

COMS30115

Rendering

Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

January 26, 2017

<http://www.carlhenrik.com>

Info

- Coursework
 - Find a partner

- Coursework
 - Find a partner
- Reddit
 - its reddit, cat gifs are encouraged
 - if you found an awesome graphics resource etc.

- Coursework
 - Find a partner
- Reddit
 - its reddit, cat gifs are encouraged
 - if you found an awesome graphics resource etc.
 - its anonymous, tell me if I'm an idiot and the lecture/coursework being stupid

Today

Images the stuff that we want out

Colour the stuff that images are made of

Geometry how we represent the world

Illumination how geometry gets its colour

Cameras what captures the image

Images

Images

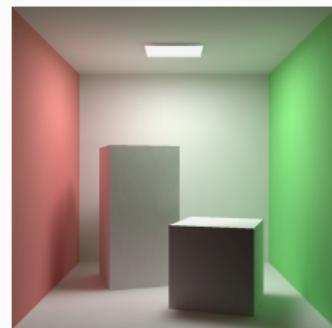
- Rectangular grid of elements
- Parametrization
 - Location (x, y)
 - Colour p



Images

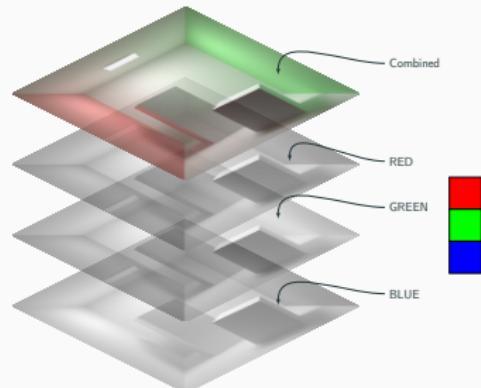
Encoding

- Indexed Images
 - GIF, PNG have this format
 - Palette (look-up)
 - Efficient if few colours in image
 - Older hardware implemented this
- (True) colour Images
 - Directly stores Colour values
 - Simpler, not limited to palette size
 - Less efficient for low colour images



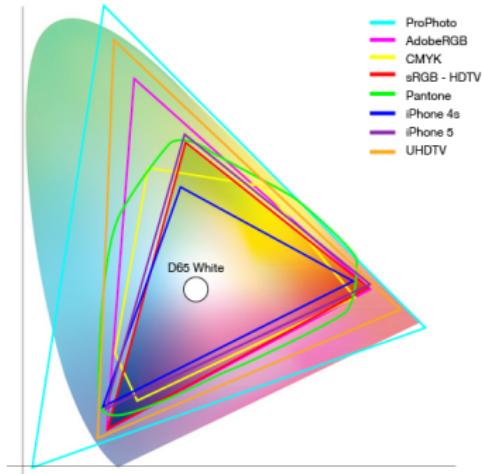
Pixels

- Colour
 - RGB - RED, GREEN, BLUE
 - HSV - HUE, SATURATION, VALUE
- Nr of bits known as the **depth**
- What is the effect of the parametrisation?
- Similarity



Gamut

Gamut Comparison



The Dress that broke the Internet¹



¹https://en.wikipedia.org/wiki/The_dress

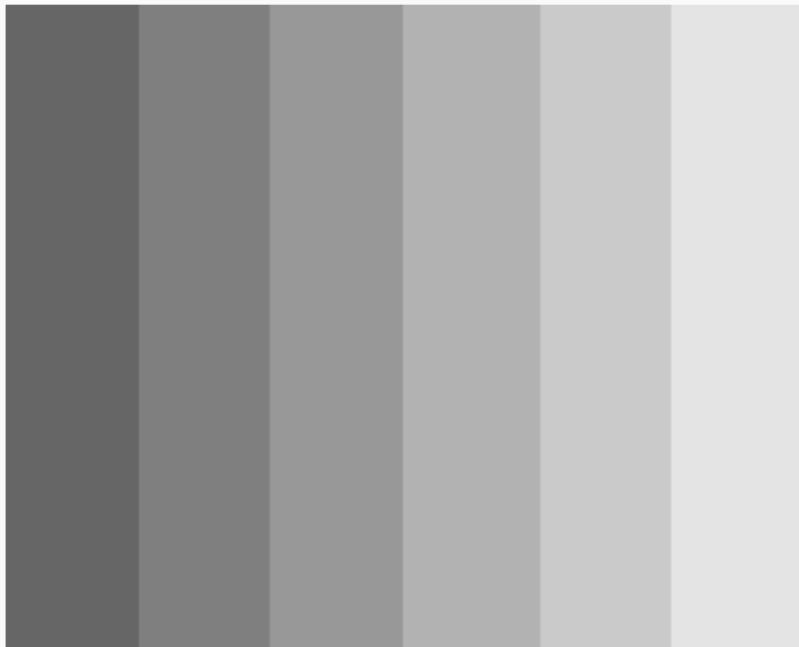
Colour

- Psychological
 - colours encode meaning
 - meaning is highly subjective
- Physiological
 - internal processes of colour
 - optical illusions etc.

Colour

- Psychological
 - colours encode meaning
 - meaning is highly subjective
- Physiological
 - internal processes of colour
 - optical illusions etc.
- *an image is interpreted by us, we never see the pixels*

Mach Bands²



²https://en.wikipedia.org/wiki/Mach_bands

Achromatic colour

Achromatic colour

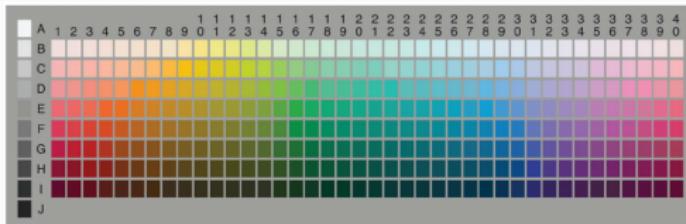
- Intensity perception relative for humans
- Perceptual difference between 0.1 and 0.11 is perceived the same as 0.5 and 0.55 (Max intensity 1.0)

$$I_0 = I_0, I_1 = r \cdot I_0, \dots, I_n = r^n I_0 = 1.0$$

$$r = \left(\frac{1}{I_0} \right)^{\frac{1}{n}}$$

$$I_j = I_0^{\frac{n-j}{n}}$$

The World Colour Survey³

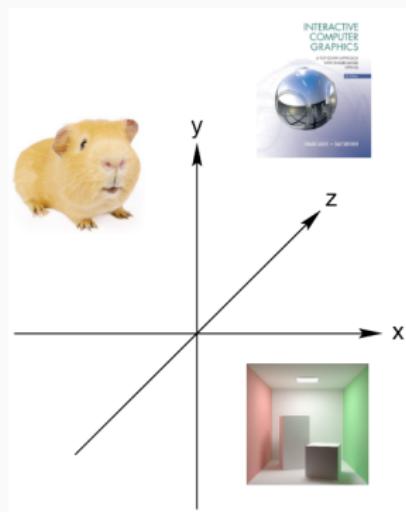


- Munsell chart
 - equal perceived distance between colours
- Is there a universal language of colour?
- 110 unwritten languages
- 2616 Monolingual participants

³<http://www1.icsi.berkeley.edu/wcs/>

Image Parametrisation

- RGB pixel $p \in \mathbb{R}^3$
- Image as a point in a vector space
- $I \in (\mathbb{R}^3)^{\text{width} \times \text{height}}$
- Typical Image size
 $1920 \times 1080 \times 3 = 6220800$



Generative Model

Parameters

- Parametrisation of an image
- Number of parameters is degrees of freedom of the **image**

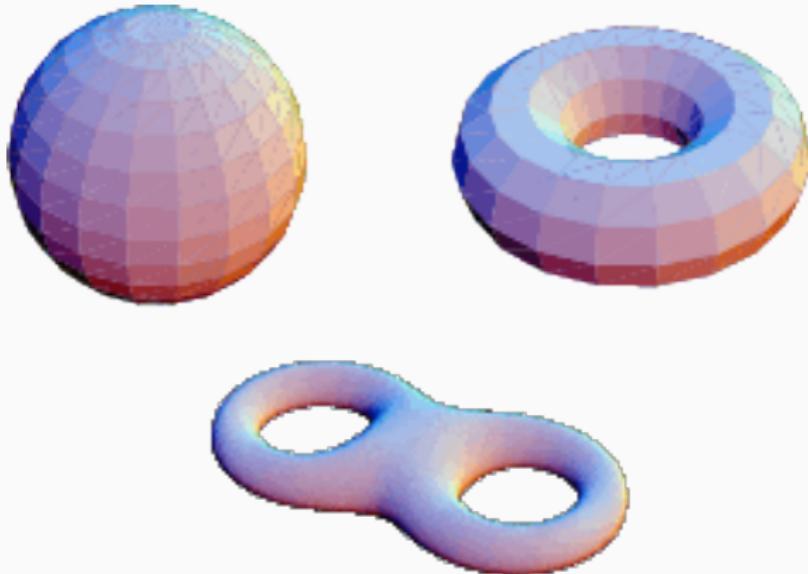


Model

- There is some underlying process that has generated the image (physics)
- Allows to control generation
- Generates visual appearance



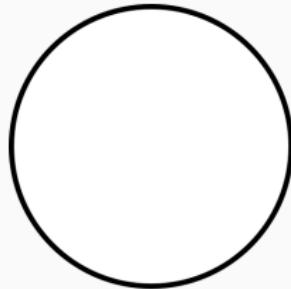
Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

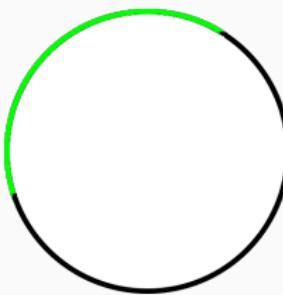
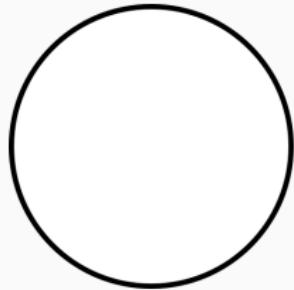
Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

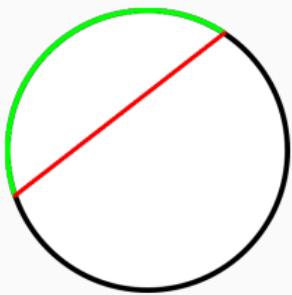
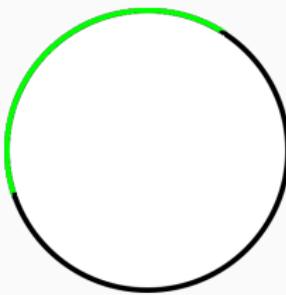
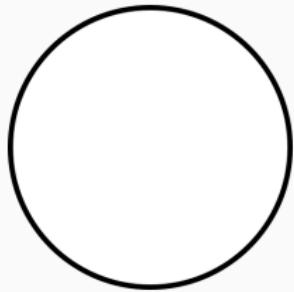
Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

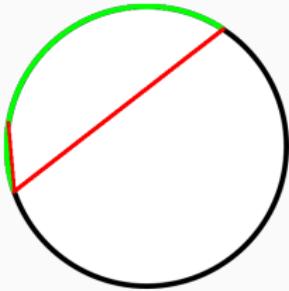
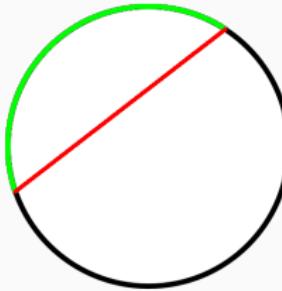
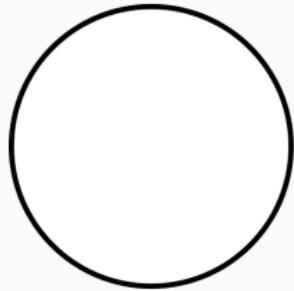
Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

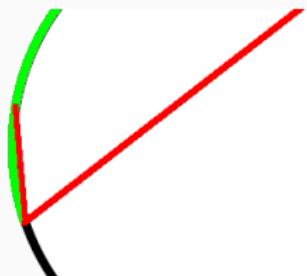
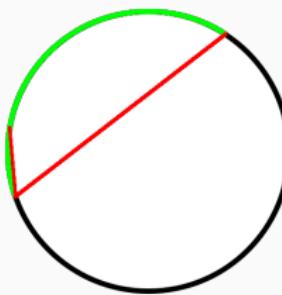
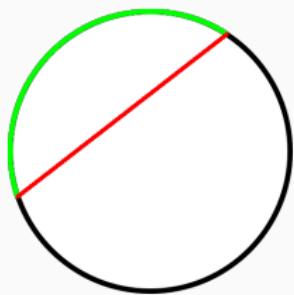
Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

Manifolds



"In mathematics, a manifold is a topological space that near each point resembles Euclidean space"⁴

⁴<http://en.wikipedia.org/wiki/Manifold>

Parametrisation

- Image space is a manifold in space of matrices with positive values
- Degrees of freedom is coordinates along manifold
 - Each parameter configuration generates valid image
- *how can we find these parameters?*

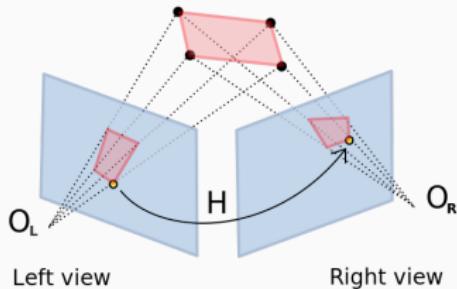
Parametrisation

- Image space is a manifold in space of matrices with positive values
- Degrees of freedom is coordinates along manifold
 - Each parameter configuration generates valid image
- *how can we find these parameters?*
 - **Light** If there is no light all environments look the same
 - **Geometry** Light bounces off surfaces to eye/camera, i.e. we see surface
 - **Surface** material, texture, colour etc. alters light
 - **Imaging device** Camera/eye characteristics

Parametrisation

- Image space is a manifold in space of matrices with positive values
- Degrees of freedom is coordinates along manifold
 - Each parameter configuration generates valid image
- *how can we find these parameters?*
 - **Light** If there is no light all environments look the same
 - **Geometry** Light bounces off surfaces to eye/camera, i.e. we see surface
 - **Surface** material, texture, colour etc. alters light
 - **Imaging device** Camera/eye characteristics
- We will study both the parametrisation and the generative mapping

Computer Vision



Computer Graphics We will assume a given parametrisation, from which to generate an image, i.e. we will study the generative mapping.

Computer Vision Take an image and recover parametrisation of this image

Computer graphics is the "forward" task of computer vision which solves an inverse problem

Assumptions

- Without making assumptions that *simplifies* our task we cannot do graphics



Assumptions

- Without making assumptions that *simplifies* our task we cannot do graphics
- Importantly, make informed assumptions
 - this is what this unit and computer graphics is about



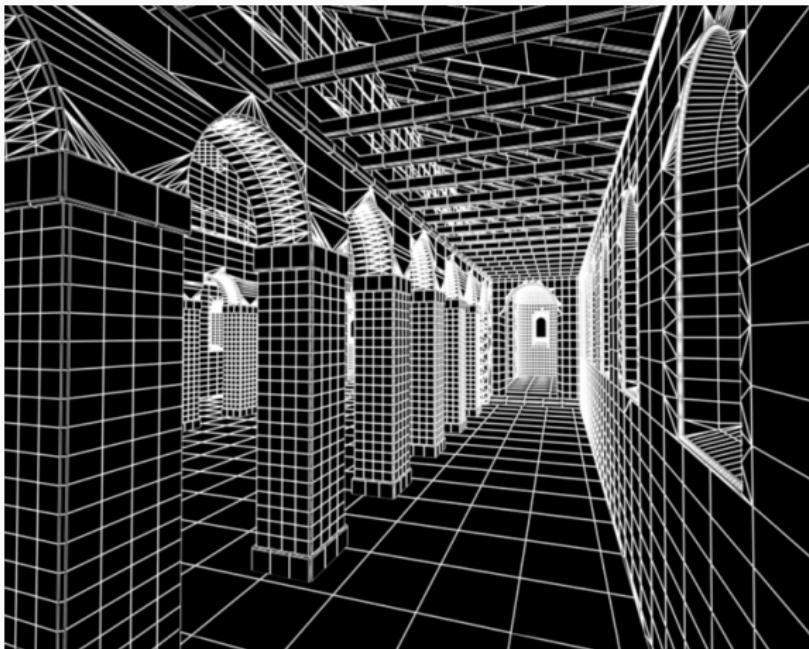
Assumptions

- Without making assumptions that *simplifies* our task we cannot do graphics
- Importantly, make informed assumptions
 - this is what this unit and computer graphics is about
- Light, Geometry, Surfaces, Cameras

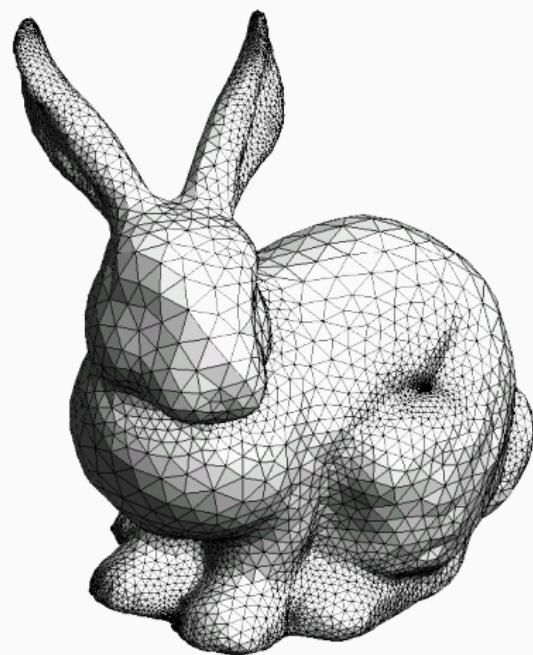


Geometry

Geometry

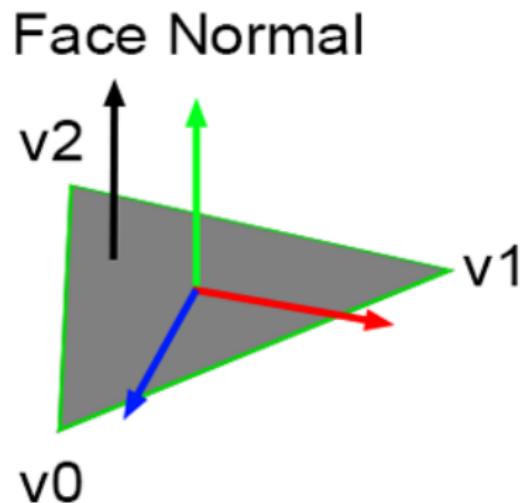


Meshes⁵



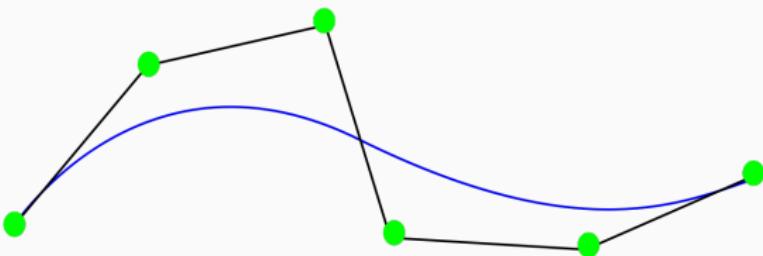
⁵https://en.wikipedia.org/wiki/Stanford_bunny

Triangles

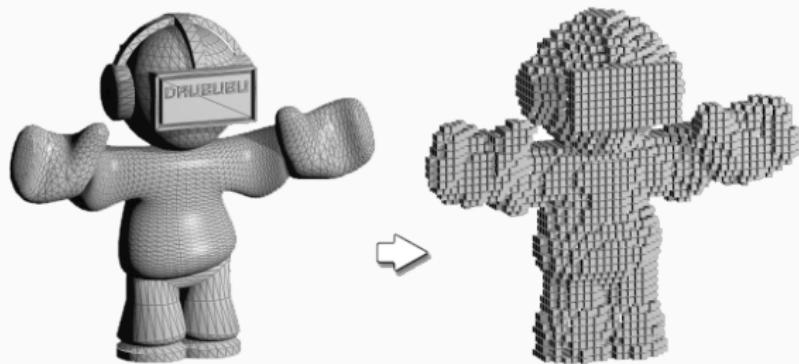


© www.scratchapixel.com

Non-planar Geometry



Voxels⁶



⁶<https://www.gamersnexus.net/gg/762-voxels-vs-vertices-in-games>

Assumptions

- The world is locally flat
- Parametrise each object as a set of triangles
- Each object is just an empty hull
- Everything has a fixed resolution

Illumination

Illumination



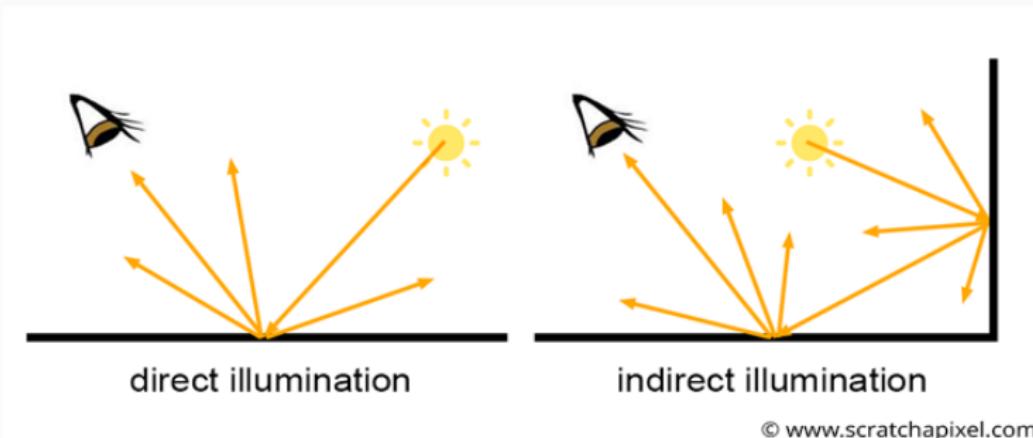
Light

- generated from lightsources (i.e. sun, lamps)
- "bounces" around in the world
- surface visible when hit by light

direct source \Rightarrow surface

indirect surface \Rightarrow surface

Illumination



© www.scratchapixel.com

Transport Problem

- Illuminating/Rendering \Rightarrow compute interaction between incident light and surface
- Compute transportation



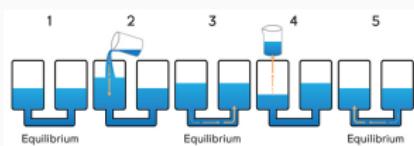
Transport Problem

- Illuminating/Rendering \Rightarrow compute interaction between incident light and surface
- Compute transportation
- Conservation principle
 - nothing is created except for at the source



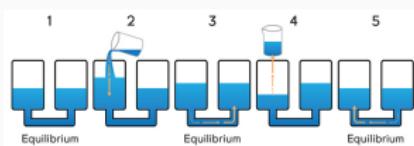
Transport Problem

- Illuminating/Rendering \Rightarrow compute interaction between incident light and surface
- Compute transportation
- Conservation principle
 - nothing is created except for at the source
- light is *very* fast



Transport Problem

- Illuminating/Rendering \Rightarrow compute interaction between incident light and surface
- Compute transportation
- Conservation principle
 - nothing is created except for at the source
- light is *very* fast
- **Assume** that it is at equilibrium



Rendering Equation ⁷

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) \cos(\mathbf{n}_x, \Psi) d\omega_\Psi$$

- Formulates transport problem of light
- Encapsulates "all" rendering techniques
- Intractable

⁷Kajiya, J. T. (1986). The rendering equation. In , ACM Siggraph Computer Graphics (pp. 143–150)

Fredholm Theory⁸

$$\phi(t) = f(t) + \int K(t, s)\phi(s)ds$$

- Fredholm integral equation
- Gives rise to a whole theory of mathematics
 - Fredholm Theory

⁸https://en.wikipedia.org/wiki/Fredholm_integral_equation

Assumptions

- Only direct lighting
- Indirect lighting can be simulated by ambient light
- All surfaces are perfectly diffuse
 - look the same from each angle
- All lights are points

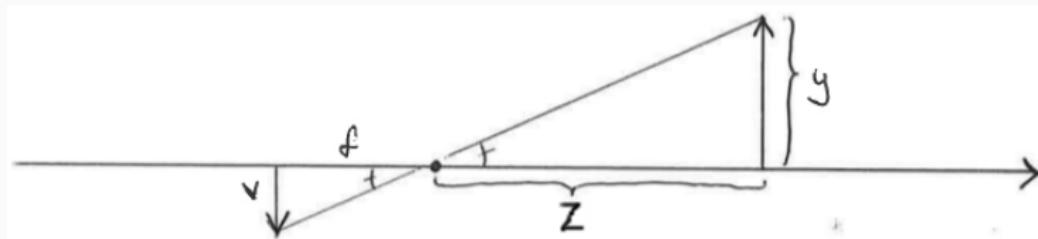
Camera

Camera

- Cameras create images
- Lens
 - captures rays
- Focal length
- Aperture

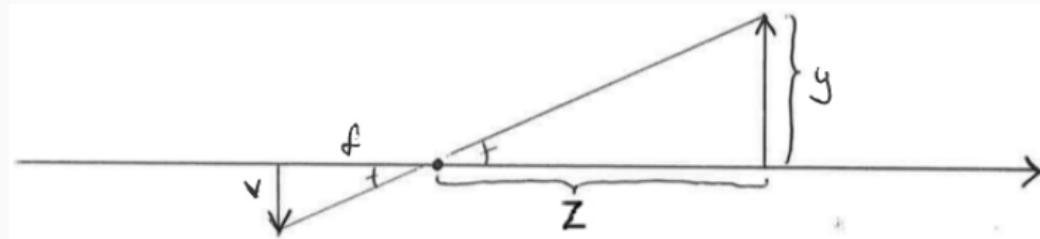


Pinhole camera

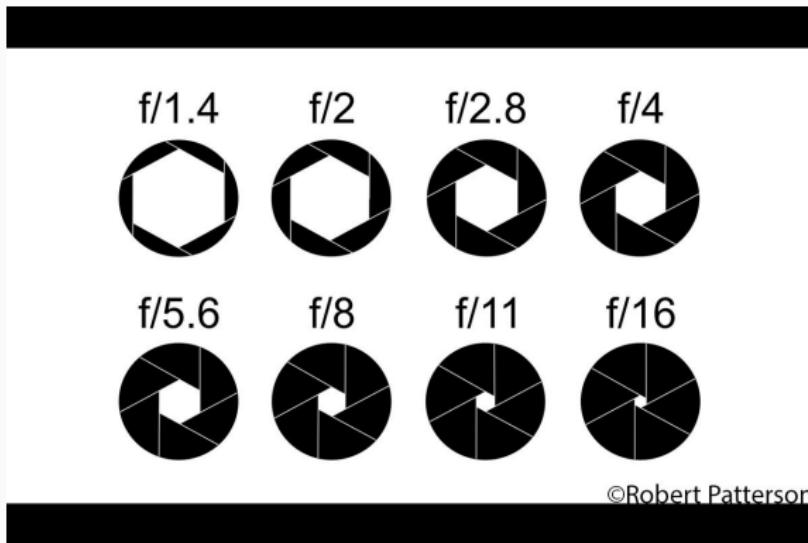


- Simplest form camera, where one ray of light lights up each pixel
- Defined by a *location*, *focal length*, *view-direction* and *up-direction*
- is this a realistic camera?

Pinhole camera

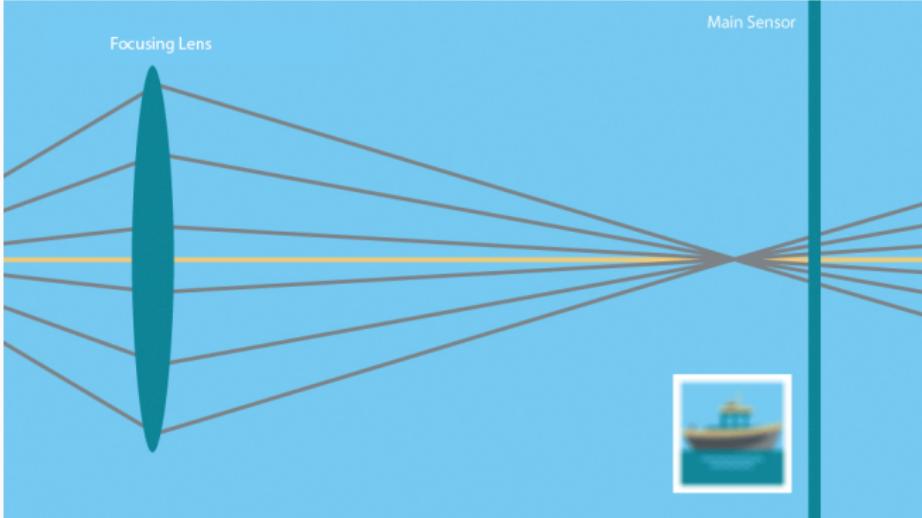


$$\frac{v}{f} = \frac{y}{z}$$
$$\Rightarrow v = \frac{f}{z}y$$



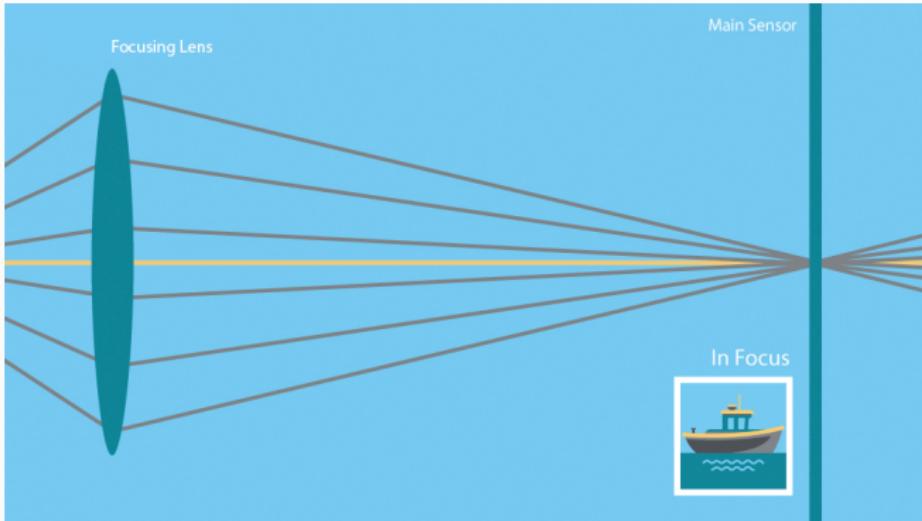
⁹<https://geneseodigitalphotography.wordpress.com/learn-about-your-camera/reference-sheets/aperture/>

Depth-of-Field¹⁰



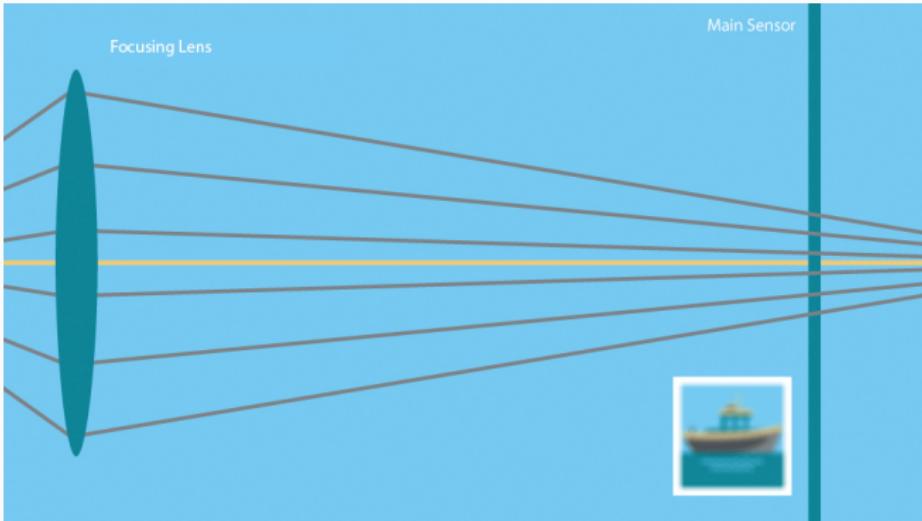
¹⁰<https://www.bhphotovideo.com/explora/photography/tips-and-solutions/how-focus-works>

Depth-of-Field¹⁰



¹⁰<https://www.bhphotovideo.com/explora/photography/tips-and-solutions/how-focus-works>

Depth-of-Field¹⁰



¹⁰<https://www.bhphotovideo.com/explora/photography/tips-and-solutions/how-focus-works>

Lens Flare

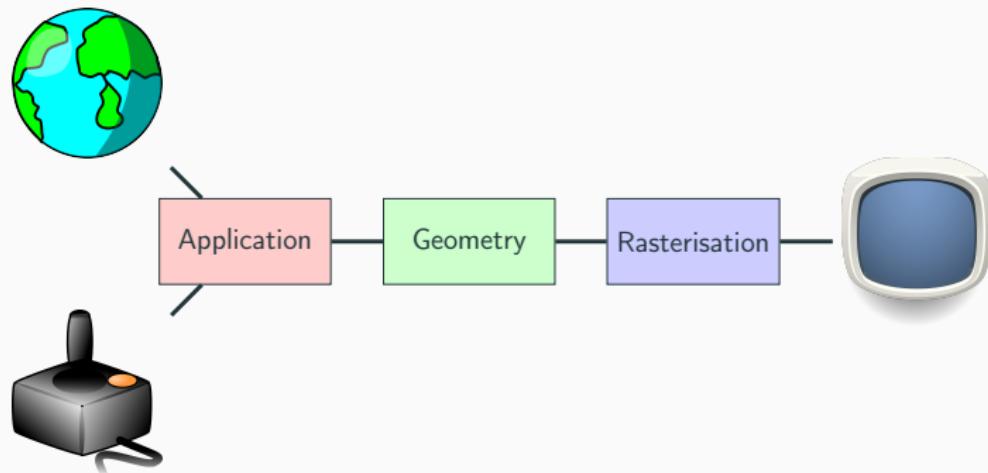


Assumptions

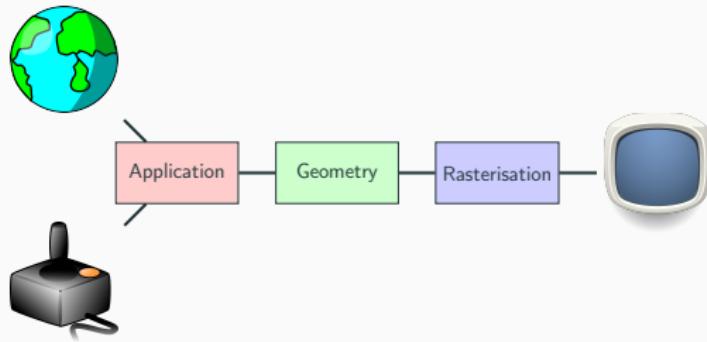
- Pinhole camera
 - everything is in focus
 - point aperture
- No lens
 - no lens flare
 - no lens distortion

Rendering Pipeline

Rendering Pipeline



Rendering Pipeline



Decide what part of geometry “might” be visible and updates geometry.

- Movement: camera, objects
- Collision Detection
- Animation: geometry, textures

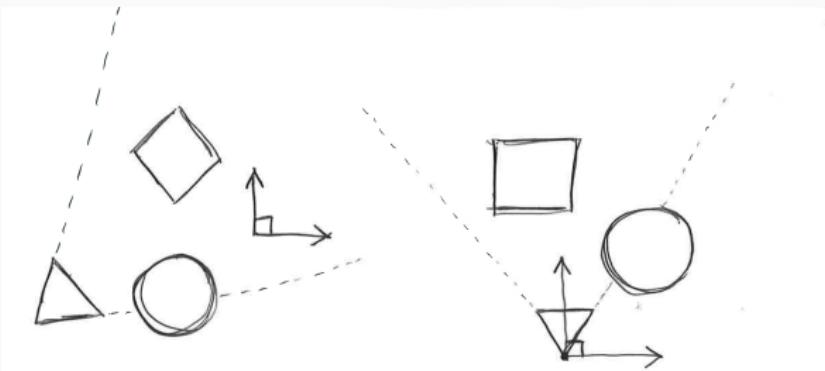
Done completely in software

Rendering Pipeline



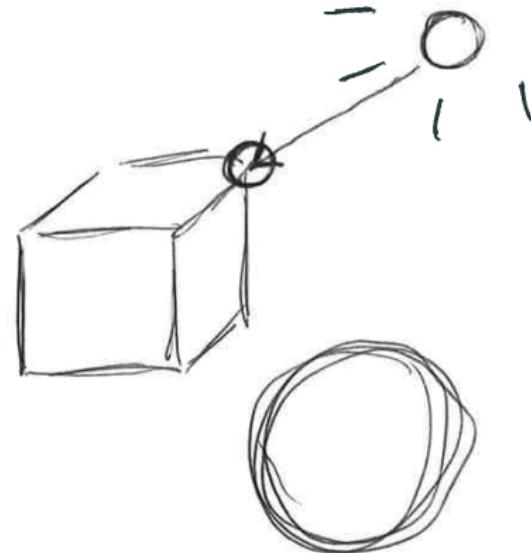
- Model & View Transformation
- Lighting
- Projections
- Clipping
- Screen Mapping

Rendering Pipeline

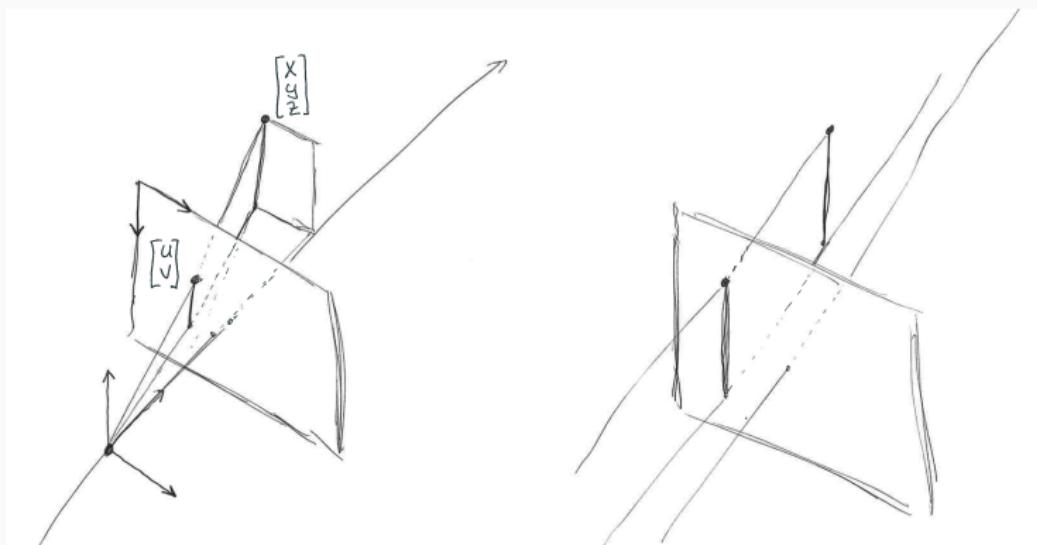


Transform world (objects) to camera-space

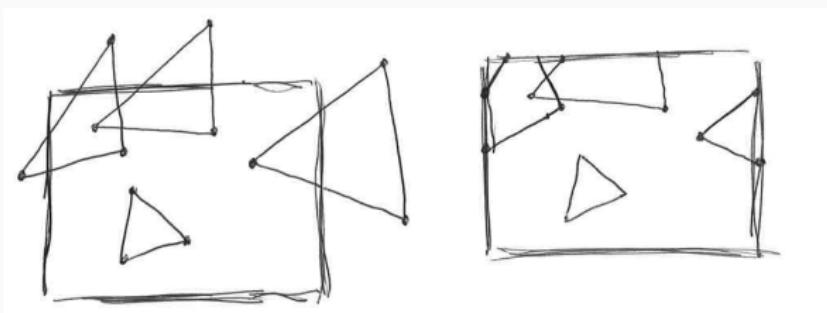
Rendering Pipeline



Rendering Pipeline



Rendering Pipeline

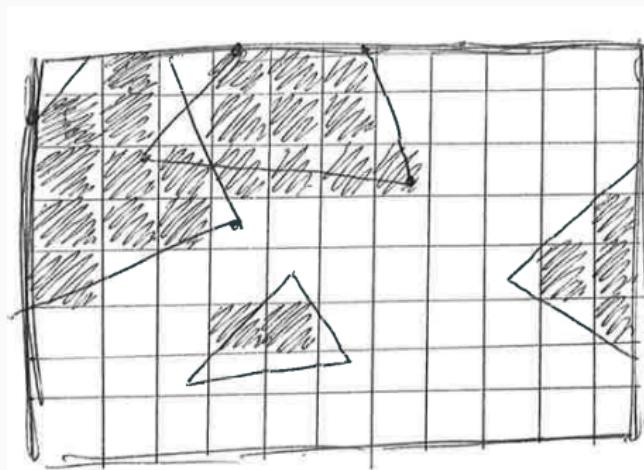


Rendering Pipeline

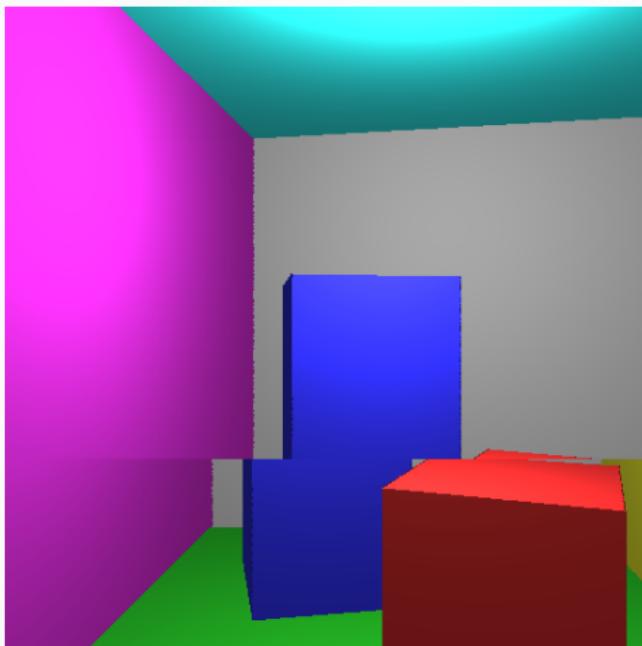


- Scan-conversion
- Draw each pixel

Rendering Pipeline



Rendering Pipeline



Summary

Summary

- An image is an array of pixels
- A pixel encodes a colour
- What are image and pixel spaces?
- We do not see an image, we see an interpretation
 - *can we use this to make assumptions?*
- We are building a generative model.
- How is the pipeline (generative model) implemented?

Next Time

- Now we are set-up for learning about computer graphics!!
 - much less abstract from now on
- Till next time Finish Lab 0
- Lecture Monday 29th of January
 - Raytracing
- Lab Monday
 - Start with coursework 1

eof