Robotica ed Animazione Digitale
La visione

Prof. Alberto Borghese

Dipartimento di Scienze dell'Informazione

borghese@dsi.unimi.it

Università degli Studi di Milano

Slide in parte tratte da: http://www.andrew.cmu.edu/course/15-491

Sommario

- La visione
- Le immagini digitali
- Il modello geometrico di una camera
- Segmentazione real-time.

Le funzioni della visione

Computer Vision

Obbiettivo: determinazione delle proprietà geometriche, fisiche e dinamiche del mondo che ci circonda mediante elaborazione di immagini o sequenze di immagini.

- *Low level vision (o early vision):* estrazione dalle immagini o sequenze di immagini delle informazioni necessarie al livello superiore (features = caratteristiche locali) utili alle elaborazioni successive.

- *High level vision:* riconoscimento, associazione di un significato semantico all’atto del vedere, ricostruzione del movimento degli oggetti.

Nel nostro caso, comportamento reattivo, principalmente low-level vision.

Processing visivo.

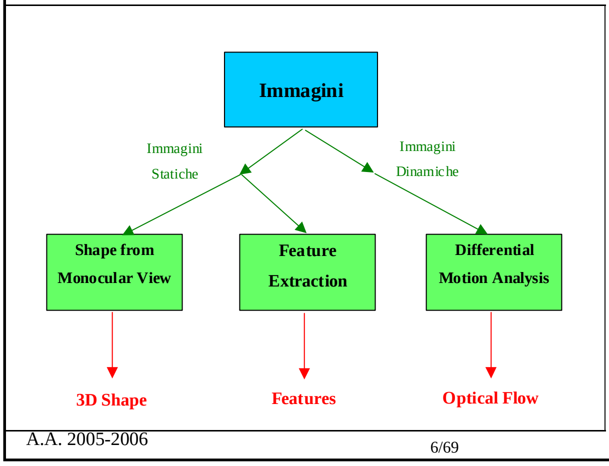
- *"Low-level vision (early vision)" – Pre-elaborazione delle immagini (estrazione di feature).*
- Calcolo del Movimento degli oggetti sull'immagine (optical flow).
- Estrazione del colore.
- Estrazione della profondita'.
- Riconoscimento di tessiture.
- Contorni (edge)
- *"Low-level vision (intermediate representations)".*
- Calcolo delle sorgenti di illuminazione e stima dell'albedo e del colore.
- Forme dai contorni (shape from edges).
- Forme da tessitura (shape from texture).
- Forme da ombreggiatura (shape from shading).
- Stereo-matching.
- Determinazione della stuttura 3D e del movimento 3D di oggetti da sequenze di immagini monoculari e da sistemi di specchi (Structure from Motion).
- Ricostruzione 3D da stereo di oggetti della scena.
- Ricostruzione di superfici.
- Parametri geometrici del sistema di visione (calibrazione).
- *"High-level vision".*
- Interpretazione e movimento (la visione artificiale deriva storicamente dall'Intelligenza Artificiale).

Visione 3D, Elaborazione di immagini e grafica

Visione 3D: Immagine/i => Ricostruzione 3D della scena statica o dinamica ed interpretazione.

- **Grafica 3D:** Modello 3D della scena, statico o dinamico => Visualizzazione.

Si incontrano sul terreno della visualizzazione 3D.



Immagini

Immagini Statiche

Immagini Dinamiche

Shape from Monocular View

Feature Extraction

Differential Motion Analysis

3D Shape

Features

Optical Flow

L'*elaborazione delle imagini* costituisce il primo livello di un sistema di visione. Fornisce le features di base.

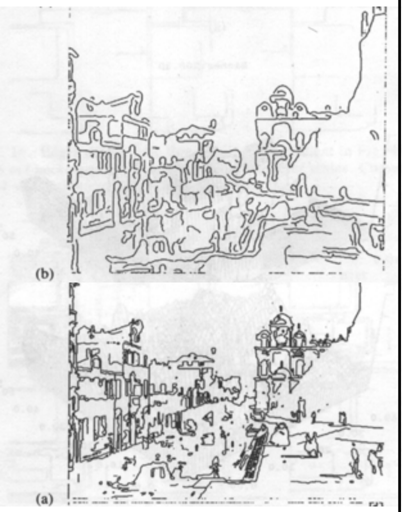
Cosa sono le features?

1) Località.

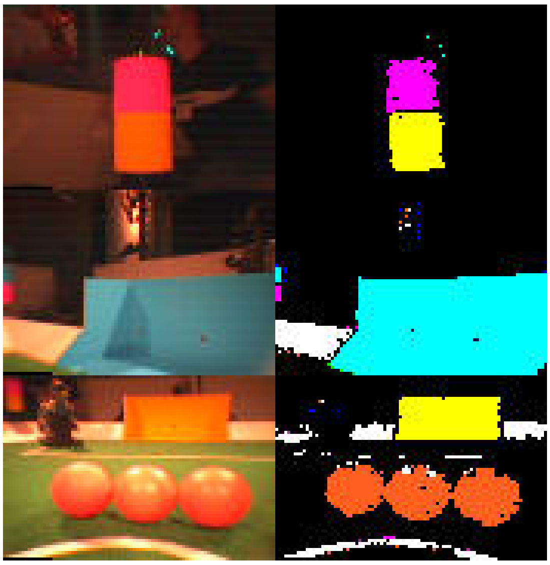
2) Significatività.

3) Riconoscibilità.

Riconoscimento dei bordi (edge)



Estrazione di regioni



Il flusso ottico

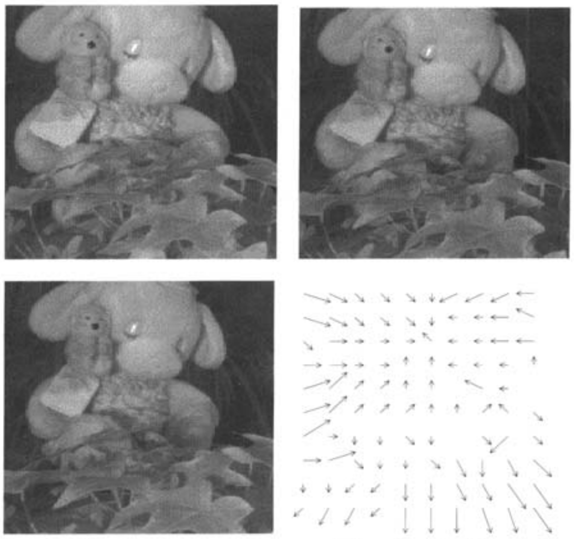


Figure 8.3 Three frames from a long image sequence (left to right and top to bottom) and the optical flow computed from the sequence, showing that the plant in the foreground is moving towards the camera, and the soft toys away from it.

I problemi di visione sono mal posti

- Perché non è facile costruire un sistema di visione?

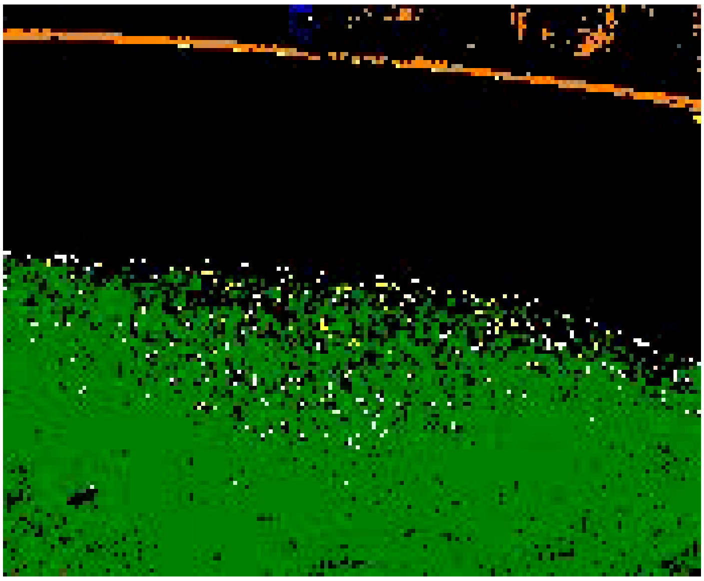
Difficoltà ad identificare esattamente le feature

- Risoluzione spaziale limitata.
- Gli oggetti reali non sono mai uniformemente illuminati.
- I contorni non sono netti.
- Le superfici non hanno albedo costante.
- L'illuminazione genera campi di irradianza "dificili".

Difficoltà ad assemblare le feature

Difficoltà ad interpretare le primitive visive

La vera Visione

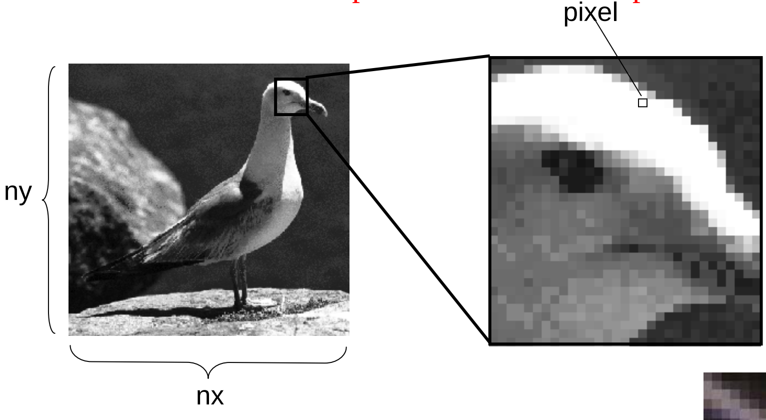


Sommario

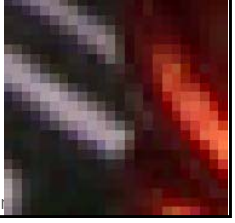
- La visione
- Le immagini digitali
- Il modello geometrico di una camera
- Segmentazione real-time.

IMMAGINI DIGITALI

Una matematica del pixel non è ancora disponibile.

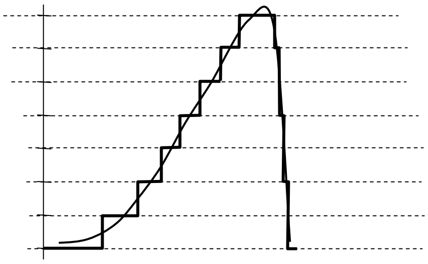


IMG = Matrice nx,ny =
$$
\begin{array}{c}
\begin{pmatrix}
142\ 174\ 164\ 144\ ..\ ..\ .. \\
107\ .... \\
... \\
...
\end{pmatrix} \quad ny \\
nx
\end{array}
$$

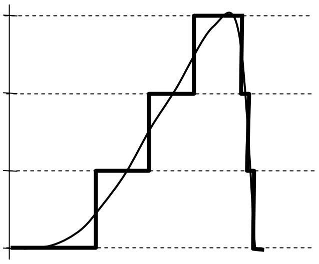


IMMAGINI DIGITALI

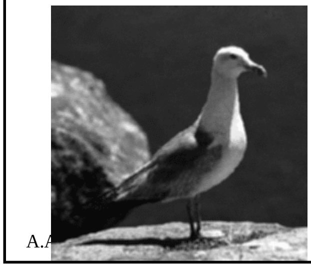
Quantizzazione: n bit => 2^n colori

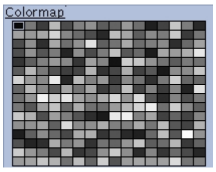


ES.: 3 bit => 2^3 = 8 '*gradini*'

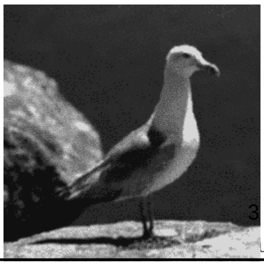


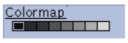
2 bit => 2^2 = 4 '*gradini*'





8 bit => 256 colori





3 bit => 8 colori

Colore

Colour is the colour which is perceived, seen, that is the colour which is reflected by the objects surface. Color is coded with three parameters (three channels).

Colour images are represented as additive mixture of Red Green Blue (additive mix).

Another two important codings are:

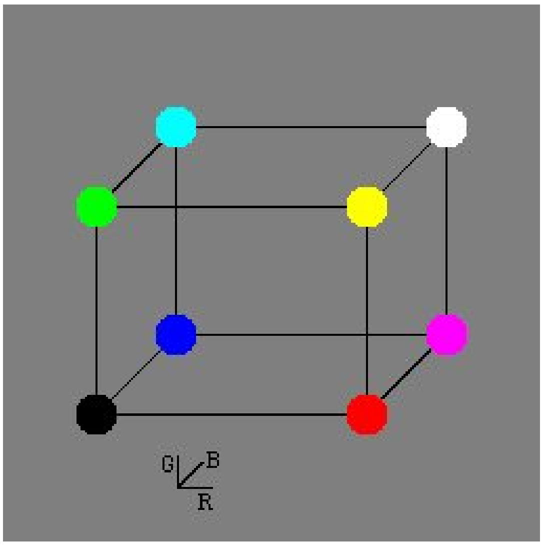
Hue. Describes the colour (red, green…)
Saturation. Quantity of the colour. It differentiates red from rose. It can be viewed as the difference from the colour and a grey with the same brightness.
Brightness. Intensità del colore, it depends on the hue and saturation. It can be viewed as the colour of the image in B/W. It is due to the illumination intensity.

Y – Brightness.	**Y**
U, V – Color.	**Cb**, **Cr**

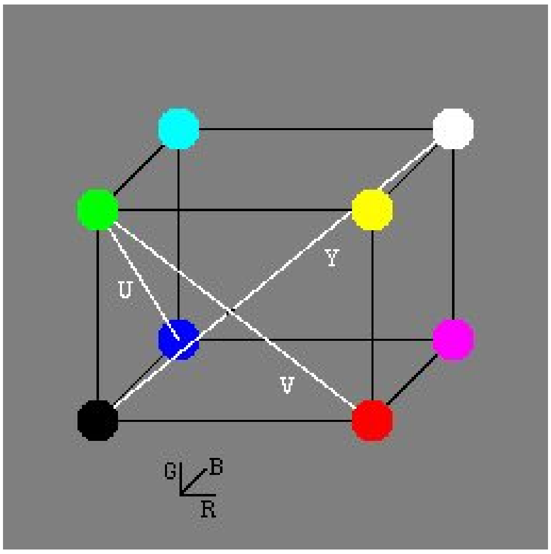
Colors (examples in RGB)

□	White	(R=255, G=255, B=255)
░	Light Grey	(R=100, G=100, B=100)
▓	Dark grey	(R=200, G=200, B=200)
⬛	Black	(R=0, G=0, B=0)
🟥	Red	(R=255, G=0, B=0)
🟨	Yellow	(R=255, G=255, B=0)
🟦	Pale blue	(R=0, G=255, B=255)
🟩	Green	(R=0, G=200, B=0)

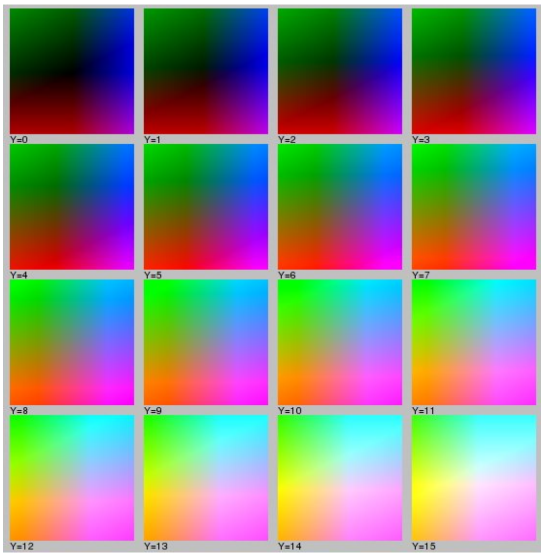
Color Spaces - RGB



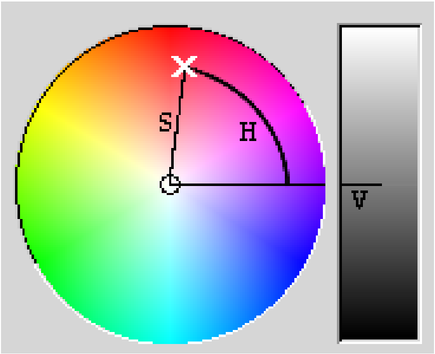
Color Spaces – YUV (YCbCr)

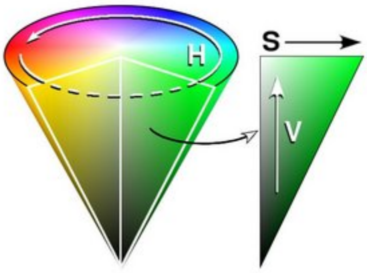


Color Spaces – YCbCr



Color Spaces – HSV





www.wordiq.com/definition/HSV_color_space

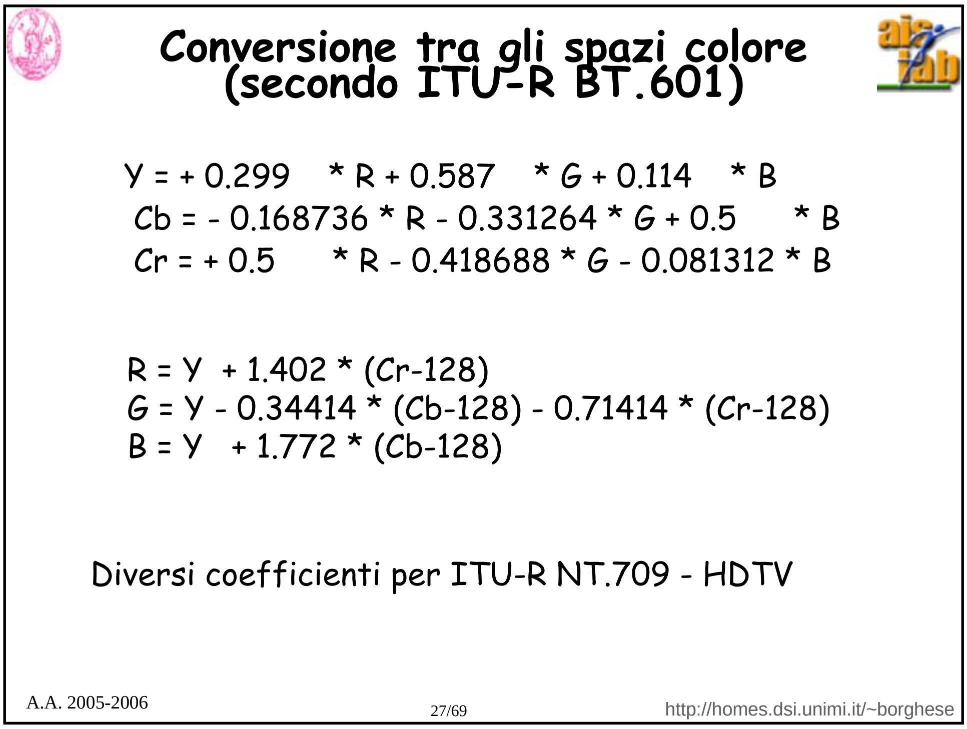
Image RGB

Image Raw

R=Y
G=Cb
B=Cr

Color Spaces - Discussion

- RGB
  - Handled by most capture cards
  - Used by computer monitors
  - Not easily separable channels
- YCbCr (YUV)
  - Handled by most capture cards
  - Used by TVs and JPEG images
  - Easily workable color space
- HSV
  - Rarely used in capture cards
  - Numerically unstable for grayscale pixels
  - Computationally expensive to calculate



Conversione tra gli spazi colore (secondo ITU-R BT.601)

$$Y = + 0.299 * R + 0.587 * G + 0.114 * B$$

$$Cb = - 0.168736 * R - 0.331264 * G + 0.5 * B$$

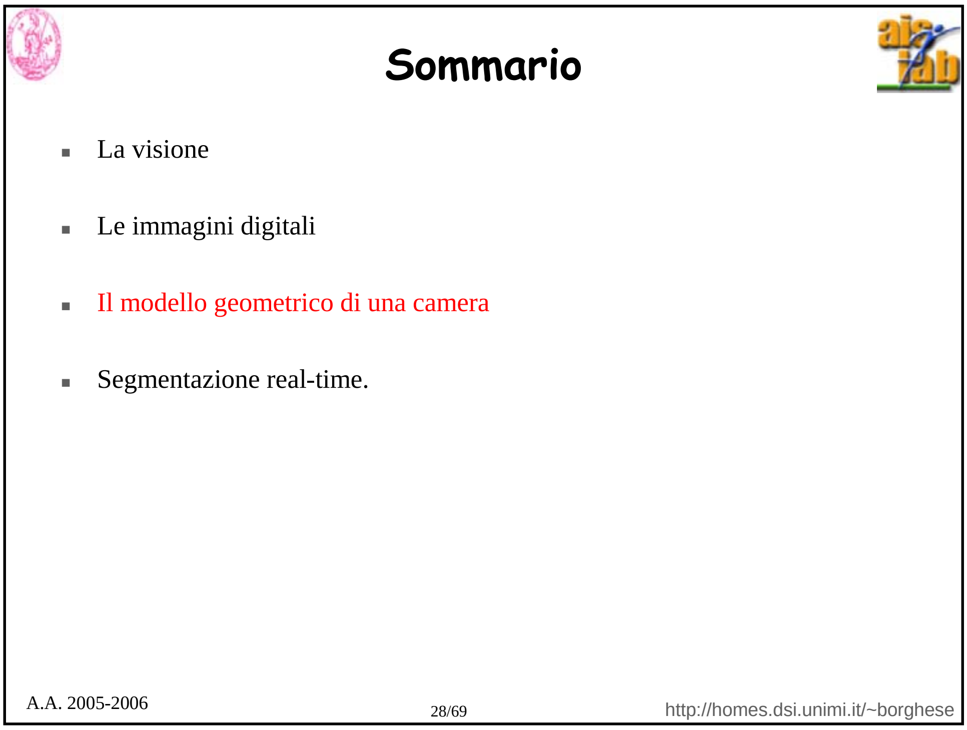
$$Cr = + 0.5 * R - 0.418688 * G - 0.081312 * B$$

$$R = Y + 1.402 * (Cr-128)$$

$$G = Y - 0.34414 * (Cb-128) - 0.71414 * (Cr-128)$$

$$B = Y + 1.772 * (Cb-128)$$

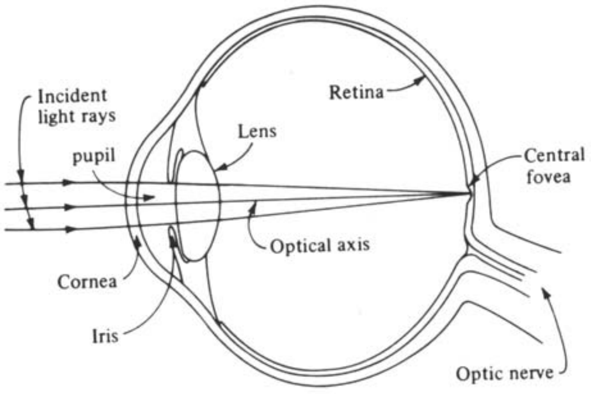
Diversi coefficienti per ITU-R NT.709 - HDTV



Sommario

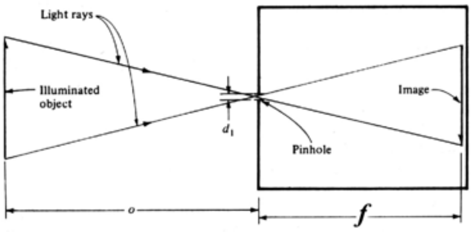
- La visione
- Le immagini digitali
- Il modello geometrico di una camera
- Segmentazione real-time.

L’occhio umano

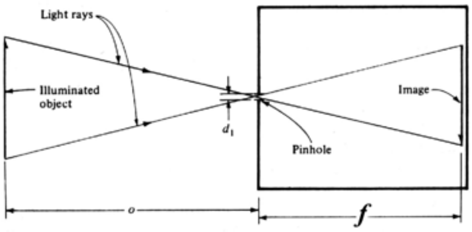


Its behavior is very similar to that of a camera

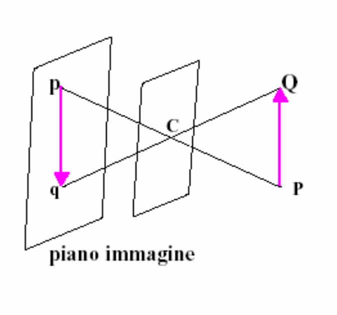
La camera come strumento di ripresa



La pin-hole camera

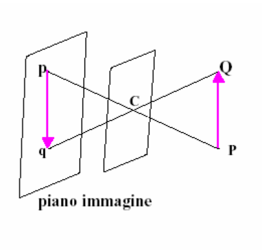


Proiezione prospettica: tutti i raggi di proiezione passano per un unico punto, detto **centro di proiezione**.

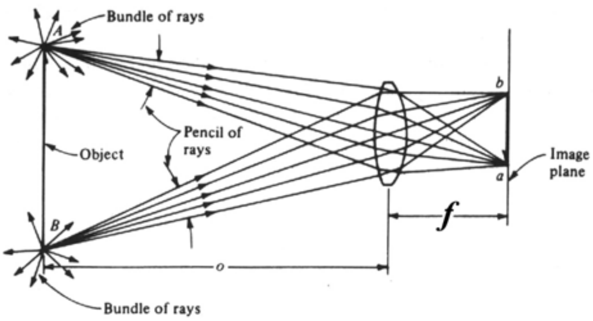


Pinhole camera

La lente

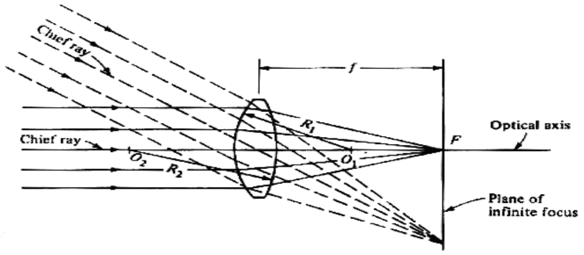


Pinhole camera



Lente convergente

Geometria dell'ottica

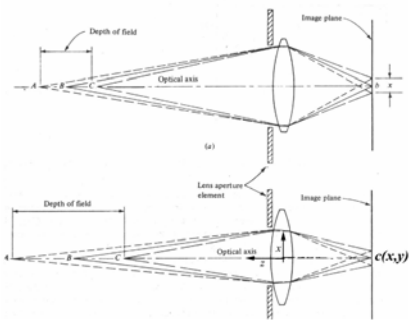


Oggetti all'infinito

- Distanza focale: distanza del piano immagine quando un oggetto si trova all'infinito.
- Asse ottico: raggio che non viene deviato dalla lente.
- Intersezione dell'asse ottico con il piano immagine dà il punto principale (F).

Messa a fuoco

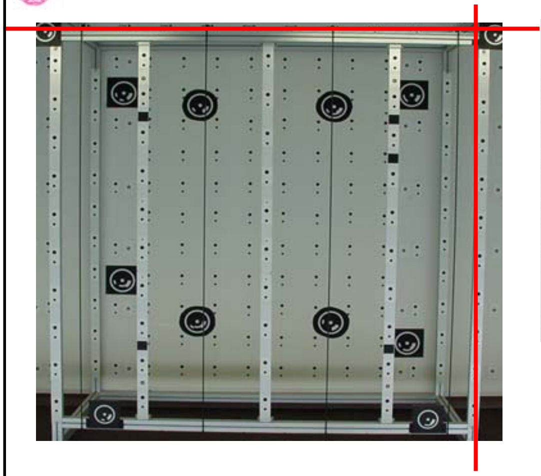
Problema della messa a fuoco

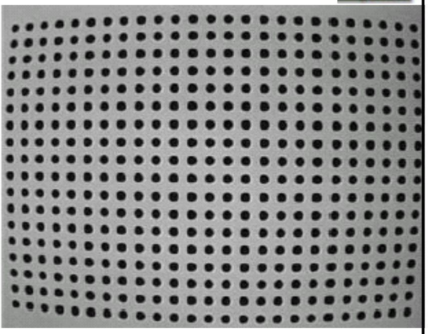


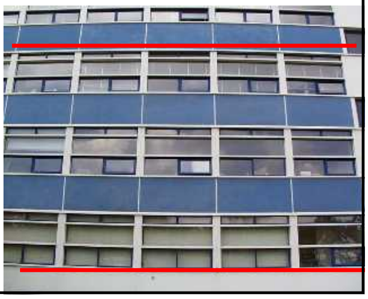
Parametri di camera (o intrinseci):

- Punto principale c(x,y) + lunghezza focale, f (3 parametri).
- Occorre conoscere anche il fattore di forma dei pixel nel caso di immagini digitali (è una costante, non un parametro).
- (Distorsioni).

Esempi di Distorsioni



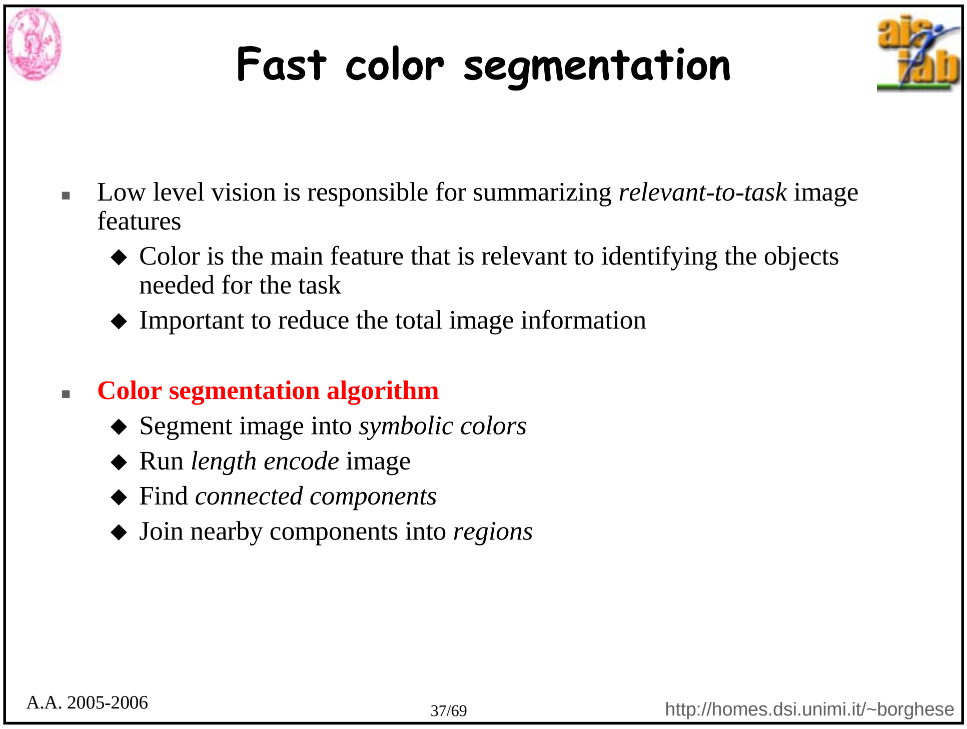




Ottime per effetti speciali, un po’ meno per delle misure…..
Le camere non sono metriche.

Sommario

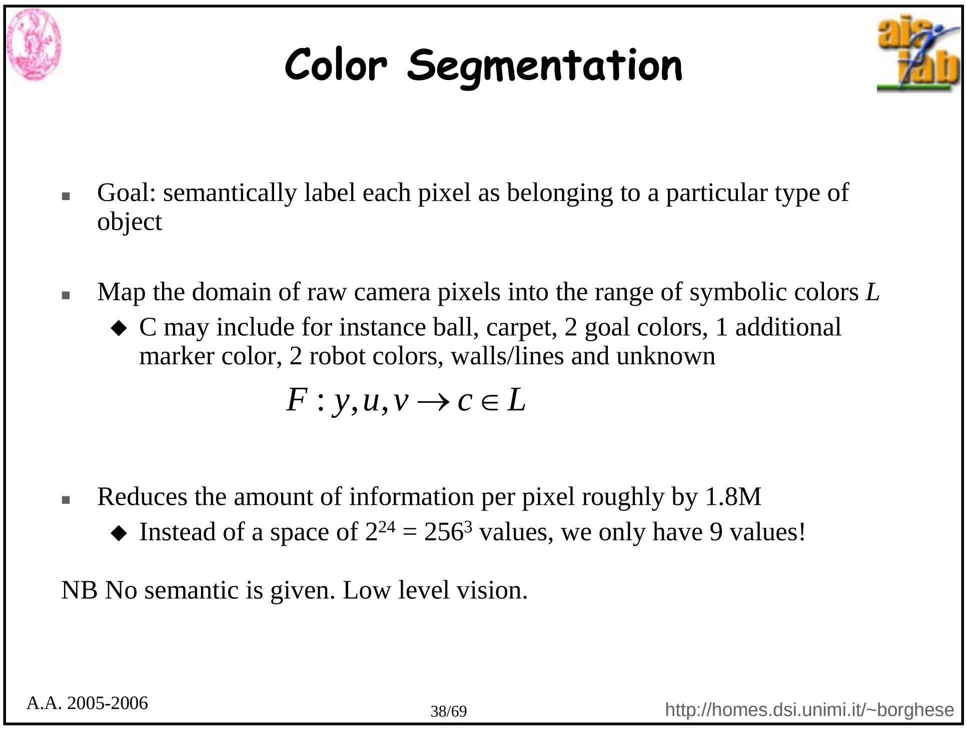
- La visione
- Le immagini digitali
- Il modello geometrico di una camera
- Segmentazione real-time.



Fast color segmentation

- Low level vision is responsible for summarizing *relevant-to-task* image features
  - Color is the main feature that is relevant to identifying the objects needed for the task
  - Important to reduce the total image information

- **Color segmentation algorithm**
  - Segment image into *symbolic colors*
  - Run *length encode* image
  - Find *connected components*
  - Join nearby components into *regions*



Color Segmentation

- Goal: semantically label each pixel as belonging to a particular type of object

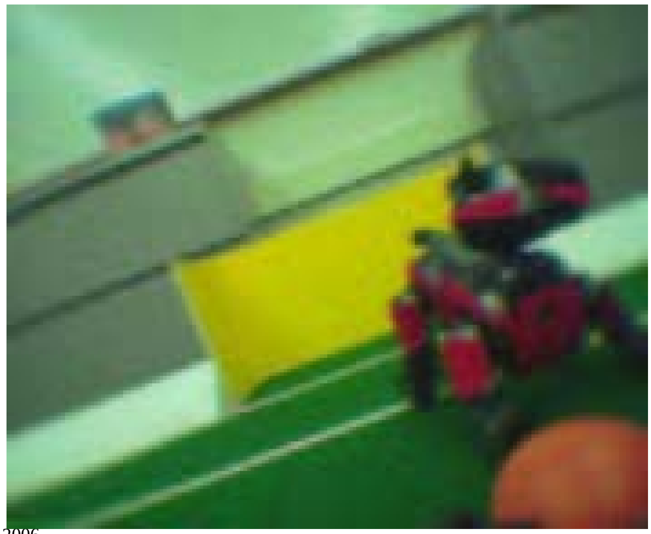
- Map the domain of raw camera pixels into the range of symbolic colors *L*
  - C may include for instance ball, carpet, 2 goal colors, 1 additional marker color, 2 robot colors, walls/lines and unknown

$$F : y,u,v \rightarrow c \in L$$

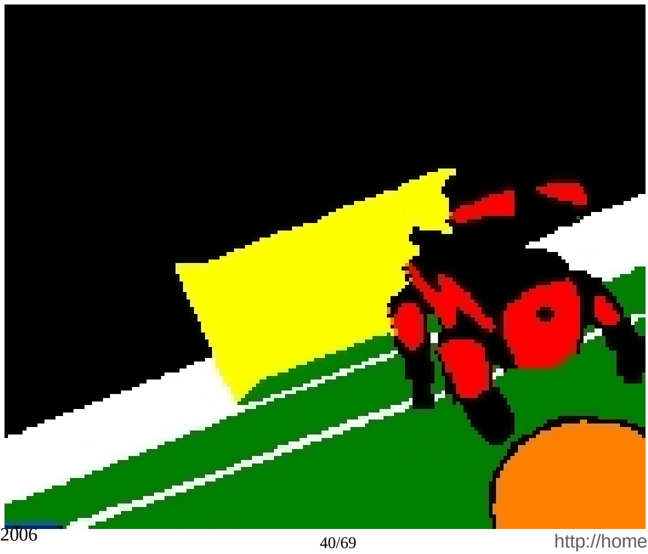
- Reduces the amount of information per pixel roughly by 1.8M
  - Instead of a space of $2^{24} = 256^3$ values, we only have 9 values!

NB No semantic is given. Low level vision.

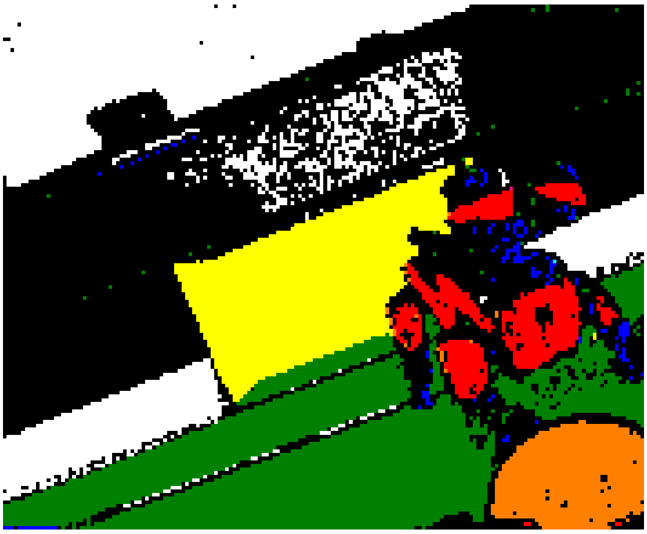
Before Segmentation



Ideal Segmentation



Result of Segmentation



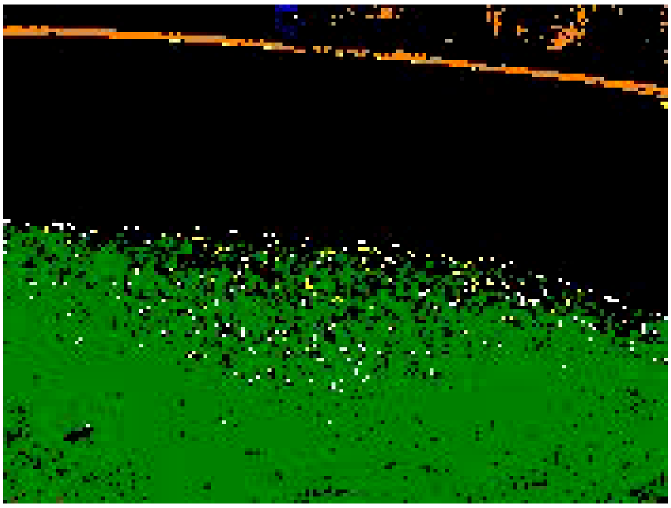
Segmentazione HW in 8 classi. Viene definita per ogni classe:

Y_{min} Y_{max}

Cb_{min} Cb_{max}

Cr_{min} Cr_{max}

Potential Problems with Color Segmentation



Color Segmentation Analysis

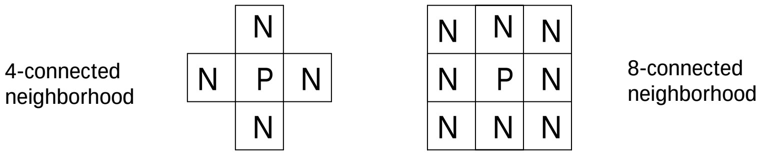
- Advantages
  - Quickly extract relevant information
  - Provide useful representation for higher-level processing
  - Differentiate between YCbCr pixels that have *similar* values

- Disadvantages
  - Cannot segment YCbCr pixels that have *identical* values into different classes
  - Generate smoothly contoured regions from noisy images

Turning Pixels into Regions

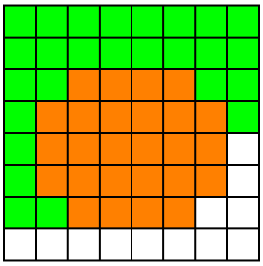
- A disjoint set of labeled pixels is still not enough to properly identify objects
- Pixels must be grouped into spatially-adjacent regions
  - Regions are grown by considering local neighborhoods around pixels

How to achieve this?

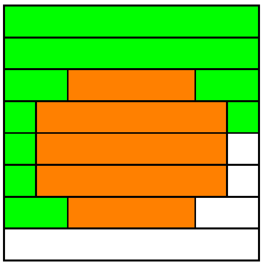


1) Run Length Encoding

- Segment each image row into groups of similar pixels called *runs*
  - Runs store a start and number of pixels belonging to the **same** color (alternatively the end point is memorized)

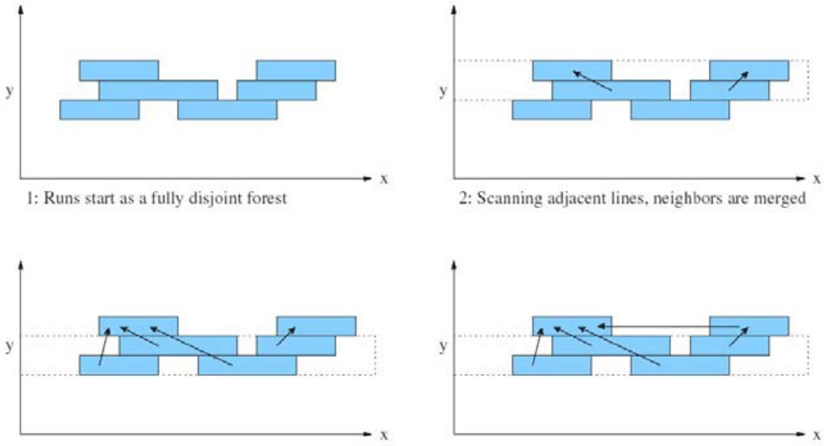


Original image



RLE image

2) Merging Regions



1: Runs start as a fully disjoint forest

2: Scanning adjacent lines, neighbors are merged

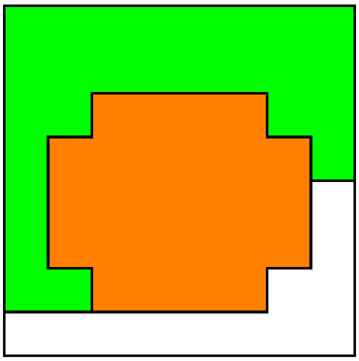
3: New parent assignments are to the furthest parent

4: If overlap is detected, latter parent is updated

Goal: a closed region is formed.

Final Results

- Runs are merged into multi-row regions
- Image is now described as contiguous regions instead of just pixels
- Objects are convexes



Data Extracted from Regions

- Features extracted from regions
  - *Centroid*
    - Mean location
  - *Bounding box*
    - Max and min (x,y) values
  - *Area*
    - Number of pixels in box
  - *Average color*
    - Mean color of region pixels

- Regions are stored by color class and sorted by largest area
- These features let us write concise and fast object detectors

Features are a function of later processing required.

High level vision - Object Detection Process

- Produces a set of candidate objects that might be this object from lists of regions
  - Given ‘n’ orange blobs, is one of them the ball?
- Compares each candidate object to a set of **models** that predict what the object would look like when seen by a camera
  - **Models** encapulate all assumptions
  - Also called filtering
- Selects best match to report to behaviors
  - Position and quality of match are also reported

Filtering Overview

- Each filtering **model** produces a number in [0.0, 1.0] representing the certainty of a match
  - Some filters can be binary and will return either 0.0 or 1.0
- Certainty levels are multiplied together to produce an overall match
  - Real-valued range allows for areas of uncertainty
  - Keeps one bad filter result from ruining the object
  - Multiple bad observations will still cause the object to be thrown out

Alternatively fuzzy inference can be used.

Object Detection Algorithm

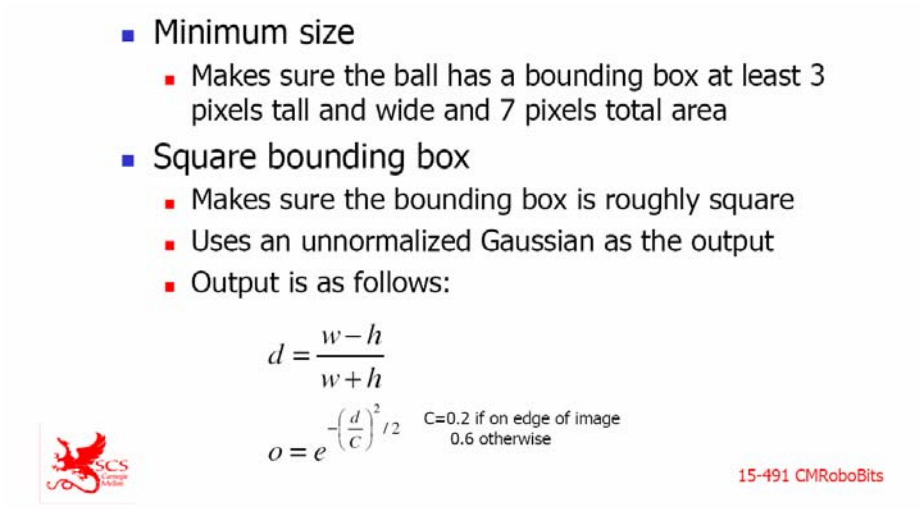
1) Produce a set of candidate objects that might be the object from the list of regions produced by low-level vision.

2) Compare each candidate object to a set of models that predict features that the object should have when seen through a camera.

3) Select the best match and compute its distance to the robot based on the robot’s camera model and kinematics.

4) Return this best match along with the quality of the estimate to the behaviors.

Riconoscimento palla – CMU – I



- Minimum size
  - Makes sure the ball has a bounding box at least 3 pixels tall and wide and 7 pixels total area
- Square bounding box
  - Makes sure the bounding box is roughly square
  - Uses an unnormalized Gaussian as the output
  - Output is as follows:

$$d = \frac{w-h}{w+h}$$

$$o = e^{-\left(\frac{d}{c}\right)^2/2}$$

C=0.2 if on edge of image
0.6 otherwise

15-491 CMRoboBits

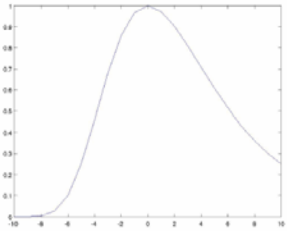
Riconoscimento palla – CMU – II

Filter

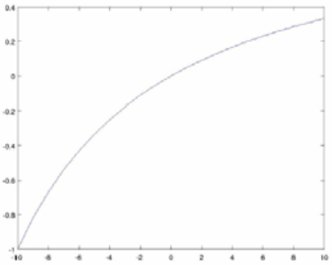
$$d = \frac{w-h}{w+h}$$

$$o = e^{-\left(\frac{d}{C}\right)^2/2}$$

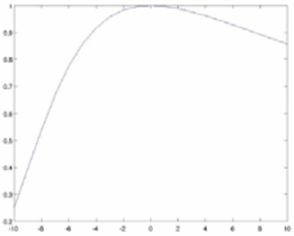
Plot: o
C=0.2



Plot: d
H=10
W=[0-20]

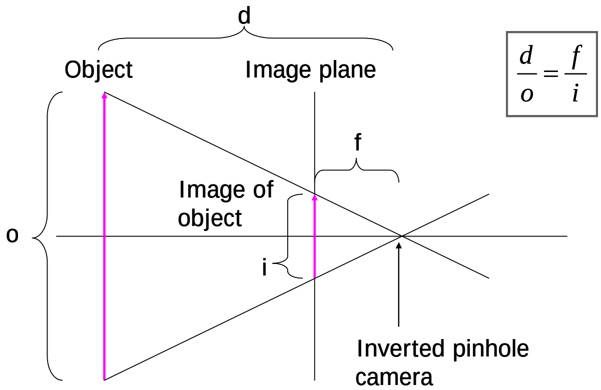


Plot: o
C=0.6



Ball Distance Calculation

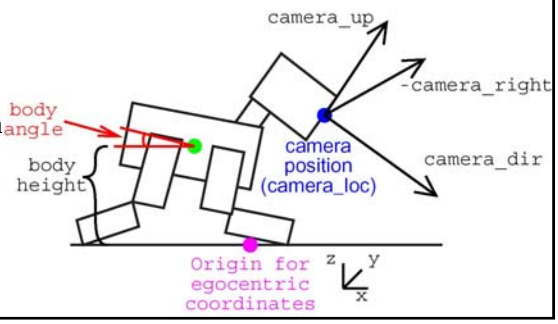
- The size of the ball is known
- The kinematics of the robot are known
- Given a simplified camera projection model, the distance to the ball can be calculated



$$\frac{d}{o} = \frac{f}{i}$$

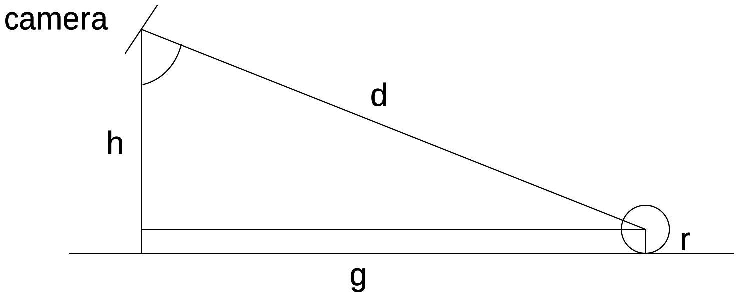
Calculation of Camera Position

- Position of camera is calculated based on body position and head position w.r.t body
- Body position is known from walk engine
- Head position relative to body position is found from forward kinematics using joint positions
- Camera position
  - *camera_loc* is defined as position of camera relative to egocentric origin
  - *camera_dir*, *camera_up*, and *camera_down* are unit vectors in egocentric space
    - Specify camera direction, up and right in the image



Ball Position Estimation

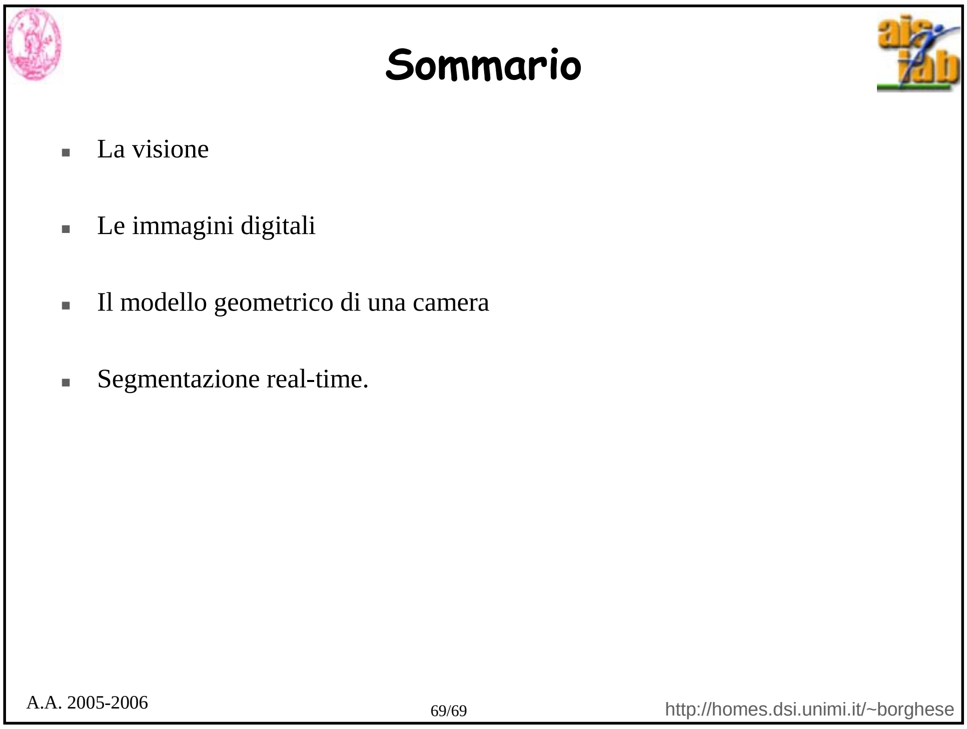
- Two methods are used for estimating the position of the ball
  - The first calculates the camera angle from the ball model
  - The second uses the robot's encoders to calculate the head angle
- The first is more accurate but relies on the pixel size of the ball
  - This method is chosen if the ball is NOT on the edge of the image
  - Partial occlusions will make this estimate worse



Additional Color Filters - tricks

- The pixels around the ball are analyzed
  - Red vs. area
    - Filters out candidate balls that are part of red robot uniform
  - Green filter
    - Ensures the ball is near the green floor
- If the ball is farther than 1.5m away
  - Average “redness” value of the ball is calculated
    - If too red, then the ball is assumed to be the fringe of the red robot’s uniform

End Result – Accurate Ball Position



Sommario

- La visione
- Le immagini digitali
- Il modello geometrico di una camera
- Segmentazione real-time.