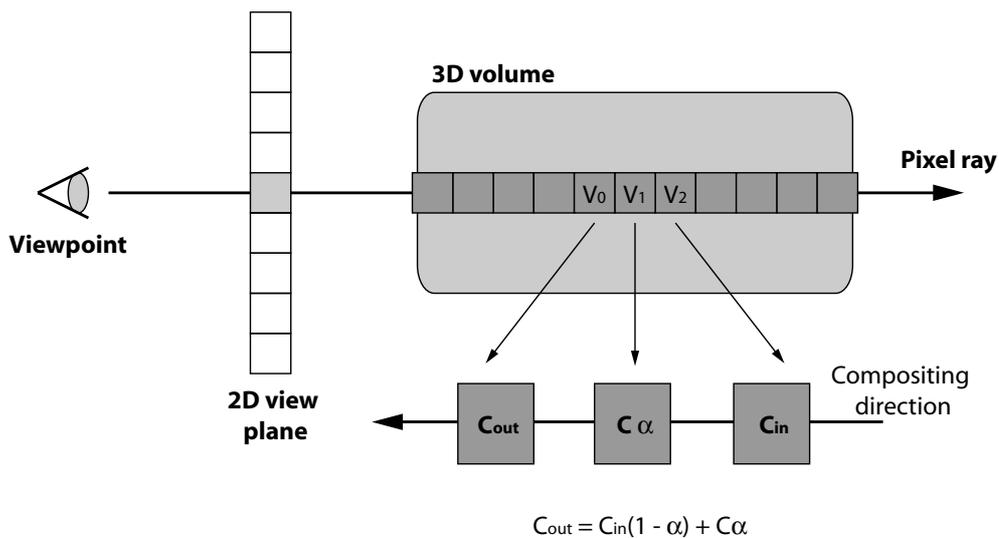


Figure 4.6: An illustration of the ray compositing process.

The direction implied by C_{in} and C_{out} is from back to front with respect to the view plane, i.e. the operation starts at the voxel that is furthest from the view plane.

Using this approach, the colour that comes out of a particular voxel is the product of the colour of the voxel and the opacity of the voxel plus the product of the incoming colour and the transparency of the voxel.

Note: The colour coming into the voxel that is furthest away from the view plane is black.

The purpose of this operation is to make voxels with high opacity values predominate, obscuring voxels that are behind them and being made visible through voxels

in front of them.

Example: Calculate the value of the shaded pixel in the view plane in Figure 4.7 by compositing the voxel values along the pixel ray.

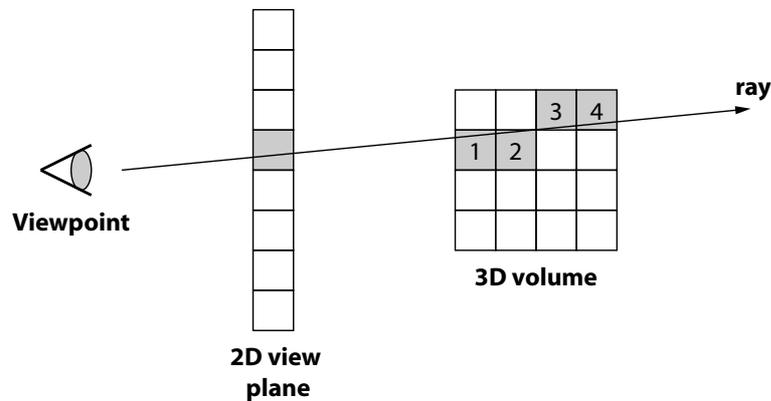


Figure 4.7: An illustration of the ray compositing process.

Given the following grey scale colour values and opacity values for the numbered voxels in the volume.

Voxel #	Colour	Opacity
1	20	0.05
2	25	0.07
3	130	0.55
4	225	0.80

- Voxel 4

- $C_{in} = 0, C = 225, \alpha = 0.8$
- $C_{out} = C_{in}(1 - \alpha) + C\alpha$
- $C_{out} = 0 \times (1 - 0.80) + 225 \times 0.8$
- $C_{out} = 180$

- Voxel 3

- $C_{in} = 180, C = 130, \alpha = 0.55$
- $C_{out} = C_{in}(1 - \alpha) + C\alpha$
- $C_{out} = 180 \times (1 - 0.55) + 130 \times 0.55$
- $C_{out} = 81 + 71.5 = 152.5$

- Voxel 2

- $C_{in} = 152.5, C = 25, \alpha = 0.07$
- $C_{out} = C_{in}(1 - \alpha) + C\alpha$
- $C_{out} = 152.5 \times (1 - 0.07) + 25 \times 0.07$

$$- C_{out} = 141.825 + 1.75 = 143.575$$

- Voxel 1

$$- C_{in} = 144, C = 20, \alpha = 0.05$$

$$- C_{out} = C_{in}(1 - \alpha) + C\alpha$$

$$- C_{out} = 144 \times (1 - 0.05) + 20 \times 0.05$$

$$- C_{out} = 136.8 + 1 = 137.8$$

Consequently, the value of the shaded pixel in the view plane is 138. It is clear that the voxels with the highest opacity values have the greatest influence on the composited pixel value.

4.3.2 Voxel Projection

This is a variant on volume rendering that involves traversing the data set and projecting each voxel onto the view plane. This process is illustrated in Figure 4.8.

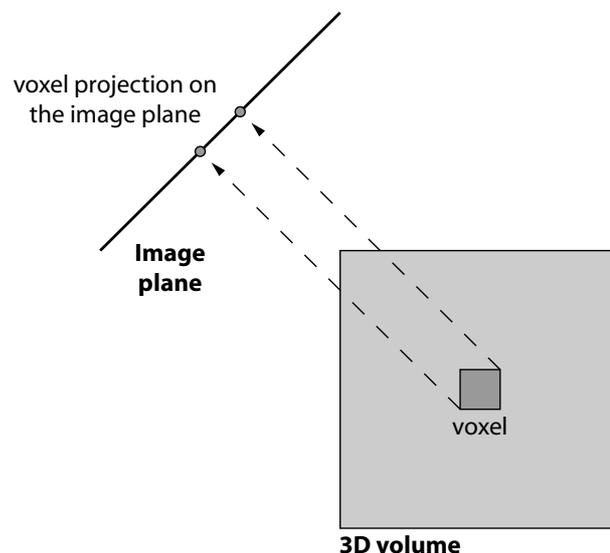


Figure 4.8: An illustration of the voxel projection process.

If the image plane is moved through the data, then the frame buffer is used as an accumulator and all pixels are updated simultaneously until the entire data set has been traversed and the pixels have their final values.

Possibly the most well know voxel projection algorithm is known as “splatting”. This term is used to describe the effect that one voxel has on the image plane. This algorithm considers how the contribution of a voxel should be spread or splatted in the image plane. There are two possible ways that splatting can be used:

1. A point in the data set at the centre of a particular voxel projects onto a single pixel. The three dimensional region surrounding the sampled voxel is filtered to determine its contribution to the pixel.
2. A single pixel value can be spread over a number of pixels in the image plane.

Both of these approaches are equivalent. Representations of the two approaches are illustrated in Figure 4.9.

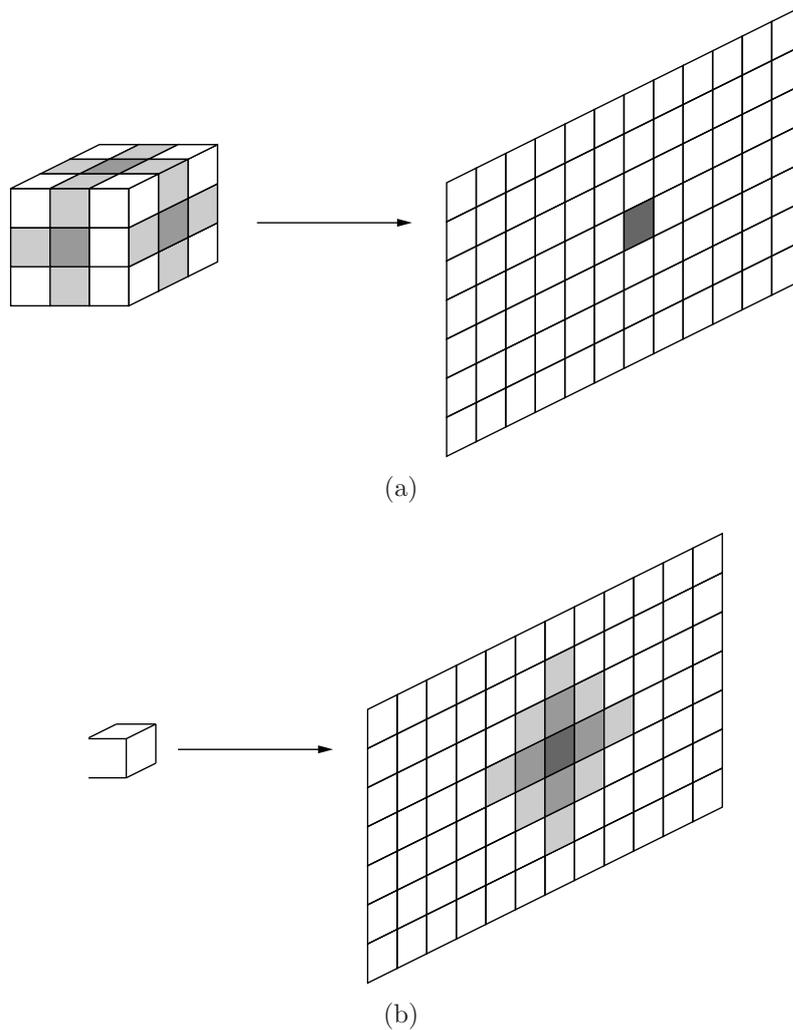


Figure 4.9: The two different approaches to splatting. (a) Many voxels are filtered over a spherical region to provide a single value for a pixel. (b) One voxel contributes value in a footprint region weighted as shown.

Reference:

- Westover L. (1990) “Footprint evaluation for volume rendering” *Computer Graphics*, **24**(4), 367-376.

4.3.3 Volume Rendering Implementation

The following example uses ray casting to generate a volume rendering of a CT data set. The program begins by loading a data set called `dataset.iso`. The dimensions of the data set are obtained by calling the relevant methods of the `DataSet` class. The dimension of the data set used in this example are:

- `width = 512 voxels`
- `height = 593 voxels`

- `depth = 512 voxels`

An initial raster scan of the data set is performed to determine the minimum and maximum voxel values. The extrema for the data set used in this example are:

- `minValue = -1024 HU`
- `maxValue = 1833 HU`

Recall that the values assigned to voxels in a CT data set are measured in Hounsfield Units. These units provide a measure of attenuation which is equivalent to density. A low value (-1024 HU) is equivalent to a low density material, e.g. air, whereas a high value (1833 HU) is equivalent to a high density material, e.g. bone.

In order to display a set of attenuation values in the form of a grey scale image, the values must be normalised in the range [0 - 255]. This is achieved by calculating two values:

1. **The offset** - this is the value that must be added to all attenuation values to ensure that each value is at least zero.
2. **The range** - this is the range of attenuation values. Dividing an offset attenuation value by the range always gives a value in the range [0 - 1].

In order to create a grey scale value in the range [0 - 255] from an attenuation value, the following equation can be used.

$$grey = \frac{255 \times (attenuation + offset)}{range} \quad (4.2)$$

A `BufferedImage` object is created to hold the 2D volume rendering and for loop is entered to calculate each pixel value for the `BufferedImage` object. At each (x, y) the ray composition algorithm is applied along the z direction. The colour of each voxel is calculated using equation 4.2 and a corresponding alpha value is calculated by dividing the colour value by 255.0. This results in a value in the range [0 - 1]. The initial alpha value is scaled using an exponential function to increase the transparency of low density voxels.

The `JFrame` is configured and the `BufferedImage` is displayed on the `JFrame` using a `JLabel`.

```

0  import java.awt.image.*;
   import javax.swing.*;
   import java.awt.*;
5  public class VolumeRenderingExample extends JFrame {
   public static void main(String args[]){new VolumeRenderingExample();}
   public VolumeRenderingExample()
10  {
   // Load the data set and get dimensions

```



(a)



(b)



(c)



(d)

Figure 4.10: Volume renderings of the same abdominal CT data set where the opacity is scaled using different exponential functions. It is clear that as the scale of the exponential function is increased, the lower density material, i.e. the soft tissue, becomes more and more transparent. In (a) the exterior of the patient and the outer covering of the patient can be seen, whereas in (d) only the skeleton is visible.

```
DataSet dataset = new DataSet("dataset.iso");
```

```
int width = dataset.getWidth();  
int height = dataset.getHeight();  
int depth = dataset.getDepth();
```

15

```

// Calculate the extrema of voxel values
int minValue = Integer.MAX_VALUE;
20 int maxValue = Integer.MIN_VALUE;

for(int x=0; x<width; x++)
    for(int y=0; y<height; y++)
        for(int z=0; z<depth; z++)
25     {
        int voxel = dataset.getVoxel(x, y, z);
        if(voxel<minValue) minValue = voxel;
        if(voxel>maxValue) maxValue = voxel;
    }

30 System.out.println("Min_value:_" +minValue+"_HU");
System.out.println("Max_value:_" +maxValue+"_HU");

// Calculate values required for normalisation
35 int range = maxValue - minValue;
int offset = - minValue;

// Create a BufferedImage to store the rendered output
BufferedImage bufferedImage = new BufferedImage(width, height,
40     BufferedImage.TYPE_INT_ARGB);

// Configure the exponential scaling function
float exponentialFactor = 1.0f;
float divisor = (float)Math.exp(exponentialFactor);
45

// Enter the volume rendering loop
float C_in, C_out = 0.0f;
for(int x=0; x<width; x++)
    for(int y=0; y<height; y++)
50     {
        C_in = 0.0f;
        for(int z=depth-1; z>=0; z--)
            {
                // Calculate the colour and opacity
55                 float C = dataset.getVoxel(x, y, z);
                C = (255*(C + offset))/range;

                float alpha = C/255.0f;
                alpha = (float)(alpha * Math.exp(exponentialFactor*alpha))/divisor);
60

                // Applied the ray compositing formula
                C_out = C_in*(1.0f - alpha) + C*alpha;
                C_in = C_out;
            }
65

// Generate a pixel value for ray and store in output image
int grey = (int)C_out;
int pixel = 0xff000000 | (grey << 16) | (grey << 8) | grey;
bufferedImage.setRGB(x, height-1-y, pixel);
70 }

```

```

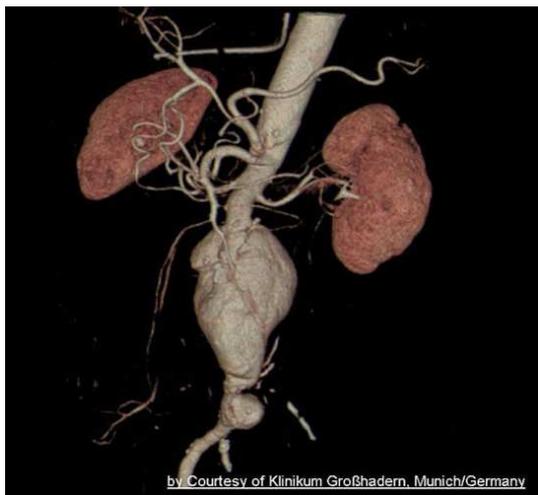
75 // Configure the JFrame and display the image
   this.getContentPane().setLayout(new BorderLayout());
   JLabel label = new JLabel();
   label.setIcon(new ImageIcon(bufferedImage));
   this.getContentPane().add(label, BorderLayout.CENTER);
   this.pack();
   this.setVisible(true);
80 }
}

```

A series of different volume renderings can be obtained by changing the exponential scaling function. Examples of these volume renderings are illustrated in Figure 4.10

4.4 Sample Volume Renderings

As mentioned earlier, volume rendering is typically used with volumetric data obtained using a medical imaging modality. The following examples, illustrated in Figure 4.11, demonstrate how volume rendering can be used to generate extremely detailed representations of human anatomy.



(a)



(b)

Figure 4.11: Two examples of high quality volume renderings obtained using commercial volume rendering software. In both cases pseudo colouring has been used to help distinguish between different voxel classes and enhance the appearance of the rendered output.

4.5 Medical Imaging Modalities

Volumetric data sets are typically acquired using a medical imaging modality that sample the characteristics of the region occupied by the patient at regular intervals. The images generated by 3D medical imaging modalities typically consist of thin slices of voxels that can be stack to create a volumetric representation of the scanned

region. This section will describe two medical imaging modalities: Computed tomography (CT) and magnetic resonance imaging (MRI).

4.5.1 Computed Tomography

Computed tomography (CT), originally known as computed axial tomography (CAT), is a medical imaging method employing tomography where digital geometry processing is used to generate a 3D volumetric representation of a scanned region from a large series of 2D X-ray images taken around a single axis of rotation.

X-ray slice data is generated using an X-ray source that rotates around the object. X-ray sensors are positioned on the opposite side of the circle from the X-ray source. Many data scans are progressively taken as the object is gradually passed through the gantry. The scans are ultimately combined together using a mathematical procedure known as tomographic reconstruction. There have been several generations of CT scanners and these are illustrated in Figure 4.12. Examples of the techniques used by two commercial CT scanners are illustrated in Figure 4.14.

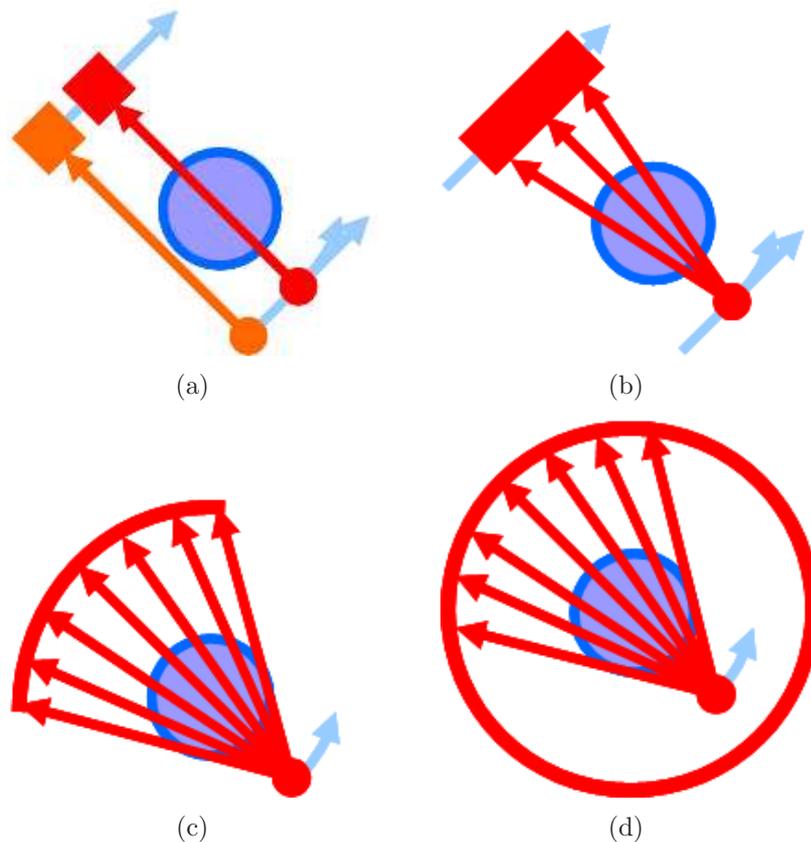


Figure 4.12: An illustration of the four generations of CT scanners. (a) The first generation: one detector, translation rotation, parallel beam. (b) Second generation: multiple detectors, translation-rotation, small fan-beam. (c) Third generation: multiple detectors, source rotation, large fan beam. (d) Fourth generation: Detector ring, source-rotation, large fan-beam.

CT is regarded as a moderate to high radiation diagnostic technique. While technical advances have improved radiation efficiency, there has been a simultaneous pressure to obtain higher-resolution image and use more complex scan techniques,

both of which require higher doses of radiation. The effective radiation dose for a chest CT is 290 times the dose for a chest X-ray.

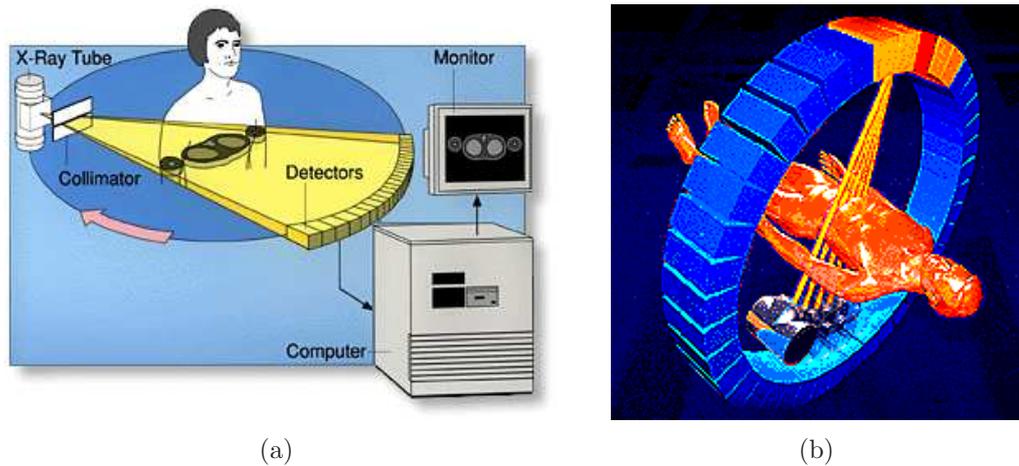


Figure 4.13: An illustration of the operation of two commercially available CT scanning systems. (a) A Siemens system, third generation. (b) A Picker system, fourth generation.

An example of an actual Siemens CT scanner is illustrated in Figure 4.14. This is a Siemens Somatom Sensation 16 scanner. It has 0.5 mm resolution and a 0.4 second rotation time. This type of high performance rotation is key for scanning the beating heart (4D medical imaging).



Figure 4.14: An image of a Siemens Somatom Sensation 16 scanner. This scanner has 0.5 mm resolution and 0.4 second rotation time.

4.5.2 Magnetic Resonance Imaging

Magnetic resonance imaging (MRI) is a non-invasive method used to render an image of the interior of an object. It is primarily used in medical imaging to demonstrate pathological¹ or other physiological² alterations of living tissues.

A CT scanner uses X-rays, a type of ionising radiation, to acquire its images. MRI, on the other hand uses non-ionising radio frequency signals to acquire its images. MRI relies on the properties of hydrogen molecules in water when influenced by a strong magnetic field. Even though the strength of the magnet is typically 100,000 the strength of the earth's magnetic field, an MRI examination poses no threat to the subject being imaged.

It should be noted that although CT and MRI both generate volumetric representations of the interior of a patient, MRI takes longer to scan a volume than CT, MRI responds differently to materials than CT and MRI does not provide as high a resolution as CT. An MRI scanner provides a maximum resolution of 1mm^3 . A modern commercially available MRI scanner is illustrated in Figure 4.15.



Figure 4.15: A three tesla Philips MRI scanner.

¹Caused by a disease.

²Relating to the functions and activities of living organisms.

4.6 Stereo 3D Visualisation

All of the 3D content discussed in this course has been rendered to a 2D display. This essentially flattens the 3D content so that it can be viewed using conventional visualisation systems, e.g. LCD and CRT displays.

In order to get that full effect of 3D content, it must be viewed using a stereo 3D visualisation system where a different views of the content are created for the left and right eyes. Using this approach the content appears to have a depth associated with it. This adds a great deal of realism to a 3D scene. An overview of the stereo viewing process is illustrated in Figure 4.16.

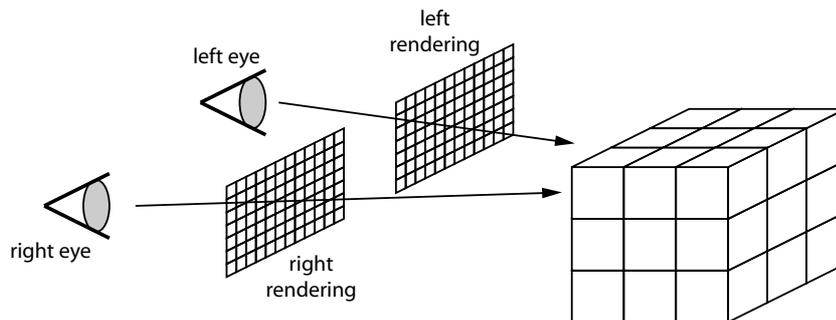


Figure 4.16: An illustration of the stereo viewing process where separate renderings are created for the left and right eyes.

One example of a stereo 3D viewing system is CrystalEyes from StereoGraphics (RealD). This system utilises a pair of LCD shutter glasses in conjunction with a CRT display and a compatible high refresh rate view card to facilitate stereo 3D viewing.

Assume that the CRT monitor has a refresh rate of 120 Hz. The CrystalEyes system displays the left eye and right eye renderings of a 3D scene in subsequent frames. A signal is sent to the CrystalEyes glasses so that right lens is opaque when the left eye image is being displayed and the left lens is opaque when the right eye image is being displayed. The CrystalEyes equipment is illustrated in Figure 4.17.



Figure 4.17: The CrystalEyes LCD shutter glasses and remote control unit.

4.7 Summary

This chapter has presented an overview of volume rendering techniques dealing with ray casting and voxel projection based approaches. A implementation of a ray casting based volume rendering system demonstrated the simplicity of the approach. It should be noted that the direct volume rendering techniques discussed in this chapter provide a very different output to the indirect volumetric rendering technique based on the marching cubes algorithm that was discussed in the previous chapter. This chapter also provided a overview of common medical imaging modalities and introduced the concept of immersive 3D stereoscopic visualisation.