GRAPHICS GROUP

Intelligent Vision Systems New Zealand

Computer Graphics and Image Processing Introduction

Part 1 – Lecture 1

COMPSCI 373

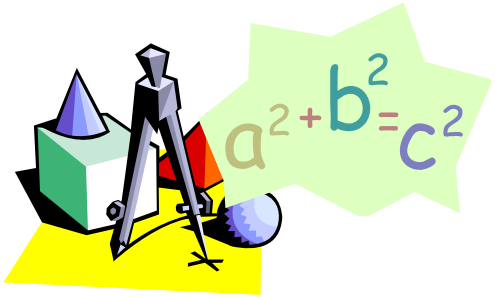
- **Lecturers: Dr Patrice Delmas (303.389)**
  - Contact details:
    p.delmas@auckland.ac.nz
    Office: 303 - 391 (3rd level CompSci building)

  A. Prof. Georgy Gimel'farb (303.391)

  Dr Burkhard Wuensche (303.453)
- Office hours: whenever the door is open
- Particularly available right after each lecture
- **Lecture time: Monday 5pm-6pm Wed/Fri 8am-9am**
- **Marking:**
  - 25% assignments, 20% test, 55% examination
- **Assignments:**
  - 8.33 marks each (date out and due date online)

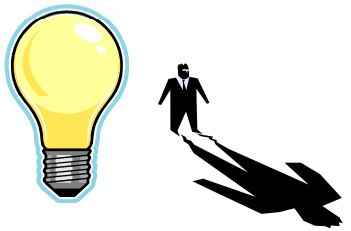
Part I Overview

Week 1: 2D and 3D Geometry

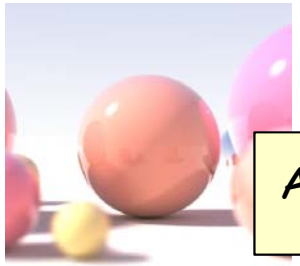


Week 2: Color Theory (Timo)

Week 3: Illumination and Shading



Week 4: Ray Tracing

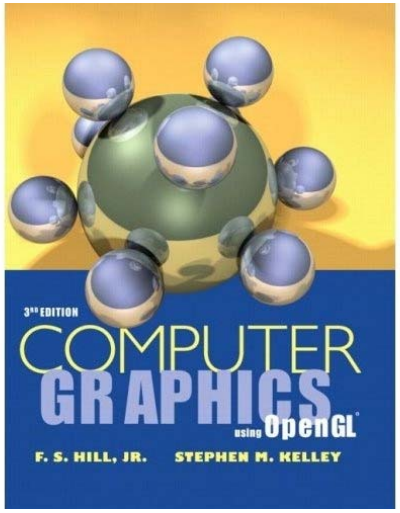


Assignment 3

Raytracing image thanks to Tim Babb

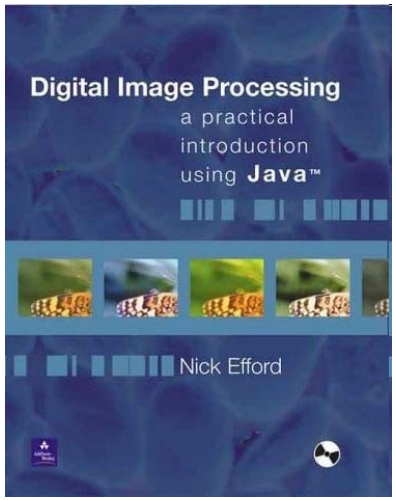
References

- F.S. Hill. Computer Graphics using OpenGL (2nd or 3rd Edition). Prentice Hall.
- Nick Efford. Digital Image Processing: A Practical Introduction Using Java. Addison Wesley.



C/C++:

- C Reference Manual
  http://www.cs.bell-labs.com/who/dmr/cman.ps
- Bruce Eckel - Thinking in C++
  http://www.ibiblio.org/pub/docs/books/eckel/
- C++ for Java Programmers
  http://www.jgcampbell.com/cpp4jp/cpp4jp.pdf
- See also CS373 resources page



THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

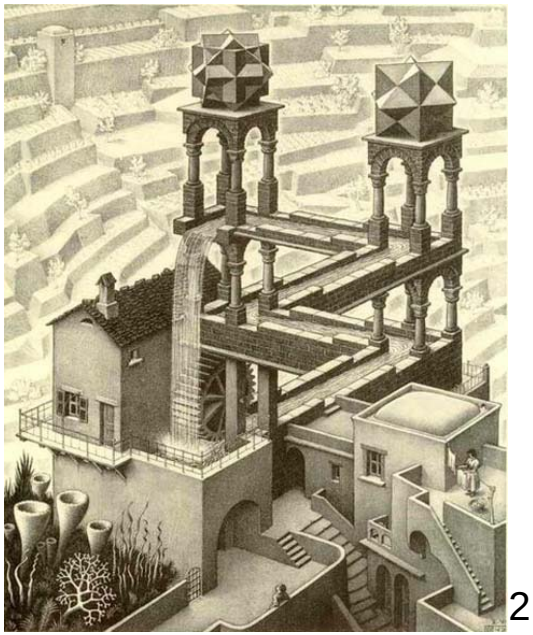
IMPORTANT ANNOUNCEMENT

Departmental Policy on Cheating on Assignments

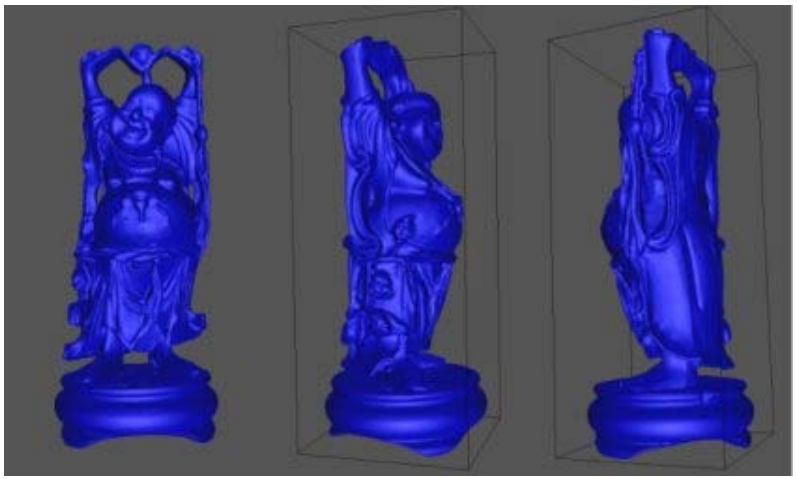
1. ***The Computer Science Department uses many ways to check that the work students submit for marking is their own and was not produced by, or copied from, someone else. In particular, for most programming assignments, the department uses a program comparison program to automatically compare all submissions from students. Also Turnitin.com may be used on essays and reports. This detects similarity to online material and submitted works in its own database.***
2. ***All assignments where plagiarism is detected are checked for similarity by the course supervisor or another suitable person associated with the course.***
3. ***New policy on cheating means you may run into very serious trouble (beyond just a 0 mark to your assignment)***

For more details see

http://www.cs.auckland.ac.nz/administration/policies/CheatingPolicy.php



2D



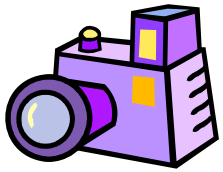
3D

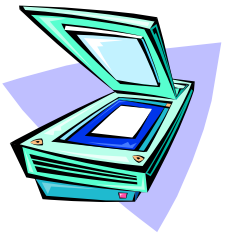
4D

DIGITAL IMAGES

How to Get Digital Images?

1. **2D real → 2D digital**
2. **3D real → 2D digital**
3. **4D real → animated 2D digital**

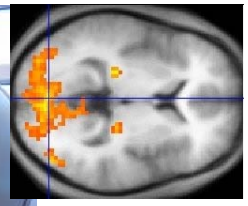




4. **3D real → 3D digital**

Stereoscopic Camera

3D Scanner



MRI

2D slices
of 3D volume

5. **4D real → 4D digital**

Motion Capturing

6. **Synthetic** or **computer generated**

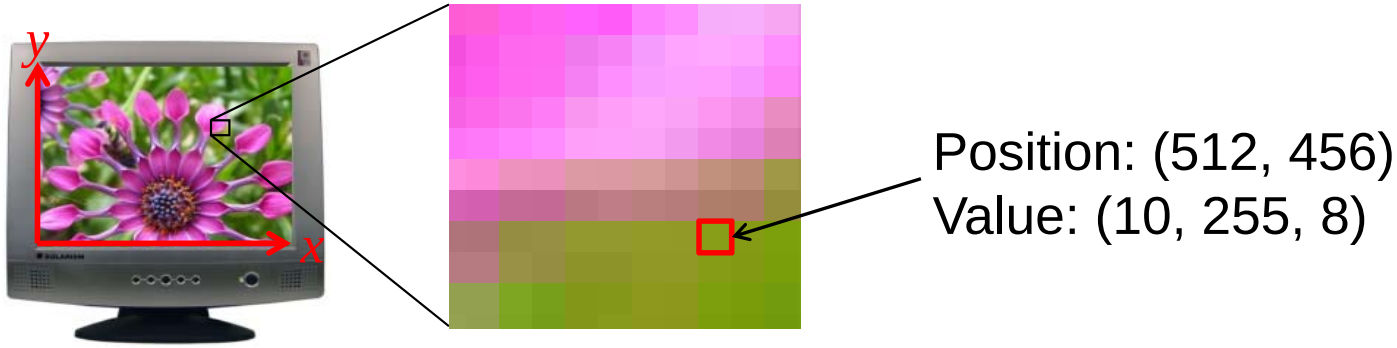
What to Do with Digital Images?

- **Image Processing**: process them to get new images
- **Computer Vision**: analyze them to get information about what is in the image
- **Computer Graphics**: generate them

		OUTPUT	
		Descriptions	**Images**
INPUT	**Descriptions**		Computer Graphics
	Images	Computer Vision	Image Processing

Pixels and Resolution

- **Pixel** or **pel** (picture element)
  Position (x,y) + signal value v (greyscale or colour)



  Origin (0,0) of pixel coordinates sometimes in top left corner

- **Resolution:** how many pixels? width × height
- **Spatial resolution**: image pixels per cm or inch (in x and y)
  Can be used to convert pixel coordinates into physical coordinates

Encoding of Colors

- **Bit-depth:** number of bits used to represent each pixel's value (typically 1, 8, 24 or 32)
- **Binary image:** bit depth is 1;
  only code values 0 (black) and 1 (white)
- **Scalar/monochrome/greyscale image:**
  - scalar code values
    (e.g. just a single number per color)
  - only grey values (from black to white)
    and no colour
- **Vector-valued image**
  - vector code values
    (e.g. several numbers per color)
  - All the colors can be represented

Defining Images Mathematically

Images can be defined on an $M \times N$ arithmetic grid (or lattice)

$$\mathbf{R}_{M,N} = \{(x, y) : 1 \le x \le M \ \wedge 1 \le y \le N\}$$

- Pixel coordinates x and y with $x = 1,\ldots,M$; $y = 1,\ldots,N$
- Image as a function f: $\mathbf{R} \rightarrow \mathbf{V}$
- $\mathbf{V}$ is a set of signal values, e.g. grey levels or colors
- **Example**: pixel at position (100, 100) has value 255, i.e. f (100, 100) = 255

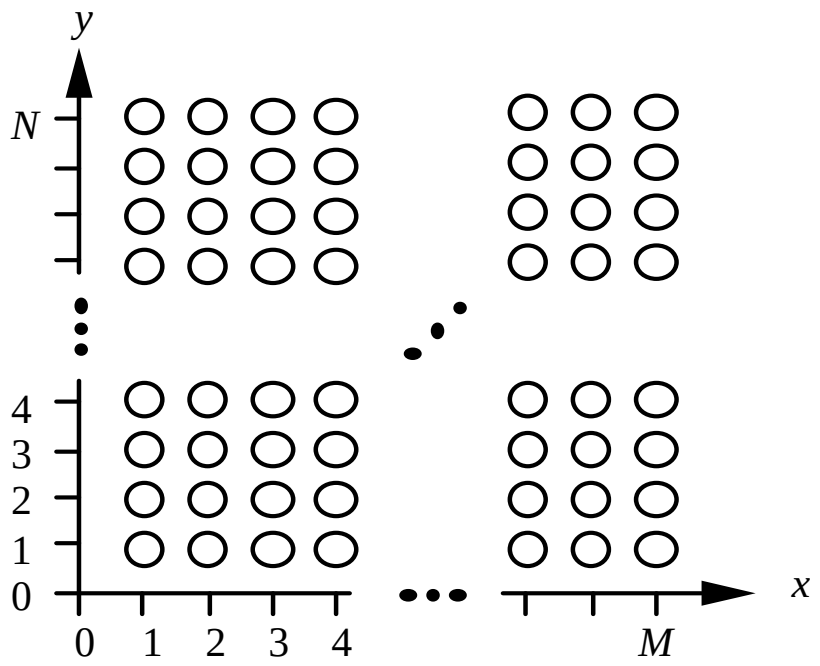
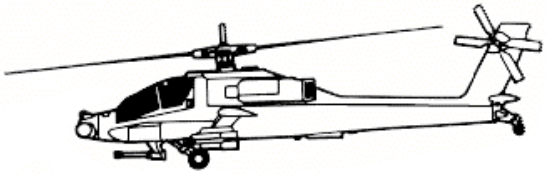


Image Processing

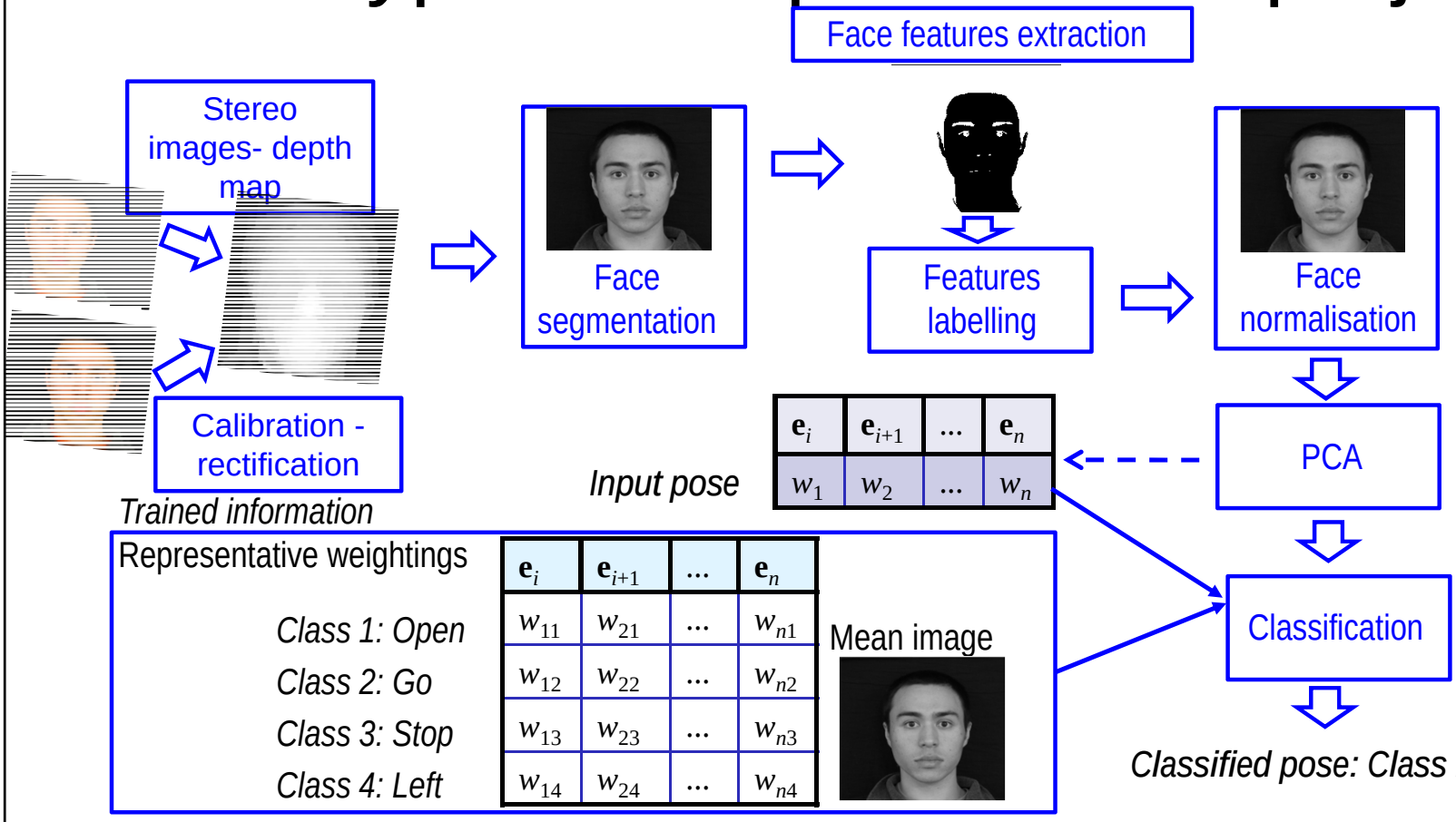
- **Geometric transformations**: resizing, rotation, deformations, …
- **Color transformations**: quantization, conversion, color adjustment, …
- **Compositing**: combination of two or more images
- **Many other operations**

Computer Vision

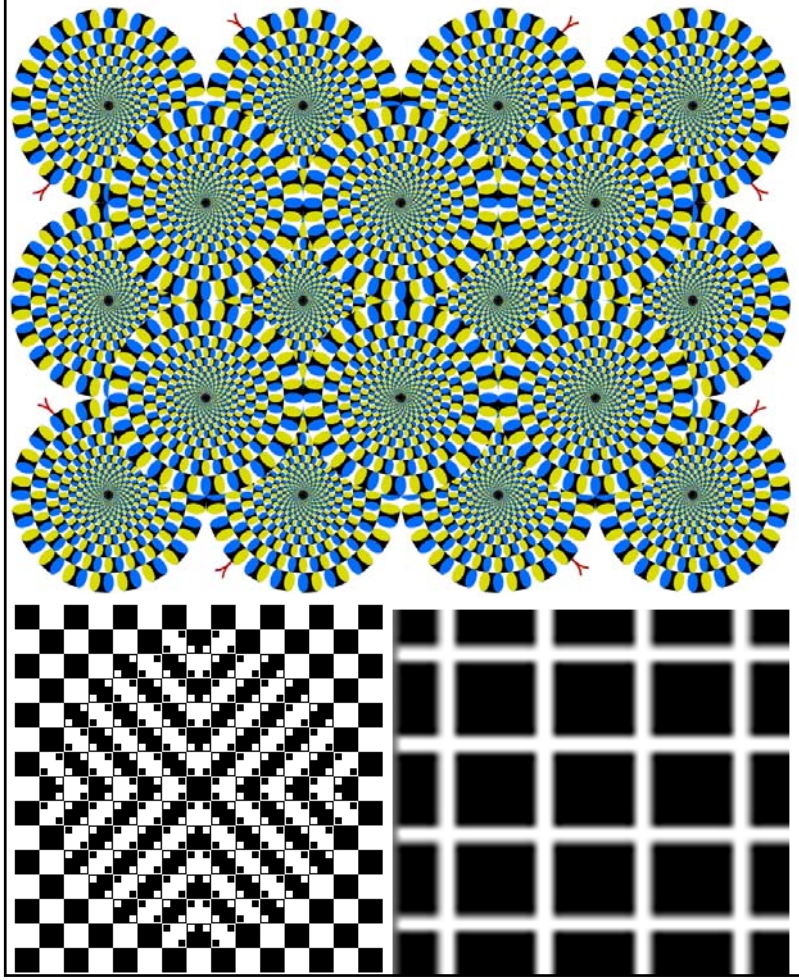
- Use computer to do things similar to human vision, using image processing, artificial intelligence, biology & physics
- Usually dealing with 2D images of 3D scene
- Often real-time and part of a larger system (e.g. robot)
- E.g. scene reconstruction, image restoration, object recognition, tracking, motion estimation, event detection, …



A Typical computer vision project



Human vs. Computer Vision



Human Vision

- Subjective, unstable, inaccurate in measurements
- Involves active interaction with environment
- Exploits experience, knowledge, context
- But unique capabilities to describe and understand…
- Real-time

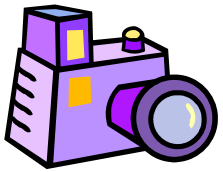
Computer Vision

- Objective accurate measurements
- But low capabilities to describe and understand…
- Rarely works effectively real-time

Generating Digital Images

Need a 3D scene consisting of:

- Objects (3D models)
- Light sources
- Camera (viewer, eye)

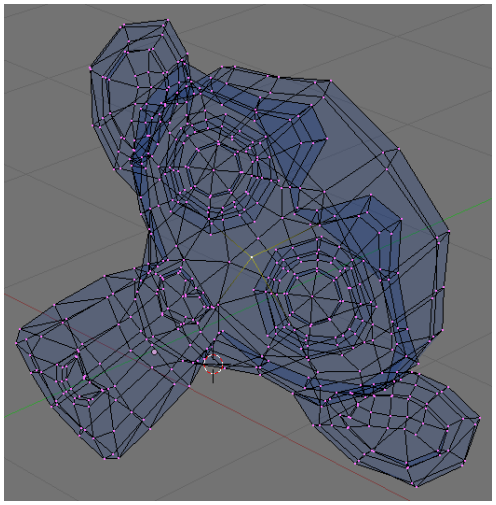


- Parameters specifying how light interacts with the scene
  - Light parameters
    e.g. type of light (light bulb, spotlight), color, brightness
  - Material parameters
    e.g. color, texture, transparency
  - Rendering parameters
    e.g. algorithm, quality, resolution

Objects (3D Models)

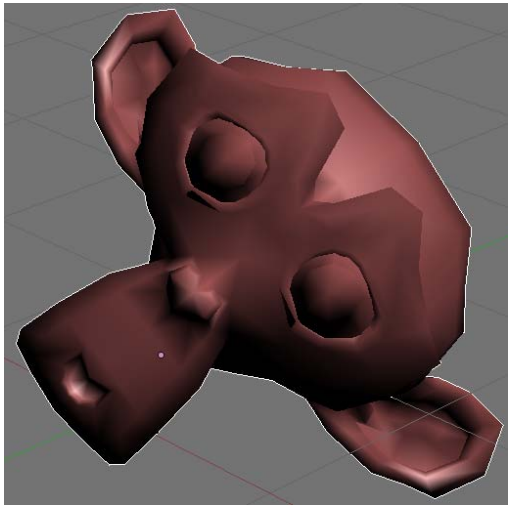
Objects are made up of parts (often also of other objects):

- **Vertices**: 3D points used to define model
- **Edges**: lines between vertices
- **Faces**: polygons bounded by edges
- **Mesh**: surface made up of connected polygons

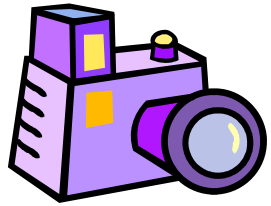


Wireframe model showing mesh with vertices, edges and faces

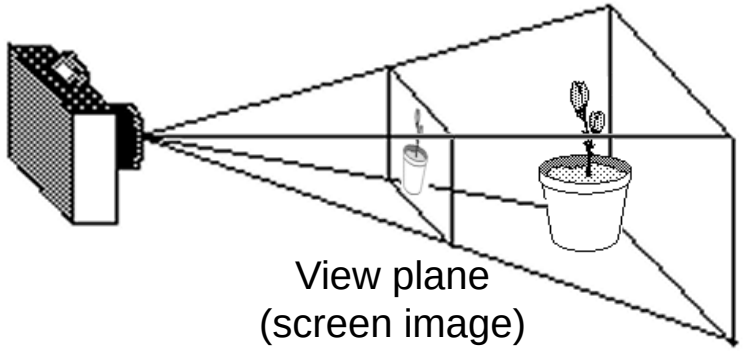
Smoothed and shaded solid model



The Camera



- **View point**: where is the camera?
- **View direction**: which direction does it point?
- **View orientation**: where does the top of the camera point? E.g. is the photo taken upside down?
- **Projection**: what type of lens does the camera have?



- **Rendering**:
  - Project scene onto view plane in front of the camera
  - Compute the color of each pixel on the view plane

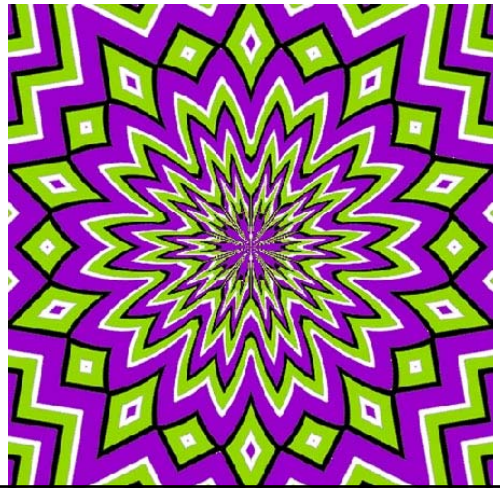
SUMMARY

Summary

- Things we can do with images:
  - **Image Processing**: process them to get new images
  - **Computer Vision / Image Analysis:** analyze them to get higher-level information (context, emotional content, numbering/counting)
  - **Computer Graphics**: generate them
- Images:
  pixels with encoded colors, (spatial) resolution,
  can be represented on arithmetic grid
- For generating images:
  3D scene with objects (vertices, edges, faces, meshes), light sources, camera

Quiz

1. What is the spatial resolution of an image?
2. What is Computer Vision?
3. What are the differences between Computer Vision and human vision?
4. What does a 3D scene consist of?
5. What are vertices, edges and faces?



Super Quiz

1. What is camera calibration?
2. How much time did each CG still shots of Gollum-LOTR take to be generated in 2001?
3. Same question for the Hobbit
4. What is the cheapest 3D camera commercially available if any?
5. Cite 3 famous (most-used) software packages for image processing/computer graphics?
6. What was the price of a medium ranged 2Mpixels camera in 2002?