Robots and

People

Vlad Estivill-Castro(2016)

Robots for People

--- A project for intelligent integrated systems

Perception Concepts

Vision

Chapter 4 (textbook)

What is the course about?

- textbook
- **Introduction to Autonomous Mobile Robots**
- second edition

Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza

Robots and people

Principles of Cameras

- Cameras have many similarities with the human eye
  - The light goes through an opening (iris – lens) and hits the image plane (retina)
  - The retina is attached to light-sensitive elements (rods, cones - silicon circuits)
  - Only objects at a particular range are in focus (fovea) – depth of field
  - 512x512 pixels (cameras), 120×10^6 rods and 6×10^6 cones (eye)
  - The brightness is proportional to the amount of light reflected from the objects

Image Brightness

- Brightness depends on
  - reflectance of the surface patch
  - position and distribution of the light sources in the environment
  - amount of light reflected from other objects in the scene onto the surface patch
- Two types of reflection
  - Specular (smooth surfaces)
  - Diffuse (rough surfaces)
- Necessary to account for these properties for correct object reconstruction $\Rightarrow$ complex computation

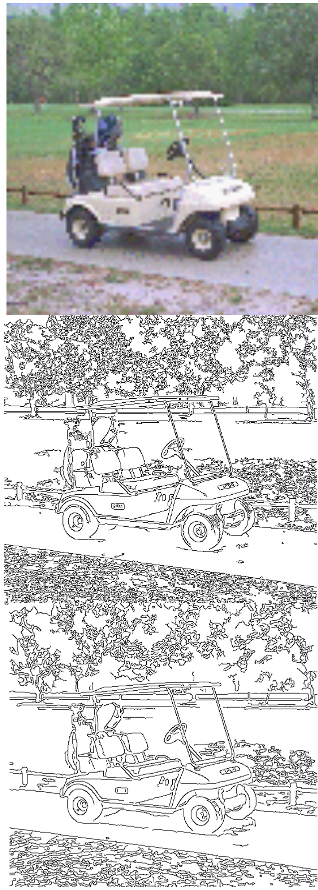
Early Vision

- The retina is attached to numerous rods and cones which, in turn, are attached to nerve cells (neurons)
- The nerves process the information; they perform "early vision", and pass information on throughout the brain to do "higher-level" vision processing
- The typical first step ("early vision") is edge detection, i.e., find all the edges in the image
- Suppose we have a b&w camera with a 512 x 512 pixel image
- Each pixel has an intensity level between white and black
- How do we find an object in the image? Do we know if there is one?

Robots and People

Edge Detection

- Edge = a curve in the image across which there is a change in brightness
- Finding edges
  - Differentiate the image and look for areas where the magnitude of the derivative is large
- Difficulties
  - Not only edges produce changes in brightness:
    - shadows, noise
- Smoothing
  - Filter the image using convolution
  - Use filters of various orientations
- Segmentation: get objects out of the lines



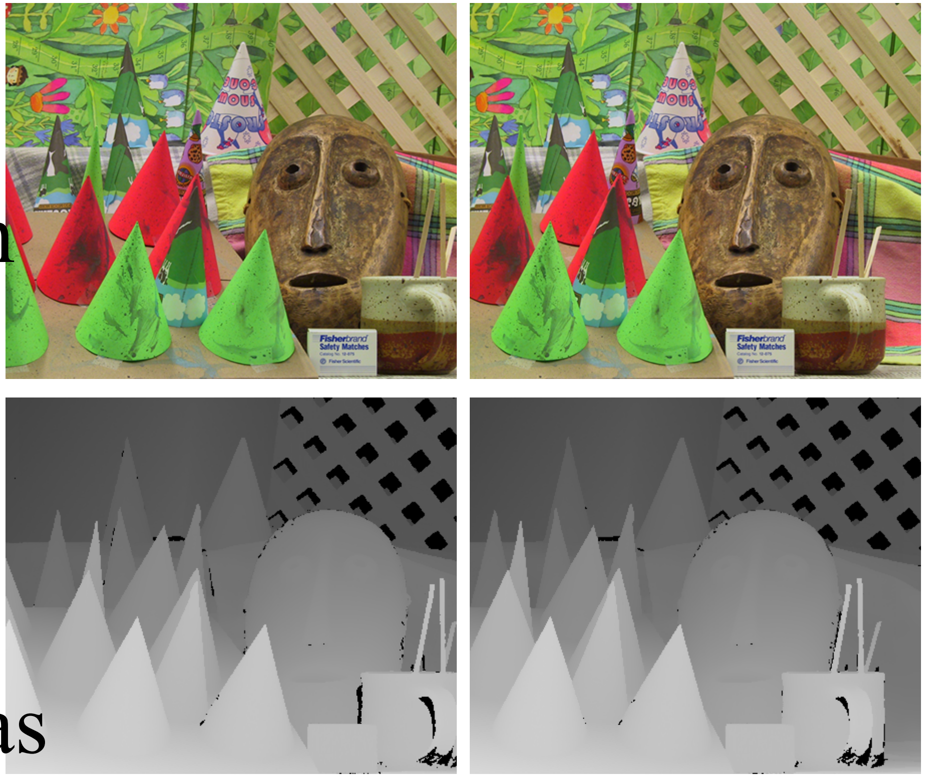
Model-Based Vision

- Compare the current image with images of similar objects (**models**) stored in memory
- Models provide prior information about the objects
- Storing models
  - Line drawings
  - Several views of the same object
  - Repeatable features (two eyes, a nose, a mouth)
- Difficulties
  - Translation, orientation and scale
  - Not known what is the object in the image
  - Occlusion

Vision from Motion

- Take advantage of motion to facilitate vision
- Static system can detect moving objects
  - Subtract two consecutive images from each other ⇒ the movement between frames
- Moving system can detect static objects
  - At consecutive time steps continuous objects move as one
  - Exact movement of the camera should be known
- Robots are typically moving themselves
  - Need to consider the movement of the robot

Stereo Vision



- 3D information can be computed from two images
- Compute relative positions of cameras
- Compute disparity
  - displacement of a point in 3D between the two images
- Disparity is inverse proportional with actual distance in 3D

Biological Vision

- Similar visual strategies are used in nature
- Model-based vision is essential for object/people recognition
- Vestibular occular reflex
  - Eyes stay fixed while the head/body is moving to stabilize the image
- Stereo vision
  - Typical in carnivores
- Human vision is particularly good at recognizing shadows, textures, contours, other shapes

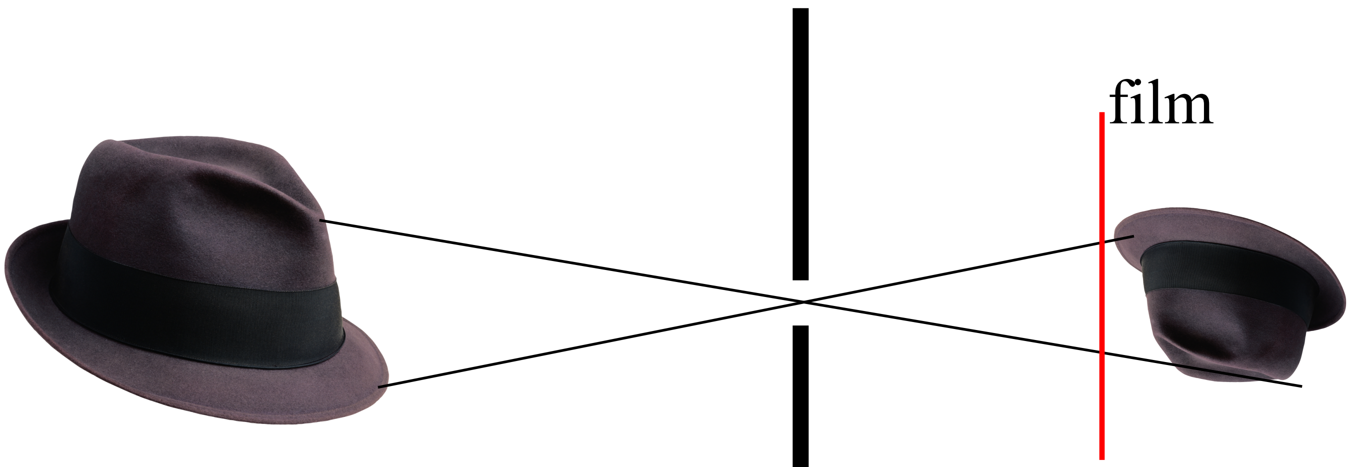
Vision for Robots

- If complete scene reconstruction is not needed we can simplify the problem based on the task requirements
- Use color
- Use a combination of color and movement
- Use small images
- Combine other sensors with vision
- Use knowledge about the environment

This lecture is a summary for

- **Section 4.2 in the book**
- Pinhole Model
- Perspective Projection
- Stereo Vision
- Optical Flow
- Color Tracking

The pinhole camera



The barrier blocks most of the rays

this reduces blurring

the opening is known as **aperture**

captures **pencil of** rays – all rays trough single hole

***the point is called* Center of Projection** or

Optical Center

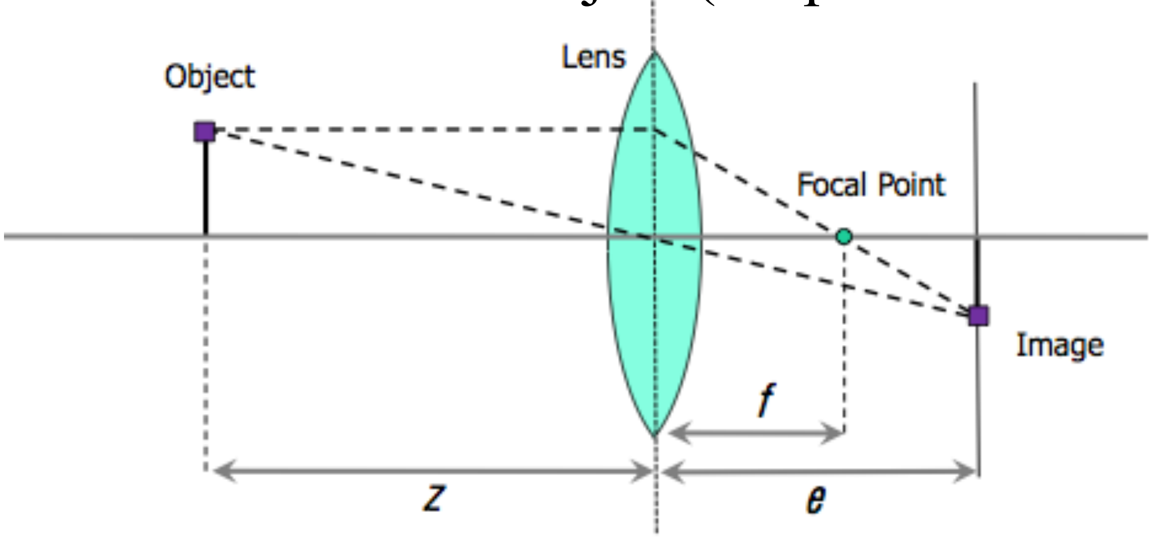
the image is formed in the **image plane**

Problems

- Objects appear blurry
  - reduce the aperture
  - demands increase of exposure
    - increase need for objects to remains still
- Solution
  - Add a lens to focus light into the film
  - The distance between the plane of the lens and the ***focal point*** (where the rays converge) is called
    - FOCAL LENGTH ***f***
  - ***Optical Axis*** (the ray that does not change)
  - The rays parallel to the optical axis converge at focal point

The THIN lens equation

- Derived by similar triangles (see book and notes)
  - $1/f = 1/z + 1/e$
  - Any object point satisfying this equation is in focus
  - This formula can also be used to estimate roughly the distance to the object ("depth from focus")



In focus?

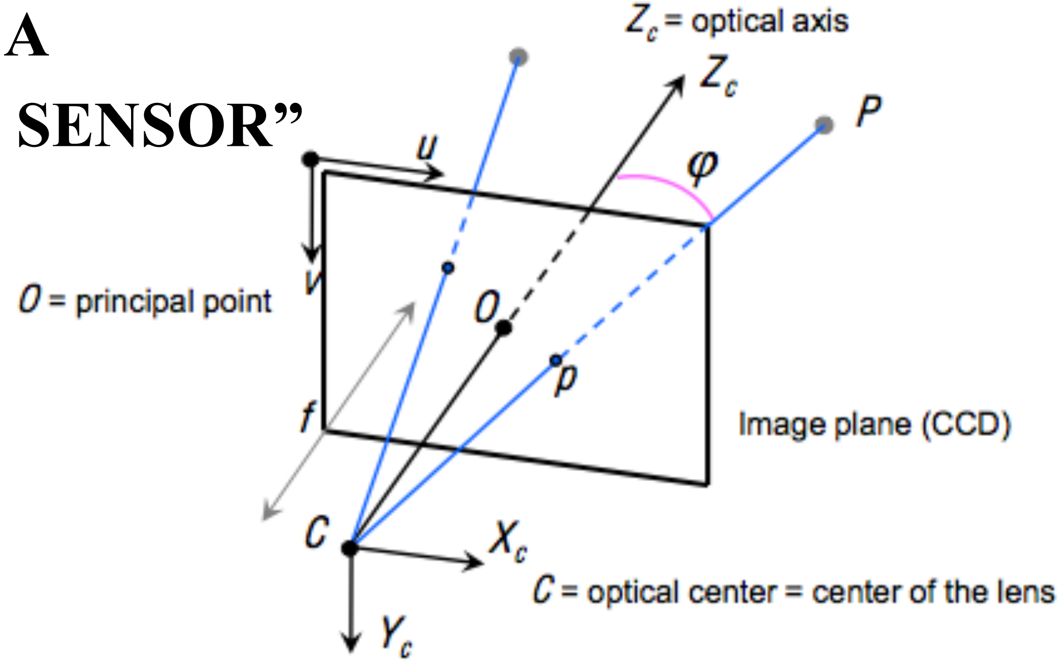
- There is a specific distance at which objects are "in focus"
  - other points project to a "blur circle" in the image
  - the blur circle has radius $R = L\delta/2e$
    - for objects out of focus, larger aperture L gives worse blur (δ is distance of film where this object would be in focus)
    - One can adjust the camera settings so that R remains smaller than the image resolution
    - there is increased sensitivity to blurring when object is close to lens (because of thin lens equation)

From Pinhole to Perspective Camera

- Adjust the image plane so that objects at infinity are in focus
  - $z >> f$ and $z >> L$
    - remember f is focal length, L is aperture, z is distance to object and e is distance to film
    - $f \approx e$
    - Apparent size of an object h' is $f\,h/z$, where h is the size of the object
    - This apparent size of objects on their distance from the observer is known as perspective

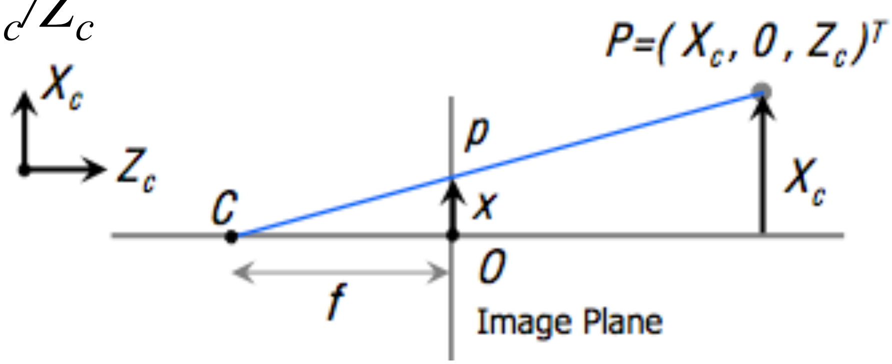
The perspective camera

- For convenience, the image plane is usually represented in front so that the image preserves the same orientation (i.e. not flipped)
- **THE CAMERA DOES NO MEASURE DISTANCES BUT ANGLES**
  - **THUS IT IS A**
  - **"BEARING SENSOR"**



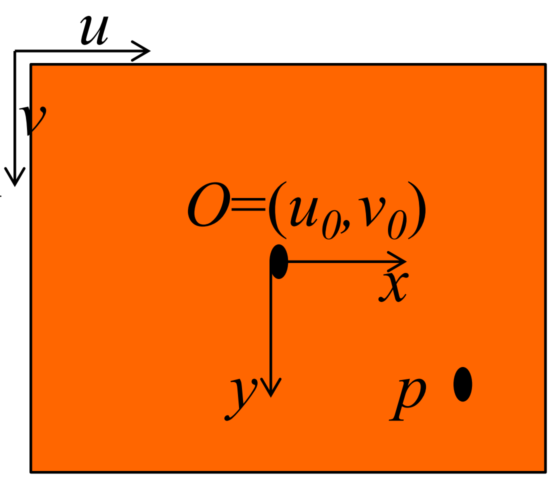
Perspective projection

- From the camera frame to the local image plane
  - the 3D scene point $P=(X_c,Y_c,Z_c)^T$ projects to the point $p=(x,y)^{\mathrm{T}}$ onto the (local) image plane
  - Analyzing similar triangles
    - $x/f=X_c/Z_c \Rightarrow x = fX_c/Z_c$
  - So, in the 3D case, we can also obtain
    - $y/f=Y_c/Z_c \Rightarrow y=fY_c/Z_c$



Perspective projection (2D General)

- From the camera frame to pixel coordinates
  - convert the coordinates of *p*, from the local image plane coordinates (*x*,*y*) to the pixel coordinates (*u*,*v*) we have to take into account
    - pixel coordinates of the camera optical center $O=(u_0,v_0)$
    - scale factors for the pixel-size in both dimensions k_u, k_v
    - so $u= u_0+ k_u x \Rightarrow u=u_0+k_u fX_c/Z_c$
    - $v =v_0 +k_v y \Rightarrow v = v_0+ k_v fY_c/Z_c$



CCD and CMOS

- In machine vision two camps firmly established:
  - CCD
    - charge-coupled device (CCD)
  - CMOS.
    - An active-pixel sensor (APS) is an image sensor consisting of an integrated circuit containing an array of pixel sensors, each pixel containing a photodetector and an active amplifier

CCD and CMOS

- CCD
  - Image quality
  - Light sensitivity
  - Real noise performance
  - Perfect global shutter
- CMOS.
  - High speed
  - On-Chip system integration
  - Low cost of manufacturing
  - High level of process innovation

Perspective projection, into image plane (CCD)

- $u = u_0 + k_u x \Rightarrow u = u_0 + k_u f X_c / Z_c$
- $v = v_0 + k_v y \Rightarrow v = v_0 + k_v f Y_c / Z_c$

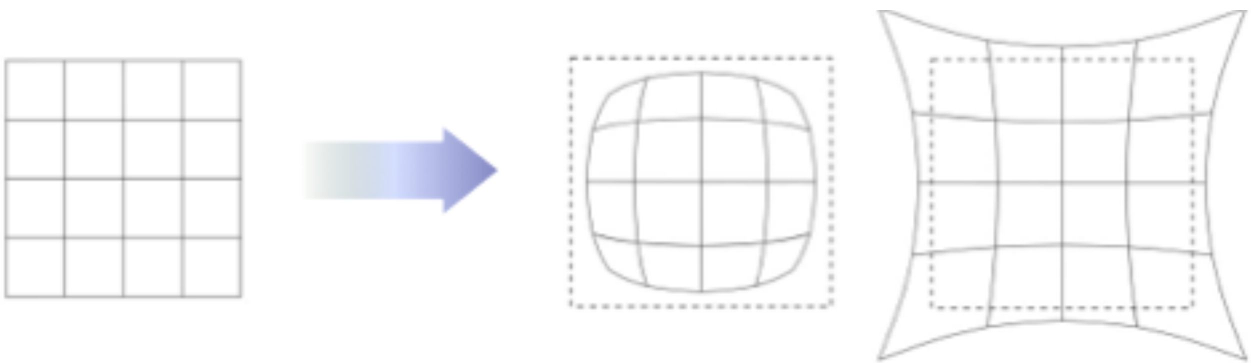
- Expressed in matrix form an as homogeneous coordinates

$$\begin{vmatrix} a_u & 0 & u_0 \\ 0 & a_v & v_0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} X_c \\ Y_c \\ Z_c \end{vmatrix} = K \begin{vmatrix} X_c \\ Y_c \\ Z_c \end{vmatrix}$$

Robots and People

In fact there is radial distortion

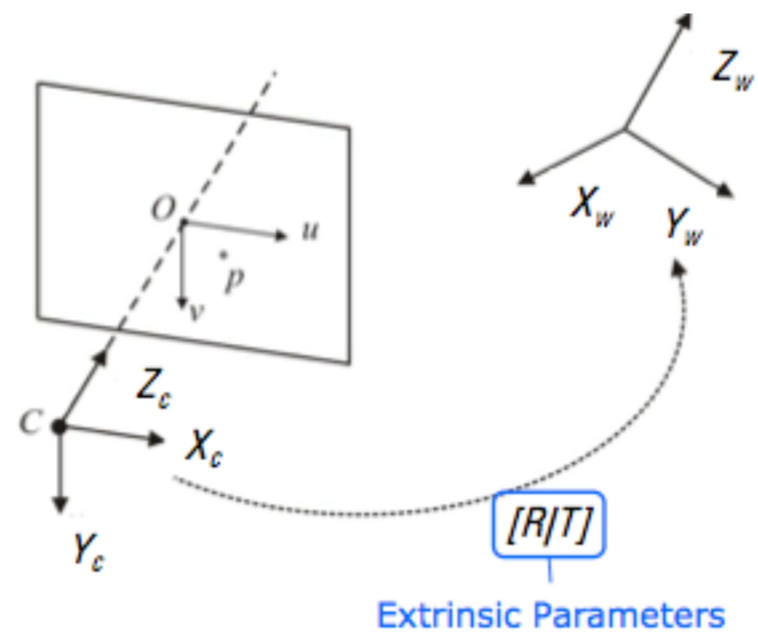
- Straight lines in the world are not straight lines in the world
  - needs a distortion model



Camera calibration

- measure all unknown parameters to form a camera model (intrinsic & extrinsic)
  - use the model to 'understand' the projection from the world to image plane
  - That is to compute the Projection Matrix for

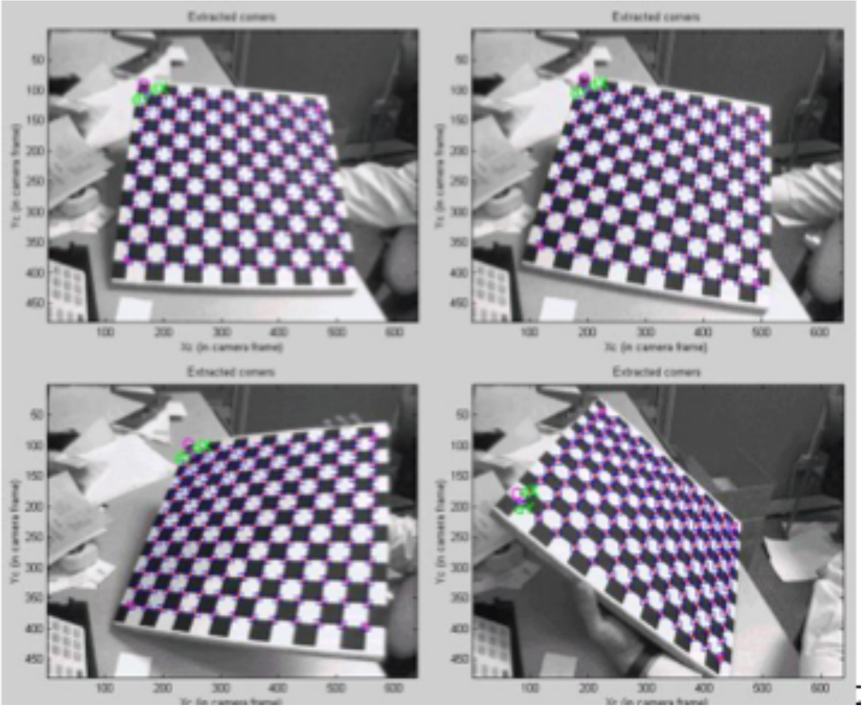
$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} R \,|\, T \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$



Camera Calibration

- By knowing the pixel coordinates of the image points *p* and the *3D* coordinates of the corresponding scene points *P*, it is possible to compute the unknown (matrix) parameters *K, R* and *T* and by solving the perspective projection equation

The 11 unknowns are identified by at least 6 points using linear least squares.
Then, the components, by using *QR* decompositions:
K as an upper triangular matrix
R as a rotation matrix
T as a translation

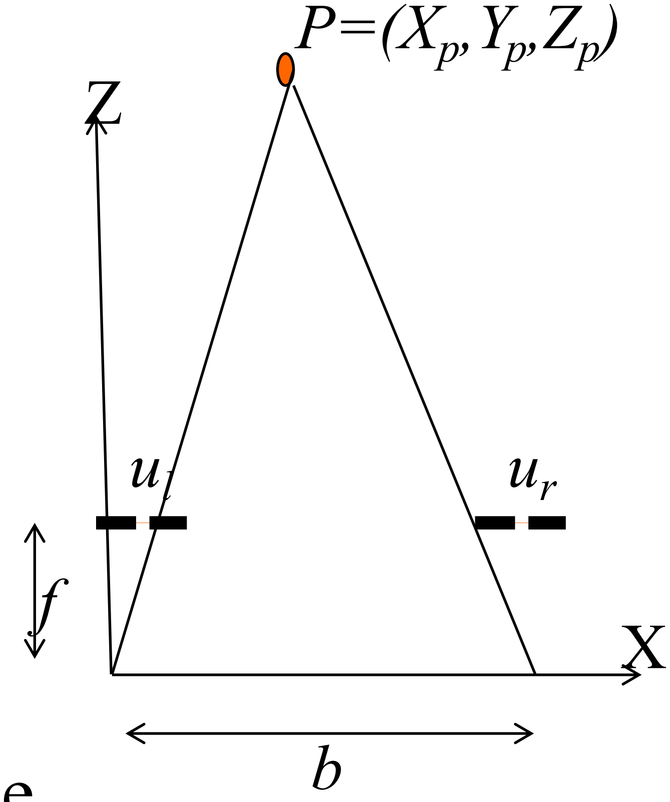


How do measure distance with cameras?

- Structure from stereo
  - use two cameras with known relative position and orientation
- Structure from motion
  - use a single moving camera: both 3D structure and camera motion can be estimated up to a scale

Stereo vision: The simplified case

- Assume both cameras re identical and aligned on a horizontal axis
  - from similar triangles
  - $f/Z_p = u_l/X_p$
  - $f/Z_p = \mathrm{u_r}/(b - X_p)$
  - $\Rightarrow Z_p = b\,f\,/(u_l - u_r)$
- Disparity $(u_l - u_r)$
- Baseline *b* distance between the optical centers of the two cameras



Stereo Vision facts

- $Z_p = b\,f\,/(u_l - u_r)$
- Depth is inversely proportional to disparity
  - foreground objects have bigger disparity than background objects
- Disparity is proportional to stereo-baseline *b*
  - the smaller the baseline *b* the more uncertain our estimate of depth
  - however, as *b* is increased, some objects may appear in one camera, but not in the other
- The projections of a single 3D point onto the left and tight stereo images are called correspondence pairs

Stereo vision, the general case

- Two identical cameras do not exist
- Aligning both cameras on a horizontal axis is very hard
  - even the most expensive stereo cameras
- In order to be able to use a stereo camera, we need the
  - **relative pose** between the cameras (rotation, translation), and
  - the **focal length**, **image center**, **radial distortion** of each
  - Use a calibration method, as mentioned before

Setting up the system of equations

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Left Camera

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + T$$

Right Camera

Robots and People

Correspondence Problem

- Which patch in the Left image, corresponds to the projection of the same 3D scene point on the Right image?
- Correspondence search could be done by testing the query-patches at all pixel positions in the other image. Typical **similarity measures** are the Cross-Correlation and Sum of Squared Differences
- **Exhaustive** image search can be computationally very expensive! Is there a way to make the correspondence search 1-dimensional?

Robots and people

Epipolar Geometry

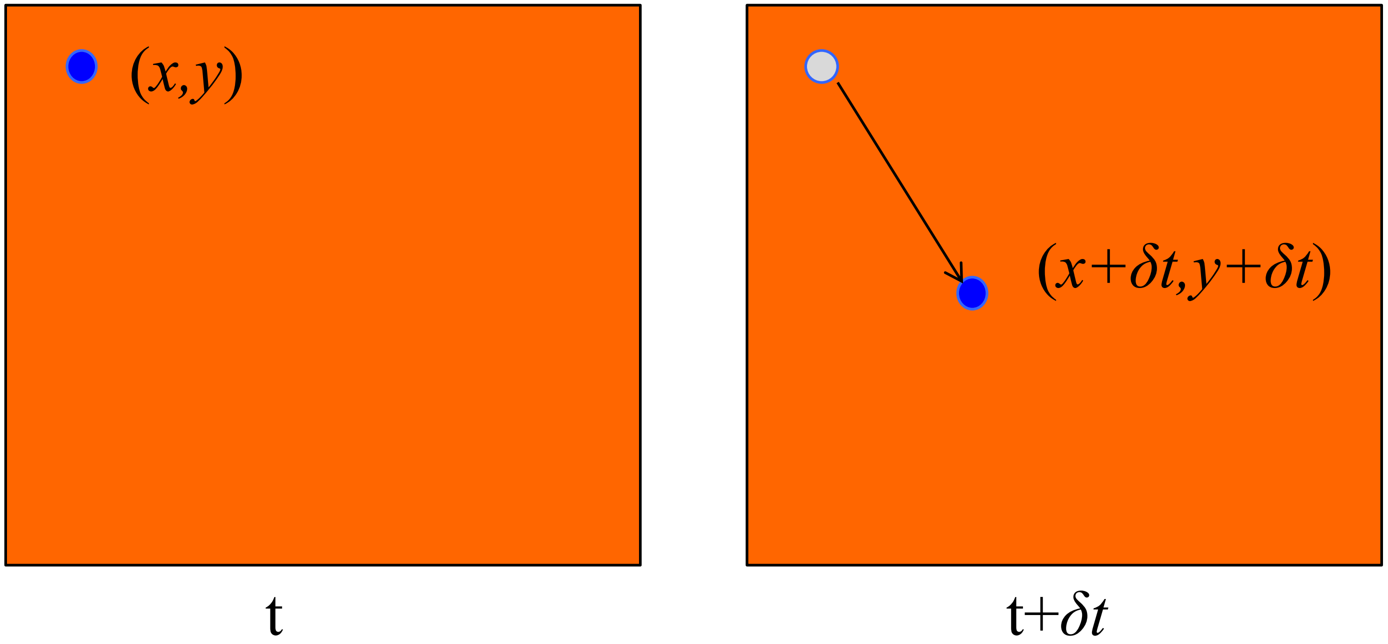
- Triangulating 2 rays from image correspondences
- Impose the **epipolar constraint** to aid matching: search for a correspondence along the epipolar line
- The epipolar plane is defined by a 3D point *P* and the optical centers.
- Thanks to the epipolar constraint, conjugate points can be searched along epipolar lines: this reduces the computational cost to 1 dimension!

Structure from motion

- Given image point correspondences,
  - $x_i \leftrightarrow x_i'$,
  - determine R and T
  - rotate and translate camera until stars of rays intersect
  - At least 5 point correspondences are needed (for calibrated cameras)

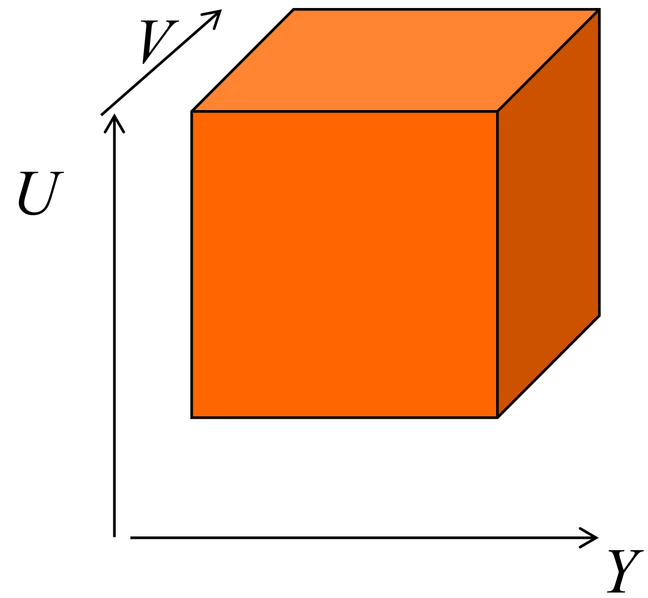
Optical Flow

- It computes the motion vectors of all pixels in the image (or a subset of them to be faster)



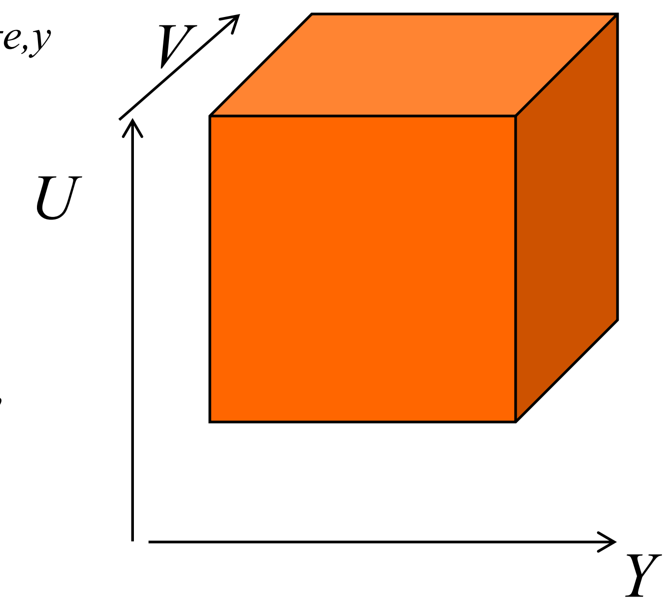
Color tracking

- Motion estimation of ball (and other robots) playing soccer by tracking blob of color



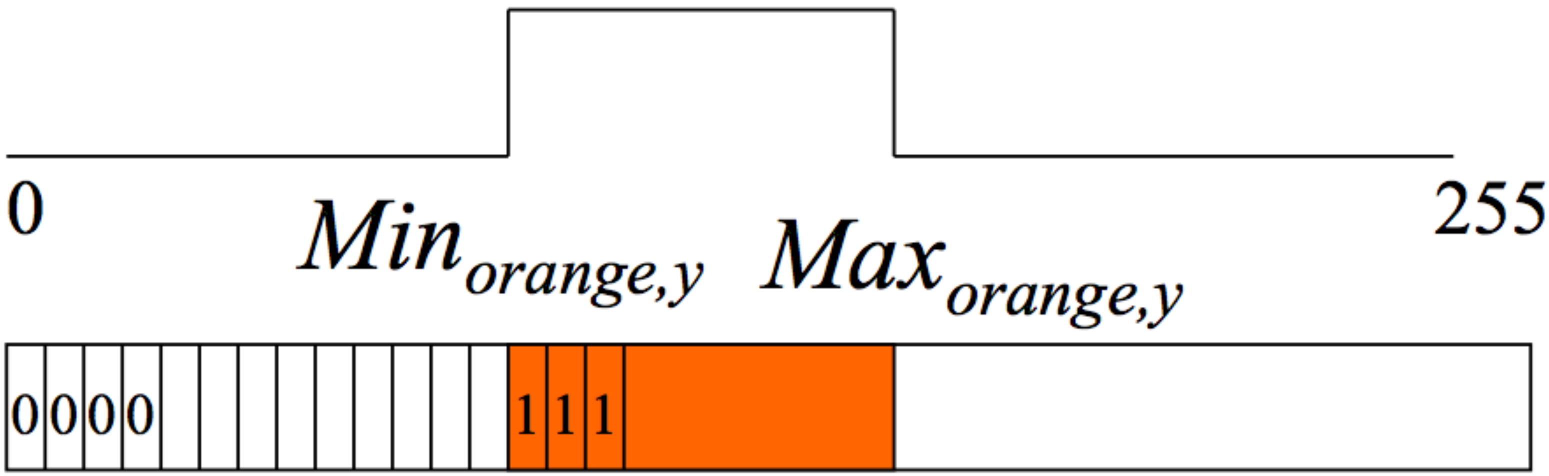
Color tracking

- pixel (y,u,v), 6 parameters
- $Min_{range,y}\ Max_{range,y}\ Min_{orange,u}$ $Max_{orange,u}\ Min_{orange,v}\ Max_{orange,v}$
- Simplification for *Color_Orange*
  - *If* $Min_{range,y} \leq y$ *AND* $y \leq Max_{orange,y}$
    - *AND*
  - $Min_{orange,u} \leq u$ *AND* $u \leq Max_{orange,u}$
    - *AND*
  - $Min_{orange,v} \leq v$ *AND* $v \leq Max_{orange,v}$
  - *Then ORANGE*



The bit-Map for a color projection

- *If* $Min_{range,y} \leq y$ *AND* $y \leq Max_{orange,y}$

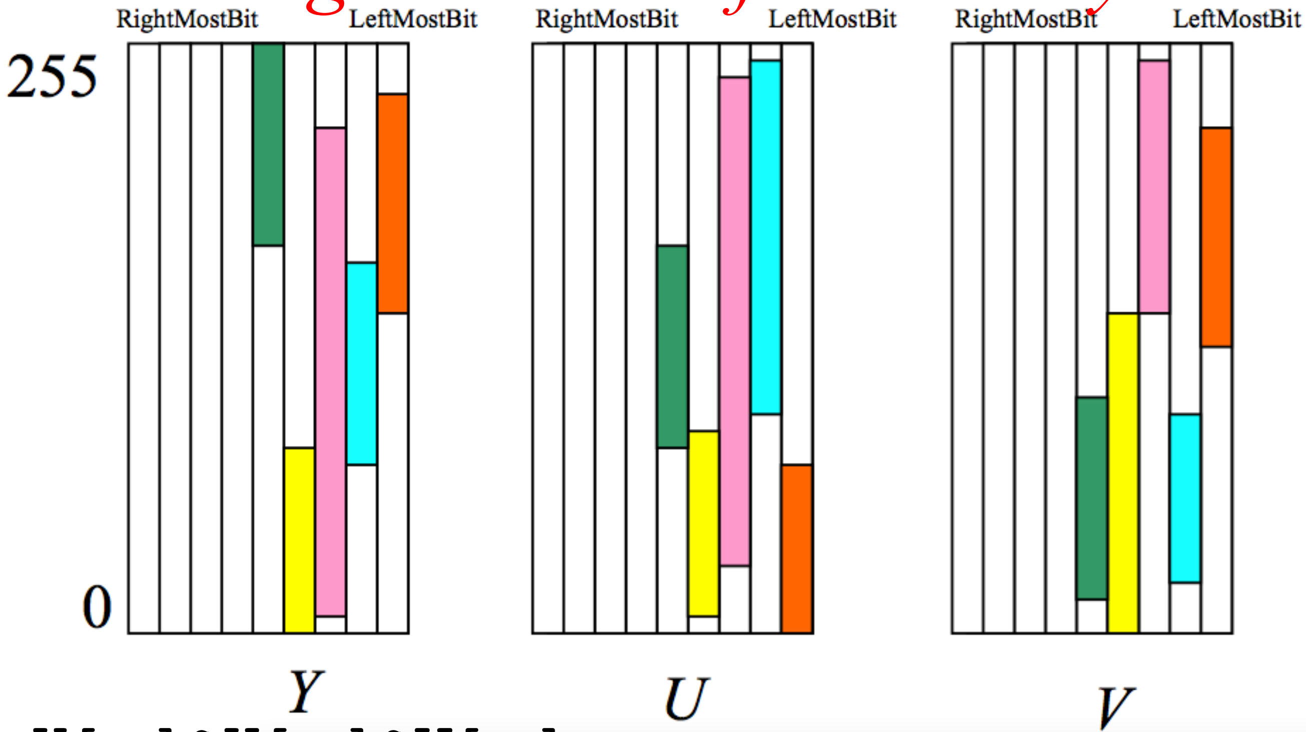


- A classifier is a table look up
  - C/C++bit-wise AND operation

```
Color_Orange=Y[y]&U[u]&V[v]
```

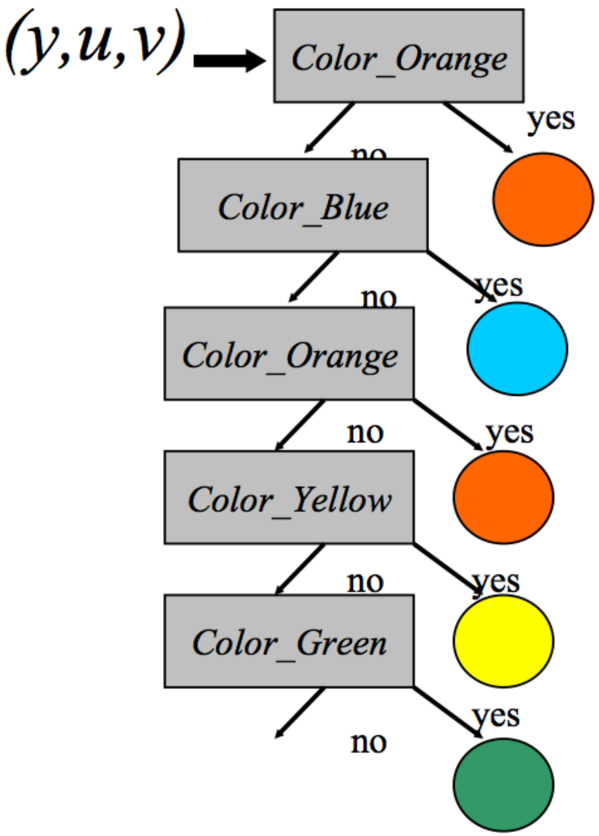
A Decision List

is a Scan through the Bits of a Memory Word




```
color=Y[y]&U[u]&V[v];
color_id=0 ;
while (color & 1 == 0)
   { color>>=1; color_id++;}
```

List Can Repeat Simple classifiers

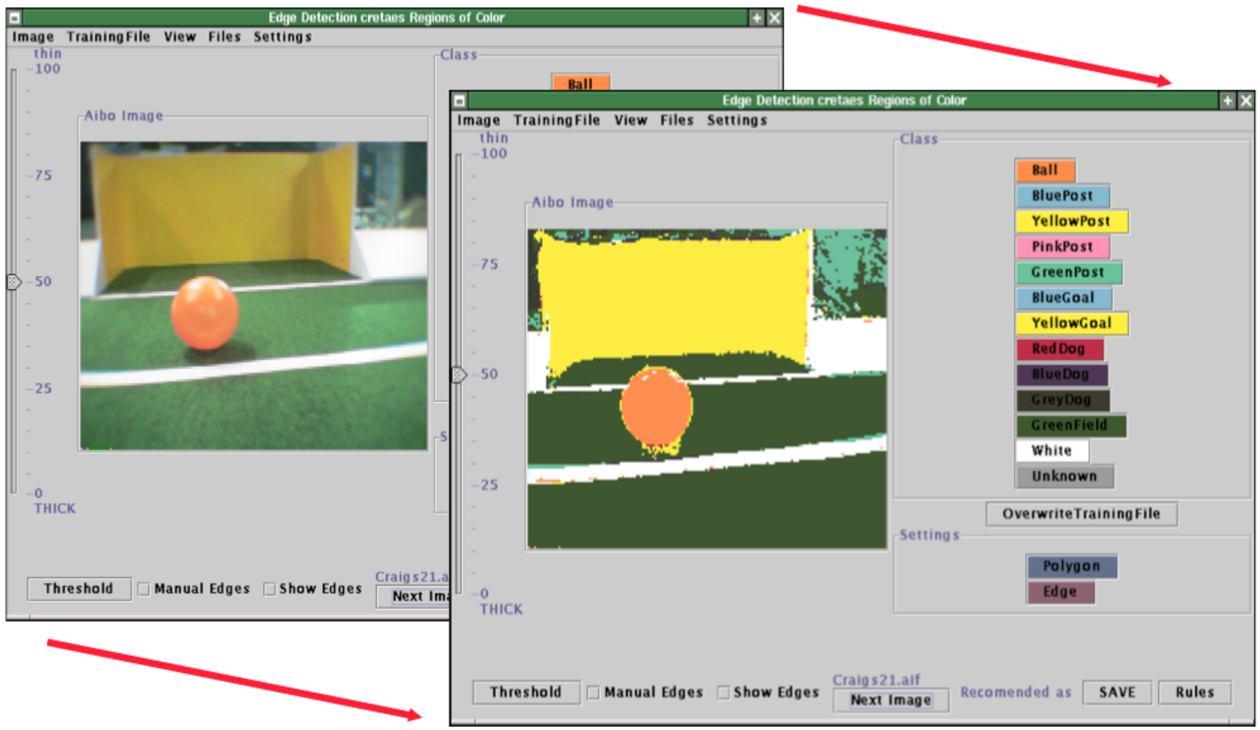


Robocup Competition

- Dynamic environment
- Very rapid analysis of images
- Legged-league
  - SONY Aibo robots
    - Image is 174x144 pixels
    - 25 frames a second
    - YUV format (3 bytes per pixel)
  - Nao
    - Image 1200 x 960
    - 25 frames a second
    - YUV 422 format

Image segmentation

- Mapping is a classifier
- *Color_class:YxUxV→Color*
- *Color_Class(y,u,v)=Orange*
- $|Y| \times |U| \times |V| = 256^3$



Comparison of Decision List

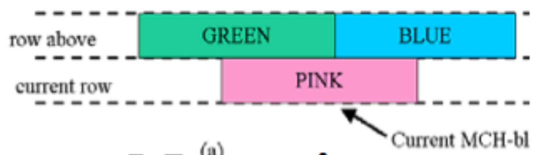
- ANN using snns were 20K times slower
- *k*-NN with Quadtrees and DT (Weka) were 2K times slower

	DL	Look-up Table	Ratio
Maximum	2.87ms	2.46ms	1.16
Average	2.33ms	1.41ms	1.65
Minimum	2.08ms	1.27ms	1.63

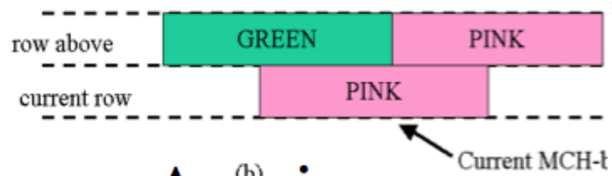
Accuracy with Decision Lists is Marginally Better

Algorithm	10-fold accuracy	Lowest accuracy per class	Largest 2-class confusion	size	Learning time	Test set accurac y
PART	99.0%	96% (yellow goal)	10 blue dog Vs gray dog	26 Rules	1.15s	99.3%
k-NN	99.3%	97% (blue dog)	8 red dog Vs gray dog	*k*=3 6,226 Instanc es	0s	99.7%
DT	98.8%	95% (yellow goal)	10 red dog Vs gray dog	34 leaves 67 nodes	1.27s	99.6%
Look-up Table	71.6%	64% (yellow goal)	45 yellow goal vs orange ball	11 rules	manual	68.2%

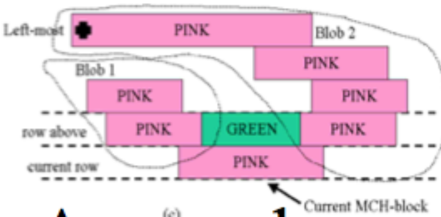
Blob Forming – Bringing contiguous blocks together



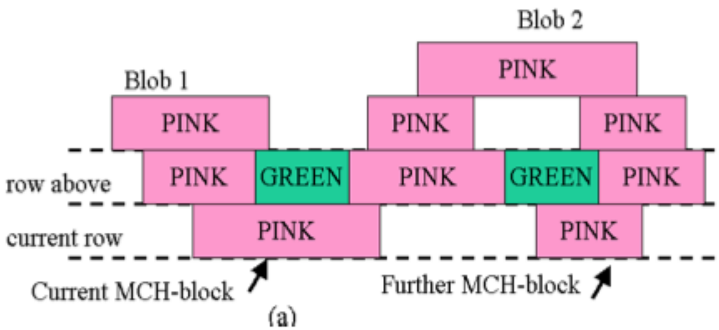
No union

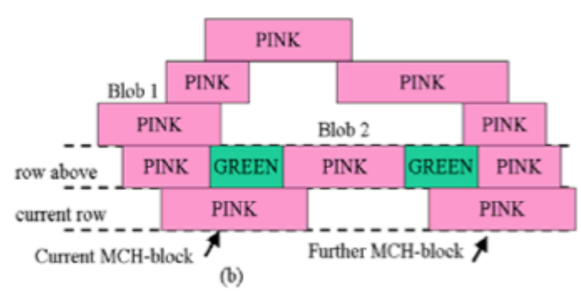


A union



A complex union





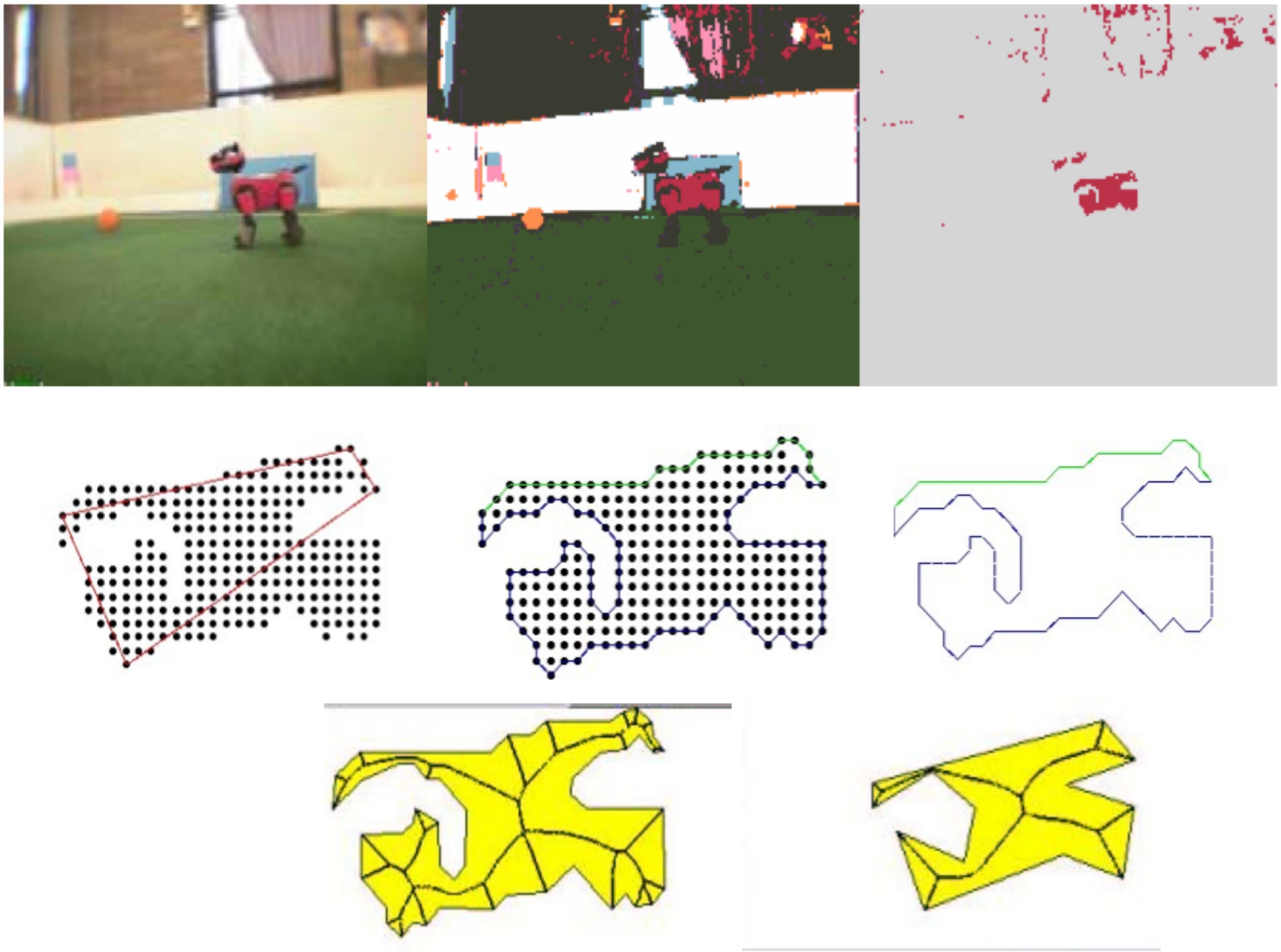
Blob information

- Minimum bounding box
  - SE & NW (4 integers)

Shape finding

- We need to find the boundary fast
  - Trace it
    - Time proportional to the number *b* of pixels in boundary
  - Simplify the boundary
    - Douglas-Peucker Algorithm
    - Time proportional to *b log b*
  - Find the skeleton
    - Use Voronoi diagram of boundary segments
    - Time proportional to high level description of boundary (less than *b*)
  - Identify the skeleton (graph isomorphism)

Illustration



Summary – stereo vision

- Stereo camera calibration compute camera relative pose
- Epipolar rectification => align images & epipolar lines
- Search for correspondences
- Output: compute stereo triangulation or disparity map
- Consider baseline and image resolution to compute accuracy

THANK YOU

Robots and people