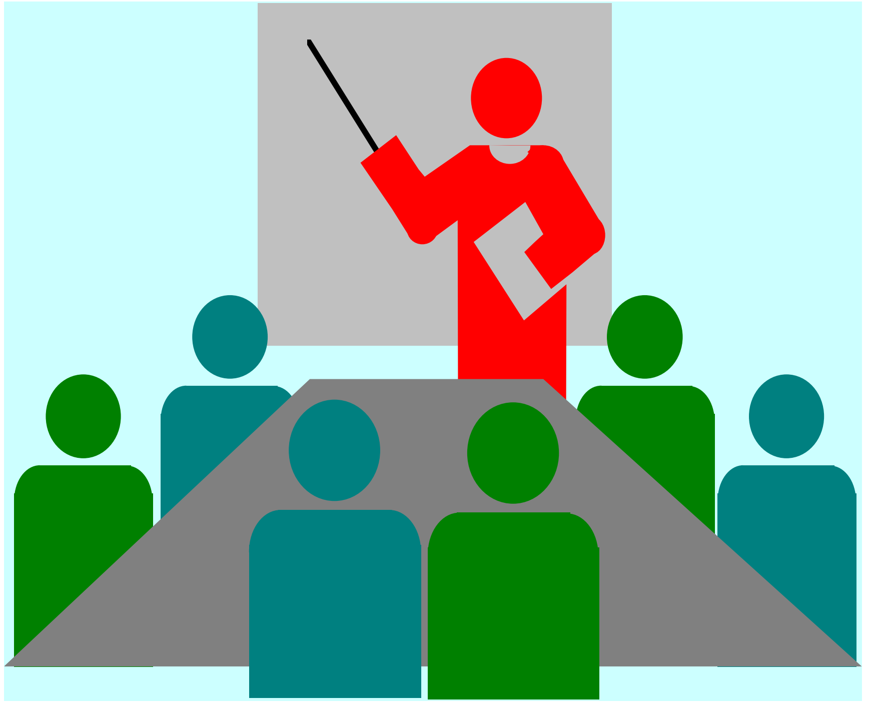
Topic 6: External Sensors for Navigation

Global Positioning & Computer Vision

6.1 Global positioning
- 6.1.1 Relative position measuring
- 6.1.2 Absolute position measuring
- 6.1.3 Kalman filtering algorithm

6.2 Computer vision
- 6.2.1 Image transformation
- 6.2.2 Image segmentation
- 6.2.3 Image understanding
- 6.2.4 Real world applications

6.3 Summary



6.1 Global Positioning

Two typical global positioning:

Relative position measurements

Odometry -- Using encoders to measure wheel rotation & steering angle to estimate the robot position based on the robot kinematics equations.

Inertial navigation -- Using gyros and accelerometers to measure rate of rotation and acceleration to obtain the position.

Absolute position measurements

Active beacons -- to compute the absolute robot position from 3 or more actively transmitted beacons (light or RF) at known sites.

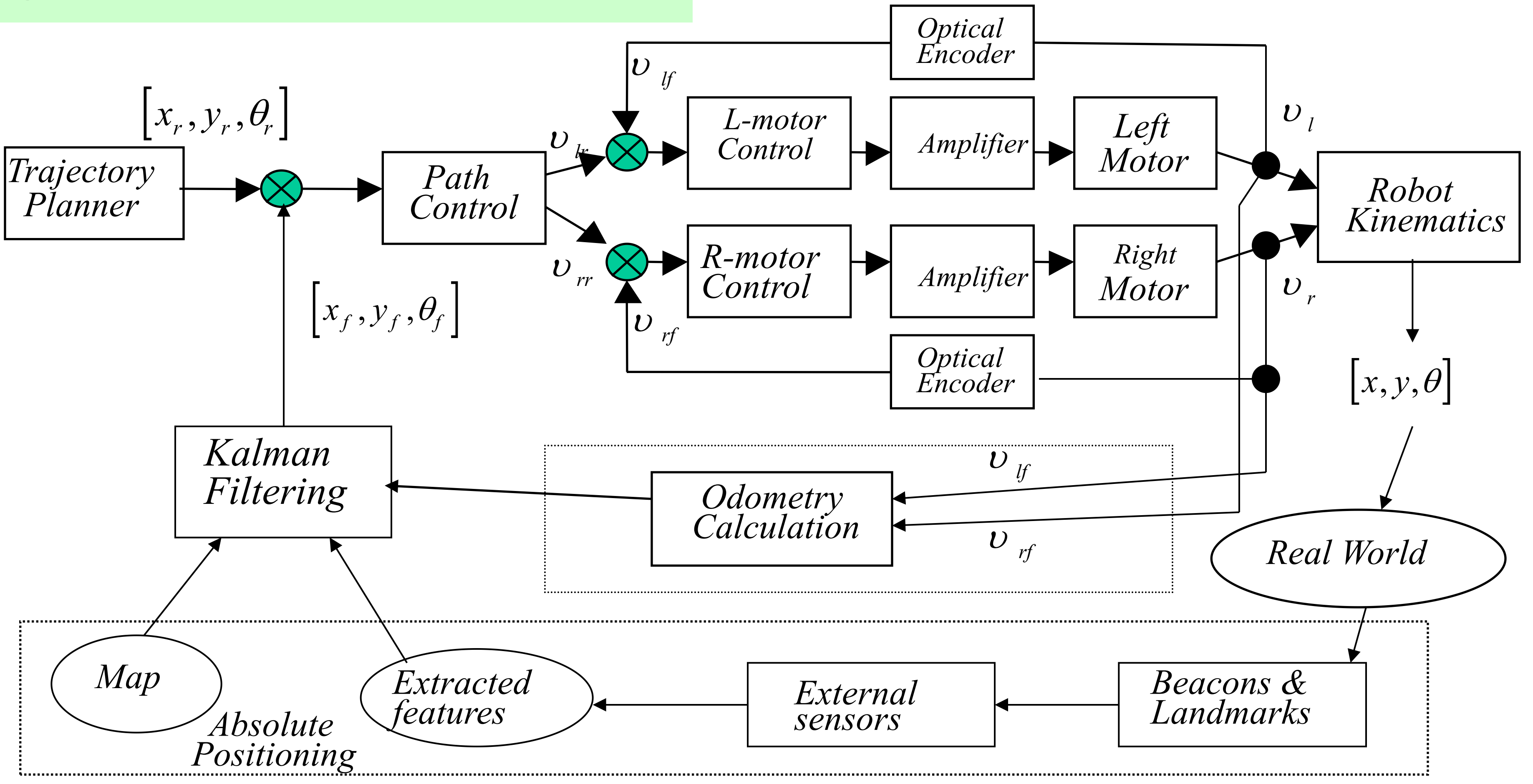
Artificial landmarks -- use specially designed objects/markers placed at the known locations in the environment.

Natural landmarks -- are distinctive features in the environment and known in advance. Their reliability are worse than with artificial landmarks.

Model matching -- the onboard sensor data is compared to a map/model of the environment to estimate the robot's absolute position.

6.1 Global Positioning

General localization Process:



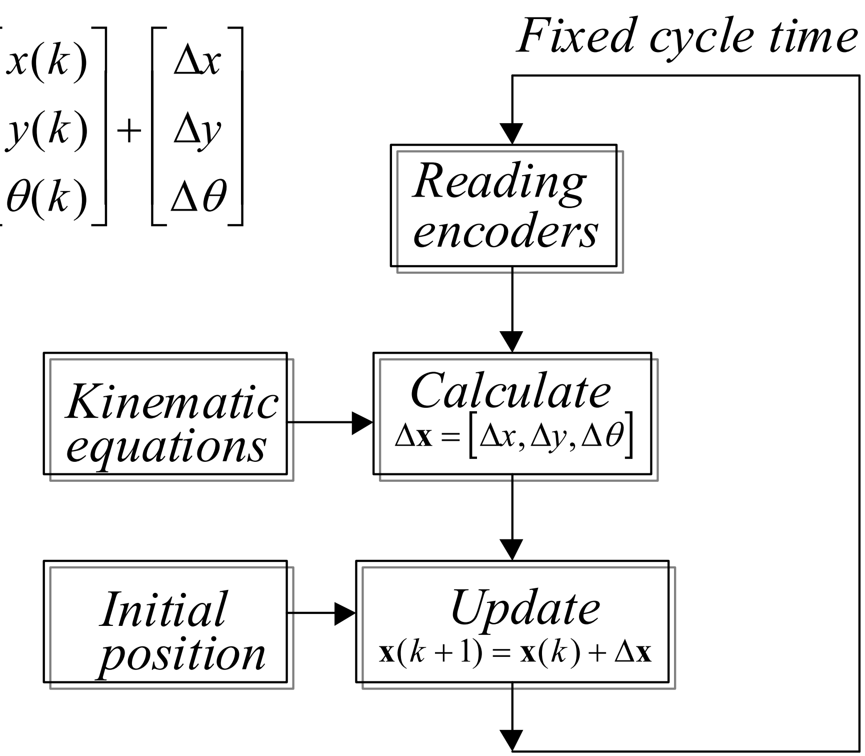
6.1.1 Relative Position Measurement

Odometry calculation process:

A general form:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \Delta\mathbf{x} \quad \text{or} \quad \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}$$

Note that $\Delta\mathbf{x}$ is depended on the wheel configuration of mobile robots.



Example:

Odometry for omni-directional mobile robots:

$$x(k+1) = x(k) + \Delta d \cos\phi$$
$$y(k+1) = y(k) + \Delta d \sin\phi$$

where $\{\Delta d, \phi\}$ is the distance & tangential direction of the path being travelled.

6.1.1 Relative Position Measurement

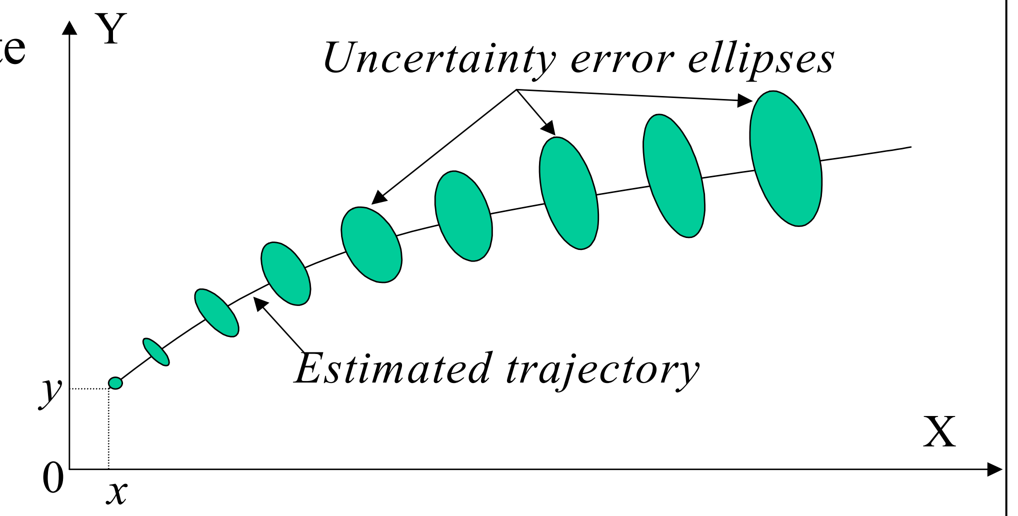
Two main errors in odometry calculation:

Systematic errors

- Backlash in motors and gearboxes
- Deviation from nominal wheel diameters
- Actual wheelbase differs from nominal wheelbase
- Misalignment of wheels
- Finite encoder resolution & sampling rate

Non-systematic errors

- Travel over uneven floor surfaces
- Travel over unexpected objects
- Wheel slippage due to slippery floors, over acceleration, fast turning



6.1.1 Relative Position Measurement

6.1.2 Absolute Position Measurements

1. Active beacons

-- used for many centuries as a reliable and accurate means for navigation

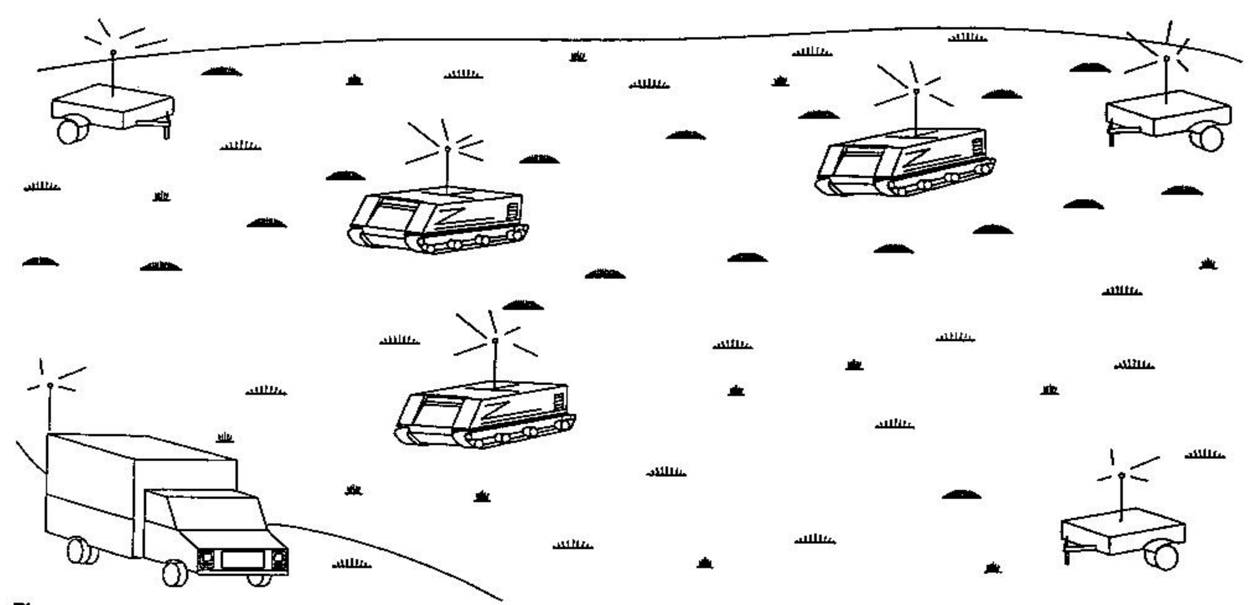
Radio Frequency Navigation Grid

The Unmanned Vehicle Control Systems Group of Kaman Sciences Corporation in the USA has developed a small-scale 1500W *Radio Frequency Navigation Grid (*RFNG*)* which covers a 30km^2 area as shown in the figure below.

-- Measuring the phase differences in received signals from a few transmitters

-- System resolution is 3cm at a 20 Hz update rate.

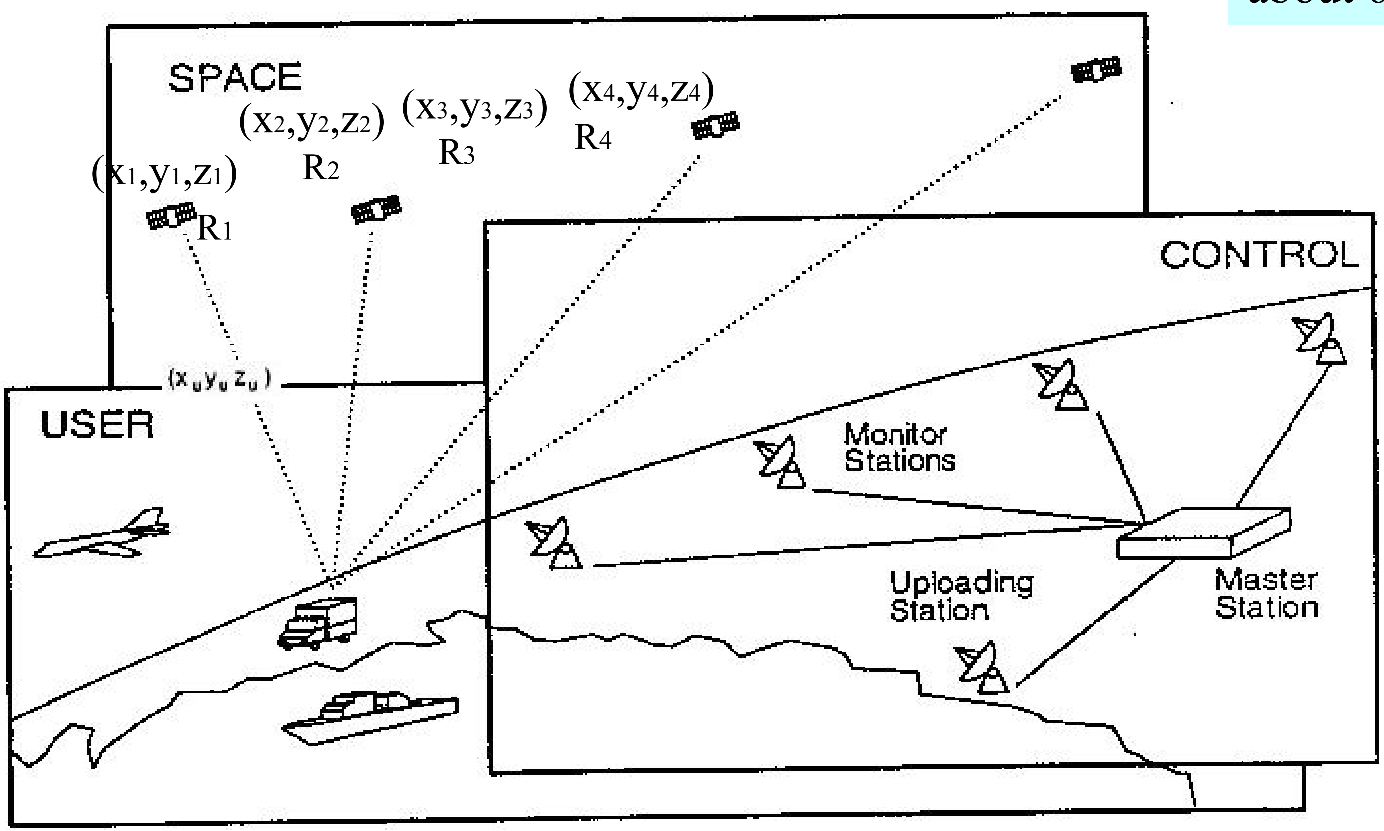
-- The remote vehicles have a position-repeatability of 1m.



6.1.2 Absolute Position Measurements

Navistar Global Positioning Systems (GPS):

24 Satellites at a height of about 6,900 nautical miles.



Main problems:

-- Accurate measurement of signal propagation time (at the light speed *c* = 0.3m/ns).

-- Time synchronisation between individual satellites & GPS receiver.

6.1.2 Absolute Position Measurements

2. <u>Artificial landmarks</u>:

-- Special geometrical shapes: rectangles, lines, circles, see Fig. (i).

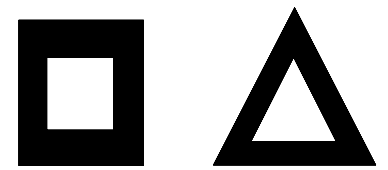
-- Barcode targets for laser scanners, see Fig. (ii).

-- They have a fixed and known position, see Fig.(iii)

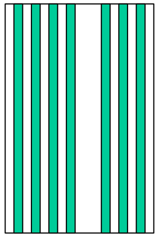
-- They should be sufficient contrast to the background in order to be easy to identify.

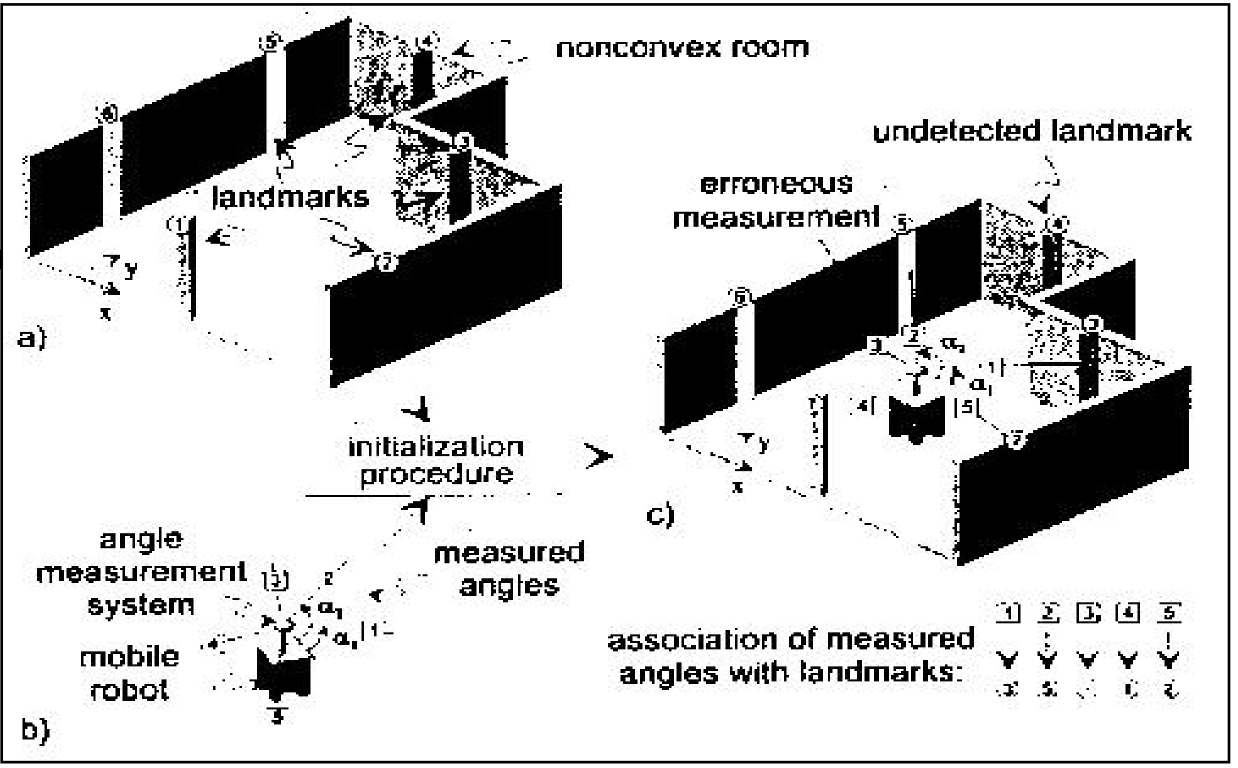
-- Their characteristics must be preinstalled in the robot's memory.

(i)



(ii)





(iii) Artificial landmarks at the Univ. of Munich

6.1.2 Absolute Position Measurements

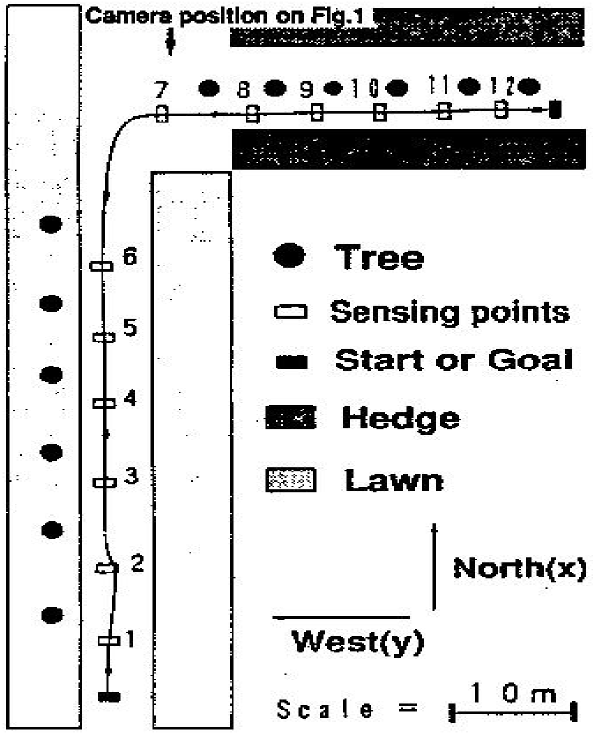
6.1.2 Absolute Position Measurements

3. Natural landmarks:

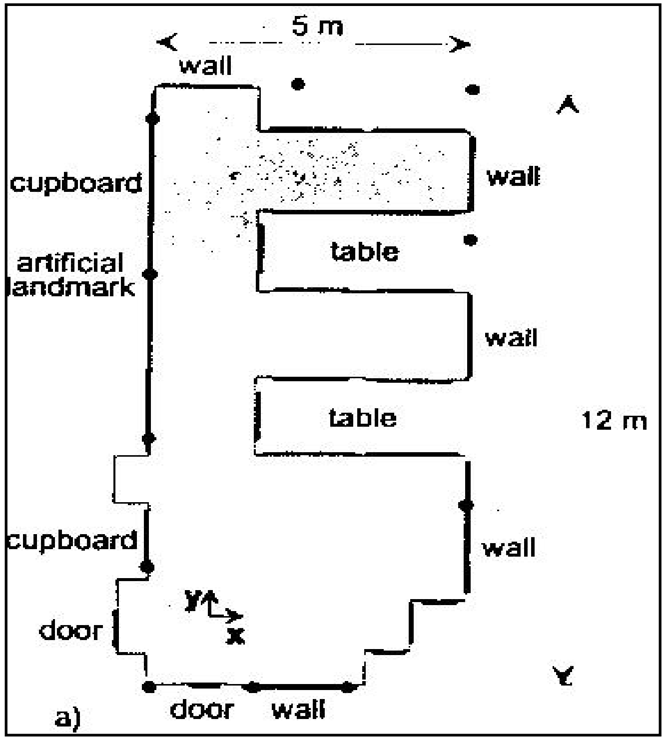
Objects or features that are already in the environment and have a function. include edges, corners, doors, wall, hallway, furniture, lights, trees, *etc.*

Natural landmark systems:

- Sensors (e.g. vision, sonar, laser,…) for detecting landmarks.
- Methods to abstract natural landmarks from their background.
- Algorithms for matching observed features with a map of known landmarks.
- A method to compute location and localization errors from matches.



Outdoor navigation



Indoor navigation

6.1.2 Absolute Position Measurements

6.1.2 Absolute Position Measurements

6. Map-based Positioning

It is a technique for a robot to create a map of its local environment. This local environment is then compared to a global map previously stored in memory. If a match is found, then the robot can compute its actual position and orientation in the environment. The pre-stored map can be a CAD model of the environment, or it can be constructed from prior sensor data.

Two methods in map-based positioning:

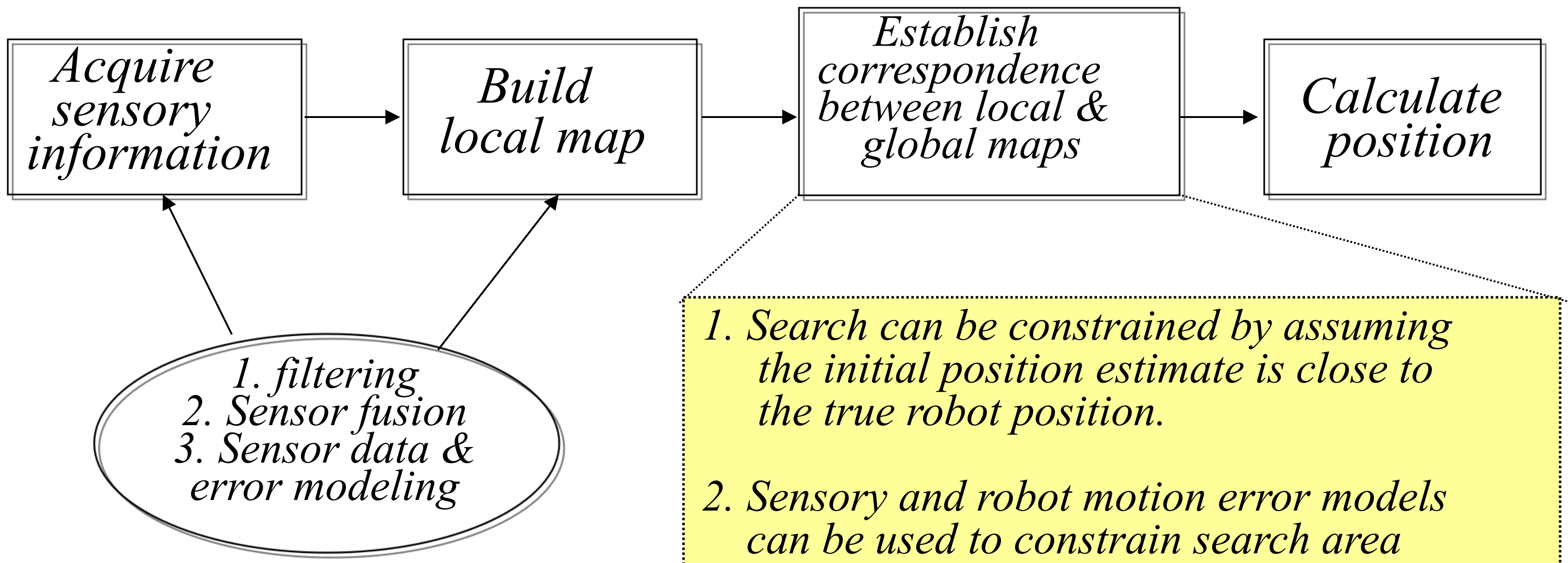
-- Using a pre-existing or pre-stored environment map and updating it.
-- Building an environment map from scratch during the robot motion.

Three steps in map building process:

-- Feature extraction from raw sensor data.
-- Fusion of data from various types of sensors.
-- Automatic generation of an environment map with different degrees of abstraction.

6.1.2 Absolute Position Measurements

General procedure for map-based positioning:



6.1.3 Kalman Filtering Algorithms

System models:

- Positon vector: $X(k) = [x(k), y(k), \theta(k)]$
- Odometry: $U(k) = [d(k), \alpha(k)]$
- Scanner observation: $B_i = [x_i, y_i, \phi_i]$
- Nonlinear system model:

$$X(k+1) = \mathbf{F}(X(k), U(k)) + W(k)$$

$$F = \begin{bmatrix} x(k) + d(k)\cos\alpha(k)\cos(\theta(k)+\alpha(k)) \\ y(k) + d(k)\cos\alpha(k)\sin(\theta(k)+\alpha(k)) \\ \theta(k) + d(k)\sin\alpha(k)/(C+D) \end{bmatrix}$$

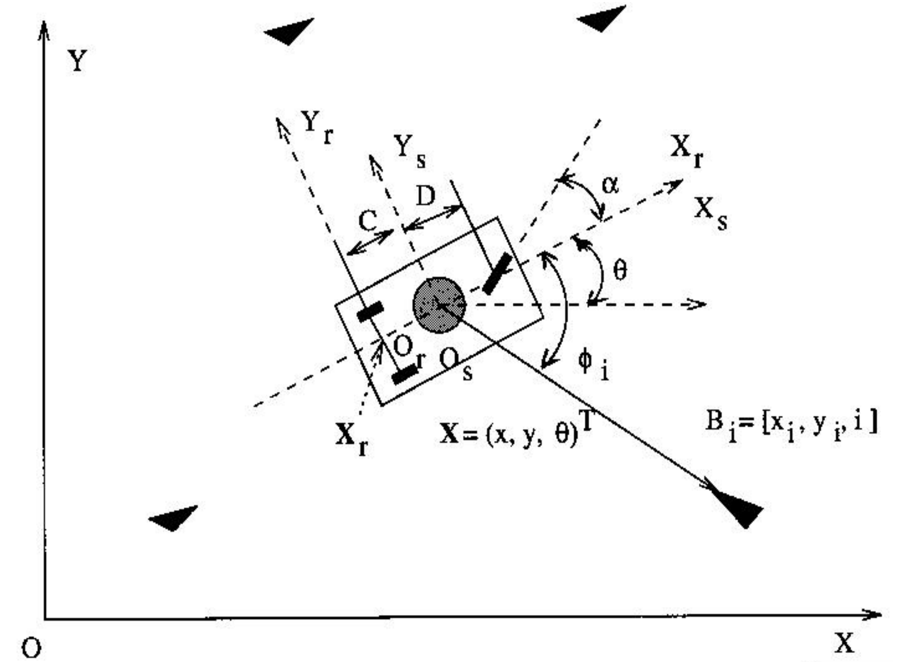
- Nonlinear observation model:

$$Z(k) = H(B_i, X(k)) + V(k)$$

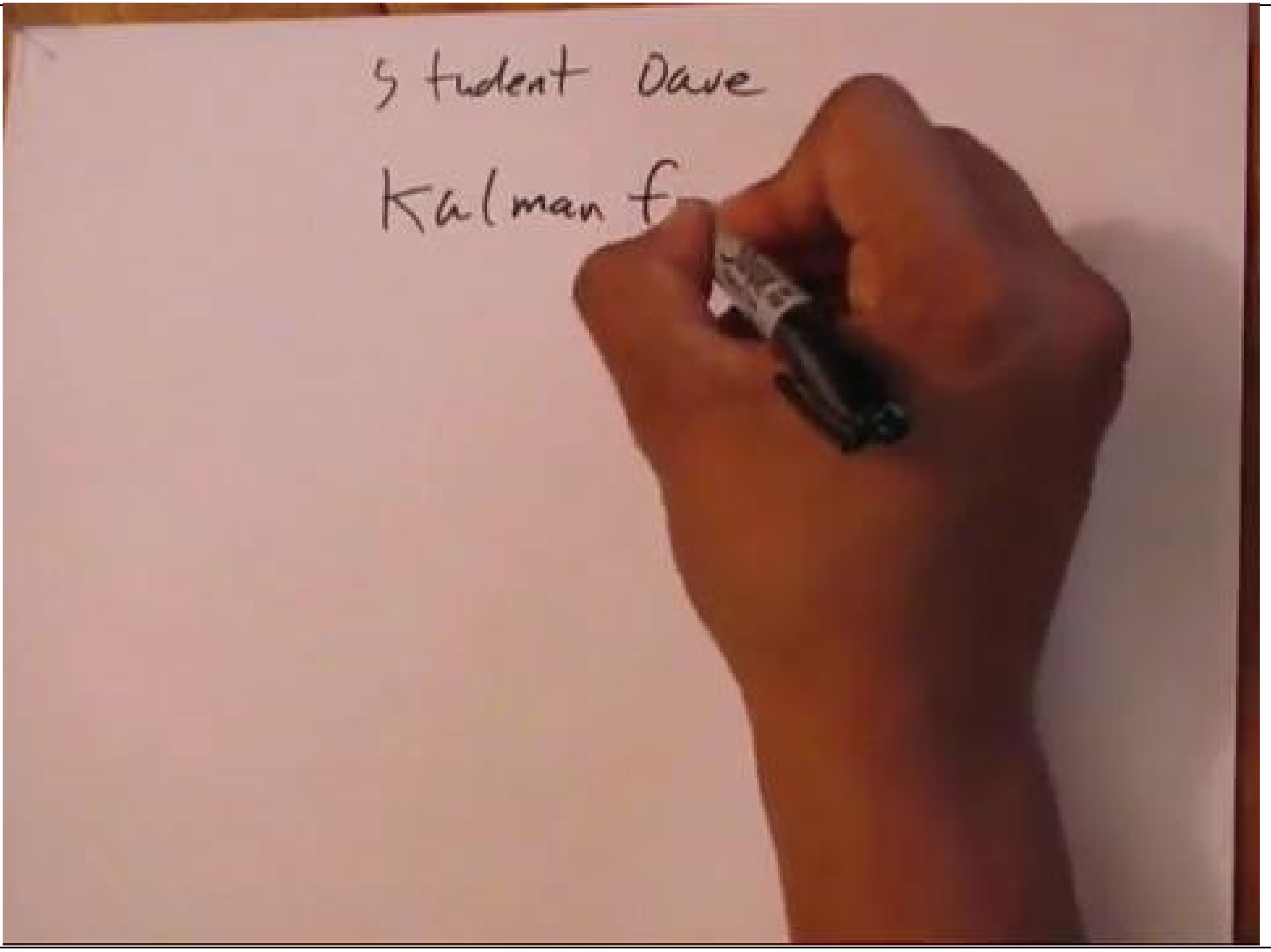
$$H(B_i, X(k)) = \arctan\frac{y_i - y(k)}{x_i - x(k)} - \theta(k)$$

- Noise model:

$$W(k) \sim \mathcal{N}(0, Q(k)), \quad V(k) \sim \mathcal{N}(0, R(k))$$



6.1.3 Kalman Filtering Algorithms



6.1.3 Kalman Filtering Algorithms

Four-step operation:

Prediction → *Observation* → *Matching* → *Correction*

Note that: Kalman filters have been widely used to integrated data from odometry and from other sensors which detect and abstract useful features from the environment.

Recursive implementation:

Step 1: **Prediction** -- predict the next position of the robot using odometry

$$\mathbf{x}(k+1/k) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))$$

$$\mathbf{P}(k+1/k) = \nabla\mathbf{f}\mathbf{P}(k/k)\nabla\mathbf{f}^{T} + \mathbf{Q}(k)$$

where $\nabla\mathbf{f}$ is the Jacobean matrix of the transition function, and is obtained by linearization

6.1.3 Kalman Filtering Algorithms

$$\nabla \mathbf{f} = \begin{bmatrix} 1 & 0 & -\Delta d(k)\sin\theta(k) \\ 0 & 1 & \Delta d(k)\cos\theta(k) \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2: Observation -- make actual measurement

The measurement of the laser scanner is

$$\mathbf{z}(k+1) = \mathbf{h}(B_i, \mathbf{x}(k))$$

The predicted angle measurement is

$$\hat{\mathbf{z}}(k+1) = \mathbf{h}(B_i, \hat{\mathbf{x}}(k+1/k))$$

Step 3: Matching -- compare real measurement with the predicted measurement

To calculate the innovation, then use

$$\mathbf{v}(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1)$$

6.1.3 Kalman Filtering Algorithms

The innovation covariance is: $\mathbf{S}(k+1) = \nabla\mathbf{h}\mathbf{P}(k+1/k)\nabla\mathbf{h}^T + \mathbf{R}(k+1)$

where $\nabla\mathbf{B}$ is the Jacobean matrix of the measurement function: $\nabla\mathbf{h} = \left[\frac{\partial\mathbf{h}}{\partial\mathbf{x}}, \quad \frac{\partial\mathbf{h}}{\partial\mathbf{y}}, \quad -1\right]$

For each measurement, a validation gate is used to decide whether it is a match or not:

$$\mathbf{v}(k+1)\mathbf{S}(k+1)\mathbf{v}^T(k+1) \le \mathbf{G}$$

If it is true, the current measurement is accepted. Otherwise, it is disregarded.

Step 4: Correction -- correct the prediction error from odometry readings

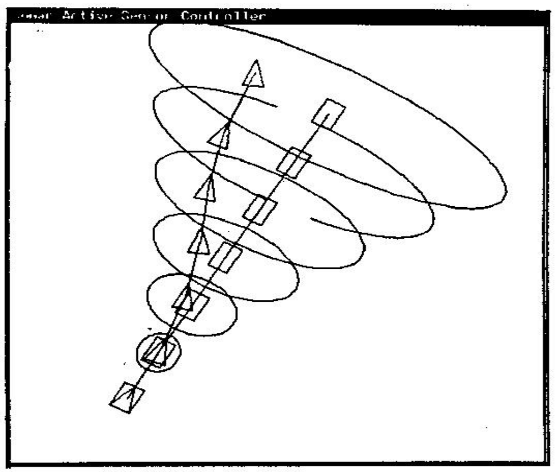
The filter gain is updated by: $\mathbf{W}(k+1) = \mathbf{P}(k+1/k)\nabla\mathbf{h}^T\mathbf{S}^{-1}(k+1)$

The robot state is then calculated by: $\mathbf{x}(k+1/k+1) = \hat{\mathbf{x}}(k+1/k) + \mathbf{W}(k+1)\mathbf{v}(k+1)$

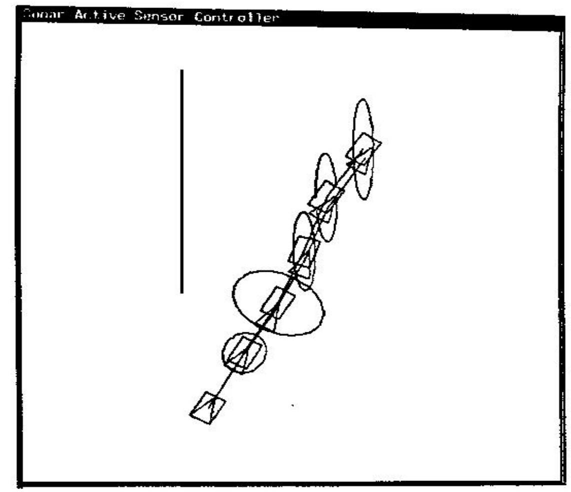
The covariance is updated by: $\mathbf{P}(k+1/k+1) = \mathbf{P}(k+1/k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$

Return to **Step 1** to recursively implement four steps above.

6.1.3 Kalman Filtering Algorithms

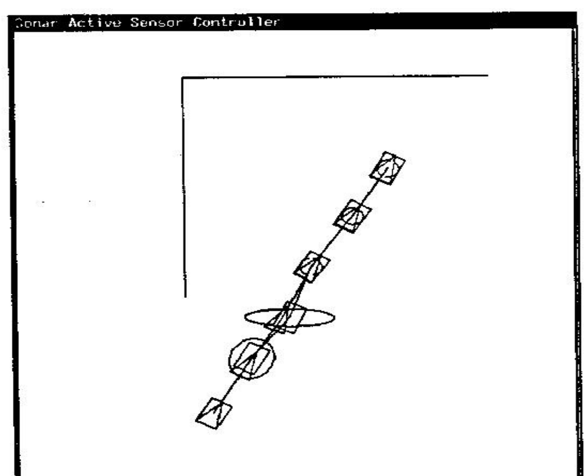


No landmark being observed

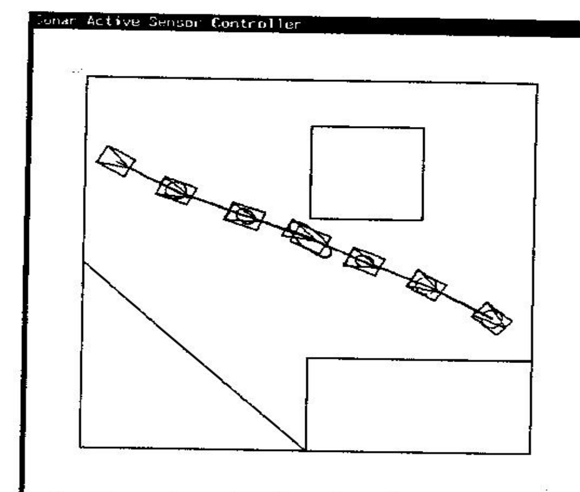


Wall being observed

A corner being observed



Multiple observations



6.2 Computer Vision

Why computer vision?

- Robots and intelligent machines must be able to see if they are to perform human-like tasks such as assembly, inspection and navigation.
- Vision provides robots with the capability to acquire information and learn from their environments autonomously.
- Vision will greatly enhance the abilities of robots or intelligent machines, comparing with other sensors.

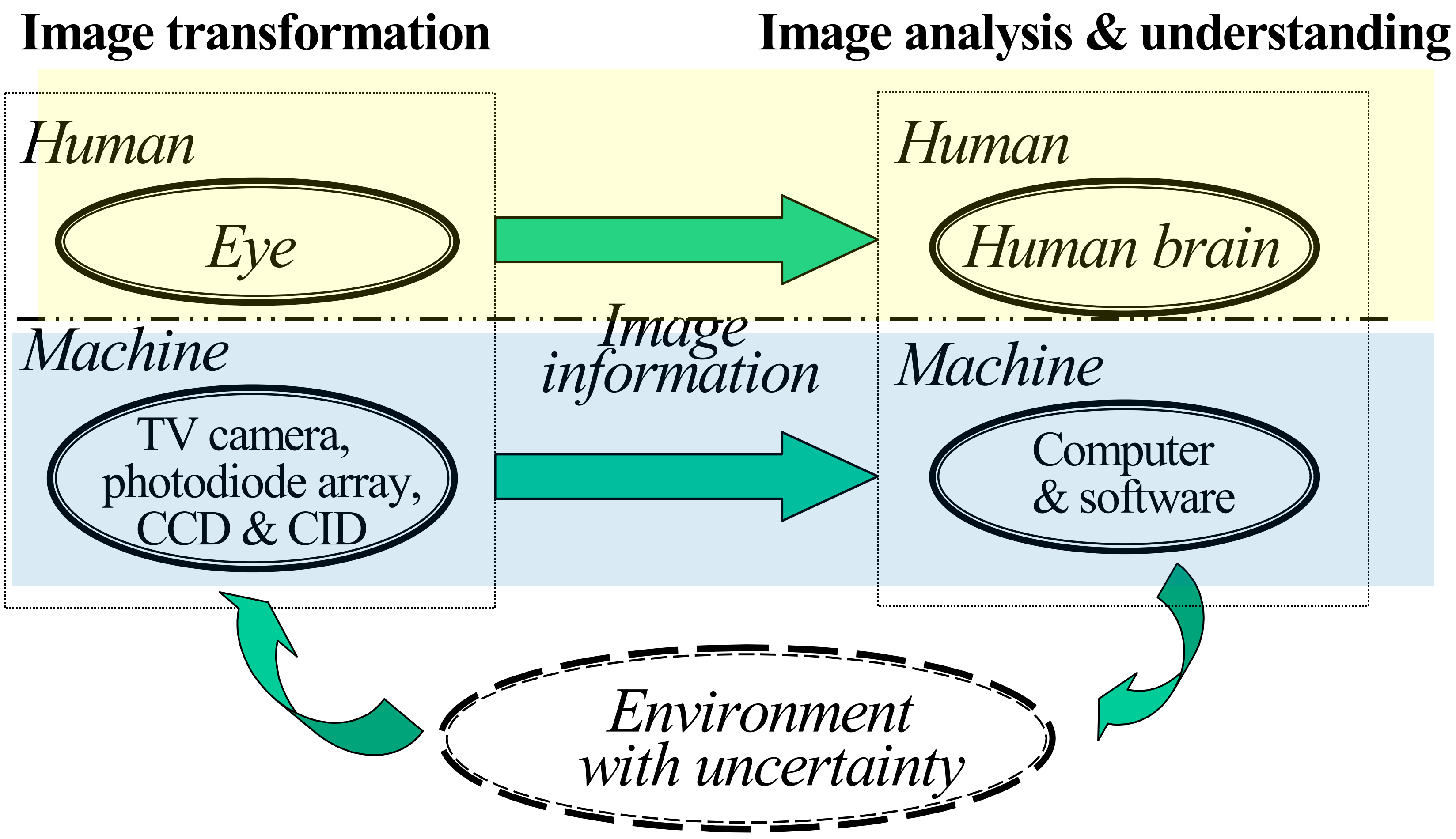
Three key processing stages of computer vision in robotics applications:

(i) image transformation;

(ii) image segmentation and analysis;

(iii) image understanding

6.2.1 Image Transformation



6.2.1 Image Transformation

Digitizing the image

- ***Photodiode array***

  -- A photodiode is a solid-state, light-sensitive device using a PN junction.

  -- Individual photodiodes for detection of the presence/absence of an object.

  -- 1D line-scan cameras & 2D dotted-matrix cameras.

- ***Charge-coupled device array***

  -- It is an array of MOSFET (*metal oxide semiconductor field-effect transistor*) devices.

  -- A set of integrated capacitors storing charges temporally.

  -- high sensitivity, small size, light weight, and lower power consumption.

6.2.1 Image Transformation

- *Charge-coupled device array*

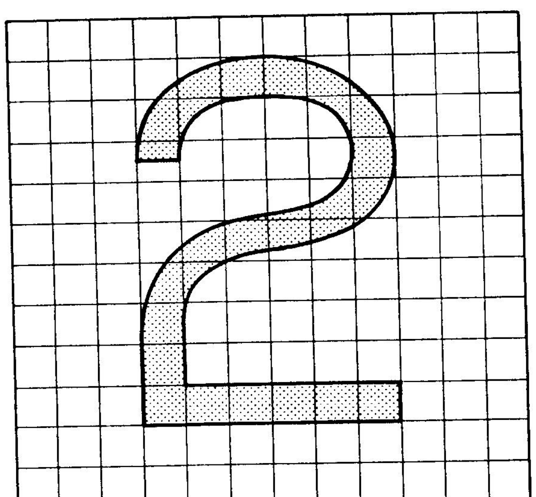
6.2.1 Image Transformation

Data structures

Matrices -- the most common data structure for low-level representation of images.

Gray image: Construct a 12 x 12 picture matrix for the digit 2 using an 8-bit gray scale code.

- A black pixel is assigned a value of 255 & a white pixel is assigned a value of 0.
- A gray pixel has a mixture of white & black, i.e. an average intensity.



0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	128	240	255	150	0	0	0	0
0	0	0	200	128	0	0	128	128	0	0	0
0	0	0	128	0	0	0	0	255	0	0	0
0	0	0	0	0	15	50	200	225	0	0	0
0	0	0	25	200	225	175	128	0	0	0	0
0	0	0	200	56	0	0	0	0	0	0	0
0	0	0	255	0	0	0	0	0	0	0	0
0	0	0	255	0	0	0	0	0	0	0	0
0	0	0	255	255	255	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

6.2.2 Image Segmentation & Analysis

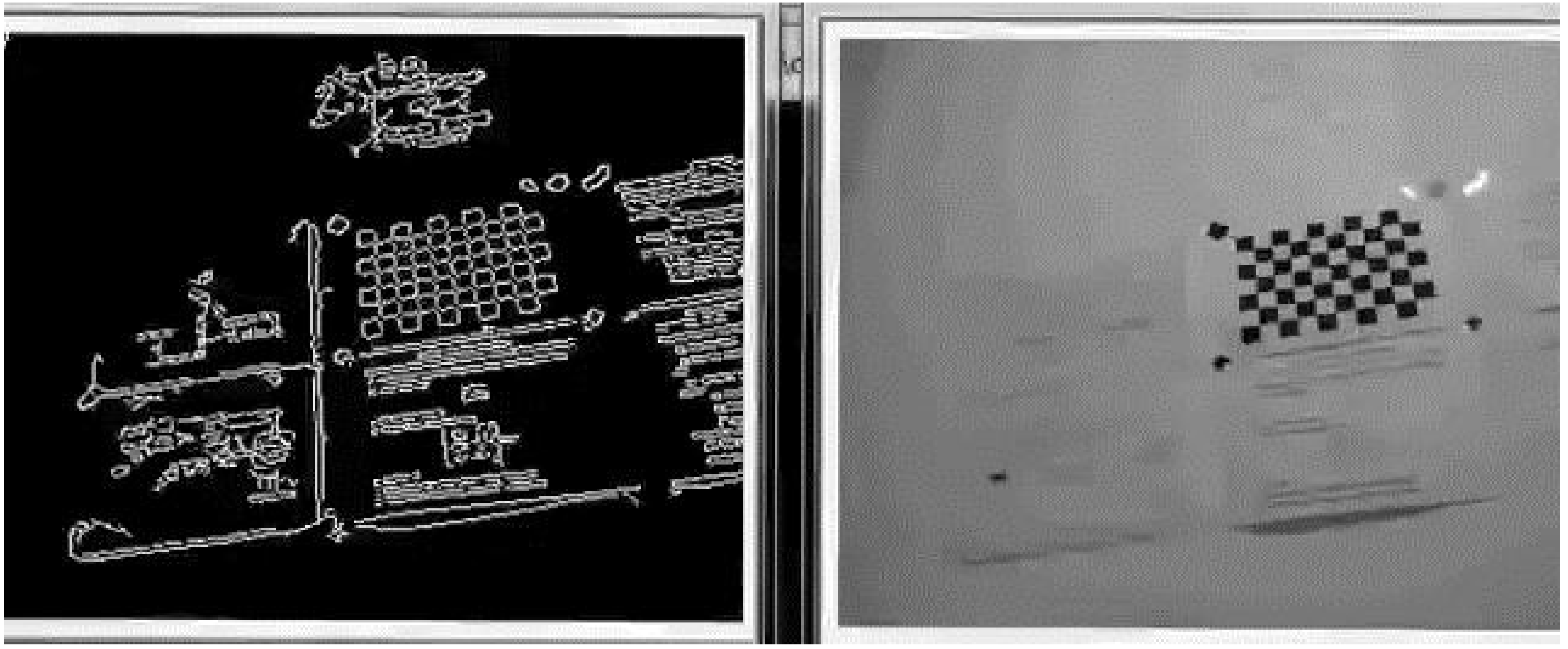
What is image segmentation

- The ***main purpose*** of image segmentation is to divide a image into parts that have a strong correlation with objects or areas in the image.
- The segmentation methods can be divided into three groups:
  - -- thresholding
  - -- edge-based segmentation
  - -- region-based segmentation (a closed boundary/edge)
- The major difficulties in image segmentation:
  - -- data ambiguity
  - -- information noise

Note: ***The more a priori information is available to the segmentation process, the better segmentation results that can be obtained.***

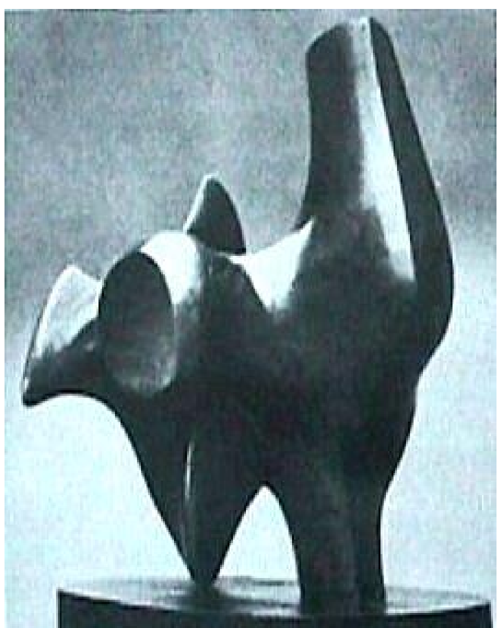
6.2.2 Image Segmentation & Analysis

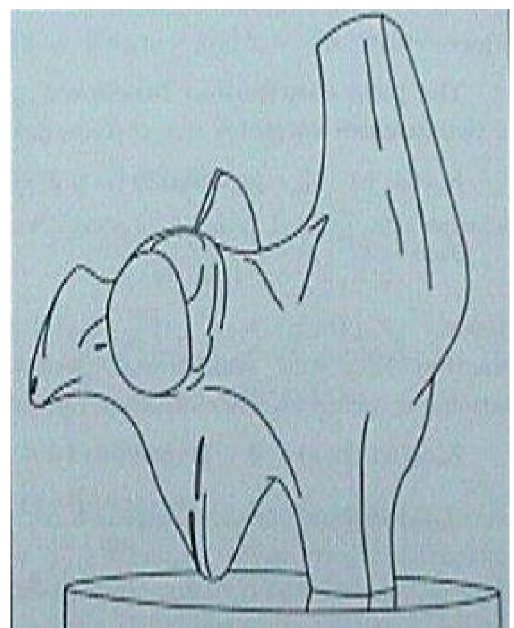
Canny Edge Detection

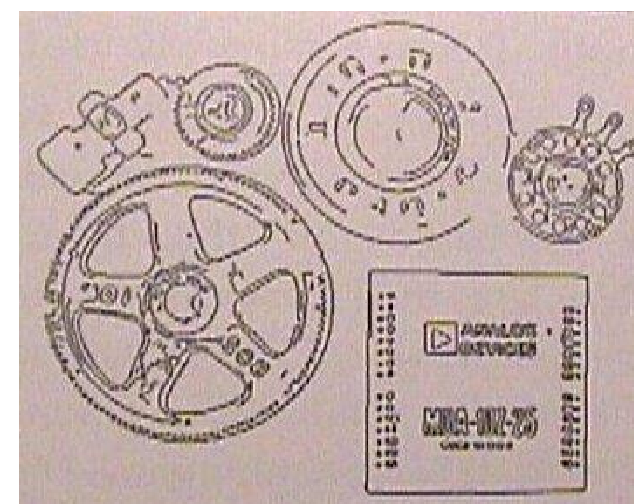


6.2.2 Image Segmentation & Analysis

Difficulties with Edge Detection:
(i) image noise;
(ii) environmental noise;
(iii) illumination & occlusion







6.2.2 Image Segmentation & Analysis

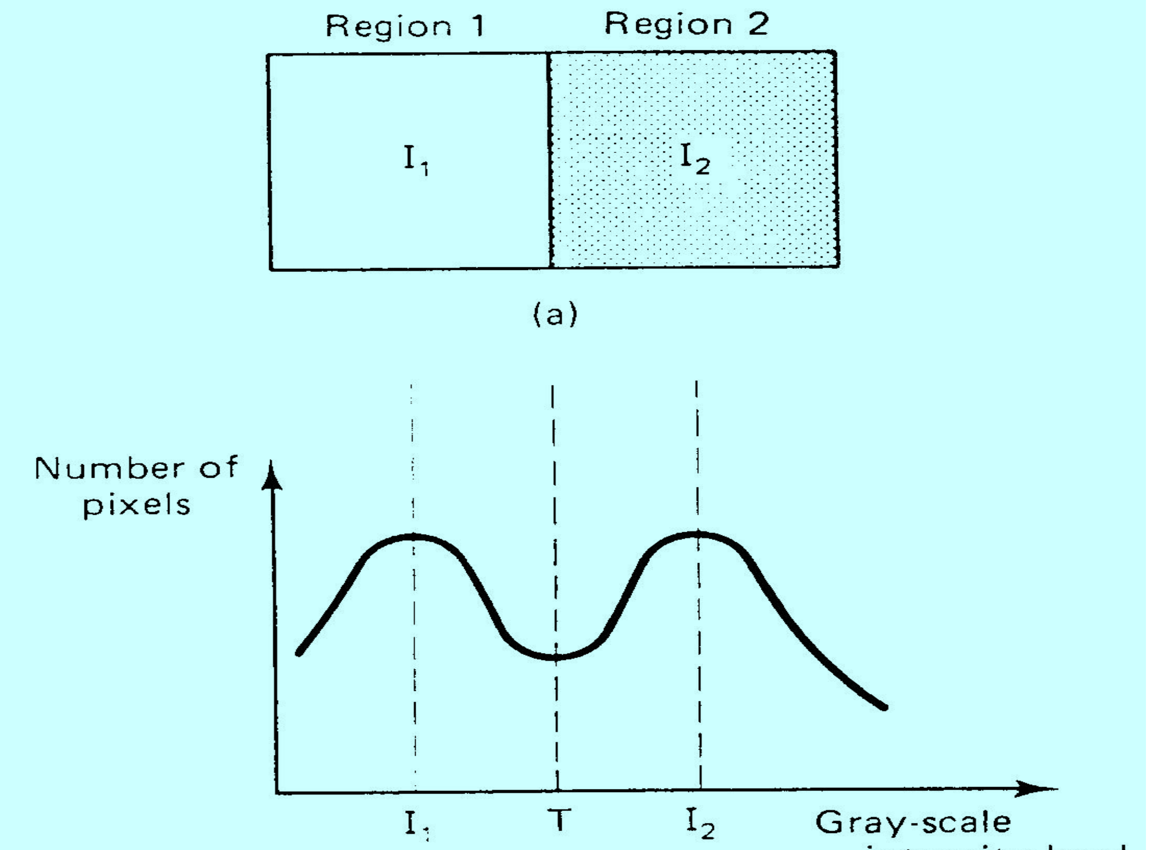
6.2.2 Image Segmentation & Analysis

- Region segmentation & analysis is to find the object shape in an image.
- Region analysis is to find groups of pixels that are similar and may belongs some features.
- Three image features are useful for region analysis:
  - ***-- grey-scale intensity,***
  - ***-- colour***
  - ***-- texture***

Three common methods to segment regions are as follows:

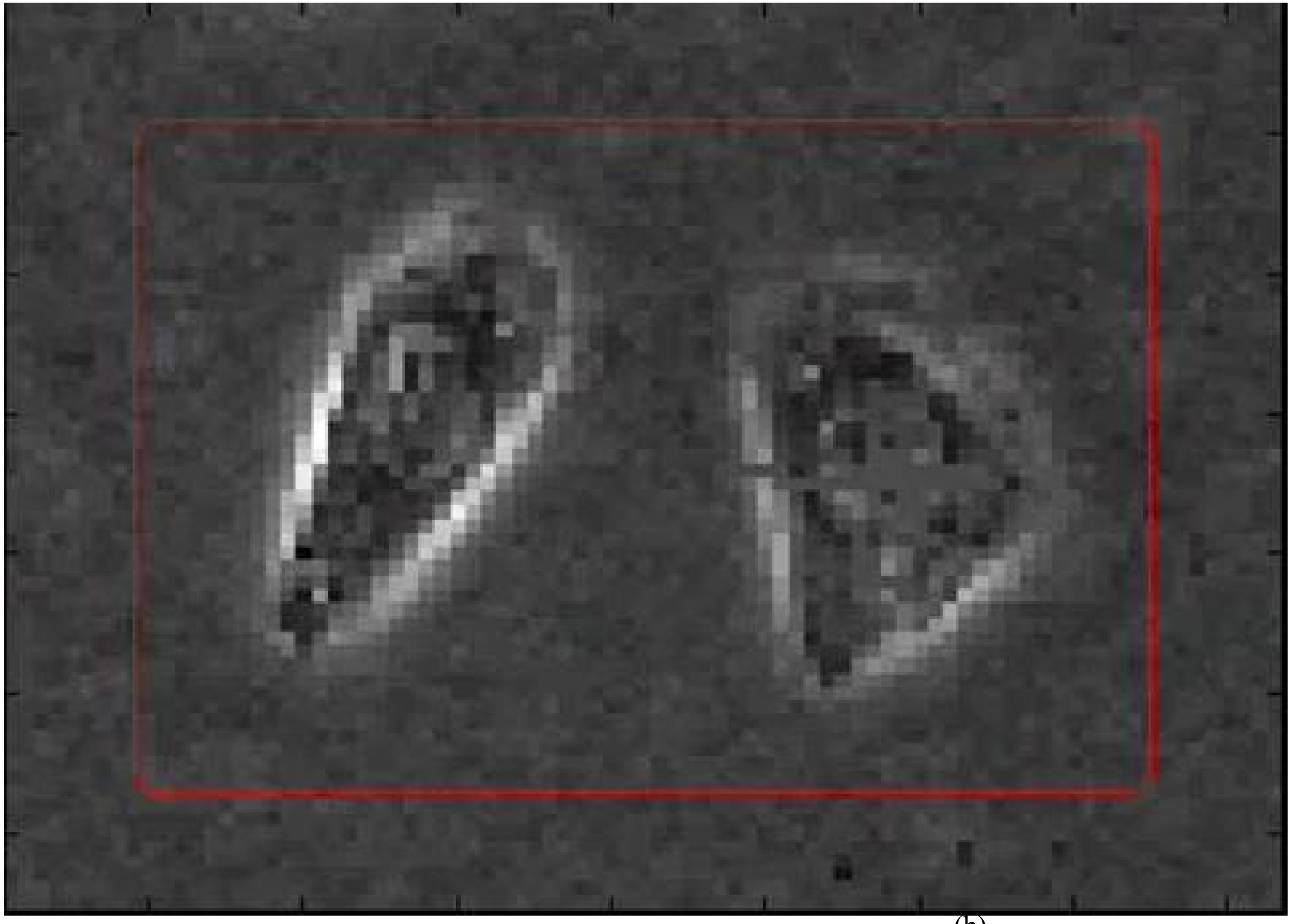
<u>1. Region splitting:</u>

-- to isolate different image features



(b)

6.2.2 Image Segmentation & Analysis



(b)

6.2.2 Image Segmentation & Analysis

2. Region growing:

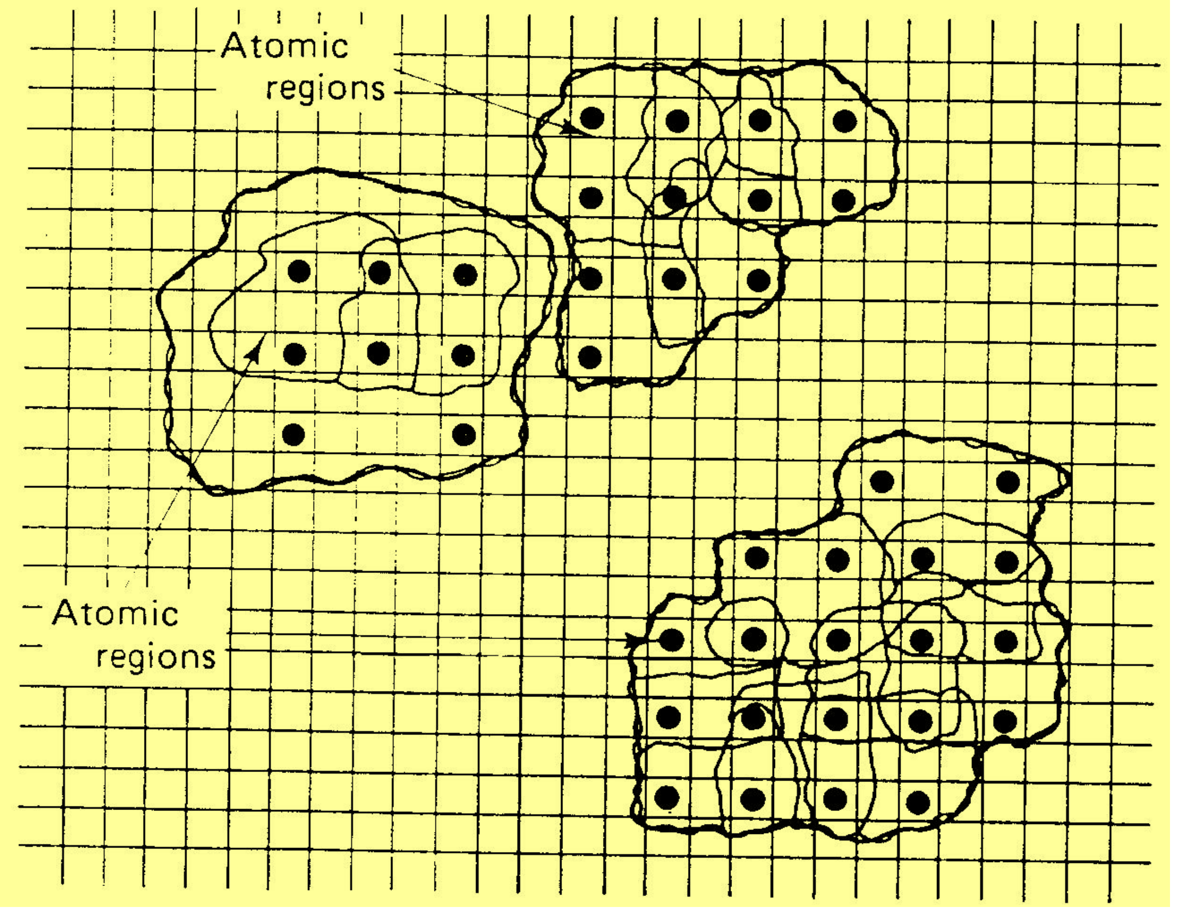
- to identify image features

  For example:

  The Blob algorithm used in the Pioneer robots for finding a colour ball

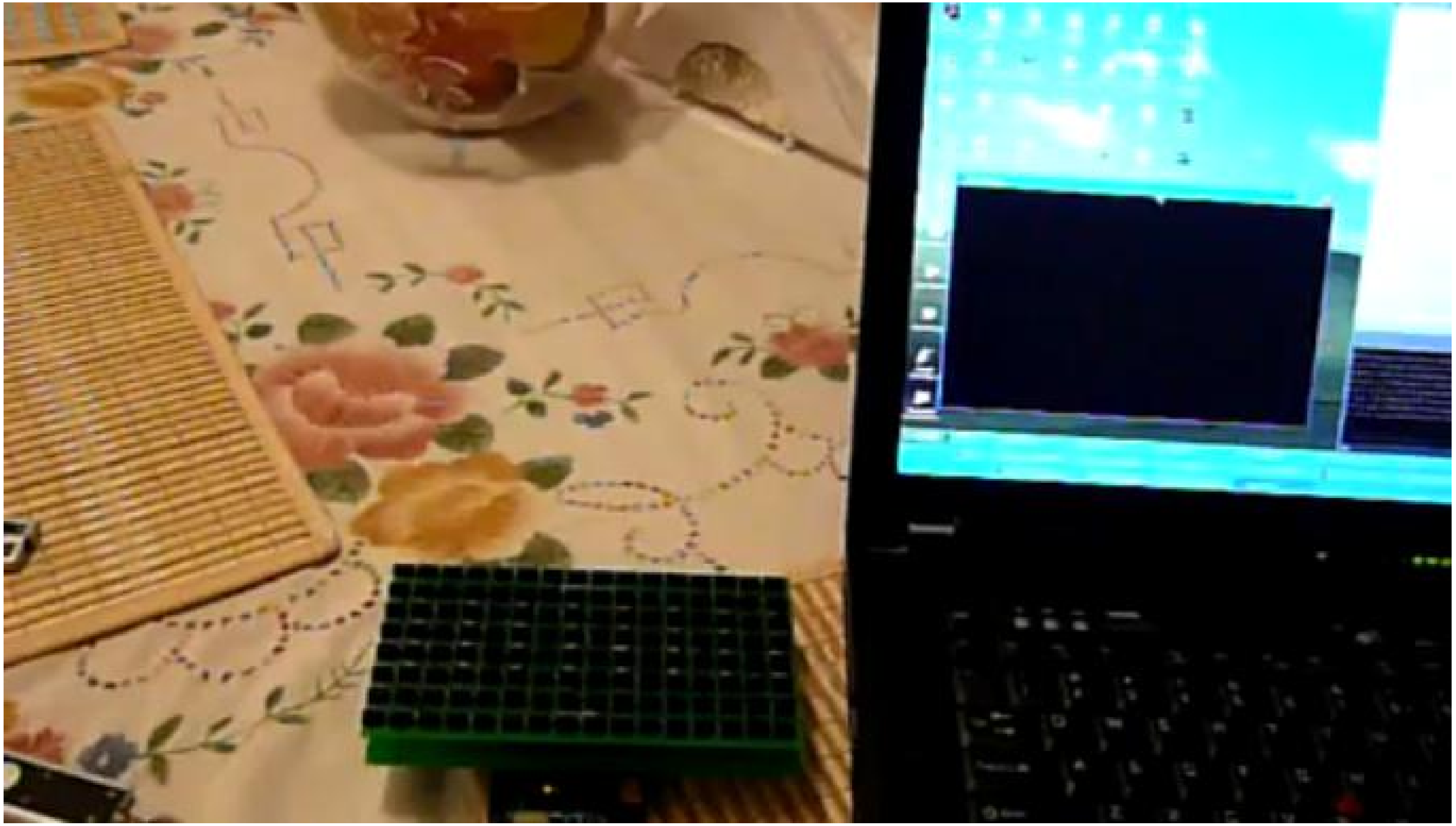
3. Watershed segmentation:

- starting with finding a downstream path from each pixel to local minima of image surface altitude.
- A catchments basin is then defined as the set of pixels whose downstream path all end up in the same altitude minima.



Region growing

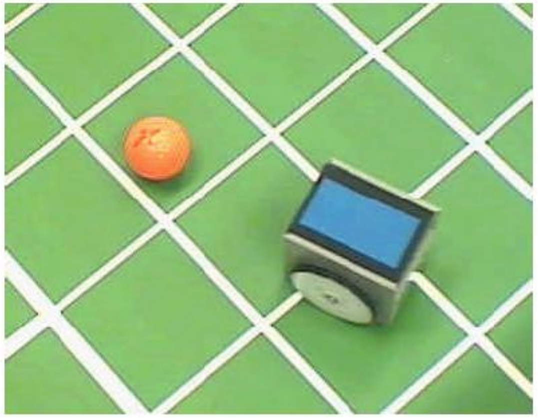
6.2.2 Image Segmentation & Analysis

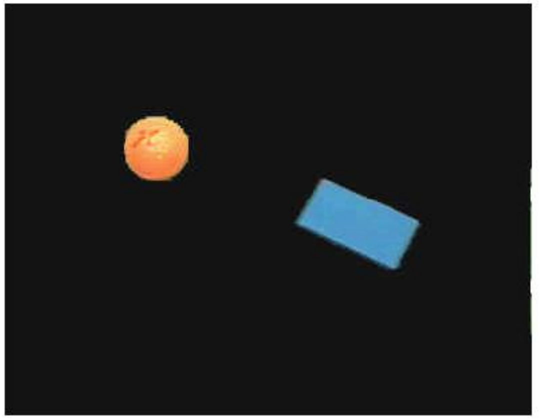


6.2.2 Image Segmentation & Analysis

Colour information

- It can be used to detect edges & regions in order to identify objects ***(as human does)***.
- A colour is defined by 3 attributes: ***hue, saturation*** and ***brightness.***
  - The colour of an object is distinguished primarily by its hue or tint.
  - Different wavelengths of light produce different hues on the same object.
  - Saturation effects the shade of a colour, measuring how a colour is diluted with white.
  - The brightness of a colour is a measure of the amount of light intensity presented.



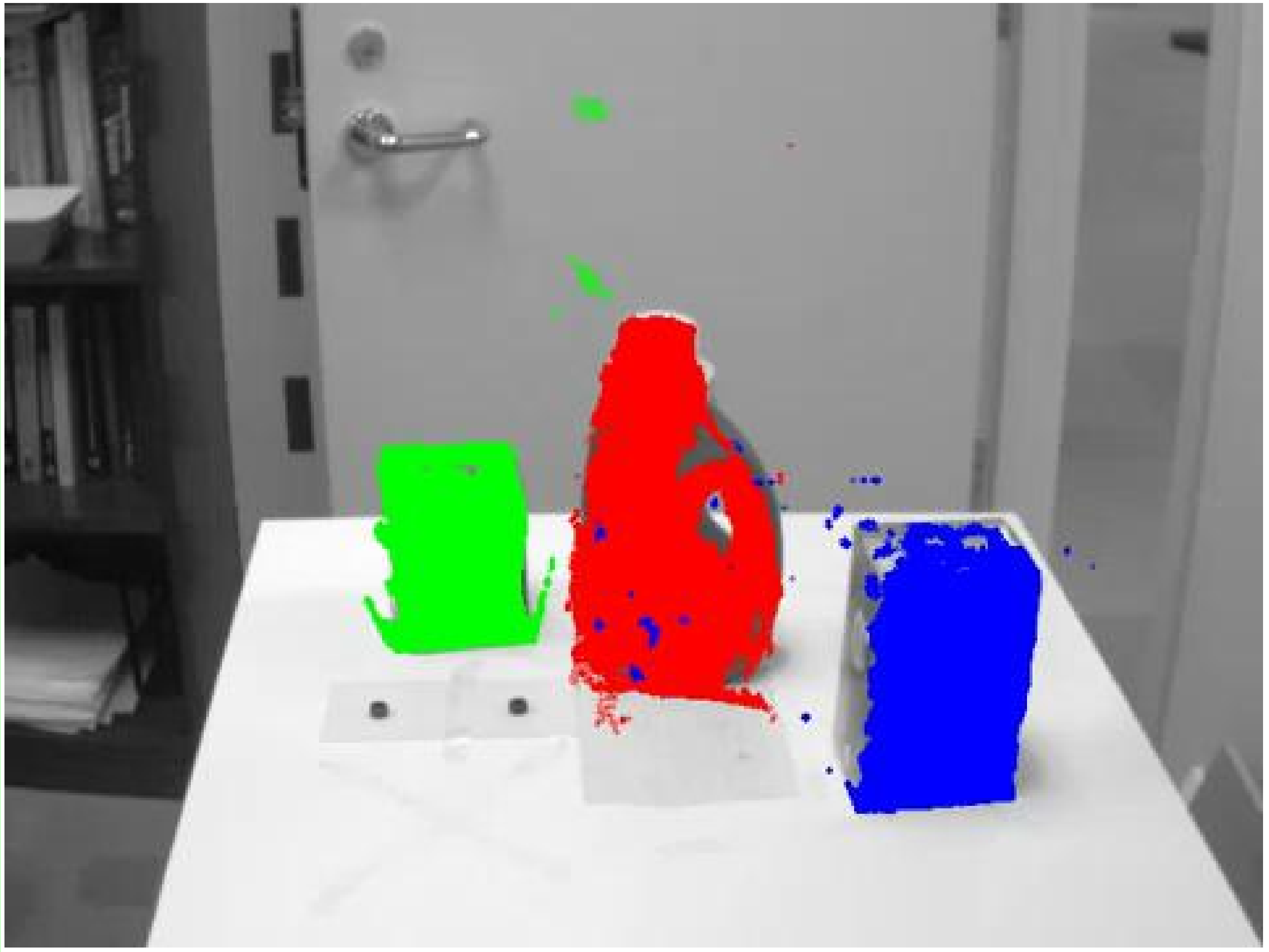


6.2.2 Image Segmentation & Analysis

6.2.2 Image Segmentation & Analysis

Texture

- Like colour, it provides another clue to identify an image (very important in aerial survey).
- Texture analysis can aid in region segmentation since objects normally have different texture.
- Many images can be broken down into some arrangements of small elements (texture primitives).



6.2.3 Image Understanding

Complex Images in the Real World

- The block world is the starting point to learn how a computer to recognise and understand simple images, but unable to deal with most real-world images.
- Most real-world images include straight & curve lines, colour, texture, shadow, etc.

Using Additional Image Information

- *Range from Motion* -- Moving objects or image devices

  -- Motion in vision systems can help to identify an object from its background.

  -- Motion in vision systems is useful for measuring depth or range to objects.

- *Range from Projection*

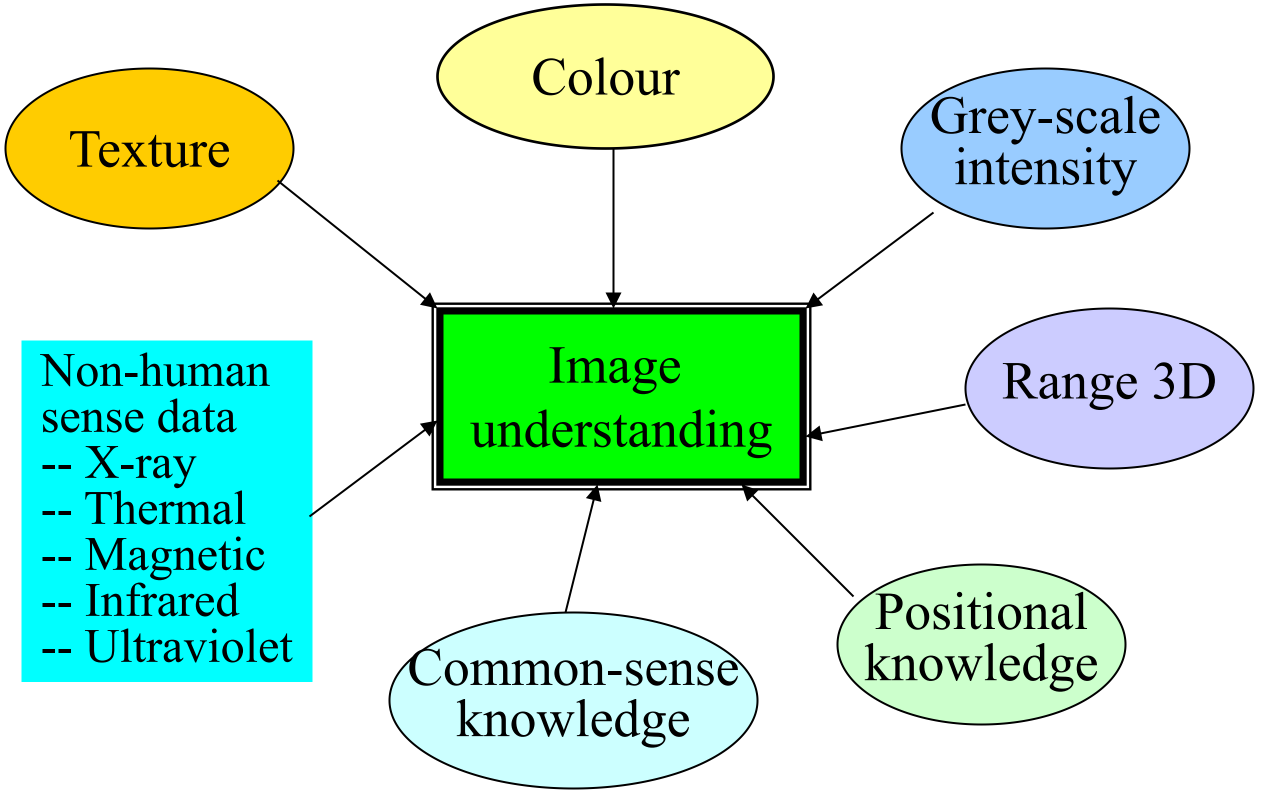
  -- It includes *X-ray image data, infrared image data, ultraviolet image data, etc.*

Using Expectational Knowledge

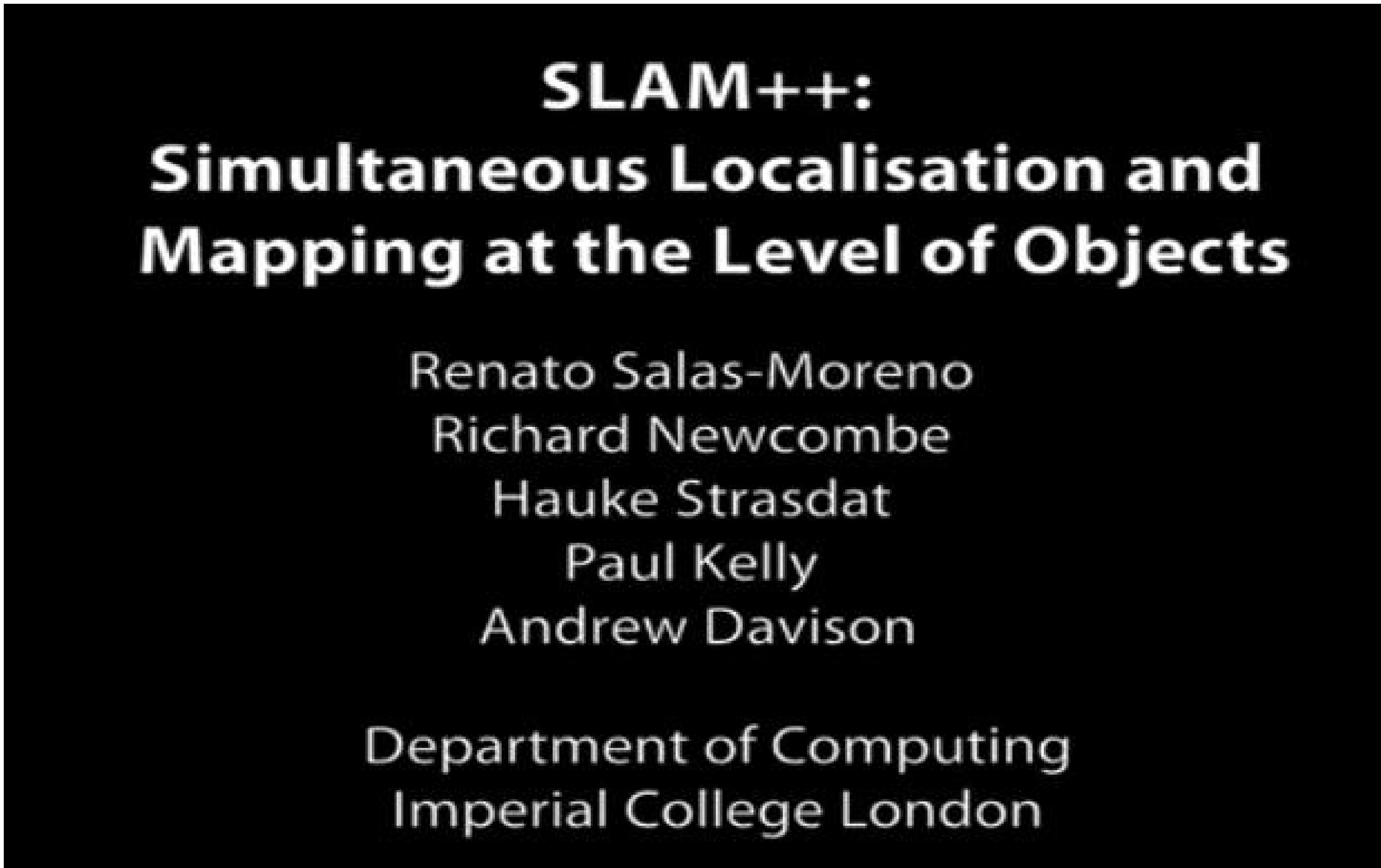
- Expectation knowledge plays an important part in the human image-understanding process.
- Expectation knowledge in vision systems can come in many different forms, see next figure.

6.2.3 Image Understanding

- Positional knowledge, relative object position, is an important expectation knowledge.
- Common-sense knowledge will help to image understanding.
  -- Tables have four legs, bicycles have two wheels, cars have four wheels, etc.



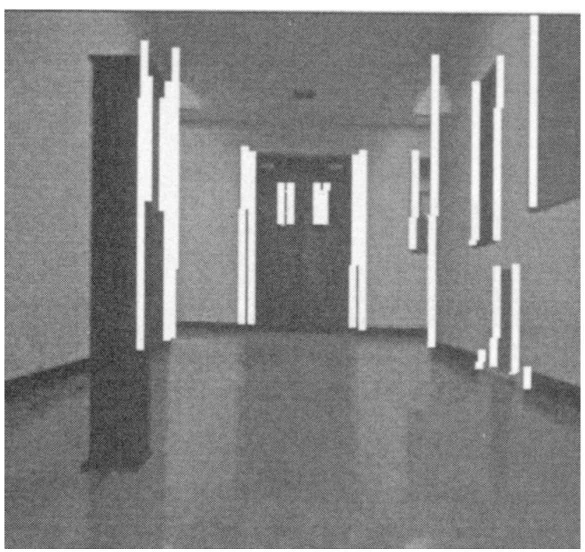
6.2.3 Image Understanding



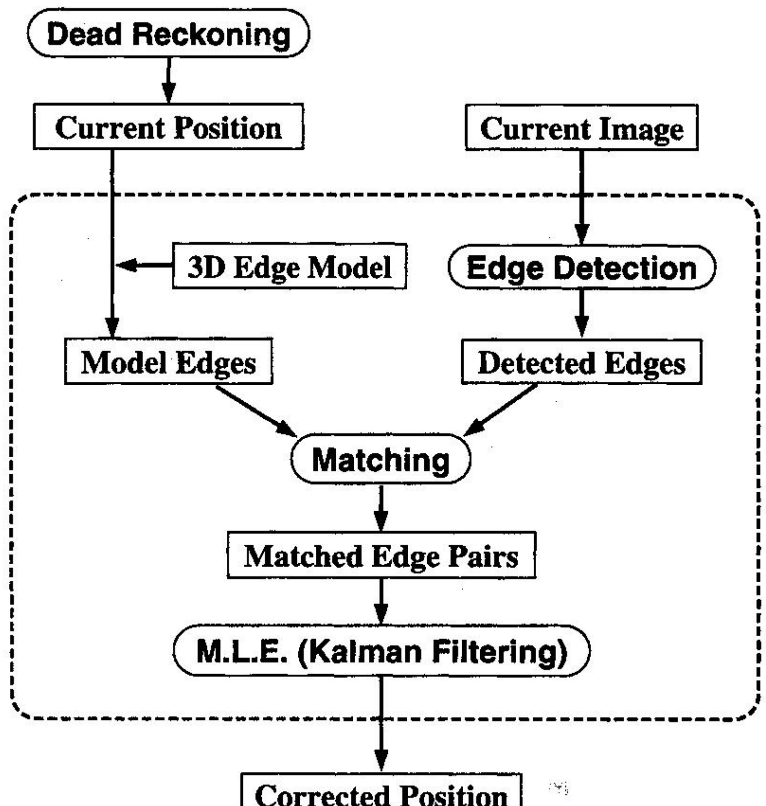
6.2.4 Real-World Applications

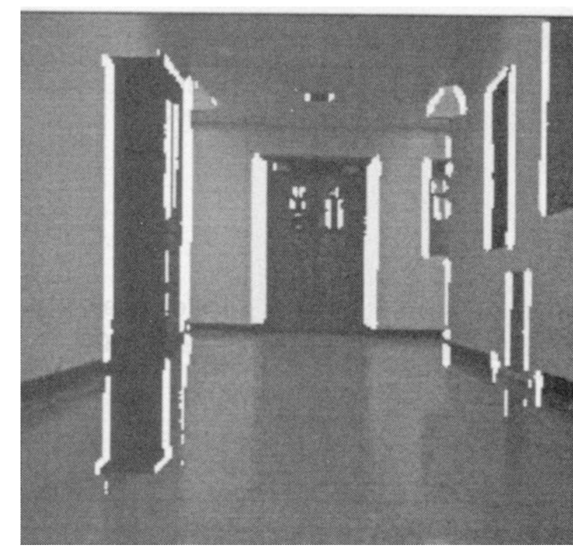
Vision-based Localisation

- The matched edges are landmarks used for Kalman Filter based localisation process.



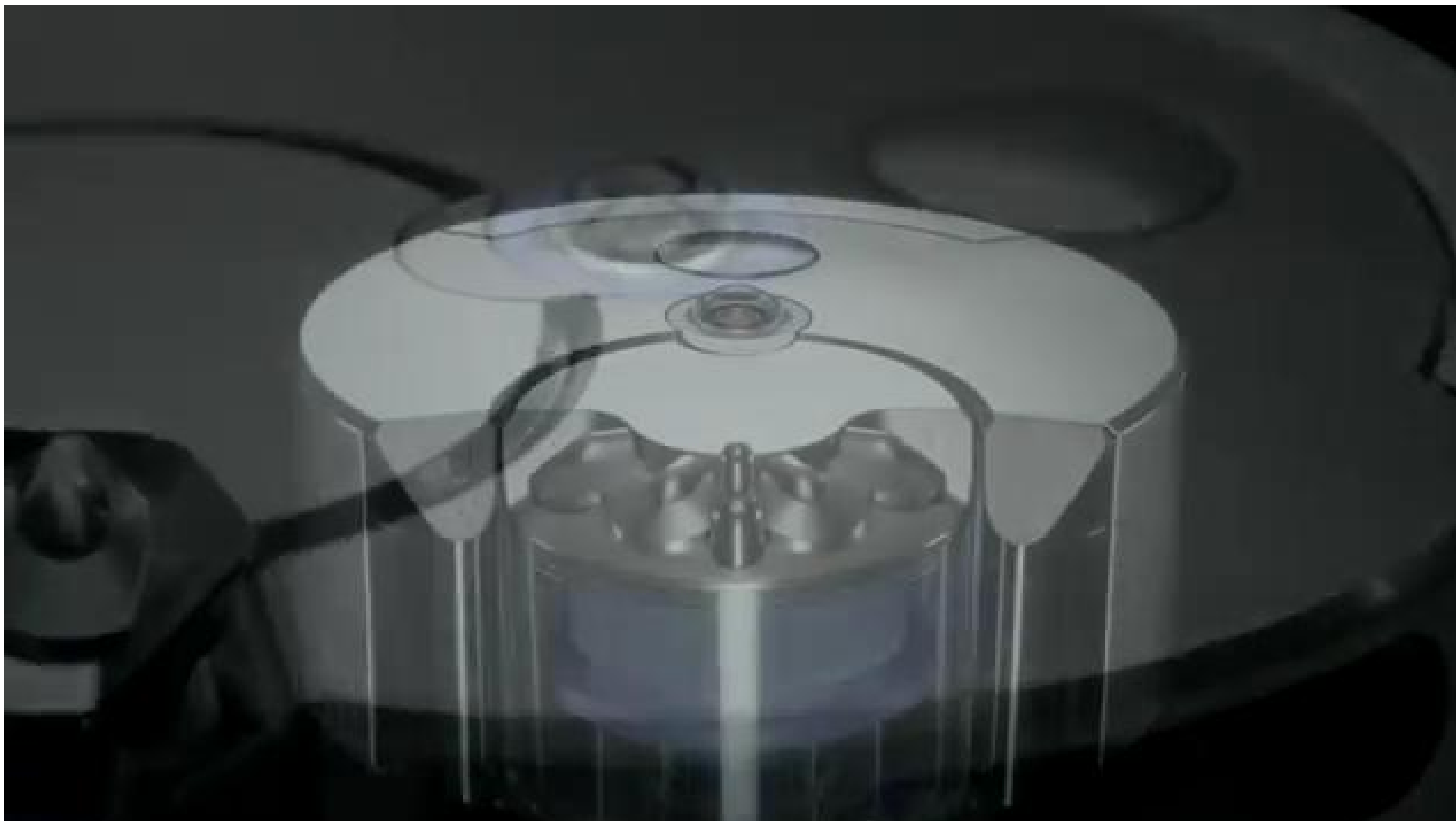
Model edges



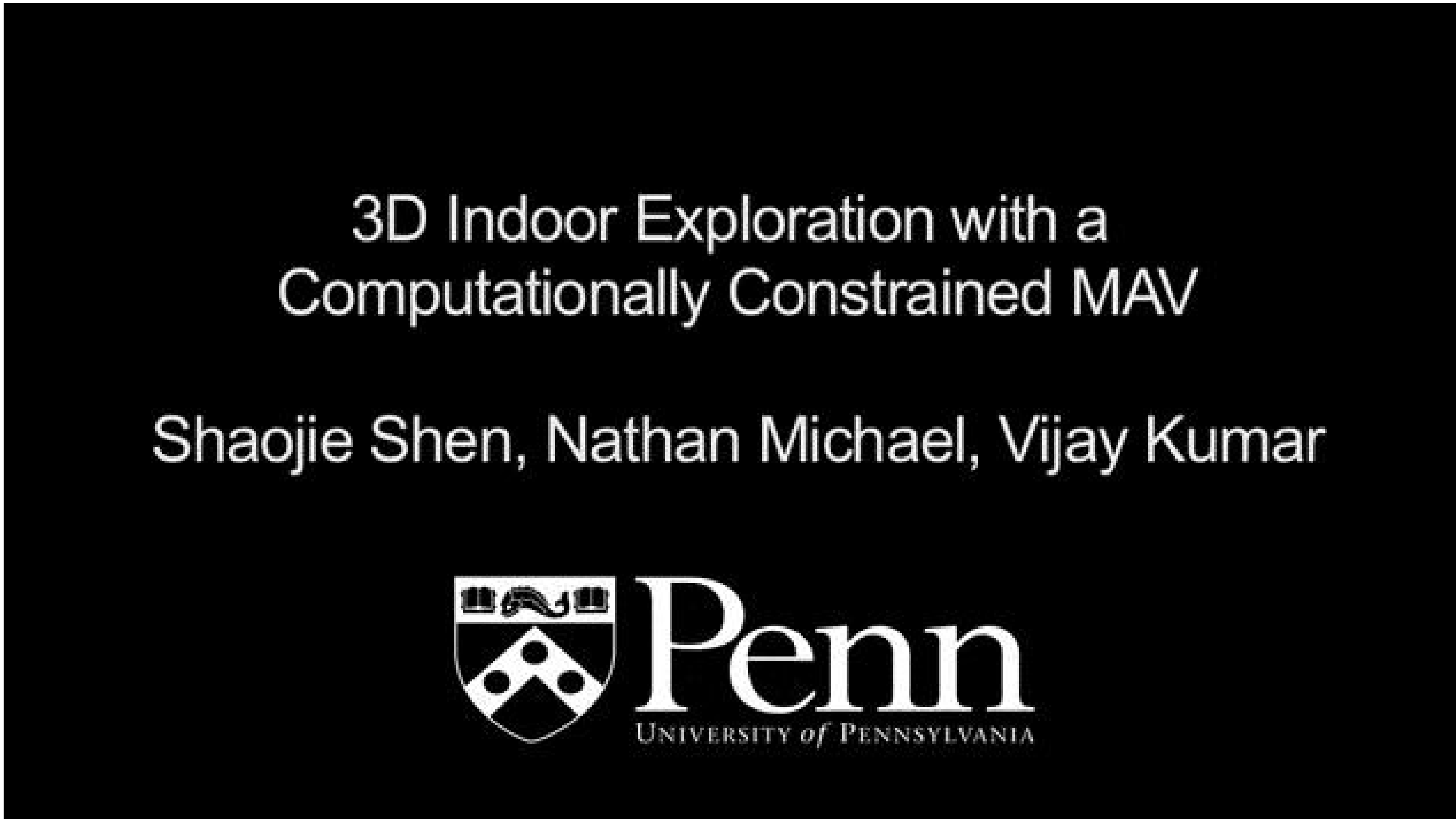


Detected edges

6.2.4 Real-World Applications



6.2.4 Real-World Applications



6.3 Summary

- Kalman filtering is an effective way to fuse multi-sensor data.
- Image transformation involves the conversion of image features, such as brightness, colour, texture, into digital signals that are stored and analysed.
- A computer stores image data in matrices made up of elements called pixels.
- Edge detection is to differentiate grey-scale data to find lines.
- Region segmentation involves the pixel similarities rather than abrupt changes.
- Region splitting and region growing are two general types of region analysis.
- Image understanding involves the interaction of several types of knowledge, including grey scale, colour, texture, motion, depth, etc.