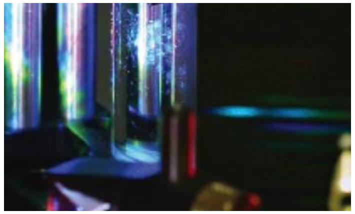
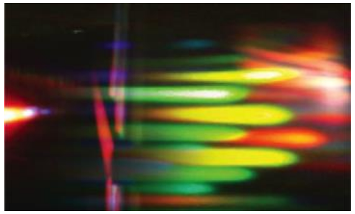
Imperial College London





CO316 Computer Vision

Lecture 1 - Introduction to Vision Systems

Su-Lin Lee, PhD
su-lin.lee@imperial.ac.uk

Prof Guang-Zhong Yang
g.z.yang@imperial.ac.uk

The Hamlyn Centre
for Robotic Surgery

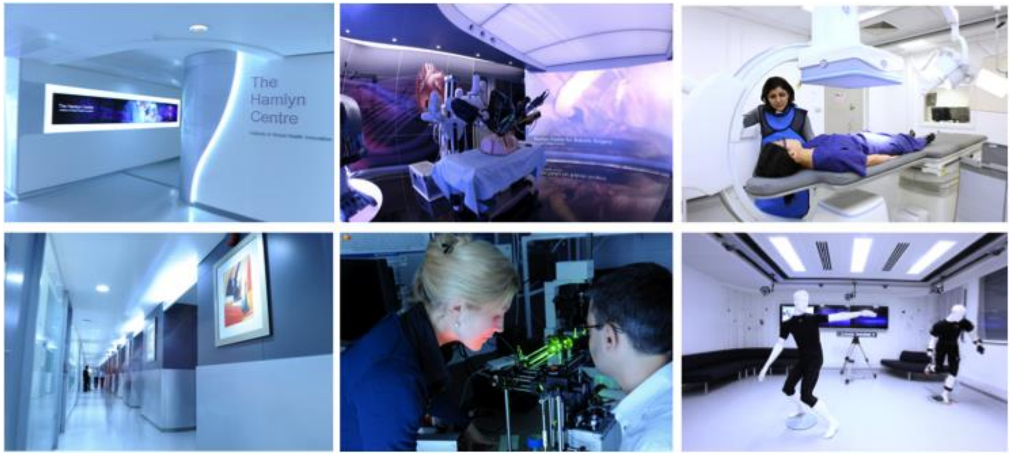
Aims, Structure and Resources

- To introduce the concepts behind computer-based recognition and extraction of features from raster images;
- To illustrate some successful applications of vision systems and their limitations;
- 20 lectures, 7 tutorials, 2 self-study and revision weeks;
- Lecture Slides – CATE (https://cate.doc.ic.ac.uk; CO316), which also includes assessed coursework details;
- Main references:
  - **Image Processing, Analysis, and Machine Vision** by Milan Sonka, Vaclav Hlavac, Roger Boyle
  - **Computer Vision** by Linda Shapiro and George Stockman
  - **Computer Vision** – a modern approach by David Forsyth and Jean Ponce
  - **Machine Vision** by Ramesh Jain, Rangachar Kasturi, Brian G. Schunck
  - **The Computer Vision Homepage** (http://www.cs.cmu.edu/~cil/vision.html)
  - **OpenCV** (http://opencv.org)

Topics

- Introduction to Vision Systems
- Pre-processing for Vision 1
- Pre-processing for Vision 2
- Shape Recognition
- Fourier Methods
- Hough Transform
- Perceptual Grouping
- Relaxation Labelling 1
- Relaxation Labelling 2
- Modelling for Vision
- Image Sequence Processing 1
- Image Sequence Processing 2
- Object Recognition

- Computational Stereo 1
- Computational Stereo 2
- Photometric Stereo
- Motion and Optical Flow
- Matching Relational Structures and High Level Vision
- Bayesian Inference in Vision (Self-study week)

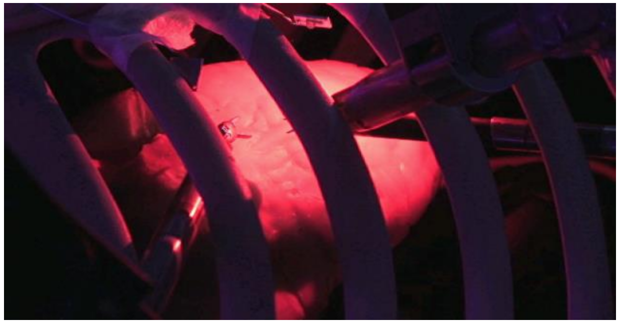


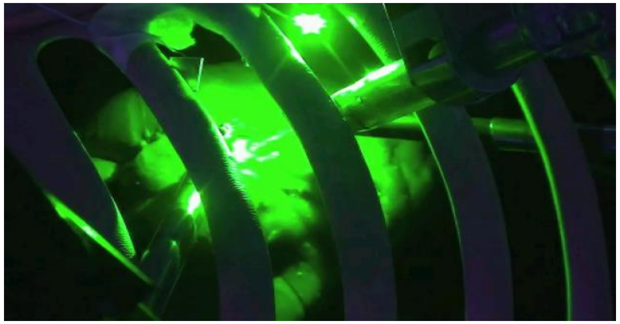
Schedule

	Tuesday 16:00	Tuesday 17:00	Friday 09:00	Friday 10:00
Week 2	Introduction to Vision Systems	Pre-Processing for Vision – Part 1	Tutorial: Edge Detection and Segmentation	Pre-Processing for Vision – Part 2
Week 3	NO LECTURE	NO LECTURE	Shape Recognition	Fourier Methods
Week 4	Hough Transform	Tutorial: Texture and Region Based Segmentation	Tutorial: Fourier Methods and Moments	Perceptual Grouping
Week 5	Relaxation Labelling – Part 1	Relaxation Labelling – Part 2	Assessed: Hough Transform and Relaxation Labelling	Modelling for Vision
Week 6	Image Sequence Processing – Part 1	Image Sequence Processing - Part 2	Tutorial: Image Sequence Processing, Matching	Object Recognition
Week 7	Computational Stereo - Part 1	Computational Stereo – Part 2	Tutorial: Computational Stereo	Photometric Stereo
Week 8	Motion and Optical Flow	Matching Relational Structures and High Level Vision	Tutorial: Photometric Stereo	Tutorial: Motion and Optical Flow
Week 9	Revision and Self-Study (Bayesian Inference for Vision) – Tutors available in Class		Revision and Self-Study	
Week 10	Revision and Self-Study Week			
Week 11	Exam Week			

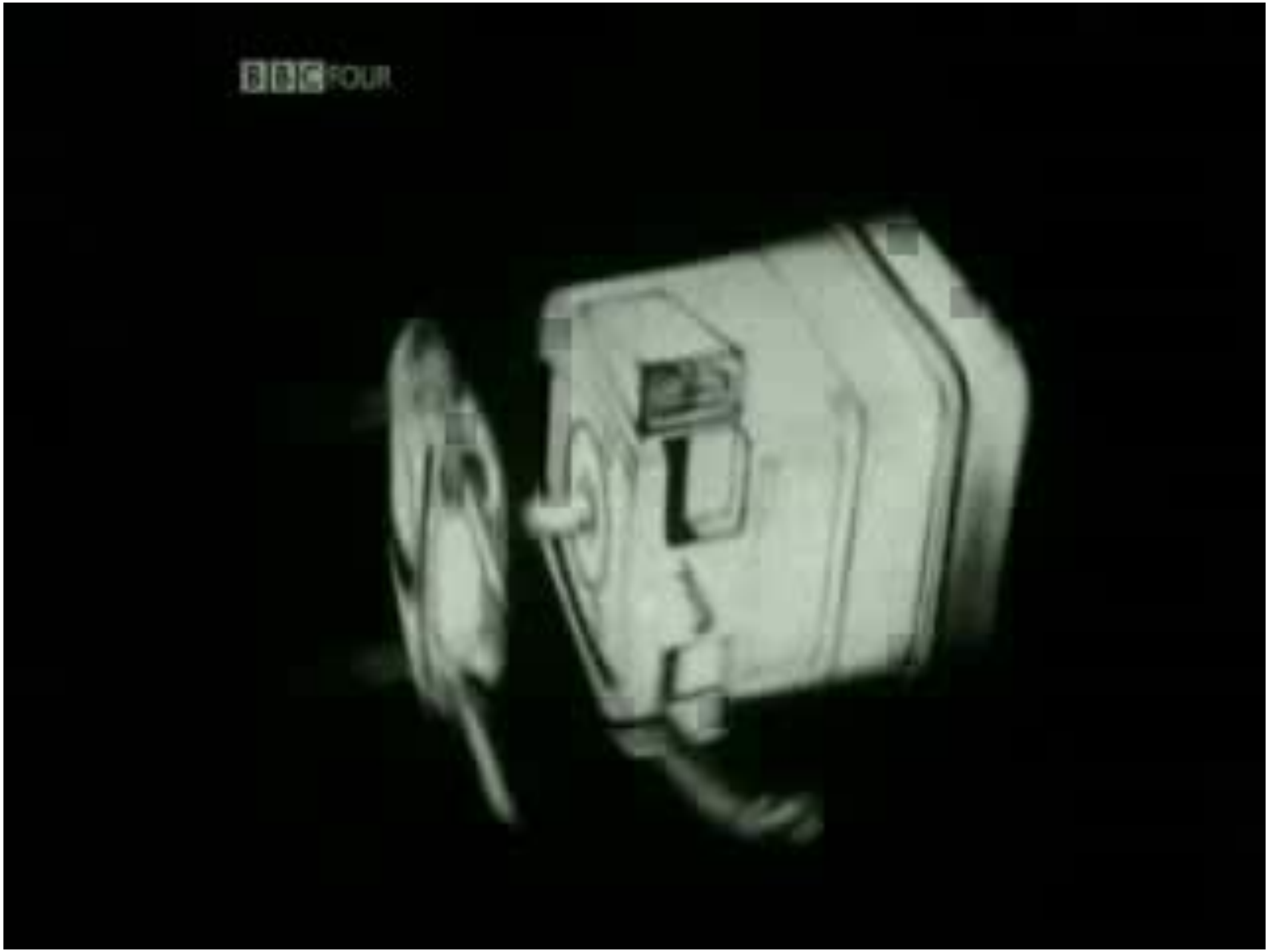
Contents - Introduction to Vision Systems

- Graphics, image processing and vision
- Application oriented vision systems
- General purpose vision systems
- Computational models for vision
- Levels of vision
- Principle of least commitment





Able Mabel: The Robotic Housemate



Video Courtesy: BBC4

Navigation: Limitations of AI

Video Courtesy: BBC

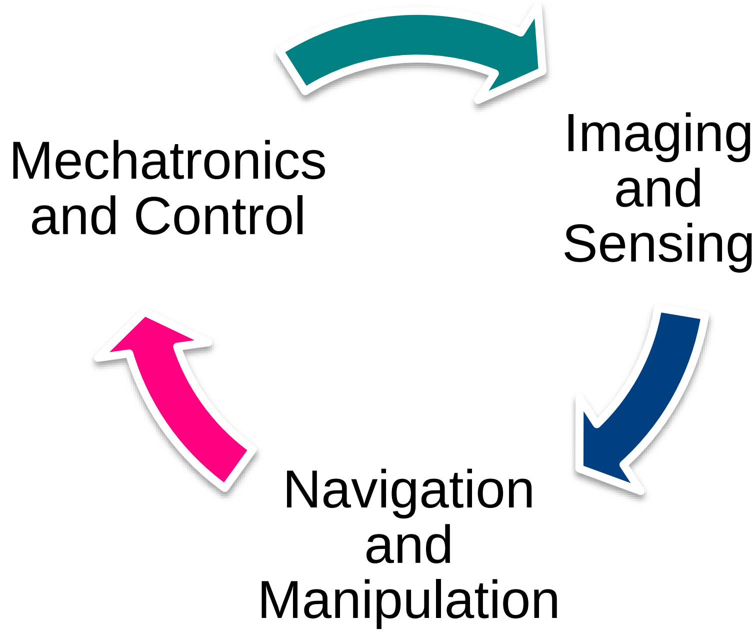
Video Courtesy: BBC4

Real-Time Monocular SLAM

Simultaneous Localisation and Mapping (SLAM) using a single hand-held camer

Image Courtesy: A Davison, Imperial College

Key Components of a Robotic Platform



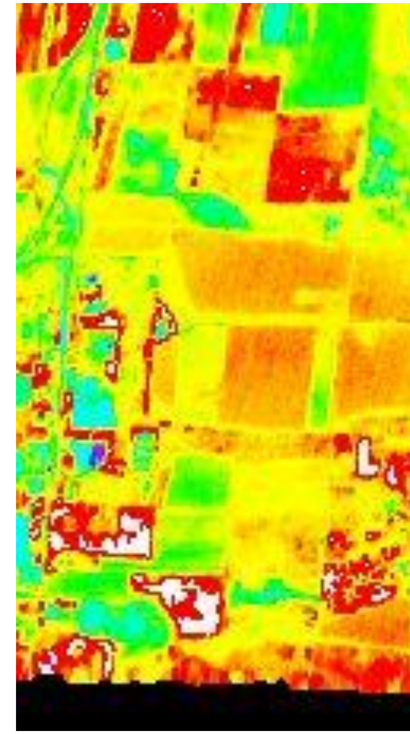
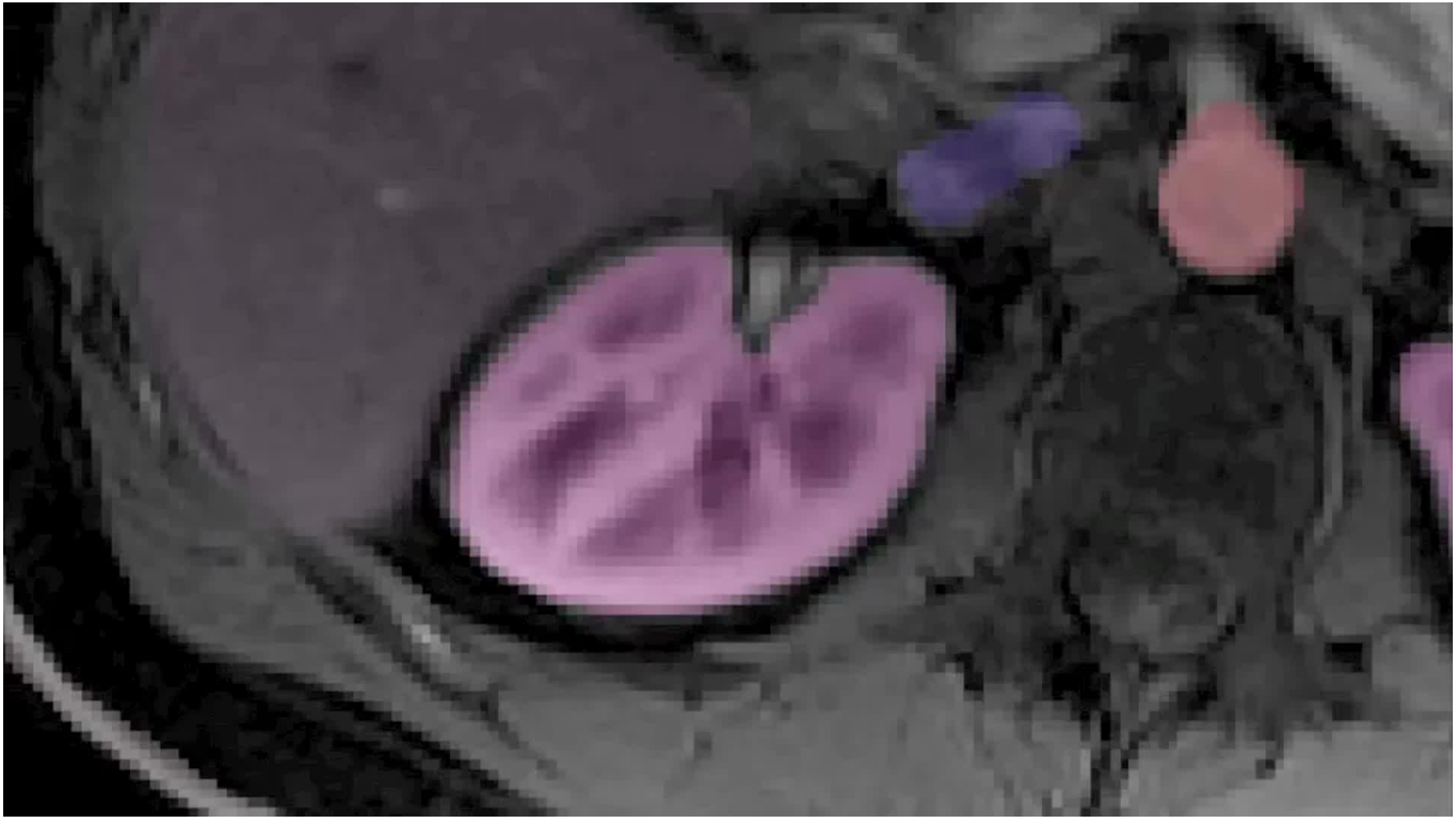


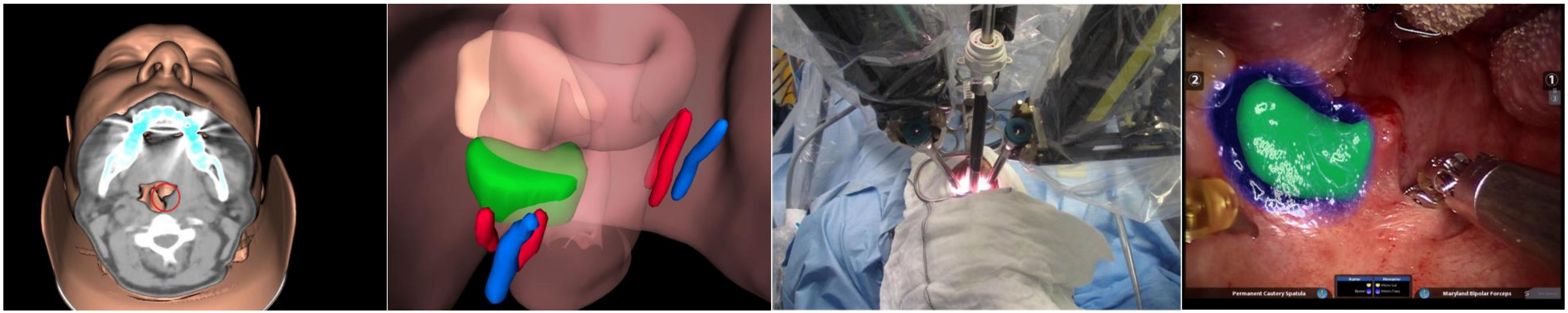
Image Courtesy and Source from: Unibots.com
NARO Japan, MIT CSAIL, and ZDNet

Navigation Inside the Body

Image and Video Courtesy: Intuitive Surgical Inc

Precision and stability





- Segmentation
- Shape modelling

- Registration
- Image fusion

- Physical based modelling
- Simulation and planning

- Tracking
- Surface Reconstruction (shape from X)

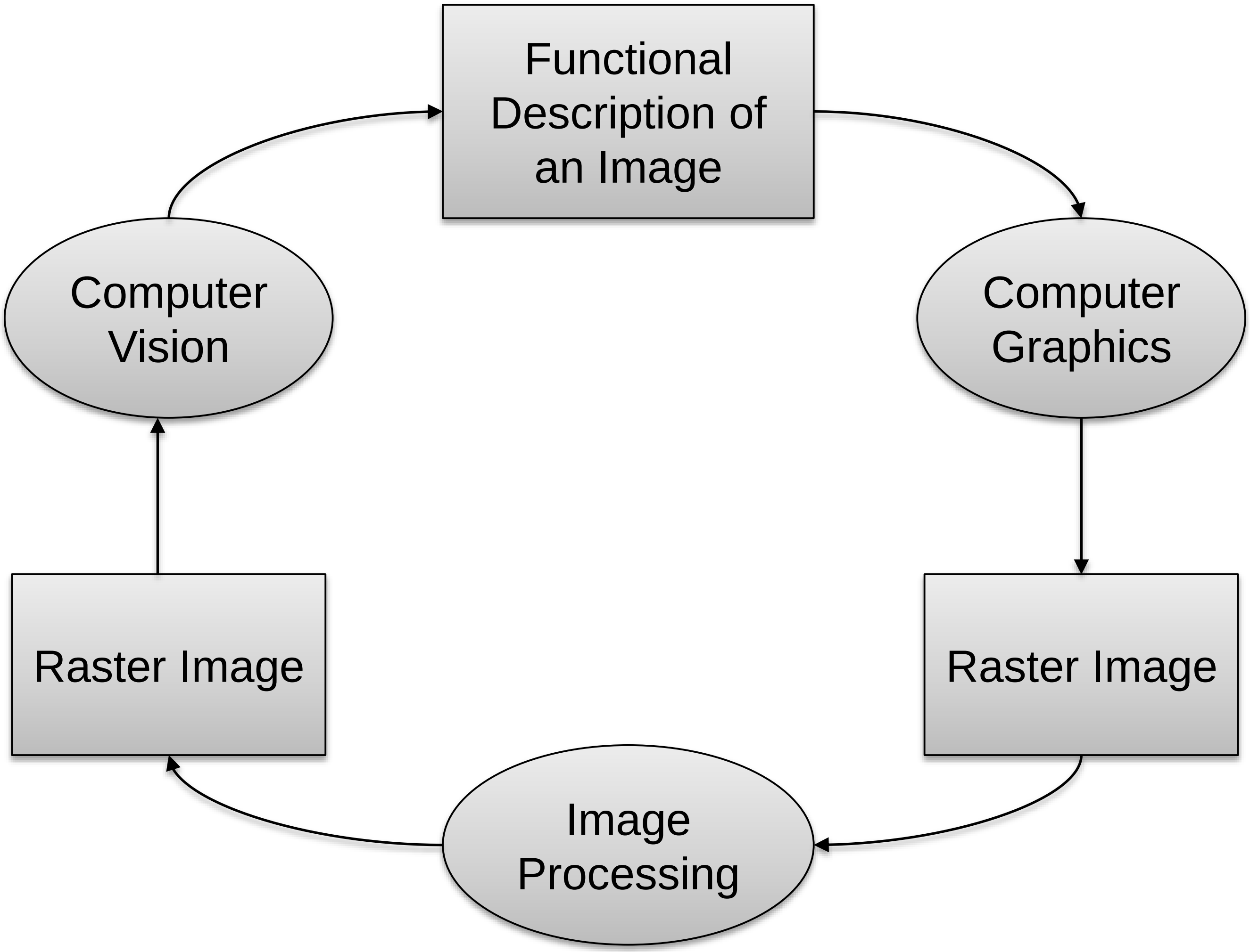
- Localisation and mapping
- Intraoperative shape instantiation

- Image fusion and AR visualisation
- Navigation and guidance

Graphics, Image Processing and Vision

- **Graphics** – image or visual representation of an object, can be in either 2D or 3D
  - Typically in raster or vector format. Raster format uses a grid of pixels where vector format is made up of paths of scalable objects (e.g., lines, shapes). Vector-based graphics can be scaled to a larger size without losing image quality (e.g. EPS, Macromedia Freehand, Adobe Illustrator, SolidWorks);
- **Image Processing** – a form of signal processing for which the input is an array of pixels (image) and the output is also an image describing specific features of the data.
  - It is useful to differentiate commonly used, often interchangeable terms, all using an image as the input - Image Processing: output=image; Image Analysis: output=measurements, e.g., path of an object moving within the image sequence; Image Understanding: output=high-level description of the scene content, which is related to what we discuss in high-level vision.
- **Computer Vision** – from image(s) or image sequence(s), to derive high-dimensional description of the objects or scenes in either numerical or symbolic representations.

Graphics, Image Processing and Vision



Graphics

- 3D representation of the object and its material properties (colour, texture, reflectance) are known
- Full control of lighting and other environmental factors
- Typical rendering methods include ray casting, ray tracing, radiosity
- Photorealistic and non-photorealistic rendering; need to consider perceptual factors to achieve realistic results

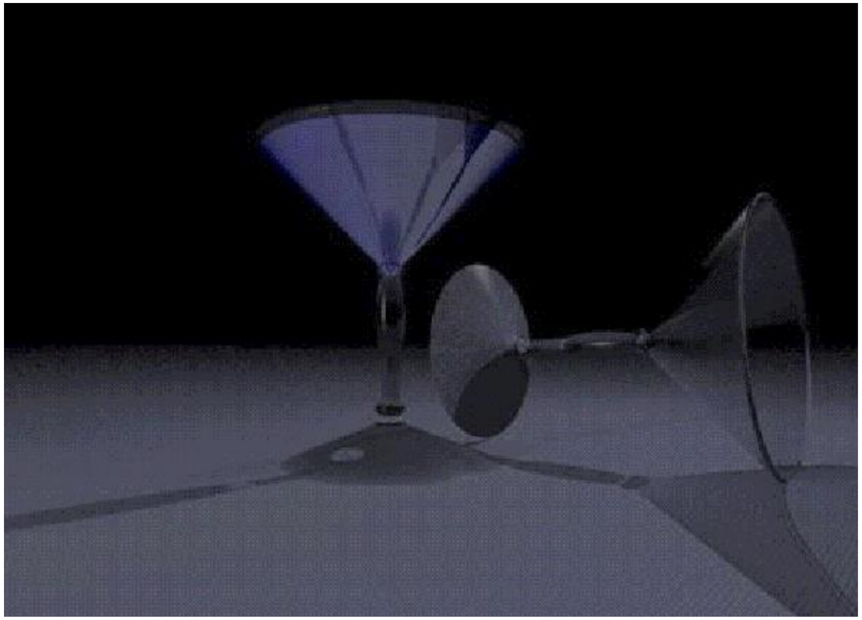


Image Courtesy: GZ Yang

Image Courtesy: SolidWorks

Image Processing

- Removal of defects such as scratches or other noise
- Improvement of contrast
- Removal of camera blur
- Removal of motion blur
- Enhancement of structure
- Enhancement of colour

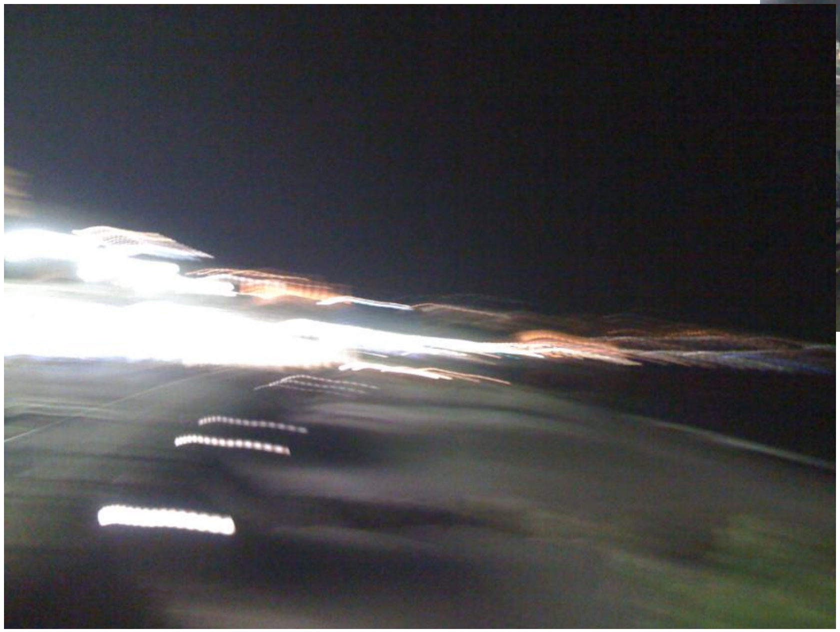
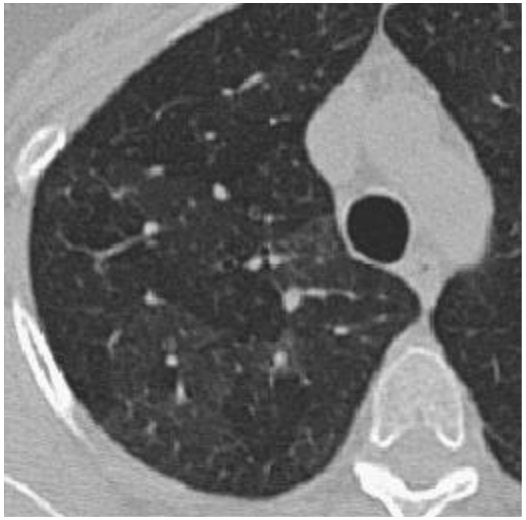


Image Processing



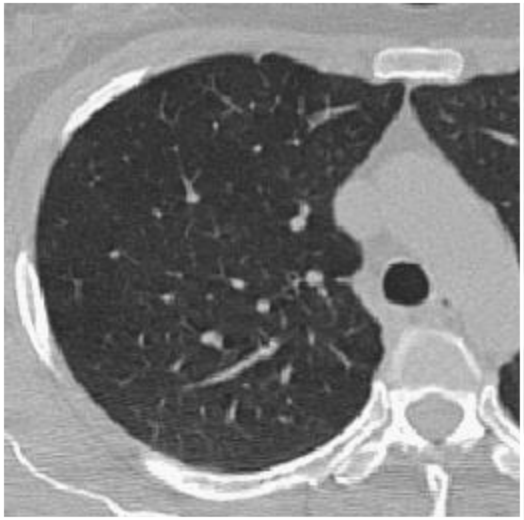
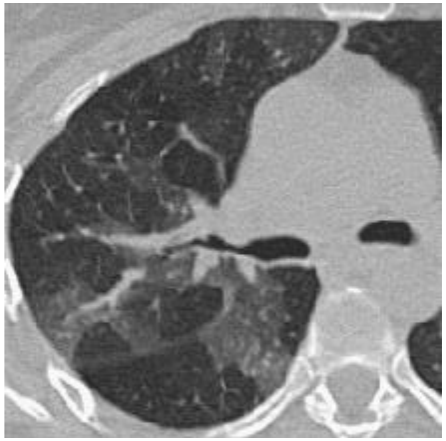
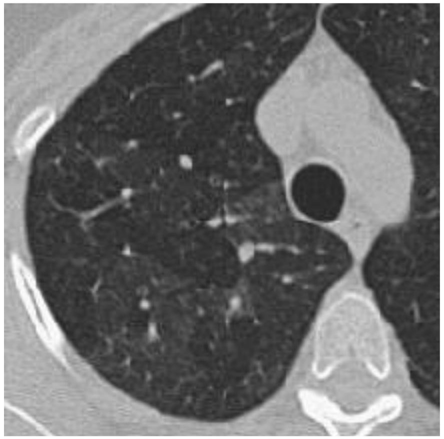
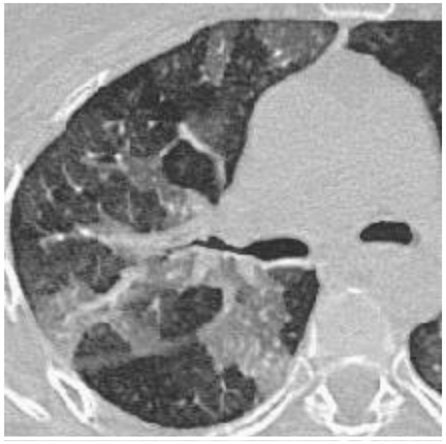
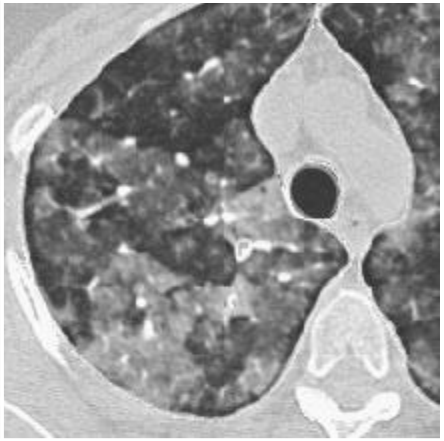
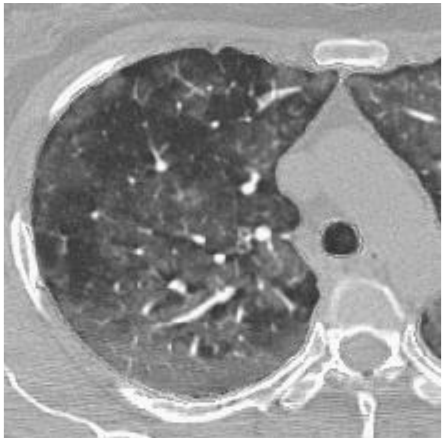


Image Processing

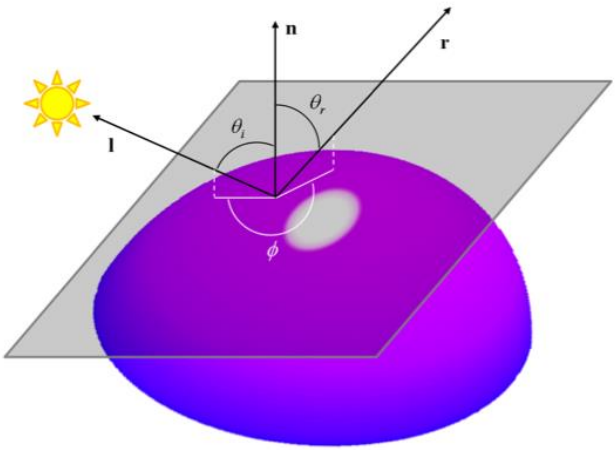




Computer Vision

- Determining the type of an object in the picture
- Assessing an object for quality
- Breaking a picture into different parts
- Constructing a 3D representation of an object
- Extracting a line representation of an object
- Reasoning about a scene to deduce hidden properties

Vision is the exact opposite process to graphics. In graphics, we start with a functional description and end up with a picture representation which is a set of pixels. In the case of graphics, we have complete information and so the problems are largely algorithmic. For vision, we do not have complete information and therefore we must use domain specific knowledge, assumptions or heuristics to achieve our goal.

Computer Vision: Examples



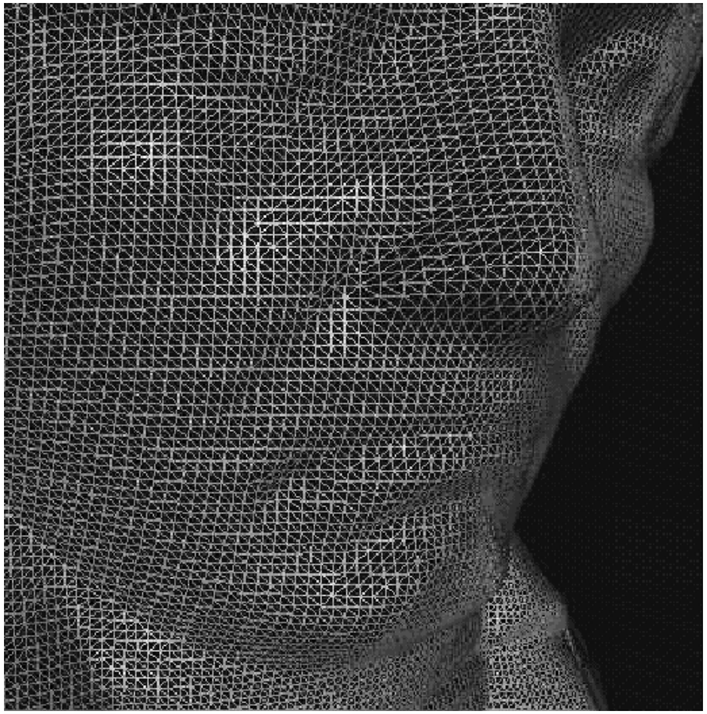
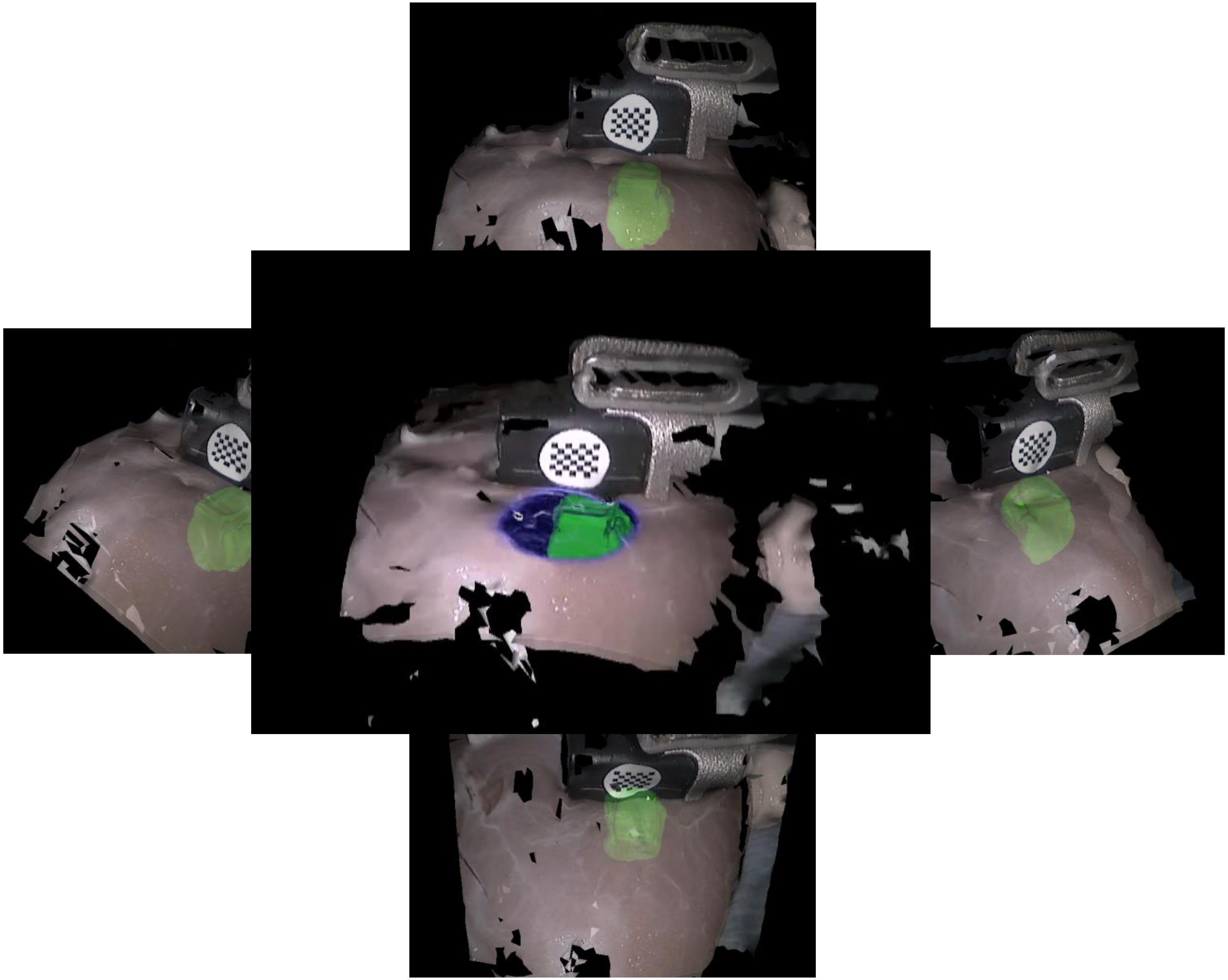
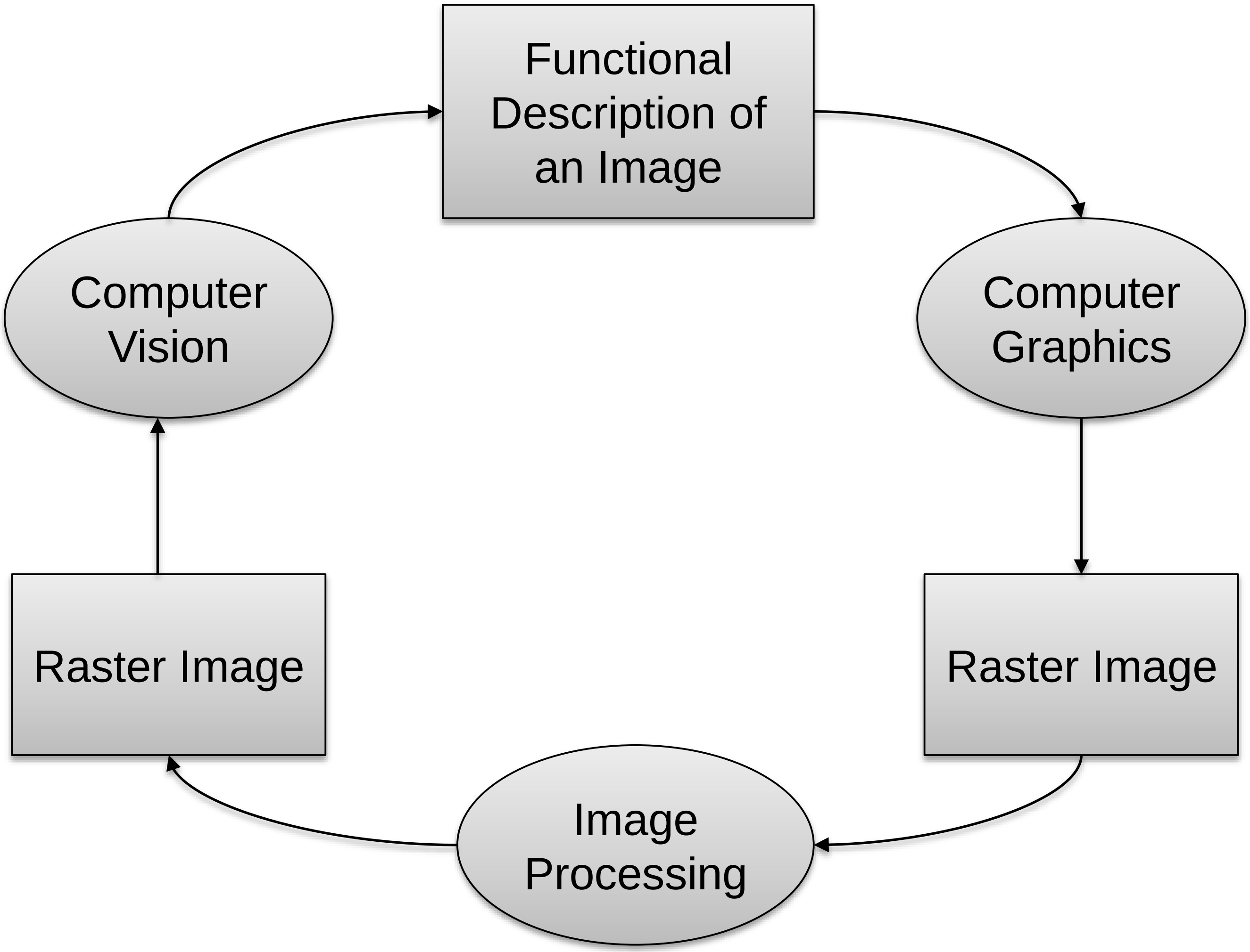


Image courtesy Karsten Schuens

Computer Vision: Examples



Graphics, Image Processing and Vision



Application Oriented Vision Systems

- Application Oriented Vision Systems
  - Vision based robots
  - Quality assessment systems

The construction of these systems is highly domain dependent, but nonetheless some techniques have been devised which have usefulness beyond their immediate application. Also included here are the so called high performance vision systems which use artificial intelligence methods. Here the identification of image features depends on heuristics which inevitably create application specificity.

General Purpose Vision Systems

- General Purpose Vision Systems
  - **Low level systems** - in which we model the micro structure of the brain as far as it is understood. Models of the visual cortex (and other brain structures) have been provided by physiologists (Hubel and Wiesel) and have formed the basis of neural net systems such as Wisard.
  - **High level behavioural models** - in which the psychologists understanding of perception is modelled.

The intention is to provide a system that can recognise image properties in a wide variety of image types. So far the only successful systems are pattern recognisers that can be trained. However, general purpose vision remains an important research goal. A common approach to constructing general purpose vision systems is to use brain modelling. That is to say we attempt to construct a system which works in a way analogous to the human vision system.

Some Questions For Vision System Design

- What information is sought from the image?
- How is it manifest in the image?
- What a-priori knowledge is required to recover the information?
- What is the nature of the computational process?
- How should the required information be represented?

Consider these problems:

Problem: Locate a house among trees
Straight lines are an intrinsic characteristic of houses, not trees

Problem: Separate land from sea in an aerial image
Uniform reflectivity is an intrinsic characteristic of sea, not land

Computational Models for Vision

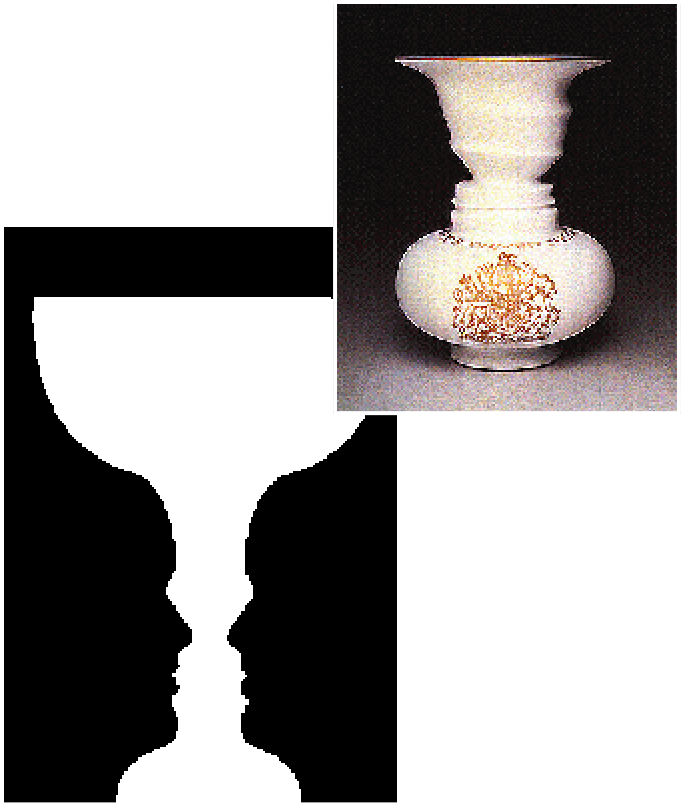
- **Intrinsic Characteristics**
  - Need to establish a relationship between physical entities and intrinsic characteristics. If, for example we wish to distinguish and extract the position of a house in a scene containing trees, we can use the fact that straight lines are an intrinsic characteristic of a house, but not of trees. Hence we would choose to extract straight lines from the raw images. Conversely, if we wish to separate sea from land for an aerial photograph, we could choose the intrinsic characteristic that water is of uniform appearance, and apply a region based segmentation algorithm tuned to extract large uniform regions. Intrinsic characteristics must be matched to established techniques.
- **Resource Limitations**
  - Consideration of computational resources and the time required to process the image. The exact nature of the computational process will be task dependent, real-time requirements need to be taken into account in most vision systems. For example, if we wish to assess the condition of a motorway surface while driving over it at 60mph, each sampled image must be processed in a fixed time, which in turn places a requirement on the hardware.

Computational Models for Vision

- **Prior Knowledge**
  - It is universally accepted that human vision depends on a vast amount of knowledge. To establish a relationship between pixel brightness and image properties, we will need to have some form of scene model, illumination model and sensor model. The scene model may include such information as the type of features we are trying to detect, or in more general cases make assumptions about properties such as smoothness or convexity. The illumination model will contain information about the position and characteristics of the light source and the surfaces reflectance properties. The sensor model will describe the position and optical performance of the cameras used, and the noise and distortion applied by the digitisation and storage media.

- **Knowledge Representation**
  - Representation is important for vision partly for the encoding of knowledge in a useful form, but also in the presentation of results in an understandable form. Typically, humans find it difficult to describe exactly the visual properties on which they make decisions.

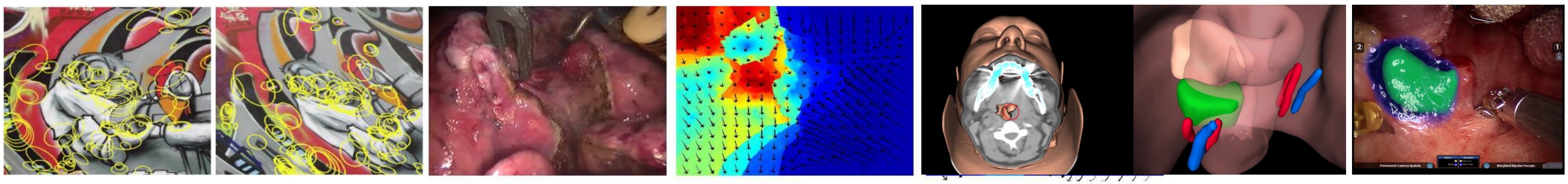
The Effect of Perception and Prior Knowledge



Levels of Vision

Low Level Vision → Intermediate Level Vision → High Level Vision

Edges, corner, depth, optical flow | Contours, regions | Objects and hidden information



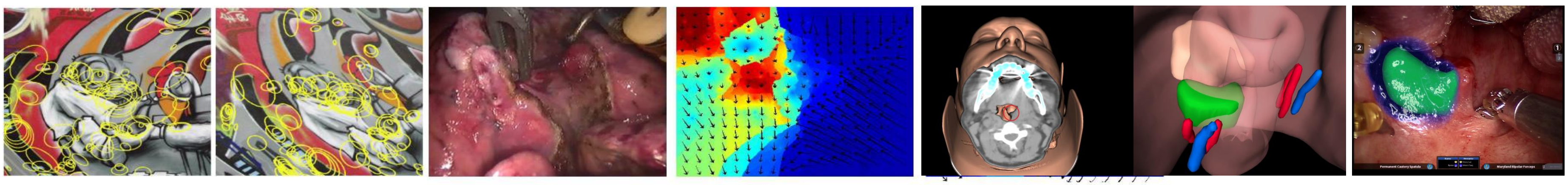
Low Level Processing

Operations carried out on the pixels in the image to extract properties such as the gradient (with respect to intensity) or depth (from the viewpoint) at each point in the image. We may for example be interested in extracting uniform regions, where the gradient of the pixels remains constant, or first order changes in gradient, which would correspond to straight lines, or second order changes which could be used to extract surface properties such as peaks, pits, ridges etc. Low level processing is invariably data driven, sometimes called bottom up. It is the area where modelling the visual cortex functioning is most appropriate.

Levels of Vision

Low Level Vision → Intermediate Level Vision → High Level Vision

Edges, corner, depth, optical flow | Contours, regions | Objects and hidden information

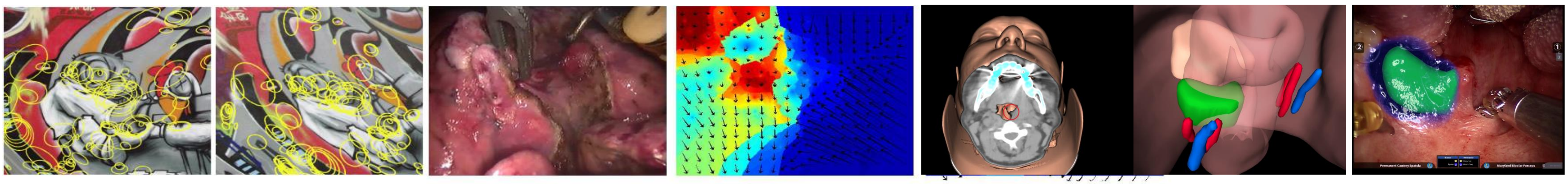


Intermediate Level Processing

The intermediate level of processing is fundamentally concerned with grouping entities together. The simplest case is when we group pixels into lines. We can then express the line in a functional form. Similarly, if the output of the low level processing is a depth map, we may further need to distinguish object boundaries, or other characteristics. Even in the simple case where we are trying to extract a single sphere, it is no easy process to go from a surface depth representation to a centre and radius representation. Since intermediate level processing is concerned with grouping, much of the recent work has concentrated on using perceptual grouping methods.

Levels of Vision

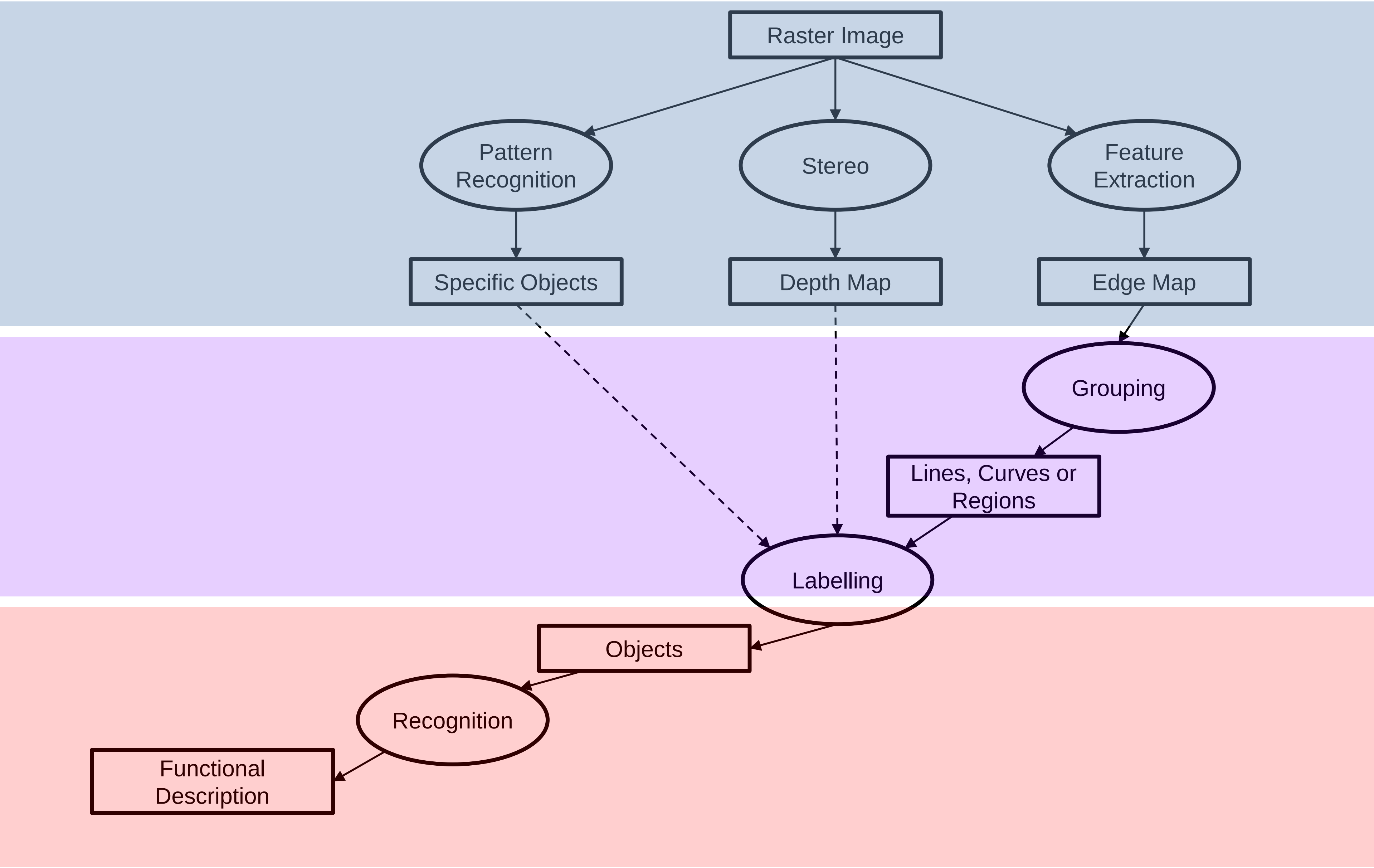
Low Level Vision	Intermediate Level Vision	High Level Vision
Edges, corner, depth, optical flow	Contours, regions	Objects and hidden information



High Level Processing

Interpretation of a scene goes beyond the tasks of line extraction and grouping. It further requires decisions to be made about types of boundaries, such as which are occluding, and what information is hidden from the user. Further grouping is essential at this stage since we may still need to be able to decide which lines group together to form an object. To do this, we need to further distinguish lines which are part of the objects structure, from those which are part of a surface texture, or caused by shadows. High level systems are therefore object oriented, and sometimes called top down. They almost always require some form of knowledge about the objects of the scene to be included.

Computer Vision Processes



Principle of Least Commitment

- From low level to high level of processing, we remove information
  - For computational efficiency, we wish to remove as much information as possible
  - For effective vision, we need to retain as much information as possible

If we adopt the three level approach, then it is clear to see that as we move from one level to the next higher we are throwing away some of the information. For example, if we extract the points on an image that we expect to form edges, by differentiation the image and setting a threshold, we eliminate a substantial number of pixels. This is desirable from the computational viewpoint, since we wish to minimise the calculations carried out on edge points; however, we run the risk removing weak but significant edge points from our image, and so making a wrong decision at a later stage. Similarly, when we extract line segments, we will reject certain segments on the basis of the edge point data, and depending on the thresholds we choose, we again may reject an edge which belongs to an object. The principle of least commitment states that we should avoid these possibilities by carrying as much information from one level to the next. The limit to which it is feasible to do this will depend on the computational resources open to us.

Conclusions

- Graphics, image processing and vision
- Application oriented vision systems
- General purpose vision systems
- Computational models for vision
- Levels of vision
- Principle of least commitment