

# SM 14/15 – T4

## Special Effects

LCC, MIERSI

*Miguel Tavares Coimbra*

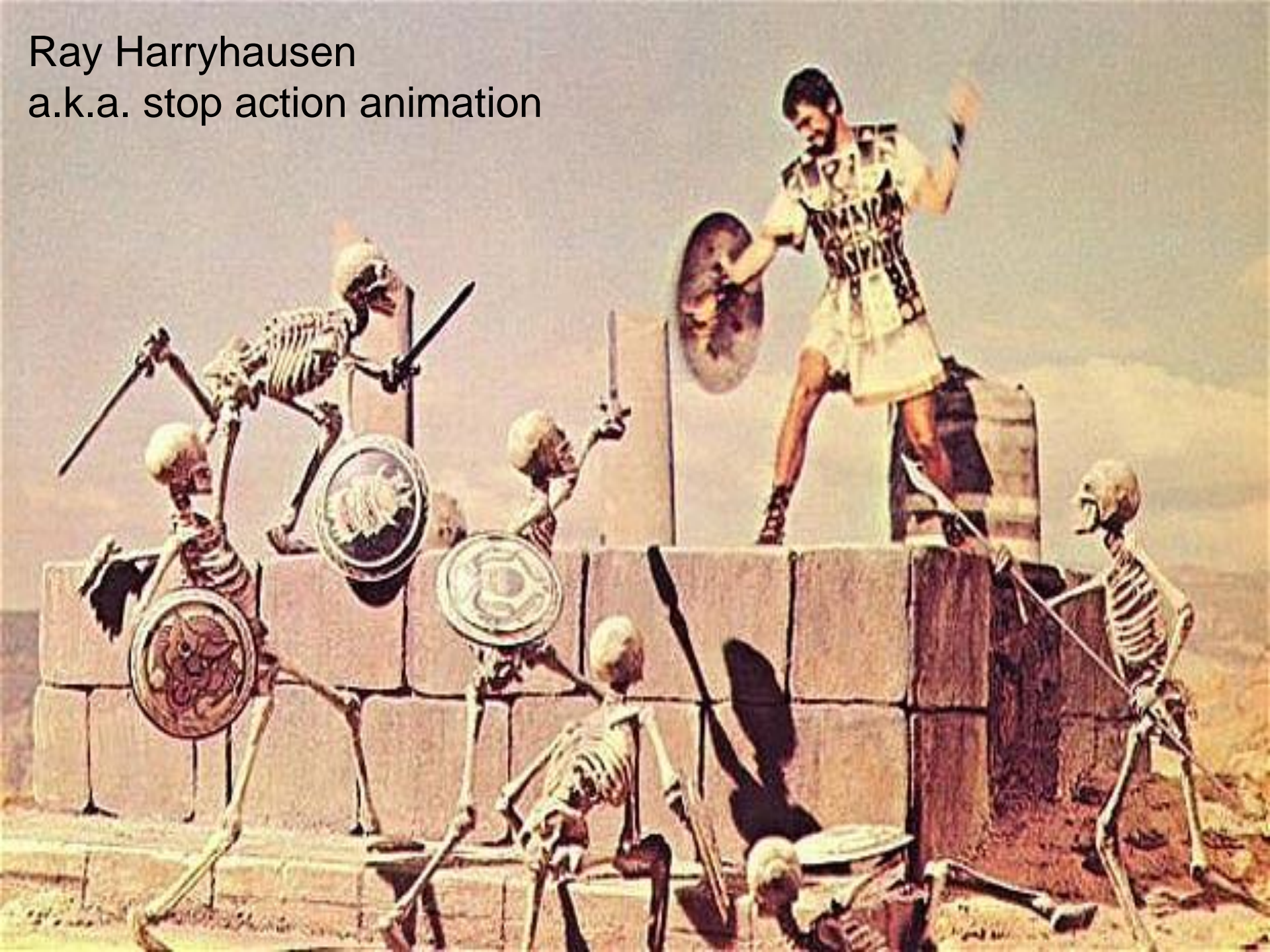


Image Processing



Computer Graphics

Ray Harryhausen  
a.k.a. stop action animation



‘Special Effects’ can mean a  
lot of things

# Today

- We will talk about **image processing**
- Computer graphics is in Lecture 8
- We will not talk about stop-action animation
  - But you should go and see “Jason and the Argonauts” anyway
  - <http://www.rayharryhausen.com/>

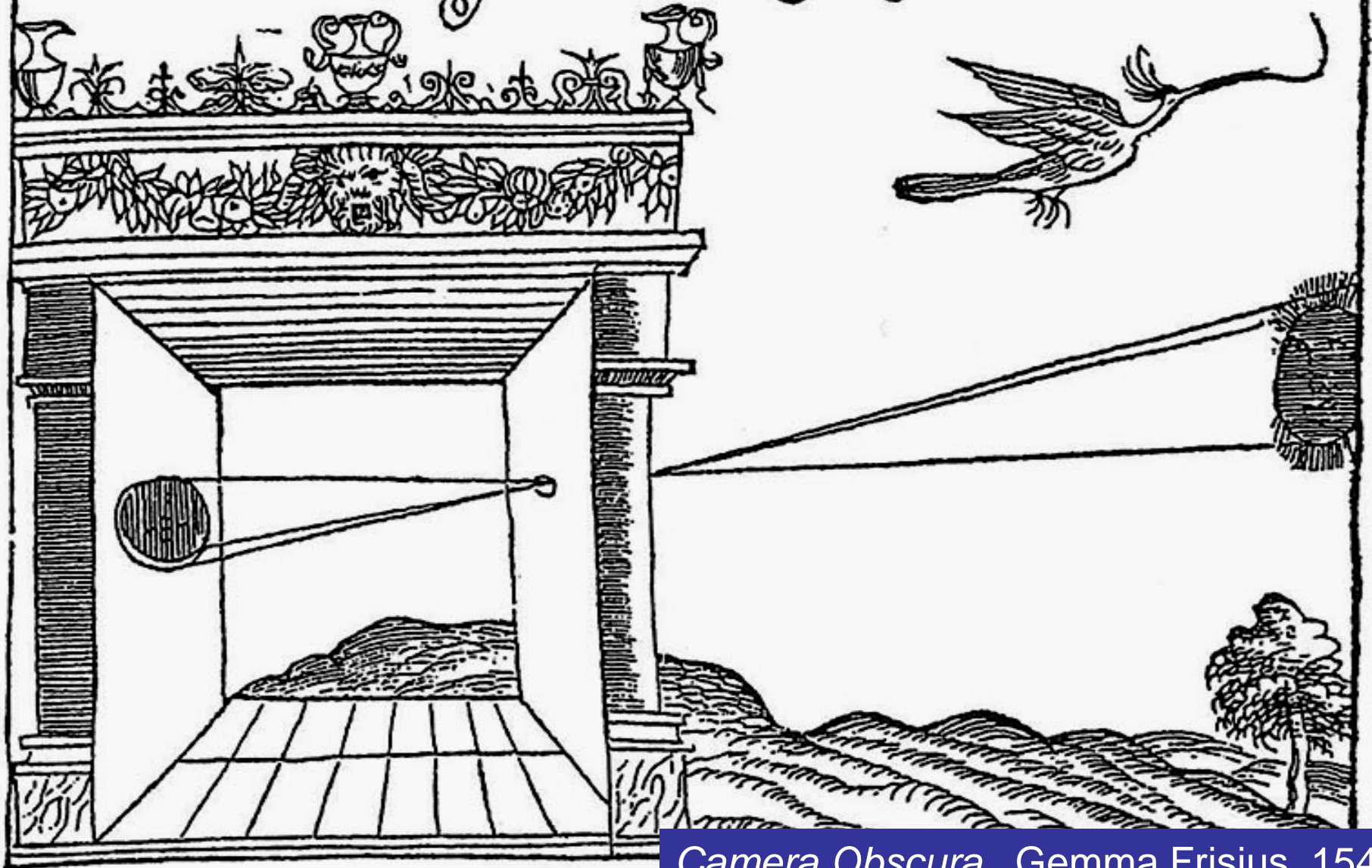
# (Some) Pieces of the Puzzle

- **Image creators (3D -> 2D)**
  - Camera (Today)
  - Computer Graphics (T8)
- **Image manipulators (2D -> 2D)**
  - Image Processing (Today)
- **Image displays (2D -> ?)**
  - 2D Screen
  - 3D Virtual Reality (T7)

How do we get 2D images of  
a real 3D world?



Solis deliquium Anno Christi 1544,  
Die 24 Januarij Louanij



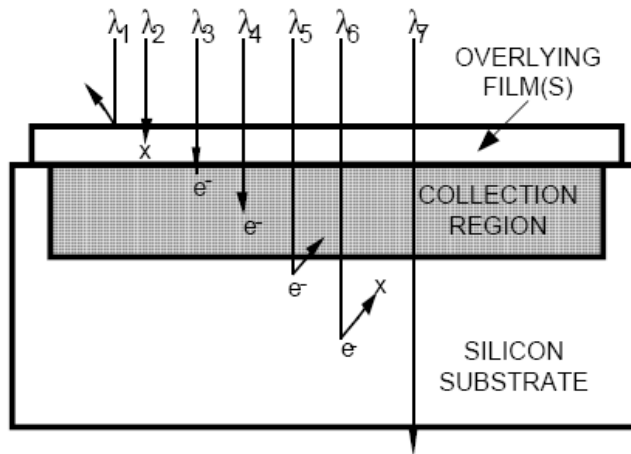
Camera Obscura, Gemma Frisius, 1544





# Image Sensors

- Convert light into an electric charge

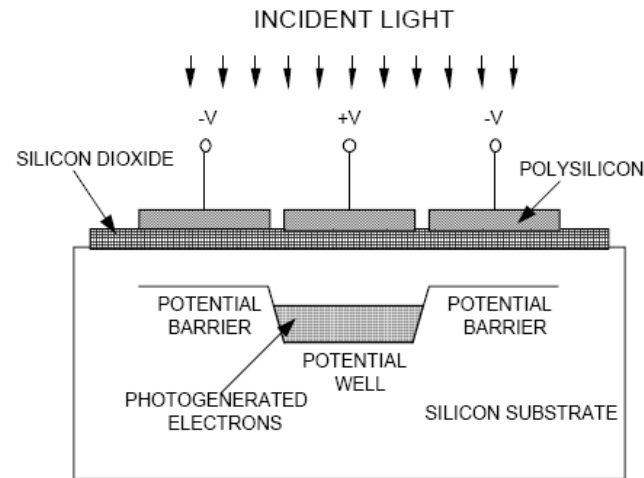
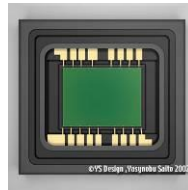


CCD (charge coupled device)

Higher dynamic range

High uniformity

Lower noise

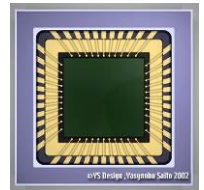


CMOS (complementary metal  
Oxide semiconductor)

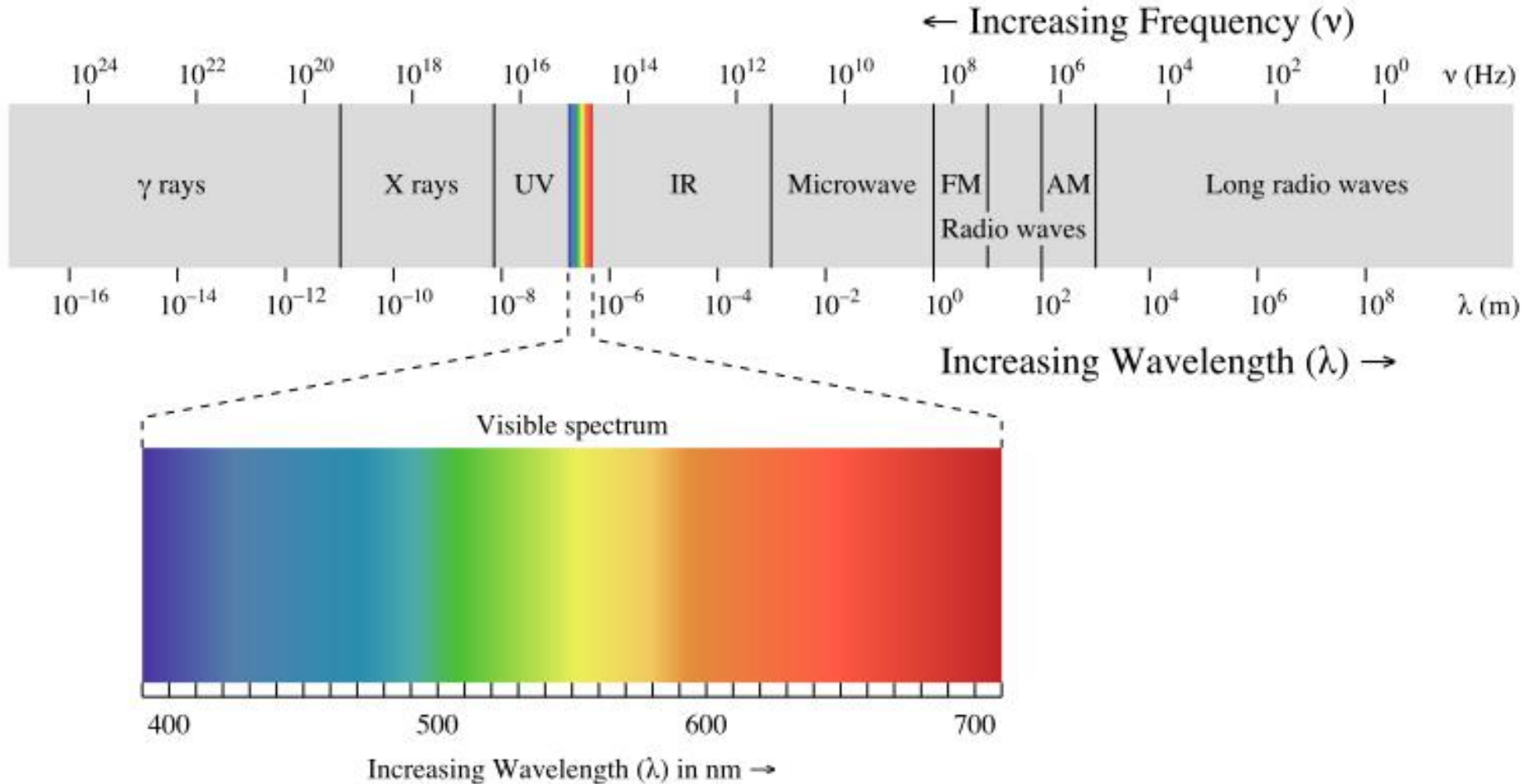
Lower voltage

Higher speed

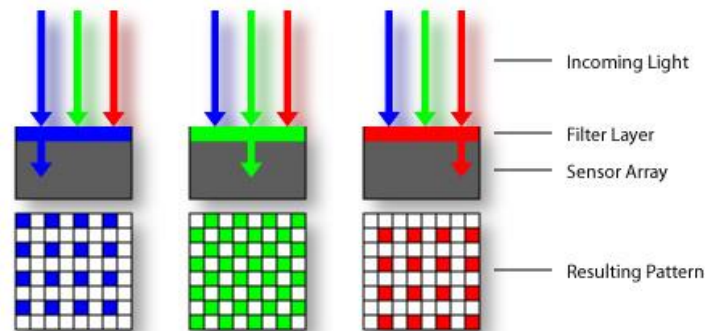
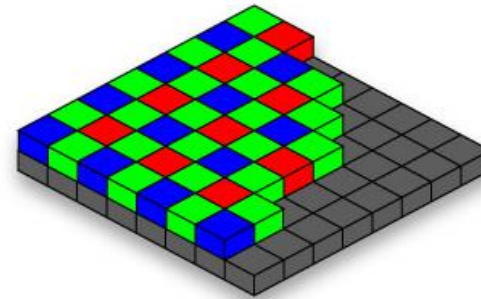
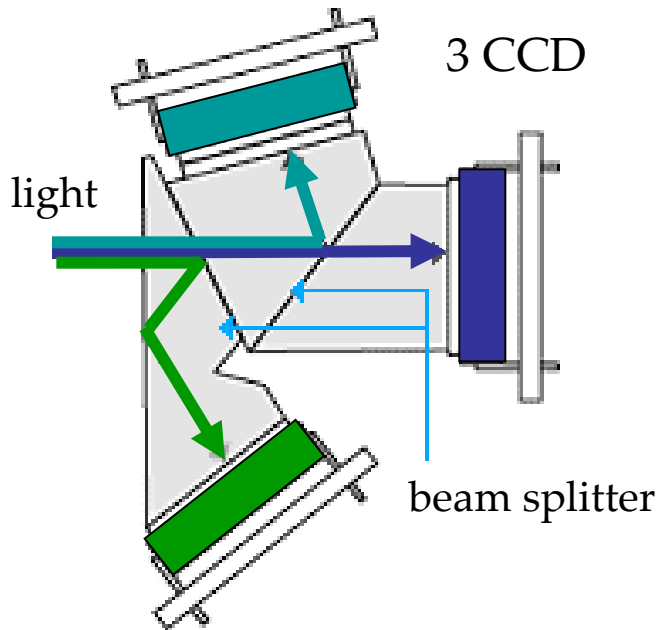
Lower system complexity



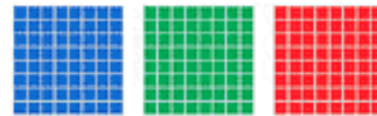
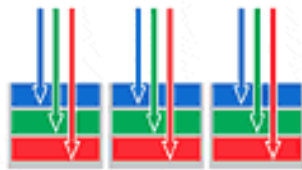
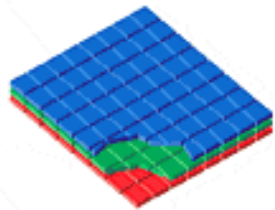
# What is Colour?



# Sensing Colour



Bayer pattern

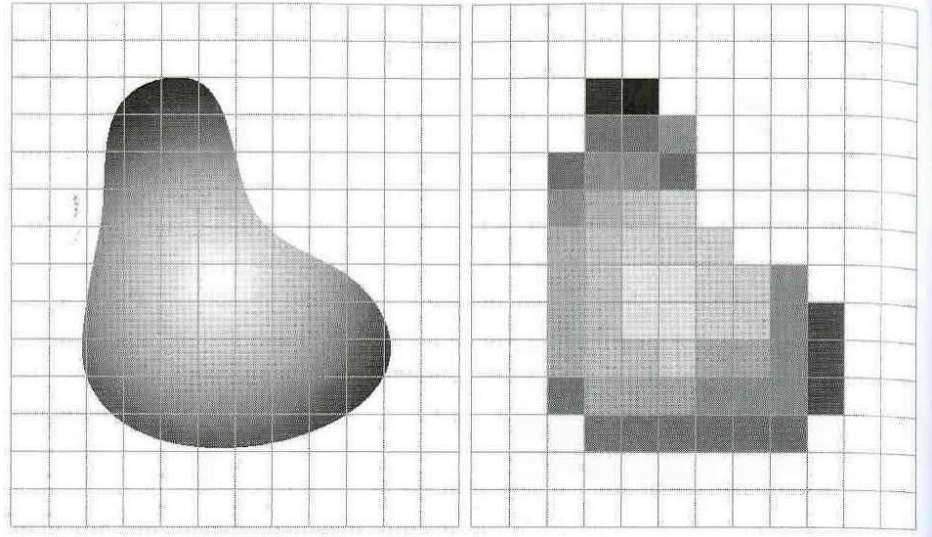


Foveon X3™

# Analog to Digital

The scene is:

- **projected** on a 2D plane,
- **sampled** on a regular grid, and each sample is
- **quantized** (rounded to the nearest integer)

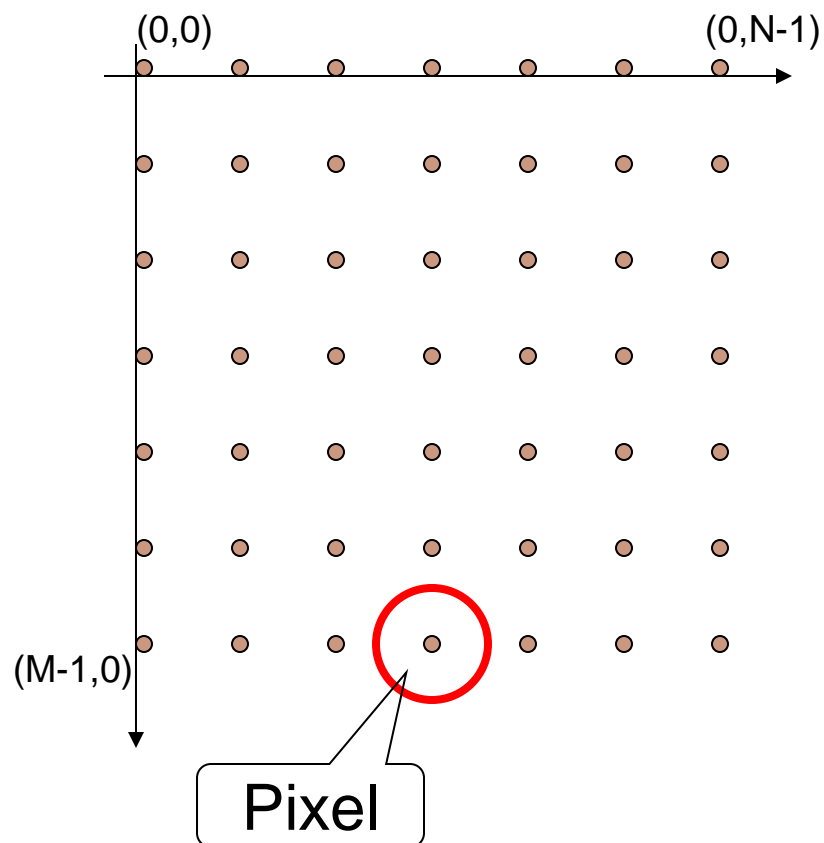


$$f(i, j) = \text{Quantize}\{f(i\Delta, j\Delta)\}$$

# Images as Matrices

- Each point is a **pixel** with amplitude:
  - $f(x,y)$
- An image is a matrix with size  $N \times M$

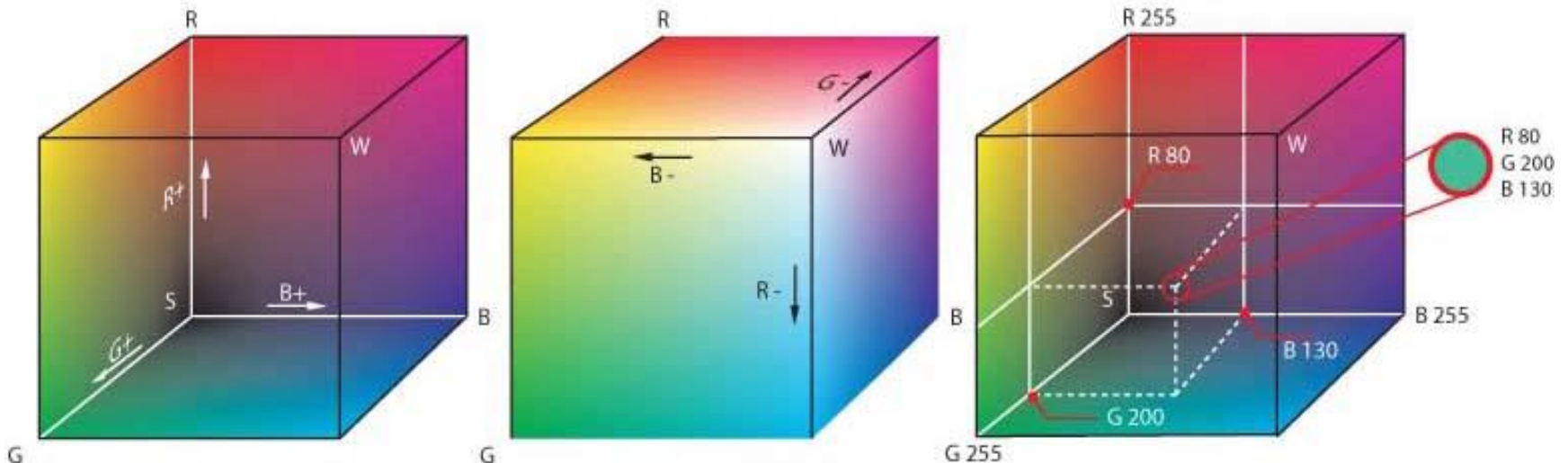
$$M = \begin{bmatrix} (0,0) & (0,1) & \dots \\ (1,0) & (1,1) & \dots \\ \dots & & \end{bmatrix}$$





# Colour Space

- **Colour space**
  - Coordinate system
  - Subspace: One colour -> One point
- **RGB is very popular**



# Manipulating Single Pixels

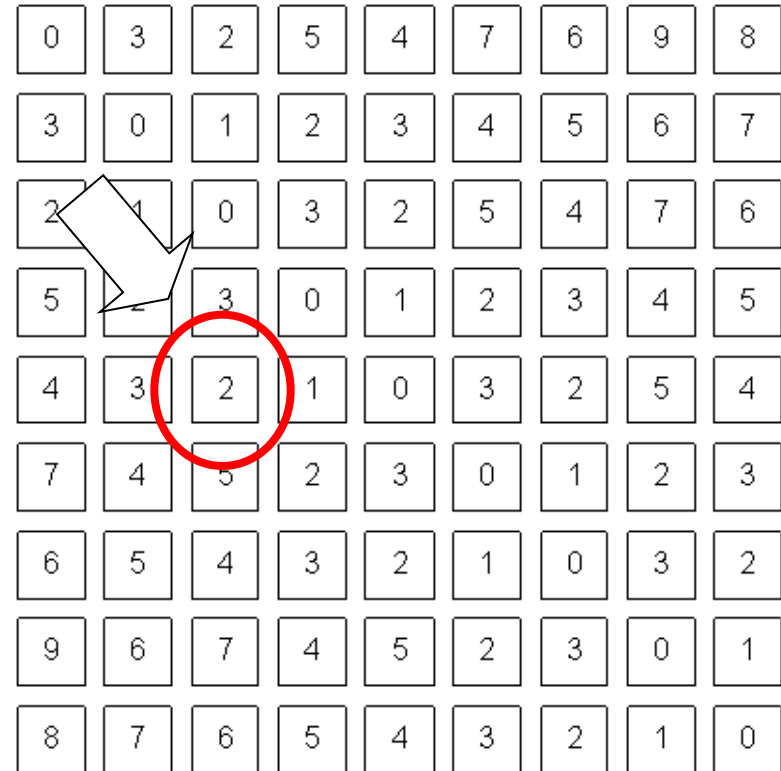
# Pixel Manipulation

- Let's start simple
- I want to change a single Pixel.

$$f(X, Y) = MyNewValue$$

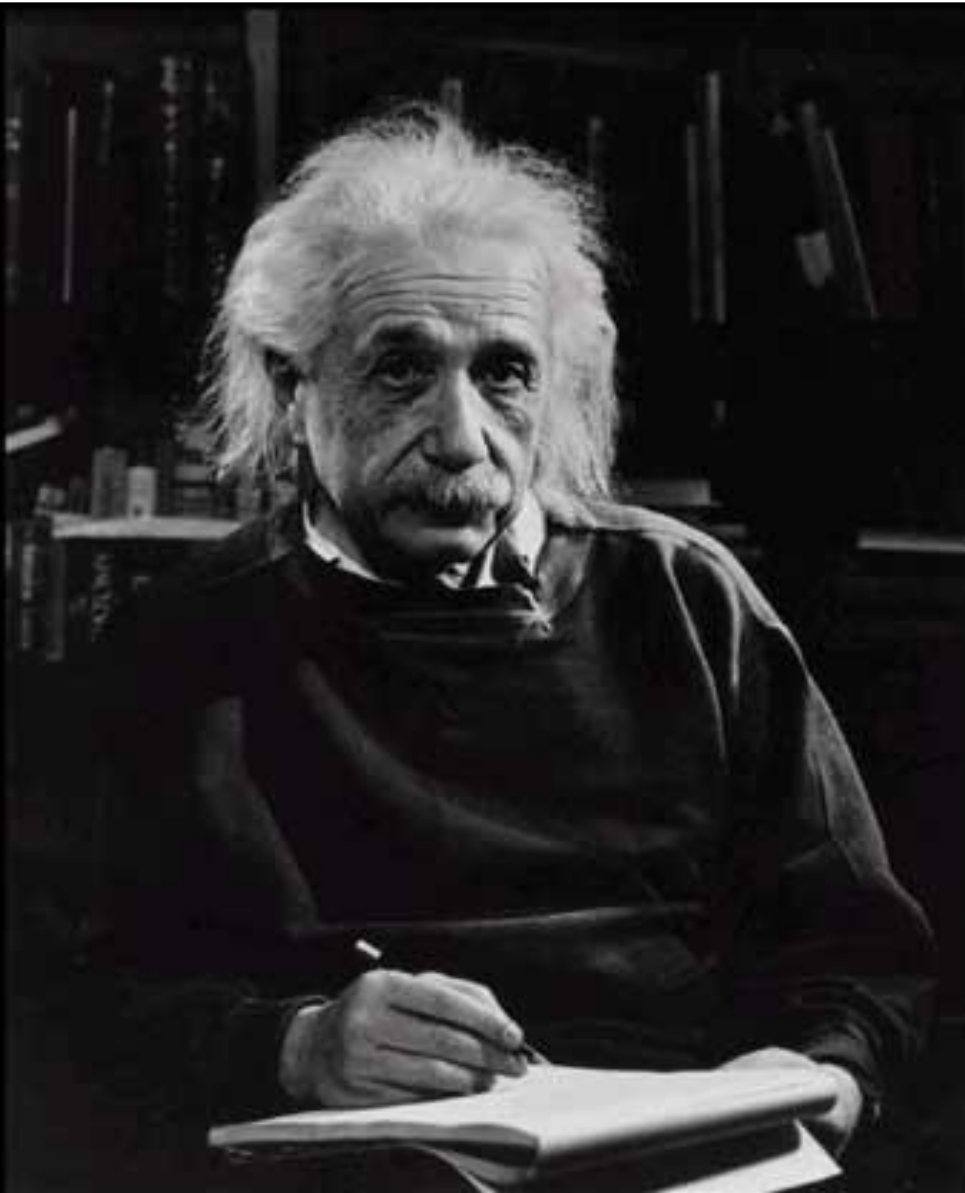
- Or, I can apply a transformation **T** to all pixels individually.

$$g(x, y) = T[f(x, y)]$$

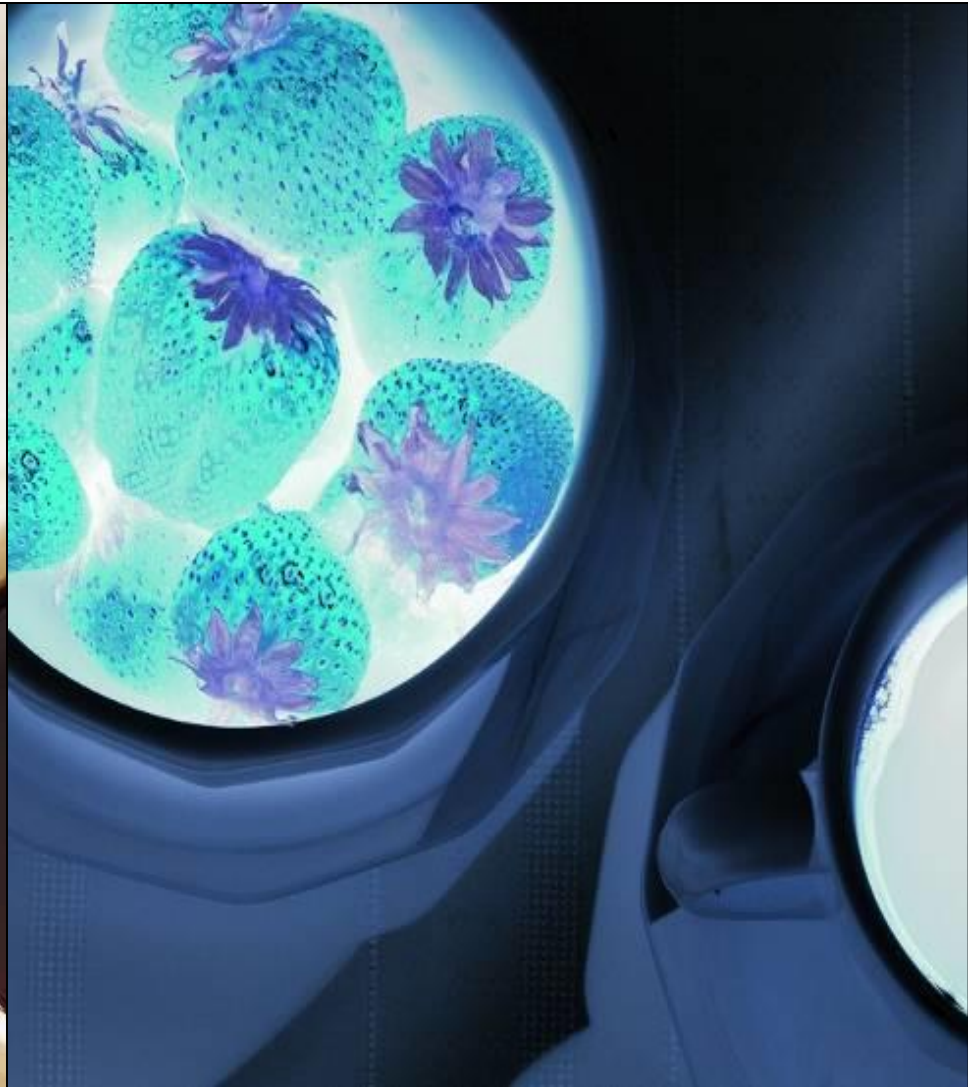


0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	4	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

# Negative



# Colour Negative



# Pseudocolour



# Colour Slicing



# Chroma Key

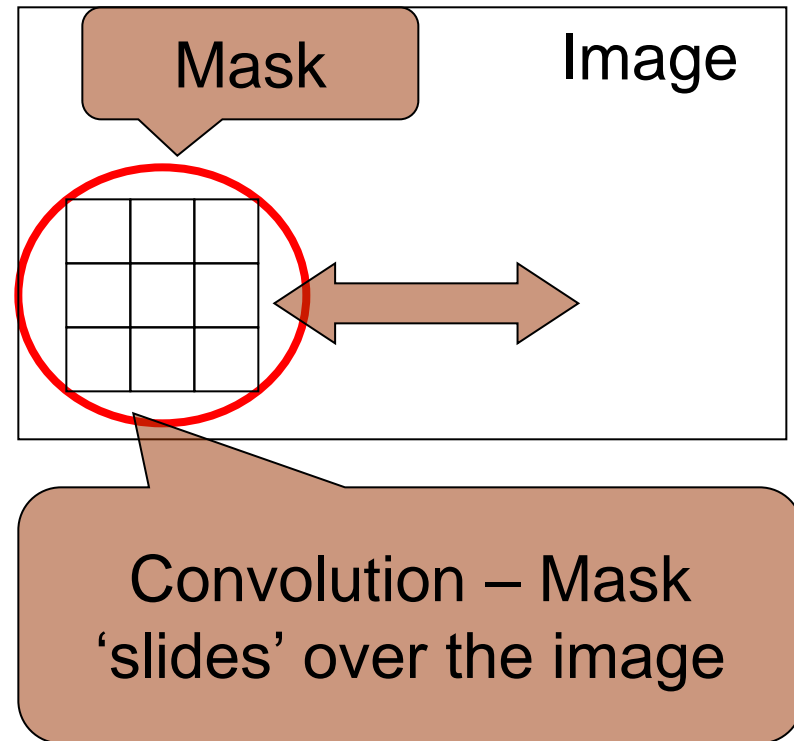




# Digital Filters

# Convolution with a Filter Matrix

- Simple way to process an image.
- Mask defines the processing function.
- Corresponds to a multiplication in frequency domain.

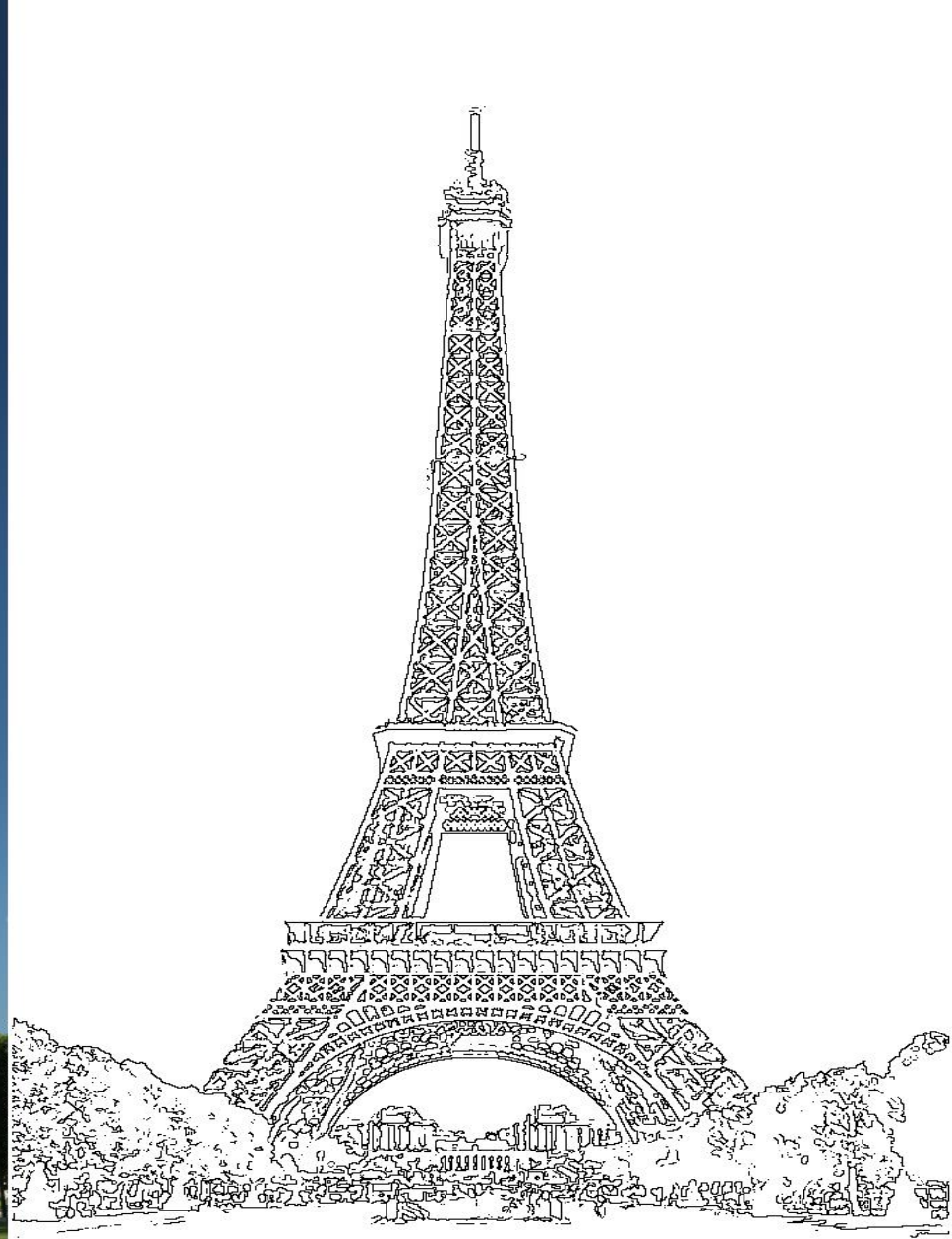


# Depth of Field Blurring



# Anti-Aliasing





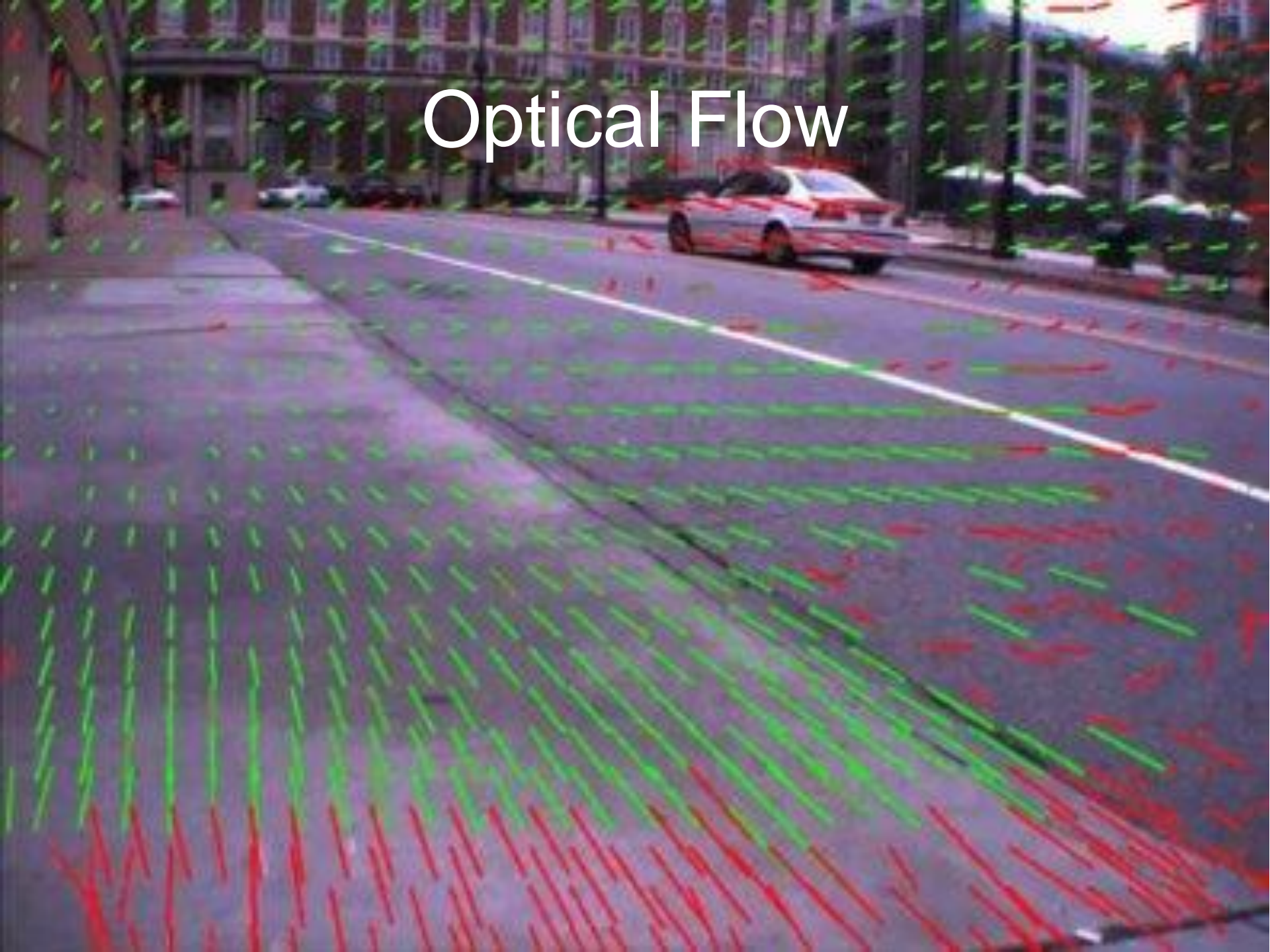
Edge Detectors



Colour Edge Detectors

# Advanced Processing

# Optical Flow







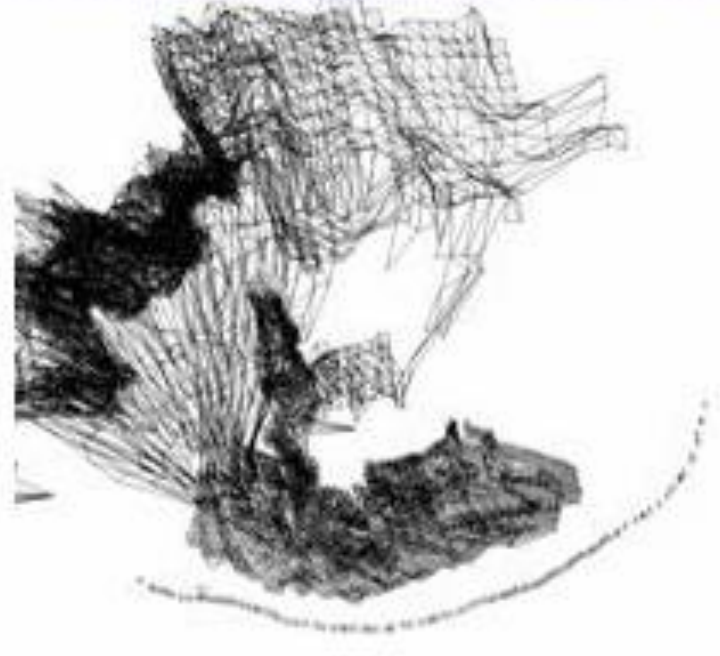
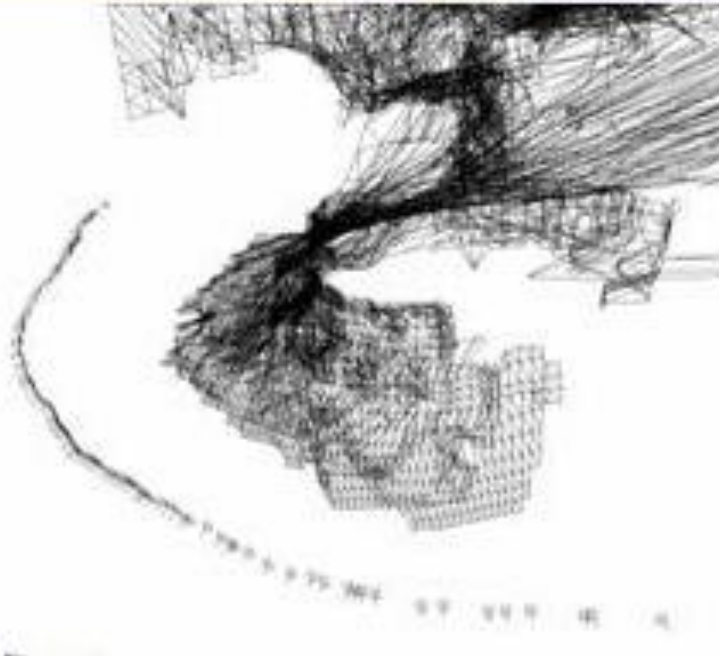
X-Rate o

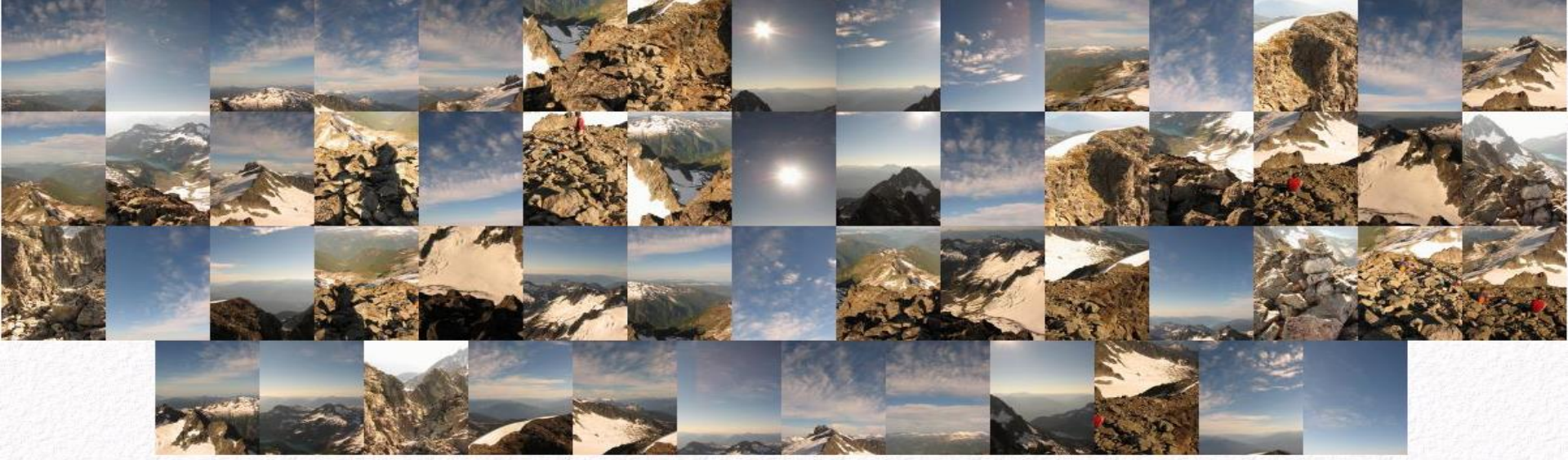
Tavollaw 3pgo



# Motion Quantification

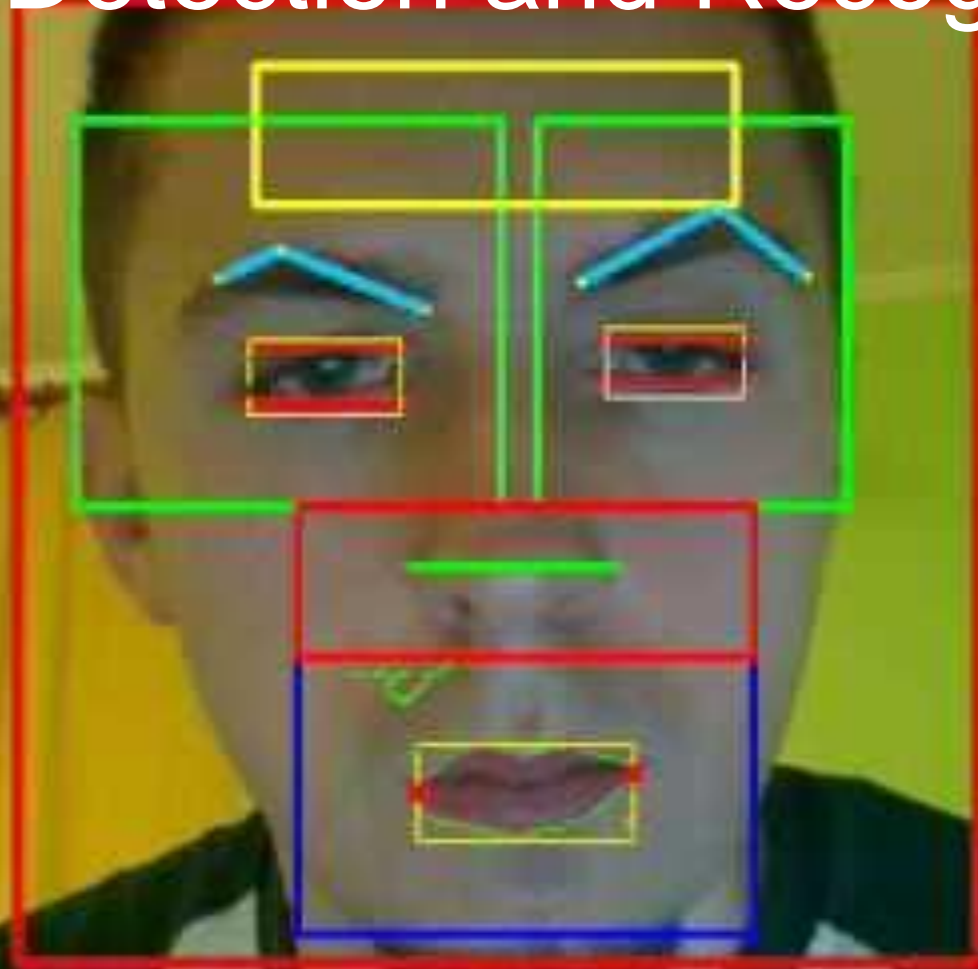
# Structure from Motion



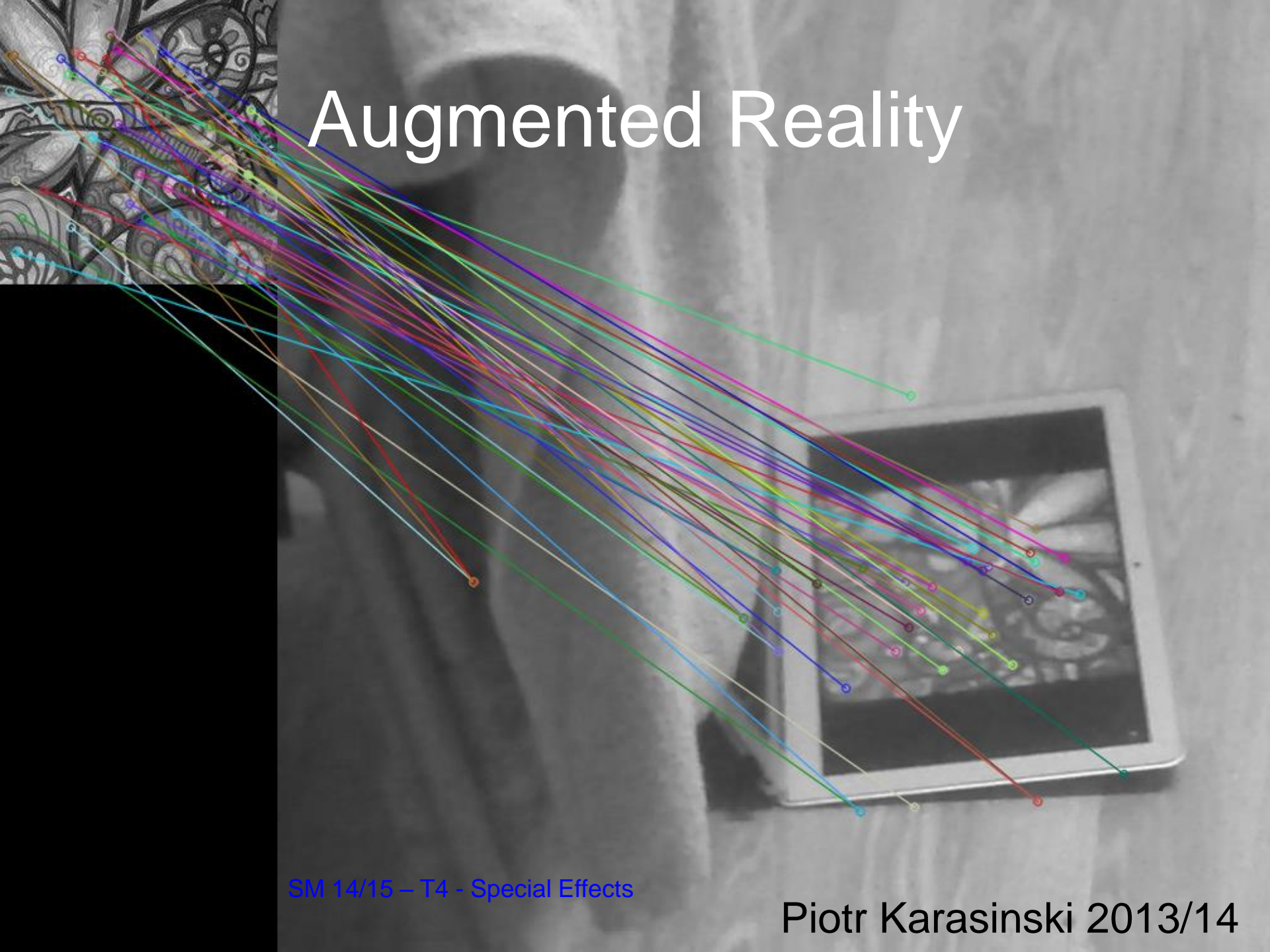


Mosaicing

# Facial Detection and Recognition



# Augmented Reality



# Crazy Stuff

That didn't really fit anywhere  
else...

# Visão Computacional - EV3

## Multimédia

Enviar som

Saudação

## Sensores

Cor:

IV:

Toque:

Cor:  IR:  Toque:

## Motores

Velocidade:



Direção:



Polaridade:



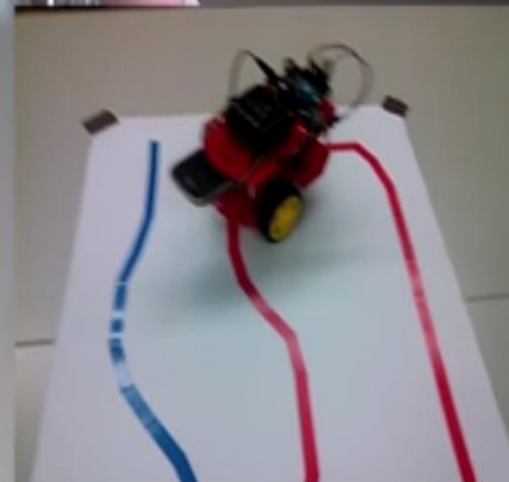
Lego Mindstorm with a smartphone camera

# Demonstration:

# Virtual Joystick

The screenshot shows a software application titled "ROBOT CONTROL" with a "Form1" window. The interface includes a "Parameters selected" panel with fields for "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59", "60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70", "71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81", "82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92", "93", "94", "95", "96", "97", "98", "99", "100". A "STOP" button is visible. A video feed window shows a person's hand holding a small yellow object with a green dot, overlaid with a white grid. A blue and yellow virtual joystick is positioned over the hand. A console window at the bottom displays the following text:

```
WindowsFormApplication1.robot.exe (32) v4.8.38109: WindowsFormApplication1.robot.exe) Loaded  
WindowsFormApplication1.robot.exe (32) v4.8.38109: WindowsFormApplication1.robot.exe) Loaded  
WindowsFormApplication1.robot.exe (32) v4.8.38109: WindowsFormApplication1.robot.exe) Loaded  
The thread 80116 has exited with code 259 (0x0101).  
WindowsFormApplication1.robot.exe (32) v4.8.38109: WindowsFormApplication1.robot.exe) Loaded  
The thread 80116 has exited with code 259 (0x0101).  
The thread 80116 has exited with code 259 (0x0101).  
The thread 80116 has exited with code 259 (0x0101).
```





How do I do all this?

# Platforms and Source Code

- **Computer Vision DCC**

- Lecture notes
- JAVA platform
- Android platform

[http://www.dcc.fc.up.pt/~mcoimbra/lectures/vc\\_1415.html](http://www.dcc.fc.up.pt/~mcoimbra/lectures/vc_1415.html)

- **OpenCV**

- Free to use, lots of algorithms, C

<http://opencv.org/>

- **Gonzalez & Woods book**