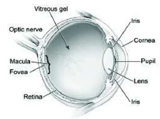
Advanced Vision

School of Informatics
UG4/MSc - 2012/13

Bob Fisher - rbf@inf.ed.ac.uk
Room: IF 1.26. Tel: 651-3441

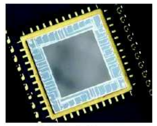
Research Interests:
3D computer vision/video analysis

Problem of Vision - complexity

Human Vision:

- Builtin 3D stereo & video
- Excellent visual reasoning
- Not well understood

CCD Photo Sensor

Computer Vision:

- Hard to get quality 3D
- Noise (environment, sensor)
- Limited, static viewpoints
- Low relative resolution
- Well understood, limited algorithms

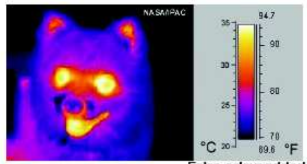
Types of Visual Sensing

1. RGB: 

RGB Channel Separation

2. Greyscale: 

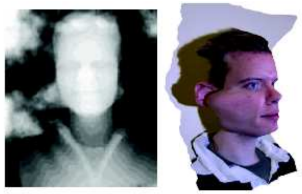
3. Video: 

4. Infrared (heat) 

False colour added

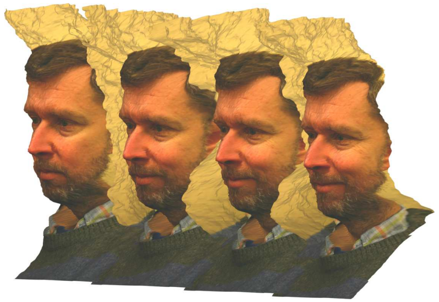
5. Infrared (night vision) 

6. 3D capture (static)



Depth Image and coloured 3D projection

7. 3D capture (video)



AV: Six Visual Systems

1. Orthographically viewed rigid 2D objects
2. Orthographically viewed non-rigid 2D objects
3. Video change detection & tracking
4. Video: human behaviour analysis
5. Recognising 3D objects from range data
6. Recognising 3D objects from stereo data

Review of 2D coordinate geometry

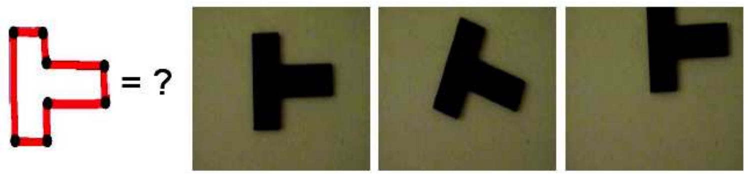
1. Object and Scene Coordinate Systems
2. Coordinate System Transformations
3. Homogeneous Coordinates I
4. Multiple Reference Frame Transformations
5. Simple 2D Rigid Part Modeling

Object and Scene Coordinate Systems

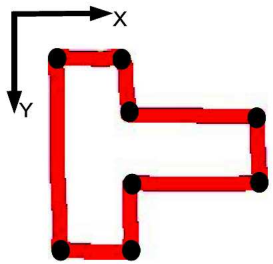
Issues:

+ Want to describe object features independently of the object's position.

+ Want to specify object position and orientation within scene



Why? Generic Model Vs. Specific Position



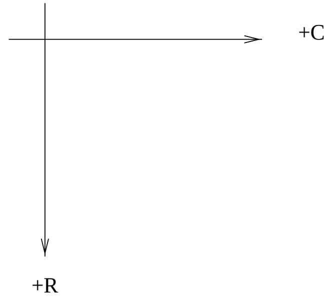
Generic: Object geometric model, aligned

Specific: Scene position in pixels, not aligned

Object and Scene Coordinate Systems II

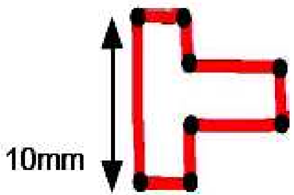
Solution: Use separate object and scene coordinate systems and link by reference frame transformations

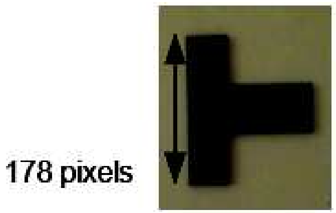
Use image coordinate system $(c, r), c \in [0, C]$, $r \in [0, R]$ for $C \times R$ image (for convenience)



Object and Scene Coordinate Systems III

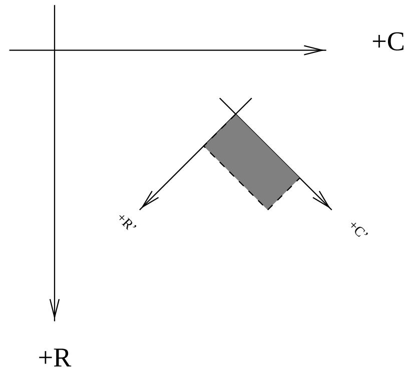
(c, r) in image (eg. in pixels) relates to (x, y) in scene (eg. in mm) using column, row scale factors ρ_c, ρ_r: $(x, y) = (\rho_c c, \rho_r r)$





Object and Scene Coordinate Systems IV

Use separate coordinate systems for object and scene

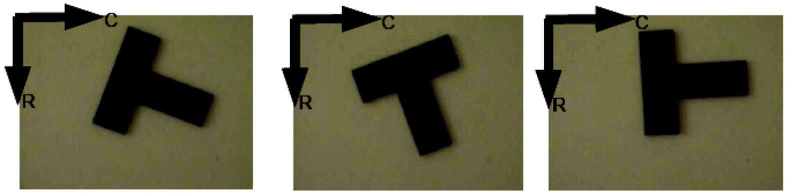


Also - image and camera coordinate systems

Coordinate System Transformations I

Placement of object relative to scene requires a coordinate system transformation

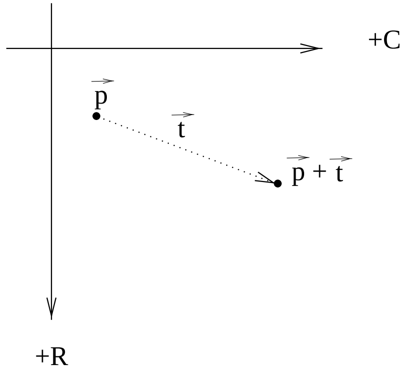
In 2D, need 1 rotation angle θ and $\vec{t} = (t_c, t_r)'$ translation (' is for transposing a row vector to a column vector and *vice versa*)

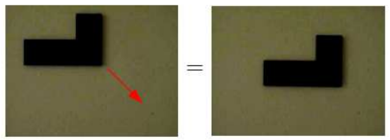


Coordinate System Transformations II

$\vec{p} = (a, b)'$ is a point in the 2D coord system

Translation of point $\vec{p} = (a, b)'$ by $\vec{t} = (t_c, t_r)'$ moves it to $\vec{p} + \vec{t} = (a + t_c, b + t_r)'$





Coordinate System Transformations III

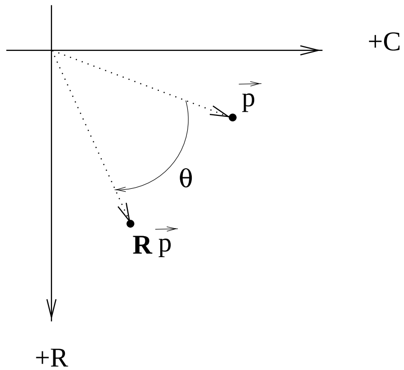
If θ is the rotation angle, let

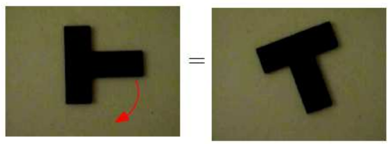
$$R = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

Sometime see $sin(\theta)$ and $-sin(\theta)$ swapped. A matter of convention about direction of rotation.

Coordinate System Transformations IV

Rotation of point $\vec{p} = (a, b)'$ by R moves it to $R\vec{p} = (a \cdot cos(\theta) - b \cdot sin(\theta), a \cdot sin(\theta) + b \cdot cos(\theta))'$

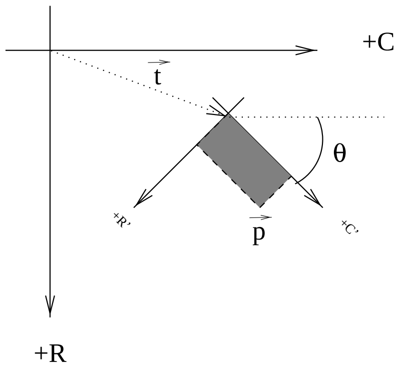




θ positive is clockwise rotation (other definition common)

Complete Transformations

Rotation & Translations: $\mathrm{R}\vec{p} + \vec{t}$



If the object local coordinate system starts at (0,0)', then the rotation & translation specify its position

Midlecture Problem

What is the position resulting from rotating the point $\vec{x} = (10, 20)'$ by $\frac{\pi}{2}$ and translating the result by $\vec{t} = (-10, 30)'$?

Homogeneous Coordinates I

Instead of 2 operations to implement the transformation, often only one operation based on homogeneous coordinates (more advanced form in later lectures)

1) Extend points $\vec{p} = (a, b)'$ to $\vec{P} = (a, b, 1)'$

2) Extend vectors $\vec{d} = (u, v)'$ to $\vec{D} = (u, v, 0)'$

3) Combine rotation and translation into one 3×3 matrix

$$\mathrm{T} = \begin{bmatrix} \cos(\theta) & -sin(\theta) & t_c \\ sin(\theta) & cos(\theta) & t_r \\ 0 & 0 & 1 \end{bmatrix}$$

Full transformation of $\vec{p}$ is now $\mathrm{T}\vec{P}$

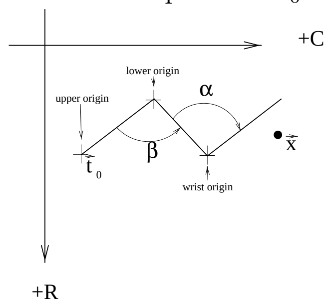
Multiple Transformations

Given 2 joint robot arm whose joint angles are α and β

$\mathrm{T}_w(\alpha)$ is the wrist joint position relative to the lower arm

$\mathrm{T}_l(\beta)$ is the lower arm position relative to the upper arm

The arm is at position T_0



RTX robot arm

Multiple Transformations II

Then, a wrist coordinate point $\vec{x}$ at the tip of the robot is at

$$\vec{y} = \mathrm{T}_0 \mathrm{T}_l(\beta) \mathrm{T}_w(\alpha) \vec{x}$$

Can also easily invert positions:

$$\vec{x} = (\mathrm{T}_w(\alpha))^{-1} (\mathrm{T}_l(\beta))^{-1} (\mathrm{T}_0)^{-1} \vec{y}$$

is the wrist coordinates of scene point $\vec{y}$

Geometric Shape Model

Here: rigid, piecewise linear / circular boundary segments

Options:

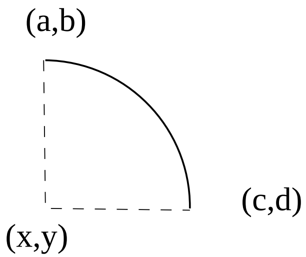
- Region representation: pixel list, quadtree
- Boundary representation
  - Curve
    - Set of boundary segments
    - Pixel list / chain code (incremental pixel list)
  - Vertices

Polycurve / Polyline Modeling

Set of vertices connected by line / curve segments

Line segment: (a,b) -L- (c,d)

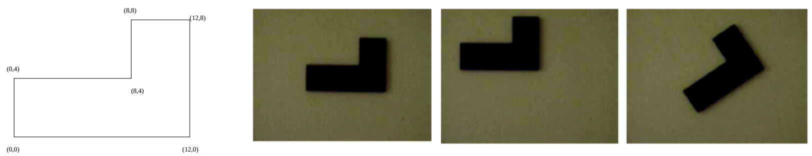
Arc segment: (a,b) -arc(x,y)- (c,d)



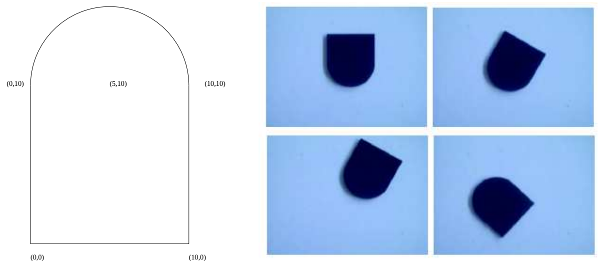
Arbitrary position in local object-centered coordinate system

Example Models

`(0,0)-L-(12,0)-L-(12,8)-L-(8,8)-L-(8,4)-L-(0,4)-L-(0,0)`



`(0,0) -L- (10,0) -L- (10,10) -arc(5,10)- (0,10) -L- (0,0)`



What We Have Learned

1. Review of Coordinate Systems
   Transformations
2. Introduction to Homogeneous Coordinates
3. Simple 2D Rigid Part Modeling