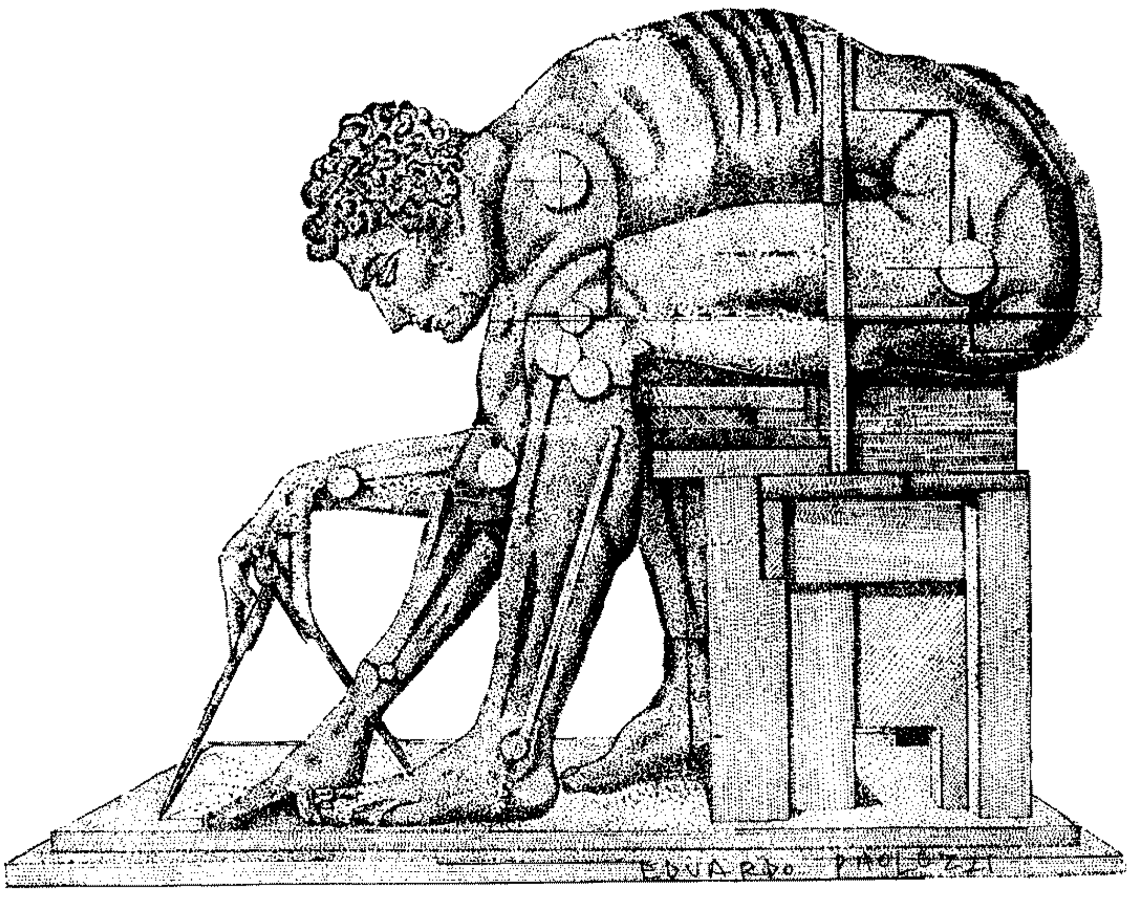
University of Cambridge
Engineering Part IIB

Module 4F12: Computer Vision

Handout 1: Introduction

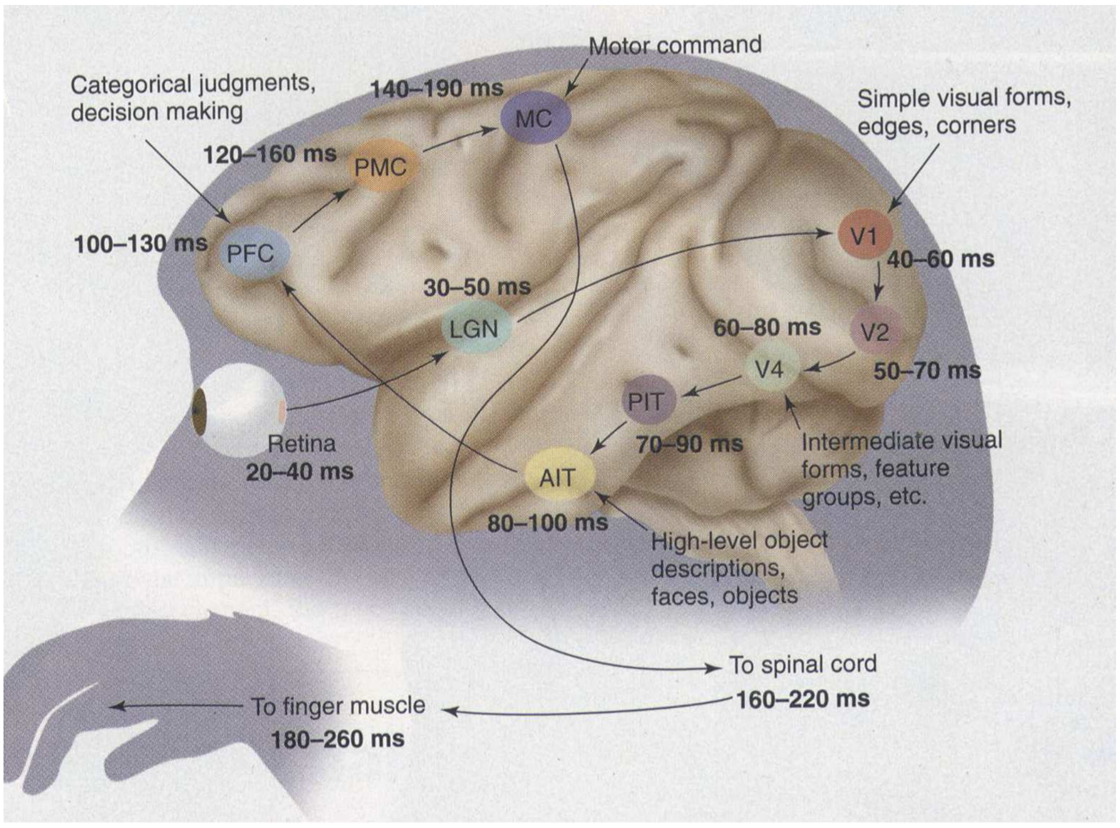


Roberto Cipolla (notes) and Ignas Budvytis (lecturer)

October 2021

What is computer vision?

Vision is about discovering from images what is present in the scene and where it is. It is our most powerful sense.



In **computer vision** a camera is linked to a computer. The computer automatically processes and interprets the images of a real scene to obtain useful information (3**R's**: recognition, registration and reconstruction) and **representations** for decision making and **action** (e.g. for navigation, manipulation or communication).

Why study computer vision?

1. Intellectual curiosity — how do *we* see?
2. Replicate human vision to allow a machine to see — many industrial, commercial and healthcare applications.

Computer Vision is not:

Image processing: image enhancement, image restoration, image compression. Take an image and process it to produce a new image which is, in some way, more desirable.

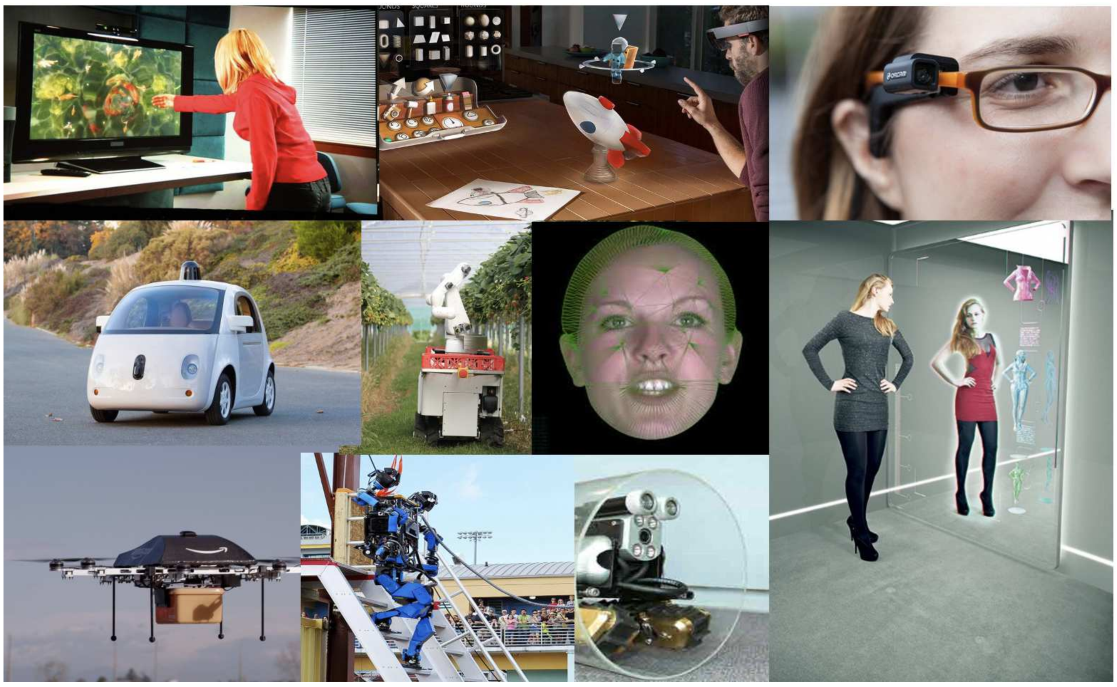
Pattern recognition: classifies patterns into one of a finite set of prototypes. There is an Infinite variation in images of objects and scenes due to changes in viewpoint, lighting, occlusion and clutter.

Applications

- Industrial and agricultural automation
  - Visual inspection
  - Object recognition.
  - Robot hand-eye coordination
- Autonomous vehicles
  - Automotive applications
  - Self-driving cars
- Human-computer interaction
  - Face detection and recognition.
  - Gesture-based and touch free interactions
  - Cashierless transactions
  - Image search in video and image databases
- Augmented reality and enhanced interactions
  - AR with mobile phones and wearable computers
- Surveillance and Security
- Medical Imaging
  - Detection, segmentation and classification
- 3D modelling, measurement and visualisation
  - 3D model building and photogrammetry
  - Human body and motion capture
  - 3D Virtual fitting and e-commerce
  - Avatar creation and talking heads

Applications

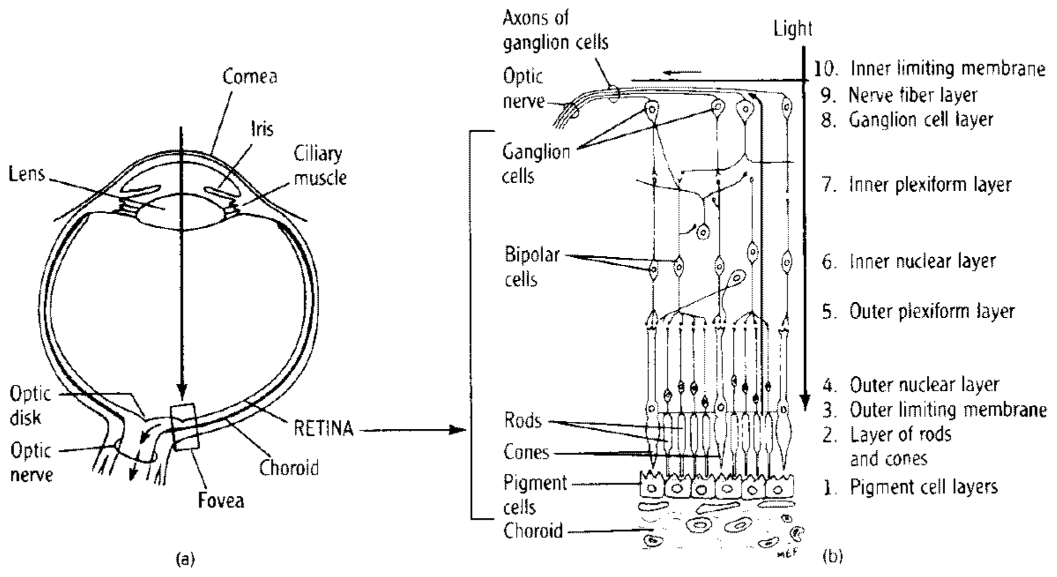
Examples of recent computer vision research that has led to new products and services.



- Microsoft Kinect - Human pose detection and tracking for game interface using gestures
- Microsoft Hololens - Smart glasses for Augmented Reality
- Orcam - Wearable camera using text-recognition to help visually-impaired
- Wayve and Waymo - autonomous driving using cameras
- Dogtooth Technologies - addressing labour shortages in fruit picking with robotics
- Pinscreen and Toshiba Europe - Photorealistic 3D avatars and Talking Heads
- Metail and Trya - Virtual fitting of clothes and shoes by estimating shape from images
- Amazon Prime Air - Drone delivery services with visual localisation and navigation
- Softbank and Boston Dynamics - Vision for robot navigation and hand-eye co-ordination
- Infrastructure visual inspection

How to study vision? The eye

Let's start with the human visual system.



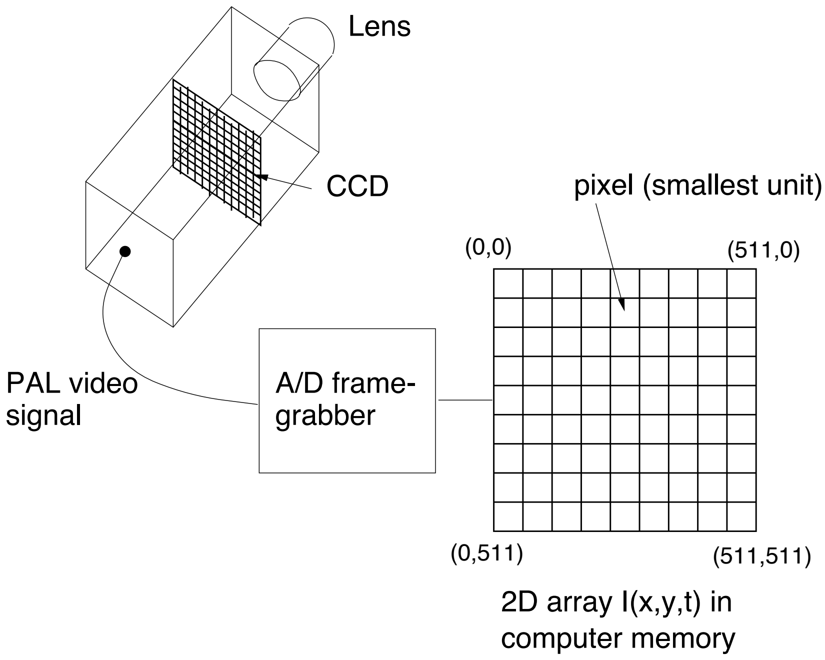
- Retina measures about 1000 mm^2 and contains about 10^8 sampling elements (rods) (and about 10^6 cones for sampling colour).
- The eye's spatial resolution is about 0.01° over a 150° field of view (not evenly spaced, there is a fovea and a peripheral region).
- Intensity resolution is about 11 bits/element, spectral resolution is about 2 bits/element (400–700 nm).
- Temporal resolution is about 100 ms (10 Hz).
- Two eyes (each about 2cm in diameter), separated by about 6cm.
- A large chunk of our brain is dedicated to processing the signals from our eyes - a data rate of about 3 GBytes/s!

Why not copy the biology?

- There is no point copying the eye and brain — human vision involves over 60 billion neurons.
- Evolution took its course under a set of constraints that are very different from today's technological barriers.
- The computers we have available cannot perform like the human brain.
- We need to understand the underlying principles rather than the particular implementation.

Compare with flight. Attempts to duplicate the flight of birds failed.

The camera



- A typical digital SLR CCD measures about 24×16 mm and contains about 6×10^6 sampling elements (pixels).
- Intensity resolution is about 8 bits/pixel for each colour channel (RGB).
- Most computer vision applications work with monochrome images.
- Temporal resolution is about 40 ms (25 Hz)
- One camera gives a raw data rate of about 400 MBytes/s.

The CCD camera is an adequate sensor for computer vision.

Image formation

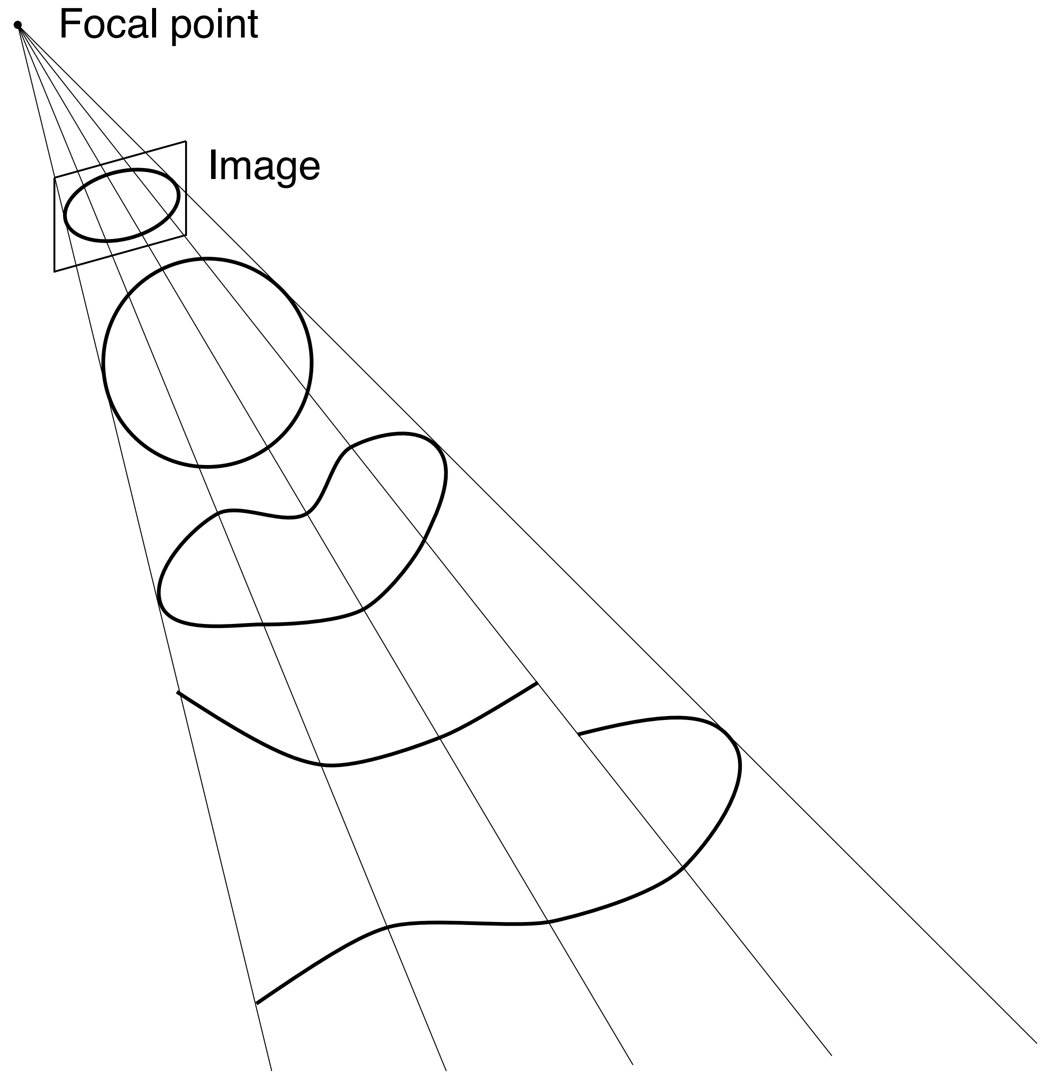
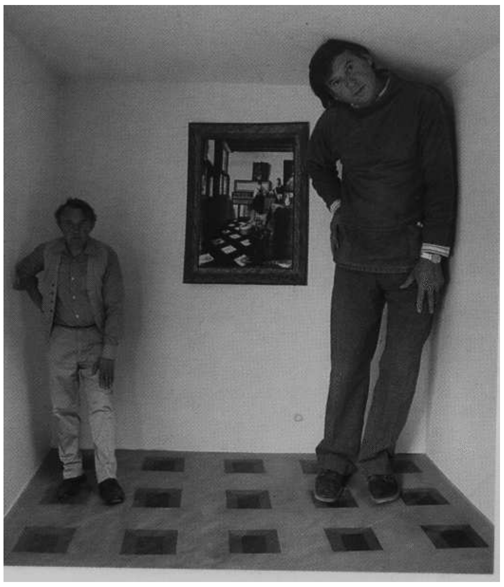
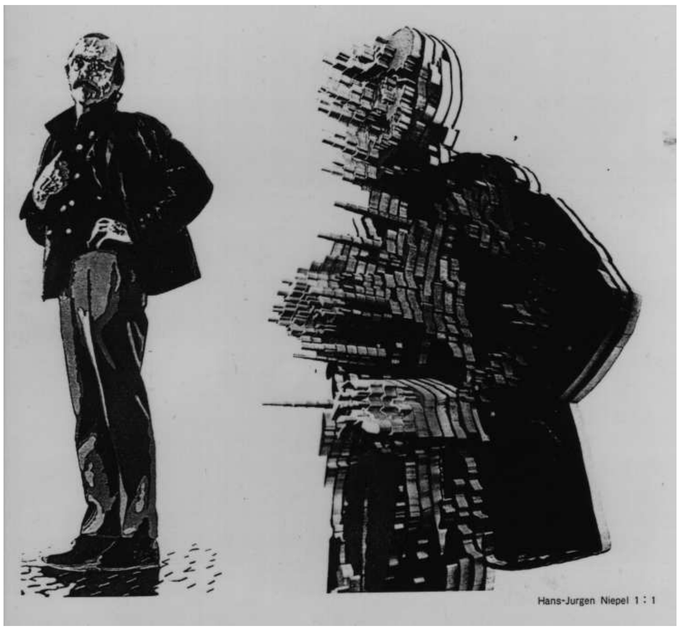


Image formation is a many-to-one mapping. The image encodes nothing about the *depth* of the objects in the scene. It only tells us along *which* ray a feature lies, not *how far* along the ray. The inverse imaging problem (inferring the scene from a single image) has no unique solution.

Ambiguities in the imaging process



Two examples showing that image formation is a many-to-one mapping. The Ames room and two images of the same 3D structure.



Vision as information processing

David Marr, one of the pioneers of computer vision, said:
" *One cannot understand what seeing is and how it works unless one understands the underlying information processing tasks being solved.*"

From an information processing point of view we must convert the huge amount of unstructured data in images into useful and actionable representations:

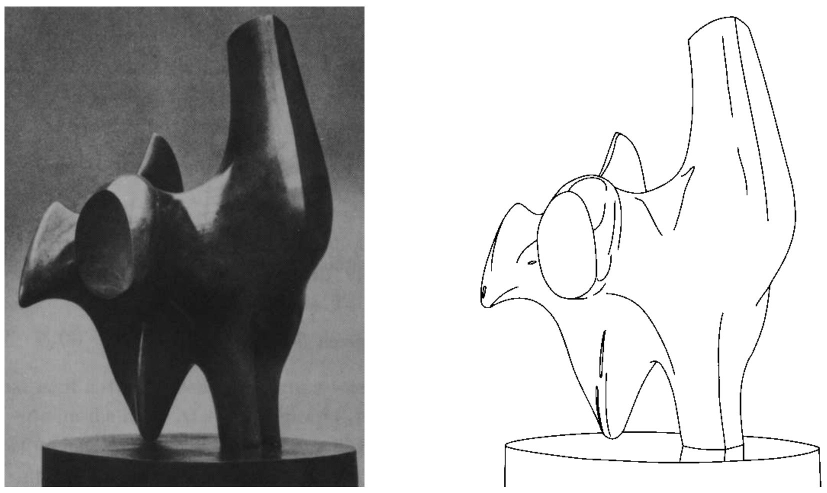
images	$\rightarrow$	generic salient features
100 MBytes/s		100 KBytes/s
(mono CCD)		
salient features	$\rightarrow$	representations and actions
100 KBytes/s		1–10 bits/s

Vision resolves the ambiguities inherent in the imaging proces by drawing on a set of constraints (AI). But where do the constraints come from? We have the following options:

1. Use more than one image of the scene.
2. Make assumptions about the world in the scene.
3. Learn (supervised and unsupervised) from the real world.

Feature extraction

The first stages of most computer vision algorithms perform feature extraction. The aim is to reduce the data content of the images while preserving the useful information they contain.



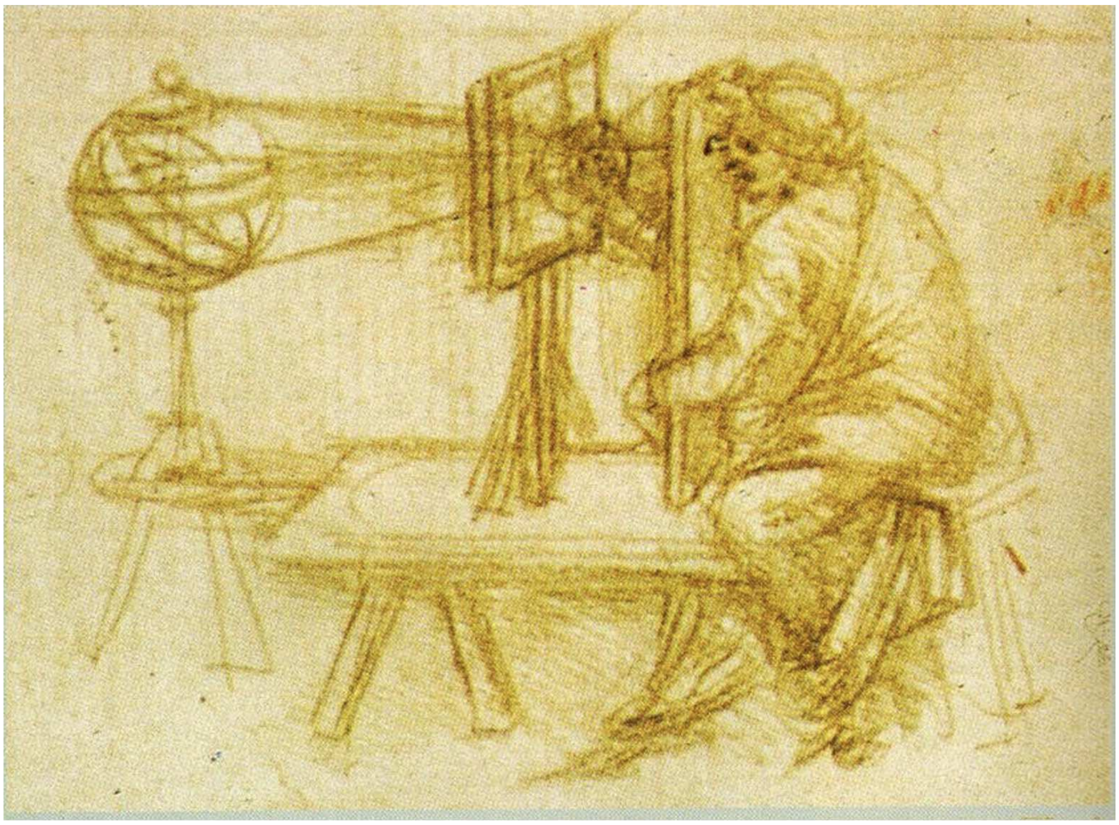
The most commonly used features are edges, which are detected as discontinuities in the image. This involves filtering (by convolution) and differentiating the image. Automatic edge detection algorithms produce something resembling a noisy line drawing of the scene.

Corner detection is also common. Corner features are localised in 2D and are particularly useful for finding correspondences in motion analysis using correlation.

Feature descriptors which are invariant to scale, orientation and lighting (e.g. SIFT) facilitate matching over arbitrary viewpoints and in different lighting.

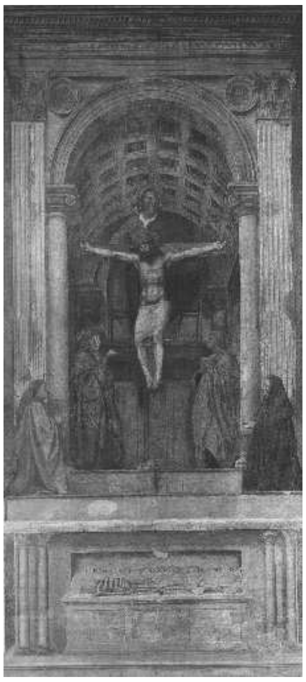
Perspective Projection

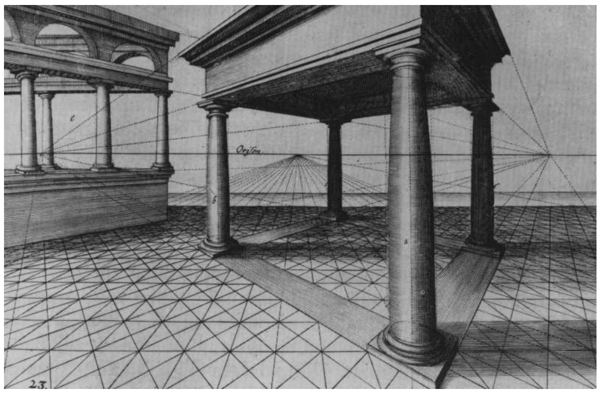
Before we attempt to interpret the image (using the features extracted from the image), we have to understand how the image was formed. In other words, we have to develop a **camera model**.



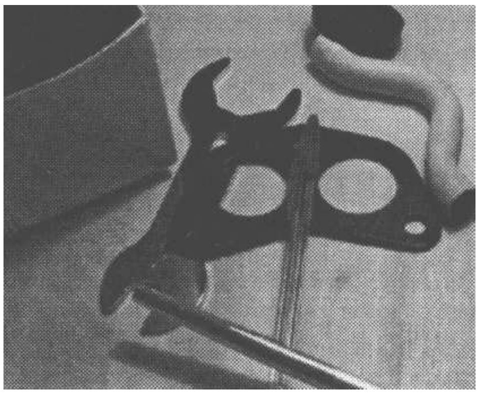
Camera models must account for the position of the camera, perspective projection and CCD imaging. These geometric transformations have been well-understood since the C14th. They are best described within the framework of **projective geometry**.

Projection and Camera models

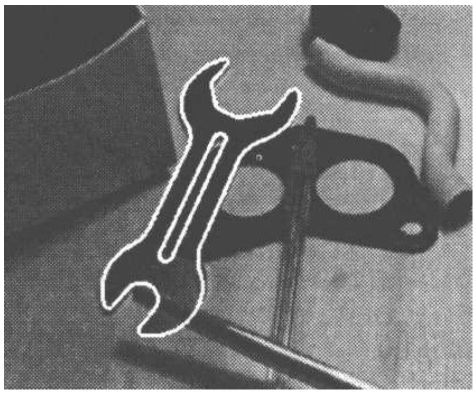




Having established a camera model, we can predict how known objects will appear in an image and can also recover their position and orientation (pose) in the scene.



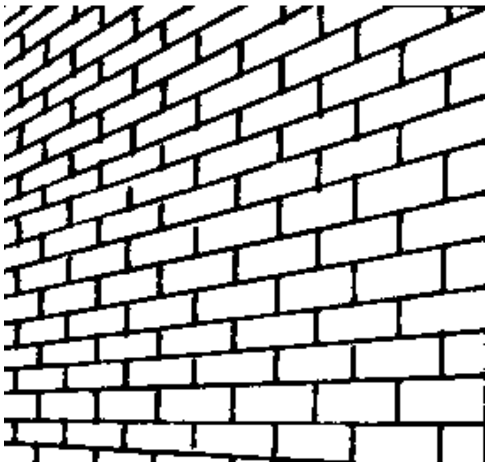
Cluttered scene



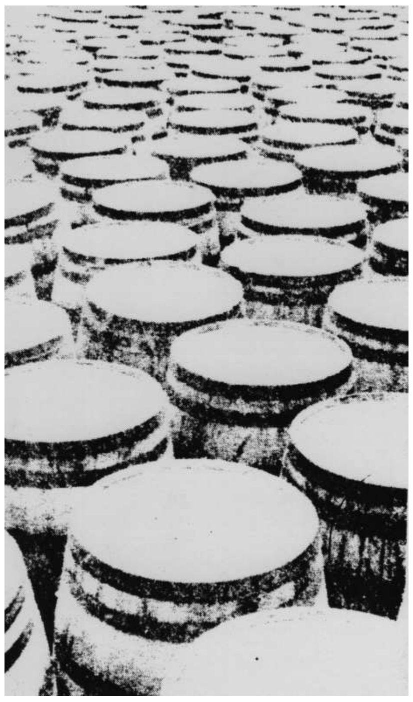
Spanner pose recovered

Shape from texture

Texture provides a very strong cue for inferring surface orientation in a single image. It is necessary to assume *homogeneous* or *isotropic* texture. Then, it is possible to infer the orientation of surfaces by analysing how the texture statistics vary over the image.



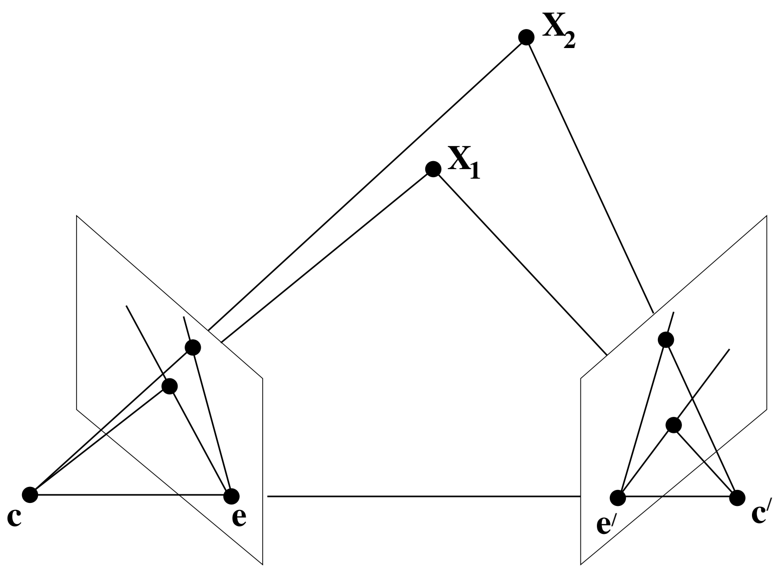
Here we perceive a vertical wall slanted away from the camera.



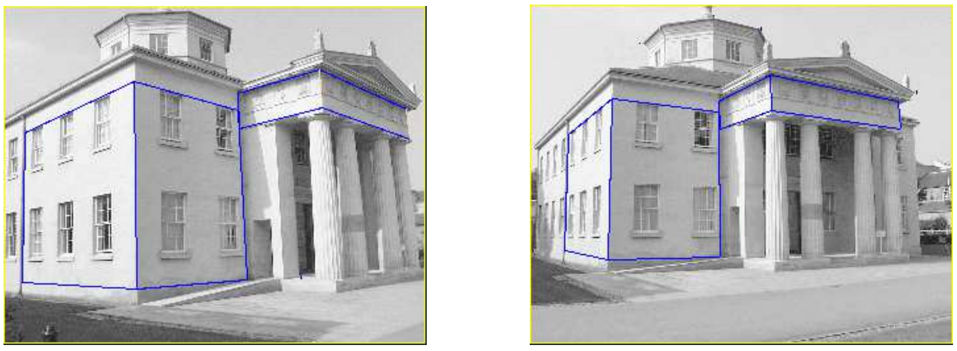
And here we perceive a horizontal surface below the camera.

Stereo vision

Having two cameras allows us to triangulate on features in the left and right images to obtain depth. It is even possible to infer useful information about the scene when the cameras are not **calibrated**.

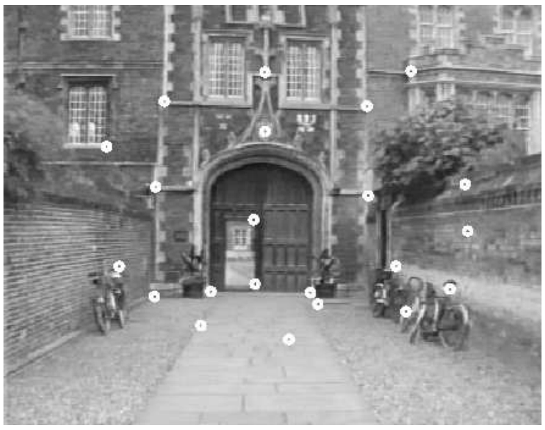


Stereo vision requires that features in the left and right image be matched. This is known as the **correspondence problem**.



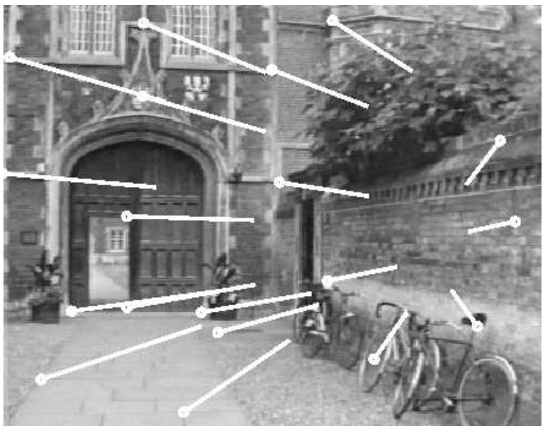
Structure from motion

Related to stereo vision is a technique known as **structure from motion**. Instead of collecting two images simultaneously, we allow a single camera to move and collect a sequence of images from different viewpoints.



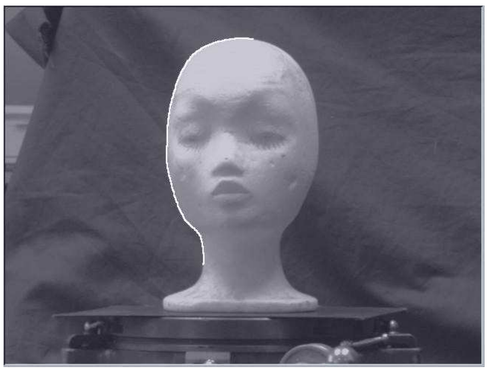
As the camera moves, the motion of some features (in this case corner features) is **tracked**.

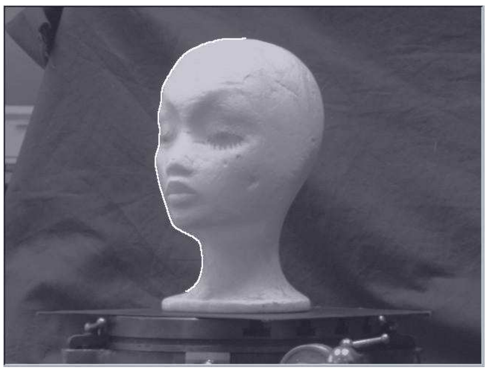
The trajectories allow us to recover the 3D translation and rotation of the camera and the 3D structure of the scene.



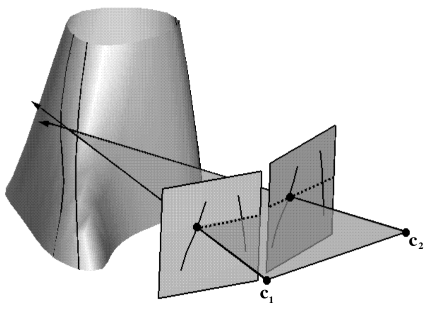
Shape from contour

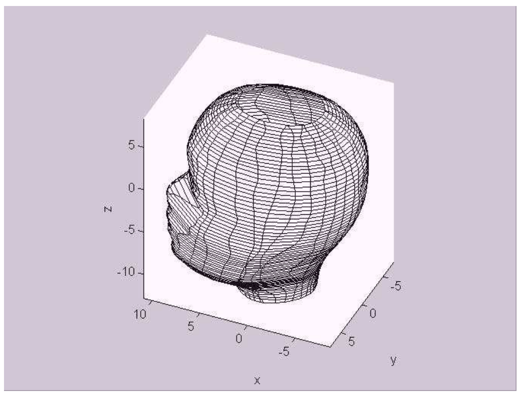
A curved surface is bounded by its **apparent contours** in an image. Each contour defines a set of **tangent planes** from the camera to the surface.





As the camera moves, the contour generators "slip" over the curved surface. By analysing the deformation of the apparent contours in the image, it is possible to reconstruct the 3D shape of the curved surface.

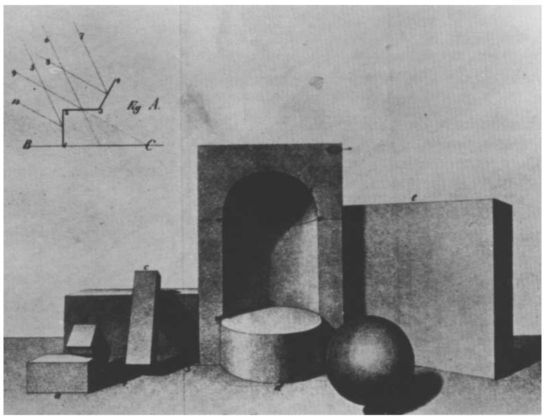


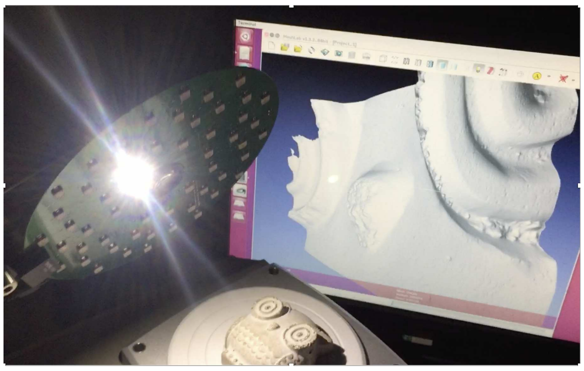


Shape from shading

It is also possible to infer the surface shape of objects from the **shading** observed in the image.

To recover shape we usually make the assumptions of a single, distant light source and a Lambertian/isotropic surface reflectance





Photometric stereo can recover accurate 3D shape from a single viewpoint from multiple shading patterns in images obtaining by changing the light source position.

Geometrical framework

The first part of the course will focus on generic computer vision techniques which make minimal assumptions about the outside world. This means we'll be concentrating on the theory of perspective, stereo vision and structure from motion.

We typically use a geometric framework:

1. Reduce the information content of the images to a manageable size by extracting salient features, typically **edges** or **blobs**. (These features are generic and substantially *invariant* to a variety of lighting conditions.)
2. Model the imaging process, usually as a perspective projection and express using projective transformations.
3. Invert the transformation using as many images and constraints as necessary to extract 3D structure and motion.

Statistical framework

Geometry alone is only a part of the solution. In the second part of the course we will introduce techniques which learn from the visual world. They are part of a statistical framework to understanding vision and for building systems which:

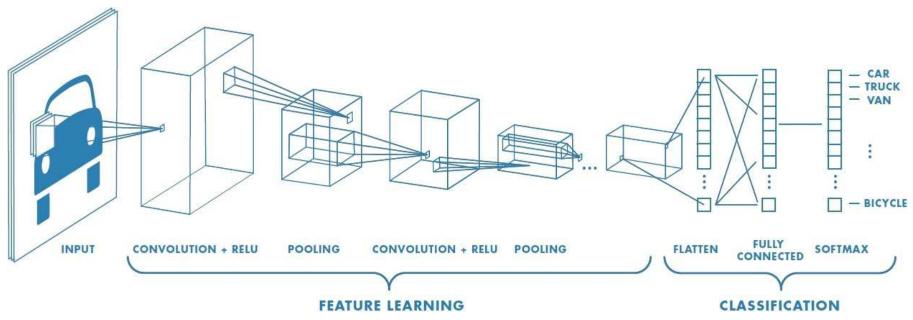
1. Have the ability to test hypotheses
2. Deal with the ambiguity of the visual world
3. Are able to fuse information
4. Have the ability to learn

Many of these requirements can be addressed by reasoning with probabilities and are the subject of other advanced courses Machine Learning.

Deep Learning for Computer Vision

We will focus on Deep Learning architectures based on **Convolutional Neural Networks** (CNNs).

CNNs have multiple layers of feature responses which are obtained by filtering/convolutions and non-linear activation functions. The weights of each filter are learned from training examples and deep networks will typically have millions (and even billions!) of parameters.



CNNs have been shown to be very effective at learning a hierarchy of features and representations for computer vision tasks. In particular they are used in many **recognition** tasks including text and face recognition, object detection and semantic segmentation.

These architectures (and the simple algorithms to train them) were first introduced in the 1980's. It is only in the last-decade that they have achieved state-of-the art performance on computer vision tasks. This is due to the availability of very large amounts of labelled training data; deeper networks and specialised computing hardware (GPUs) that can speed up the training algorithms (based on stochastic gradient descent optimisation) by many orders of magnitude.

Syllabus

1. **Introduction**
    - Computer vision: what is it, why study it and how?
    - Vision as an information processing task
    - Geometrical and statistical frameworks for vision
    - 3D interpretation of 2D images

2. **Image structure**
    - Image intensities and structure
    - 2D convolution with gaussians for low-pass filtering
    - Edge detection, the aperture problem and corner detection
    - Image pyramids, blob detection with band-pass filtering
    - The SIFT feature descriptor for matching
    - Characterising textures.

3. **Projection**
    - Orthographic projection
    - Planar perspective projection, vanishing points and lines
    - Homogeneous coordinates and the projection matrix
    - Camera calibration, recovery of world position
    - Weak perspective and the affine camera

4. **Stereo vision and Structure from Motion**
    - Recovery of depth by triangulation
    - Epipolar geometry and the essential matrix
    - Uncalibrated cameras and the fundamental matrix
    - The correspondence problem
    - Structure from motion
    - 3D shape examples from multiple view stereo and photometric stereo

5. **Deep Learning for Computer Vision**
    - Basic architectures for deep learning in computer vision
    - Detection, classification and semantic segmentation
    - Recognition, feature embedding and metric learning
    - Transformers, scaling laws for computer vision, neural architecture search
    - Self-supervised learning and pseudo-labelling

Course book: V. S. Nalwa. *A Guided Tour of Computer Vision*, Addison Wesley, 1993 (CUED shelf mark: NO 219).

Further reading

Students looking for a deeper understanding of computer vision might wish to consult the following publications, many of which are available in the CUED library.

Journals
International Journal of Computer Vision
IEEE Transactions on Pattern Analysis and Machine Intelligence

Conference proceedings
Computer Vision and Pattern Recognition Conference
International Conference on Computer Vision
European Conference on Computer Vision
British Machine Vision Conference

Books

R. Cipolla and P. Giblin *Visual Motion of Curves and Surfaces.* CUP, 1999.

D.A. Forsyth and J. Ponce. *Computer Vision - A Modern Approach.* Prentice Hall 2003.

* R. Hartley and A. Zisserman. *Multiple View Geometry.* CUP 2000.

J. J. Koenderink. *Solid shape.* MIT Press, 1990.

D. Marr. *Vision: a computational investigation into the human representation and processing of visual information.* Freeman, 1982.

* S.J.D. Prince *Computer Vision: Models, Learning and Inference.* CUP, 2012.

* R. Szeliski. *Computer Vision: algorithms and applications.* Springer, 2011.

B. A. Wandell. *Foundations of vision.* Sinauer Associates, 1995.

See also the bibliographies at the end of each handout.

Mathematical Preliminaries

Linear least squares

Consider a set of m linear equations

$$\mathrm{A}\mathbf{x} = \mathbf{b}$$

where $\mathbf{x}$ is an n-element vector of unknowns, $\mathbf{b}$ is an m-element vector and A is an $m \times n$ matrix of coefficients. If $m > n$ then the set of equations is *over-constrained* and it is generally not possible to find a precise solution $\mathbf{x}$.

The equations can, however, be solved in a **least squares** sense. That is, we can find a vector $\mathbf{x}$ which minimizes

$$\sum_{i=1}^{m} r_i^2$$

where

$$\mathrm{A}\mathbf{x} = \mathbf{b} + \mathbf{r}$$

$\mathbf{r}$ is the vector of *residuals*.

The least squares solution is found with the aid of the **pseudo-inverse**:

$$\mathrm{A}^{\dagger} = \left(\mathrm{A}^T\mathrm{A}\right)^{-1}\mathrm{A}^T$$

The least squares solution is then given by $\mathbf{x} = \mathrm{A}^{\dagger}\,\mathbf{b}$.

Mathematical Preliminaries

Eigenvectors and eigenvalues

Often the equations can be written as a set of m linear equations

$$\mathrm{A}\mathbf{x} = \mathbf{0}$$

where $\mathbf{x}$ is an n-element vector of unknowns and A is an $m \times n$ matrix of coefficients.

A non-trivial solution for $\mathbf{x}$ (up to an arbitrary magnitute) can be found if $m > n$. The solution is chosen to minimize the residuals given by $|\mathrm{A}\mathbf{x}|$ subject to $|\mathbf{x}| = 1$.

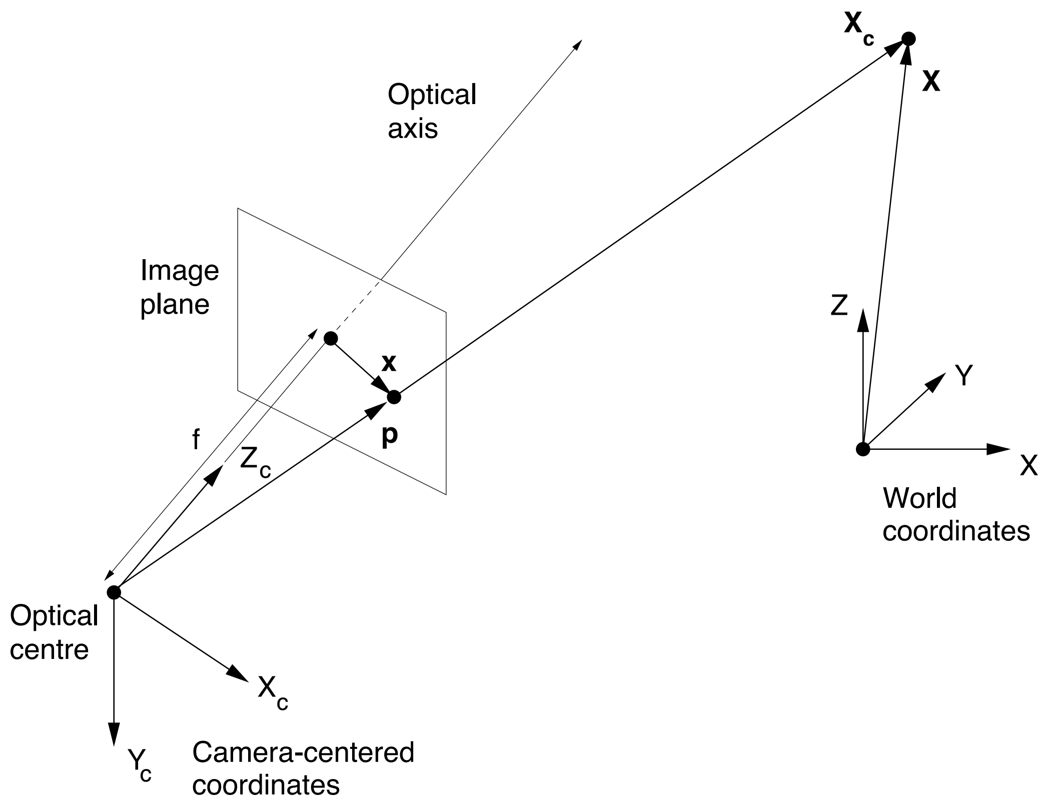
By considering Rayleigh's Quotient:

$$\lambda_1 \leq \frac{\mathbf{x}^T \mathrm{A}^T \mathrm{A} \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \leq \lambda_n$$

it is easy to show that the solution is the eigenvector corresponding to the smallest eigenvalue of the $n \times n$ symmetric matrix $\mathrm{A}^T\mathrm{A}$.

Notation

Metric coordinates



World coordinates

$\mathbf{X} = (X, Y, Z)$	Point in 3D space
$\mathbf{X}^p = (X, Y)$	Point on 2D plane
$\mathbf{X}^l = (X)$	Point on 1D line

Camera-centered coordinates

$\mathbf{X_c} = (X_c, Y_c, Z_c)$	Point in 3D space
$\mathbf{p} = (x, y, f)$	Ray to point on image plane
$\mathbf{x} = (x, y)$	Image plane coordinates

Pixel coordinates

$\mathbf{w} = (u, v)$	Pixel coordinates

Notation

Projection and transformation matrices

R	Rotation matrix (orthonormal)
$\mathbf{T}$	Translation vector (3 element)
P_r	Rigid body transformation matrix (3D)
P_p	Perspective projection matrix
P_{pll}	Parallel projection matrix (weak perspective)
P_c	CCD calibration matrix
P_{ps}	Overall perspective camera matrix
P_{wp}	Overall weak perspective camera matrix
P	Overall projective camera matrix
P_{aff}	Overall affine camera matrix
$[\;]^p$	Superscript for plane imaging matrices
$[\;]^l$	Superscript for line imaging matrices

Stereo

$\mathbf{X_c}$, $\mathbf{p}$, $\mathbf{w}\ldots$	Left camera quantities
$\mathbf{X'_c}$, $\mathbf{p}'$, $\mathbf{w}'\ldots$	Right camera quantities
$\mathbf{p_e}$, $\mathbf{p'_e}$	Rays to epipoles
$\mathbf{w_e}$, $\mathbf{w'_e}$	Pixel coordinates of epipoles
E	Essential matrix
F	Fundamental matrix

Motion

$\mathbf{v} = (\dot{x}, \dot{y})$	Image motion field
$\mathbf{U}$	Camera's linear velocity
Ω	Camera's angular velocity
Δ	Motion parallax vector