Computer Vision

- Computer vision: processing data from any modality that uses the electromagnetic spectrum which produces an image
- Image:
  - *A way of representing data in a picture-like format where there is a direct physical correspondence to the scene being imaged*
  - *Results in a 2D array or grid of readings*
  - *Every element in array maps onto a small region of space*
  - *Elements in image array are called pixels*
- Modality determines what image measures:
  - *Visible light* ➔ *measures value of light (e.g. color or gray level)*
  - *Thermal* ➔ *measures heat in the given region*
- Image function: converts signal into a pixel value

Pan-Tilt-Zoom camera

CMU Cam
(for color blob tracking)

Omnidirectional Camera

Stereo vision

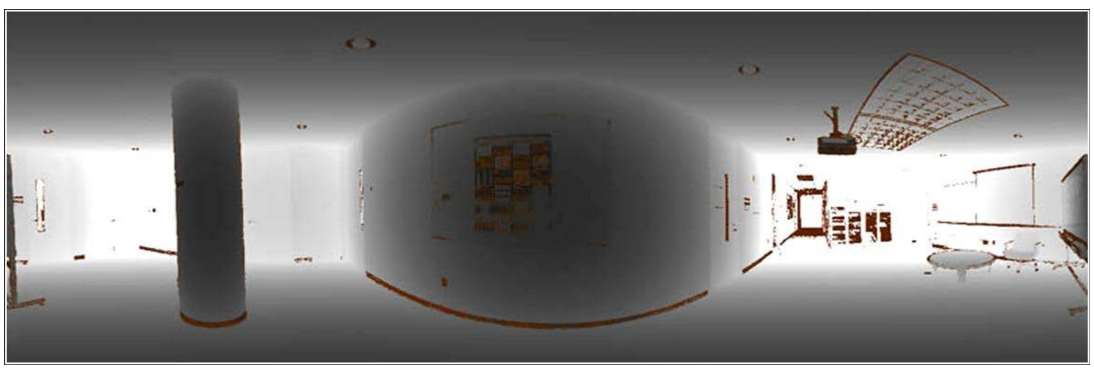
© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Types of Computer Vision

- Computer vision includes:
  - *Cameras (produce images over same electromagnetic spectrum that humans see)*
  - *Thermal sensors*
  - *X-rays*
  - *Laser range finders*
  - *Synthetic aperature radar (SAR)*

Thermal image

SAR image (of U.S. capitol building)



3D Laser scanner image

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Computer Vision is a Field of Study on its Own

- Computer vision field has developed algorithms for:
  - *Noise filtering*
  - *Compensating for illumination problems*
  - *Enhancing images*
  - *Finding lines*
  - *Matching lines to models*
  - *Extracting shapes and building 3D representations*

- *However, autonomous mobile robots operating in dynamic environments must use computationally efficient algorithms; not all vision algorithms can operate in real-time*

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

CCD (Charge Coupled Device) Cameras

- CCD technology: Typically, computer vision on autonomous mobile robots is from a video camera, which uses CCD technology to detect visible light

- Output of most cameras: analog; therefore, must be digitized for computer use

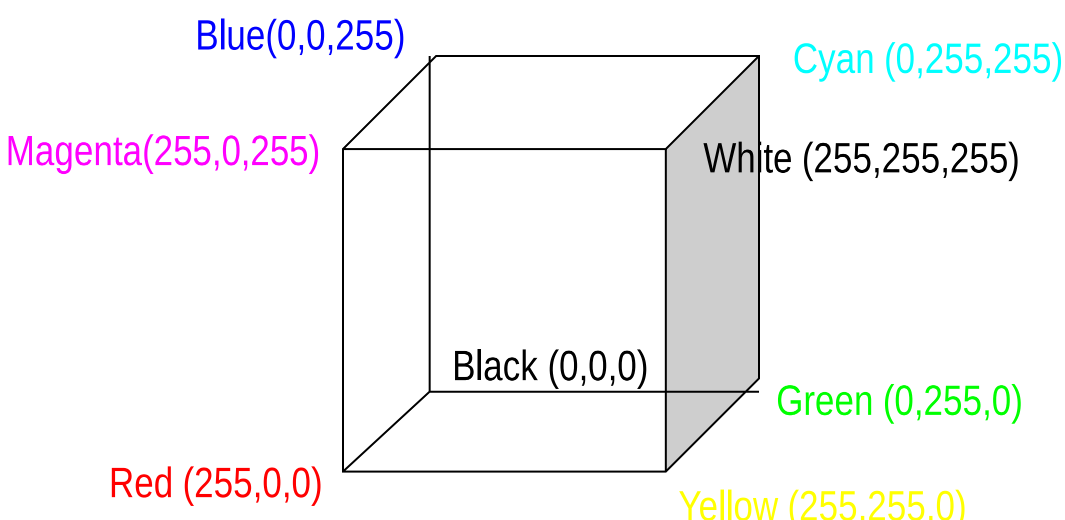
- Framegrabber:
  - *Card that is used by the computer, which accepts an analog camera signal and outputs the digitized results*
  - *Can produce gray-scale or color digital image*
  - *Have become fairly cheap – color framegrabbers cost about $200-$500.*

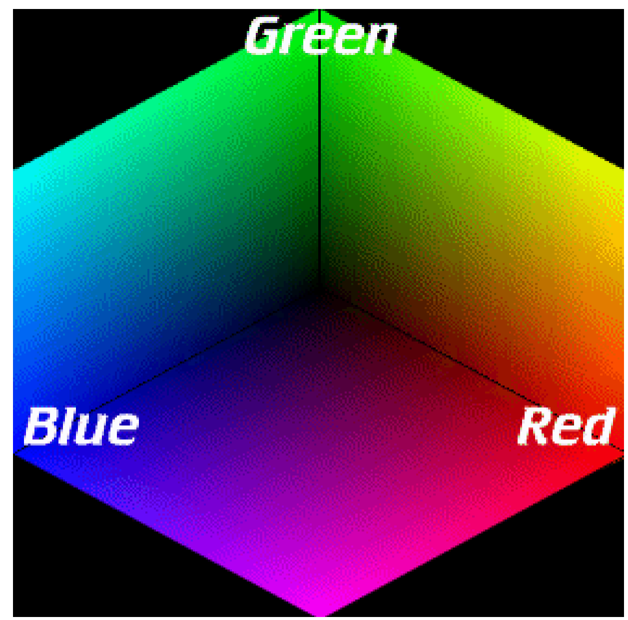
© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Representation of Color

- Color measurements expressed as three color planes – red, green, blue (abbreviated RGB)

- RGB usually represented as axes of 3D cube, with values ranging from 0 to 255 for each axis





© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Software Representation

1. Interleaved: colors are stored together (most common representation)
   - *Order: usually red, then green, then blue*

Example code:

```
#define RED 0
#define GREEN 1
#define BLUE 2

int  image[ROW][COLUMN][COLOR_PLANE];
…
red = image[row][col][RED];
green = image[row][col][GREEN];
blue = image[row][col][BLUE];
display_color(red, green, blue);
```

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Software Representation (con't.)

2. Separate: colors are stored as 3 separate 2D arrays

Example code:

```
int  image_red[ROW][COLUMN];
int  image_green[ROW][COLUMN];
int  image_blue[ROW][COLUMN];

…
red = image_red[row][col];
green = image_green[row][col];
blue = image_blue[row][col];
display_color(red, green, blue);
```

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Challenges Using RGB for Robotics

- Color is function of:
  - *Wavelength of light source*
  - *Surface reflectance*
  - *Sensitivity of sensor*

- → Color is not absolute;
  - *Object may appear to be at different color values at different distances to due intensity of reflected light*

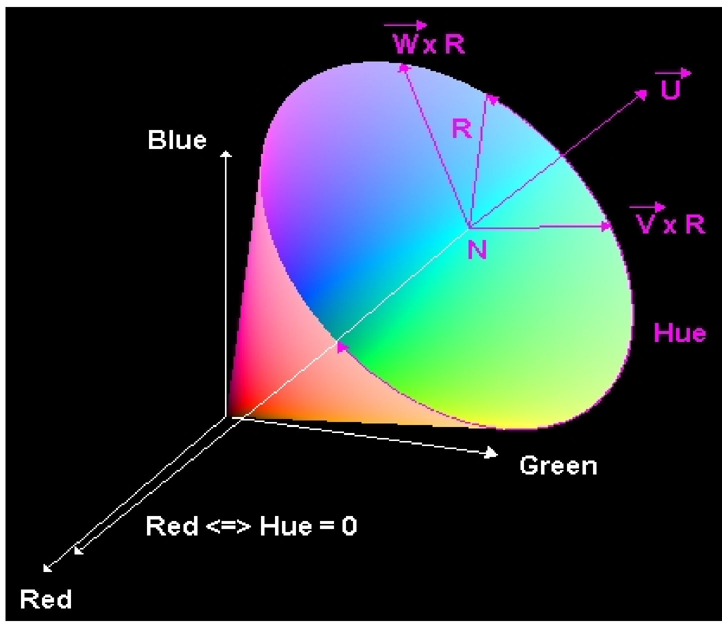
© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Better: Device which is sensitive to absolute wavelength

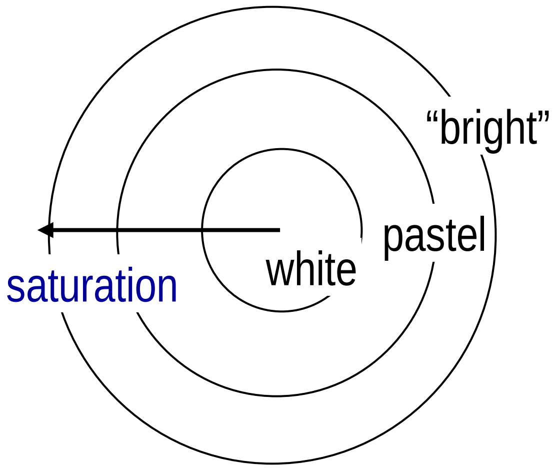
Better: Hue, saturation, intensity (or value) (HSV) representation of color

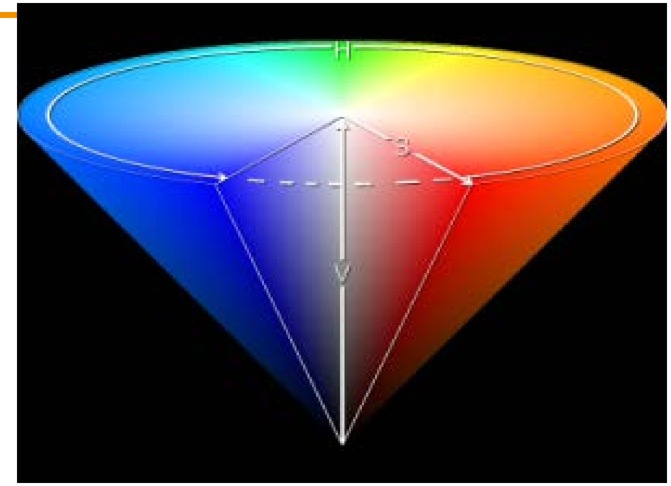
- Hue: dominant wavelength, does not change with robot's relative position or object's shape
- Saturation: lack of whiteness in the color (e.g., red is saturated, pink is less saturated)
- Intensity/Value: quantity of light received by the sensor

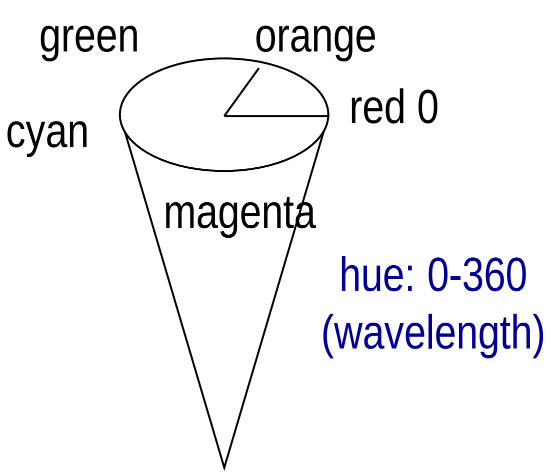
Transforming RGB to HSV

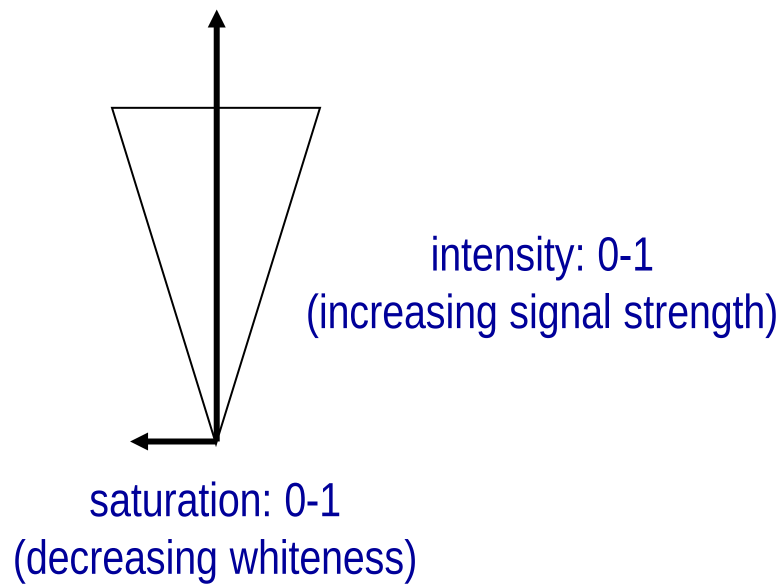


Representation of HSV









© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

HSV Challenges for Robotics

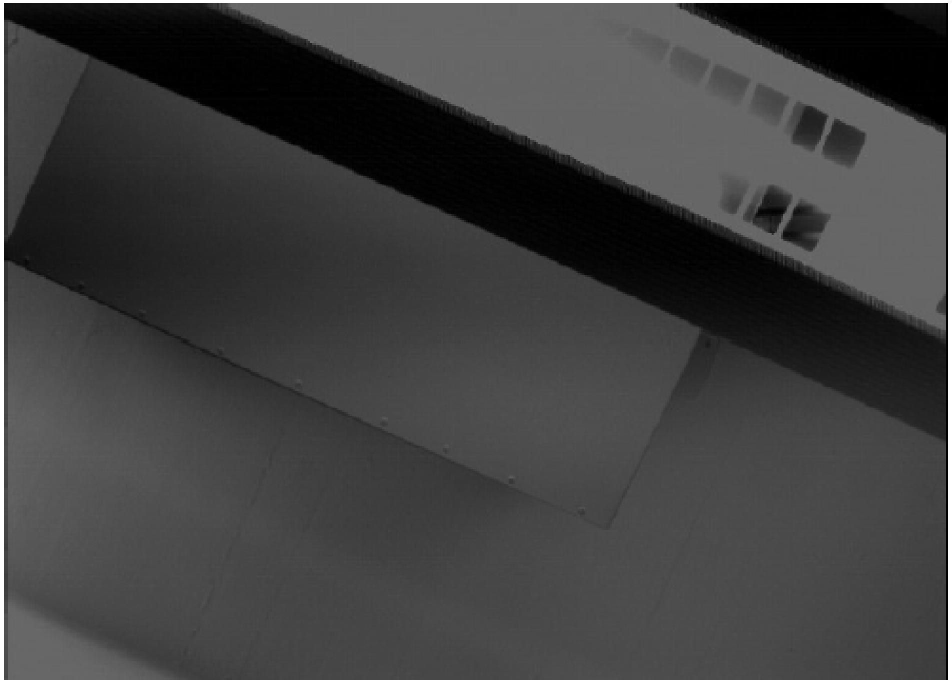
- Requires special cameras and framegrabbers
- Expensive equipment

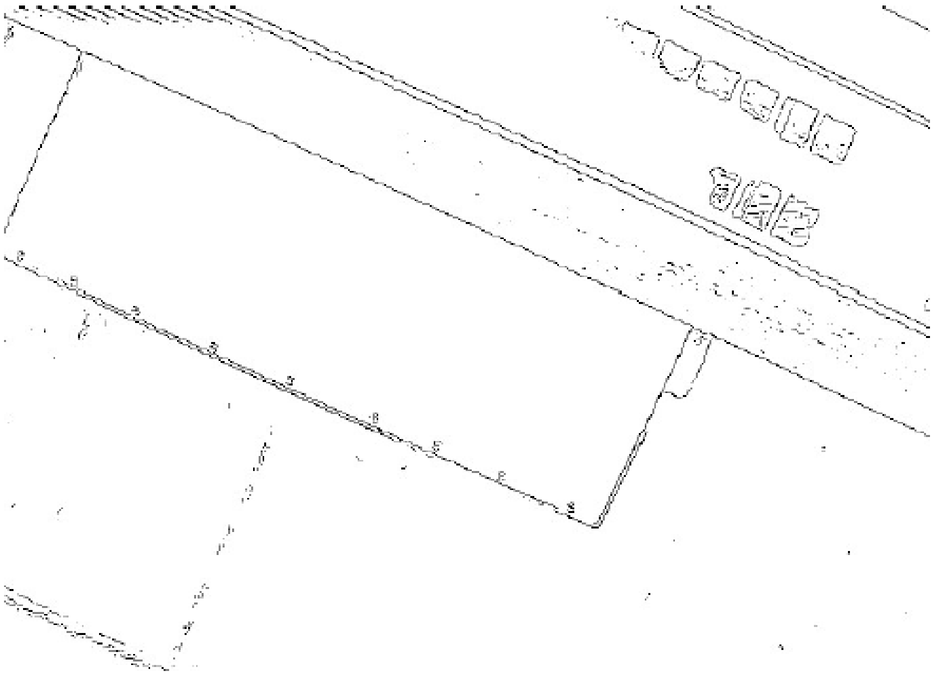
- Alternative: Use algorithm to convert -- Spherical Coordinate Transform (SCT)
  - *Transforms RGB data to a color space that more closely duplicates response of human eye*
  - *Used in biomedical imaging, but not widely used for robotics*
  - *Much more insensitive to lighting changes*

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Edge Detection

- Ultimate goal of edge detection
  - *an idealized line drawing.*
- Edge contours in the image correspond to important scene contours.





© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Region Segmentation

- Region Segmentation: most common use of computer vision in robotics, with goal to identify region in image with a particular color

- Basic concept: identify all pixels in image which are part of the region, then navigate to the region's centroid

- Steps:
  - *Threshold all pixels which share same color (thresholding)*
  - *Group those together, throwing out any that don't seem to be in same area as majority of the pixels (region growing)*

Example Code for Region Segmentation

```
for (i=0; i<numberRows; i++)
  for (j=0; j<numberColumns; j++)
    { if (((ImageIn[i][j][RED] >= redValueLow)
        && (ImageIn[i][j][RED] <= redValueHigh))
        && ((ImageIn[i][j][GREEN] >= greenValueLow)
        && (ImageIn[i][j][GREEN] <= greenValueHigh))
        && ((ImageIn[i][j][BLUE] >= blueValueLow)
        && (ImageIn[i][j][BLUE] <= blueValueHigh)))
        ImageOUT[i][j] = 255;
      else
        ImageOut[i][j] = 0;
    }
```

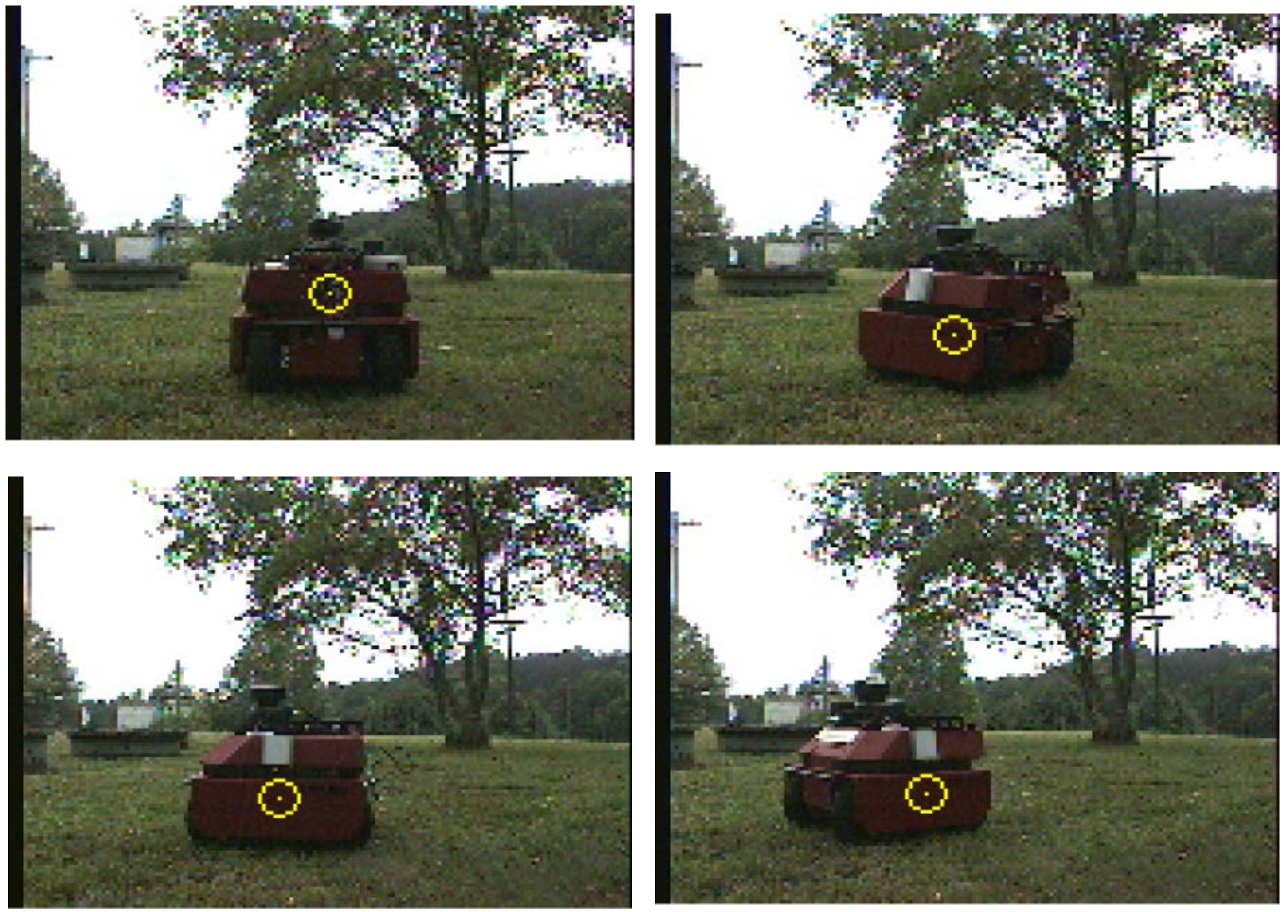
Note range of readings required due to non-absolute color values

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Example of Region-Based Robotic Tracking using Vision

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

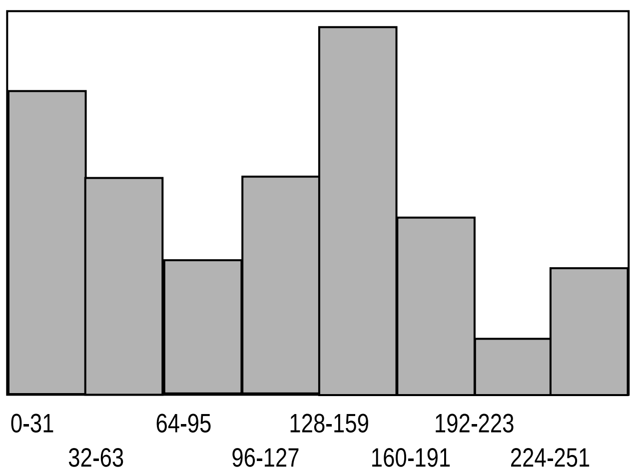
Another Example of Vision-Based Robot Detection Using Region Segmentation



© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Color Histogramming

- Color histogramming:
  - *Used to identify a region with several colors*
  - *Way of matching proportion of colors in a region*
- Histogram:
  - *Bar chart of data*
  - *User specifies range of values for each bar (called buckets)*
  - *Size of bar is number of data points whose value falls into the range for that bucket*
- Example:



Color Histograms (con't.)

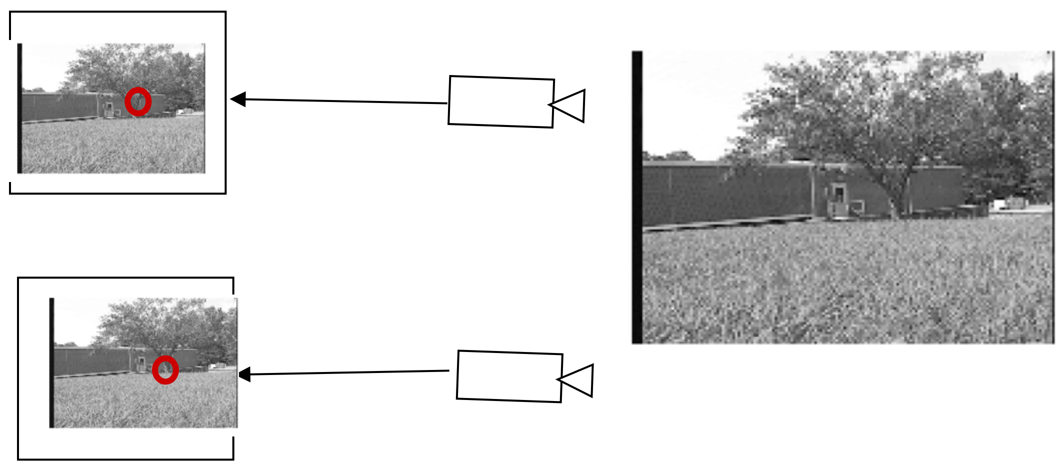
- Advantage for behavior-based/reactive robots: Histogram Intersection
  - *Color histograms can be subtracted from each other to determine if current image matches a previously constructed histogram*
  - *Subtract histograms bucket by bucket; different indicates # of pixels that didn't match*
  - *Number of mismatched pixels divided by number of pixels in image gives percentage match = Histogram Intersection*
- This is example of local, behavior-specific representation that can be directly extracted from environment

Range from Vision

- Perception of depth from stereo image pairs, or from optic flow

- Stereo camera pairs: range from stereo

- Key challenge: how does a robot know it is looking at the same point in two images?
  - *This is the correspondence problem.*



© R. Siegwart, I. Nourbakhsh (Adapted by Parker)

Simplified Approach for Stereo Vision

- Given scene and two images
- Find interest points in one image
- Compute matching between images (correspondence)
- Distance between points of interest in image is called disparity
- Distance of point from the cameras is inversely proportional to disparity
- Use triangulation and standard geometry to compute depth map

- Issue: camera calibration: need known information on relative alignment between cameras for stereo vision to work properly

© R. Siegwart, I. Nourbakhsh (Adapted by Parker)