CMRoboBits:
Creating an Intelligent AIBO Robot
Vision I

Instructors: Dr. Paul E. Rybski and

Prof. Manuela Veloso

TA: Juan Fasola (and Matt Russo)

15-491, Fall 2005

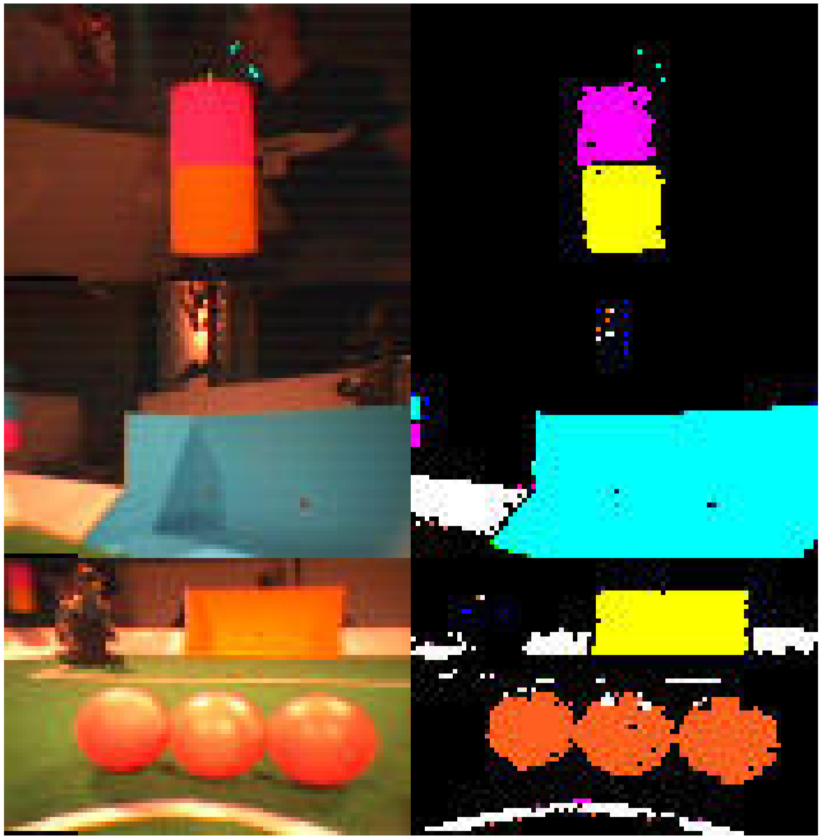
http://www.andrew.cmu.edu/course/15-491

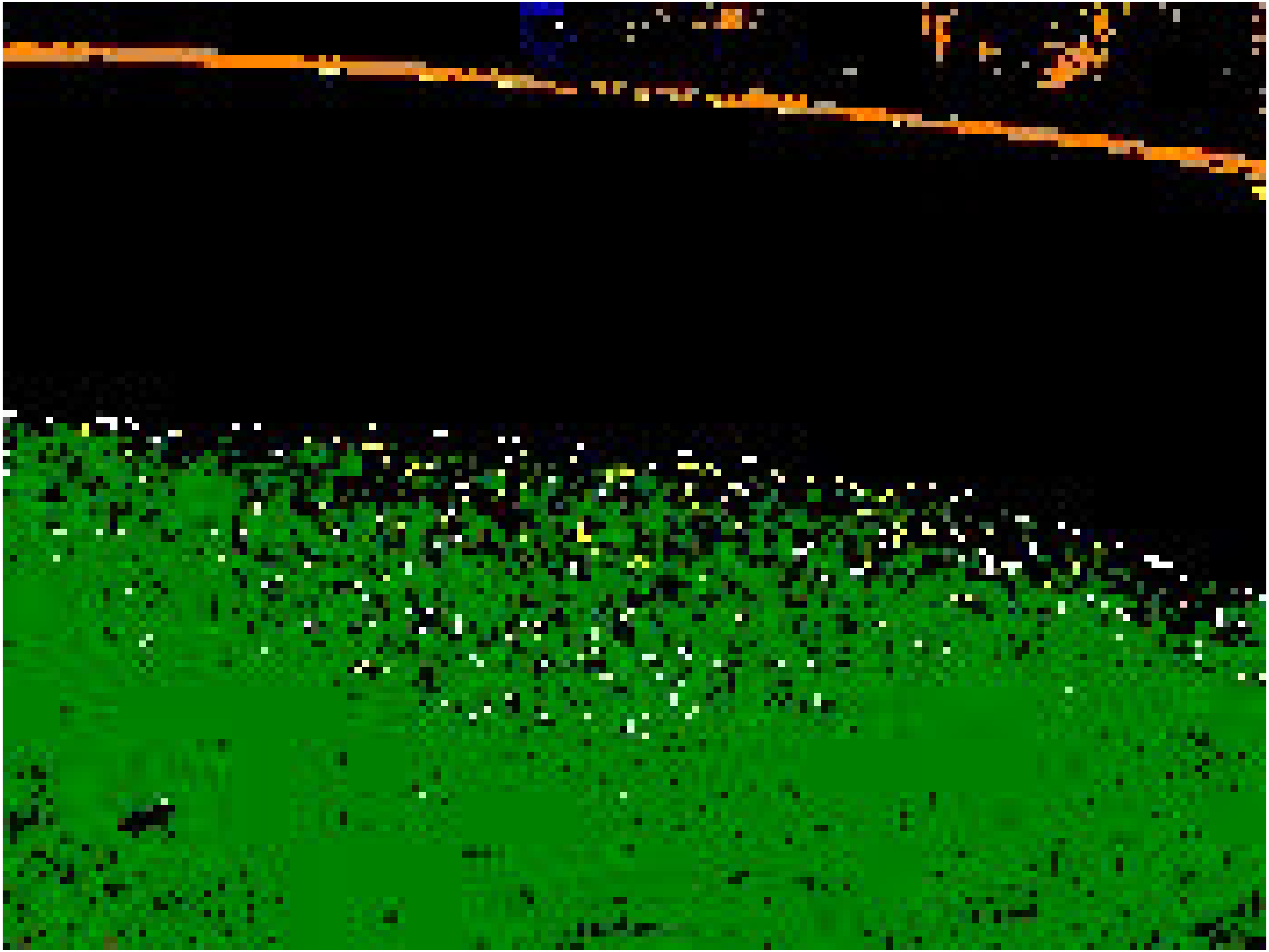
Computer Science Department

Carnegie Mellon

AIBO Vision

- Goals of this lecture
  - Illustrate the underlying processing involved with the AIBO vision system
  - Describe the high-level object recognition system
  - Provide enough background so that you can consider adding your own object detectors into the AIBO vision system





What is Computer Vision?

- The process of extracting information from an image
  - Identifying objects projected into the image and determining their position
  - The art of throwing out information that is not needed, while keeping information needed
- A very challenging research area
  - Not a solved problem!

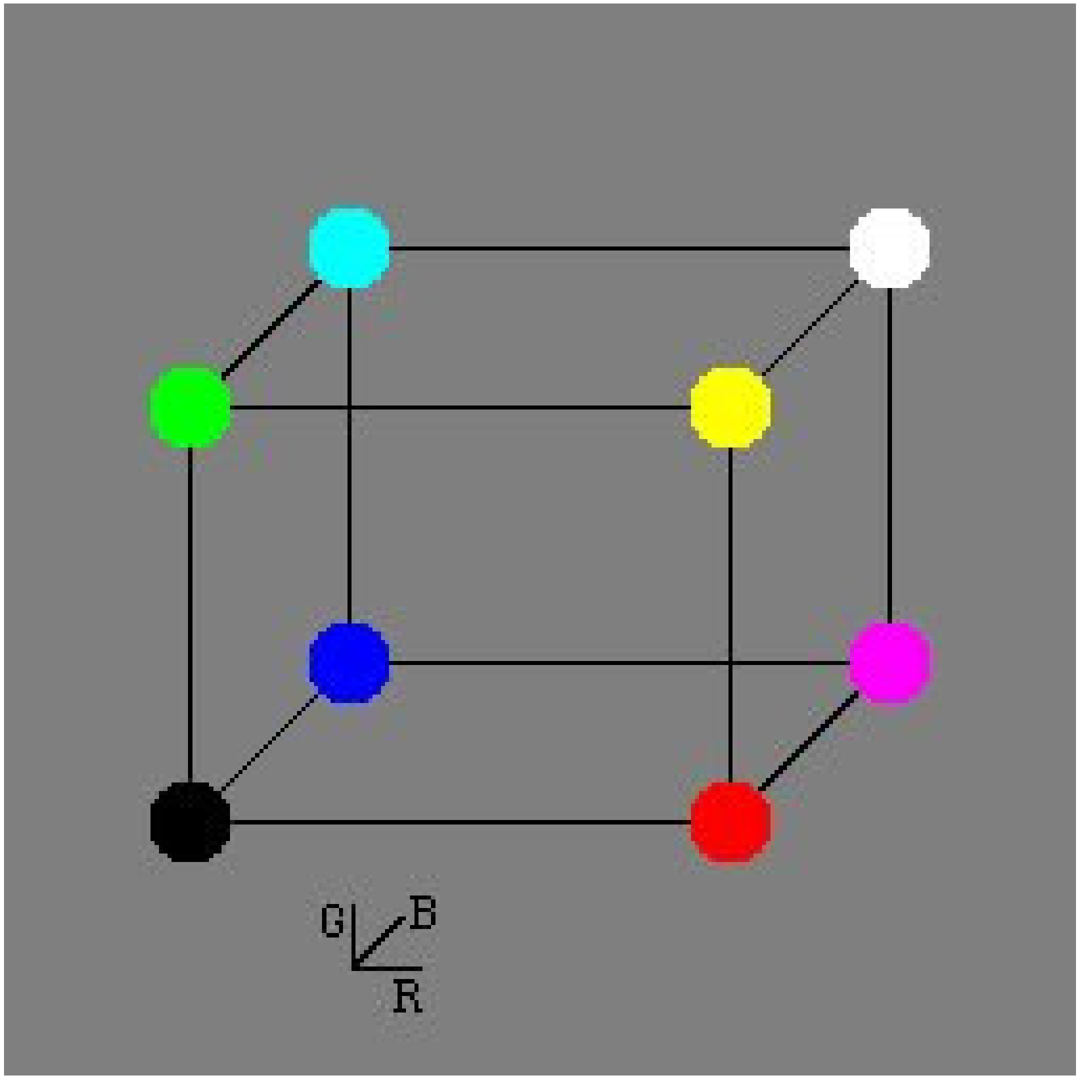
AIBO Vision

- AIBO camera provides images formatted in the *YUV* color space
- Each image is an array of 208 x 160 p*ixels*

Color Spaces

- Each pixel is a 3 dimensional value
  - Each dimension is called a *channel*
- There are multiple different possible color representations
  - RGB – R=red, G=green, B=blue
  - YUV – Y=brightness, UV=color
  - HSV – H=hue, S=saturation, V=brightness

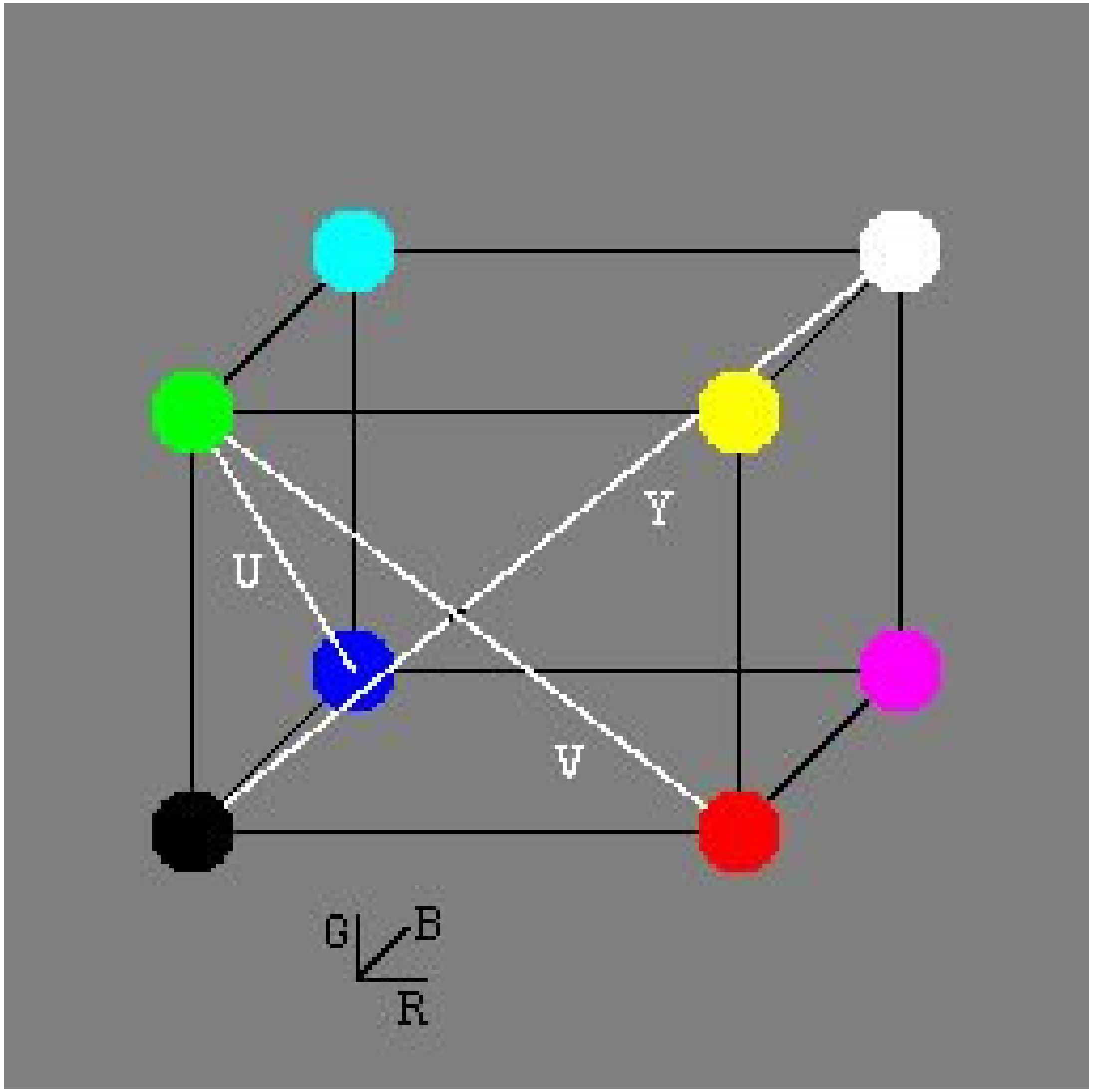
Color Spaces - RGB



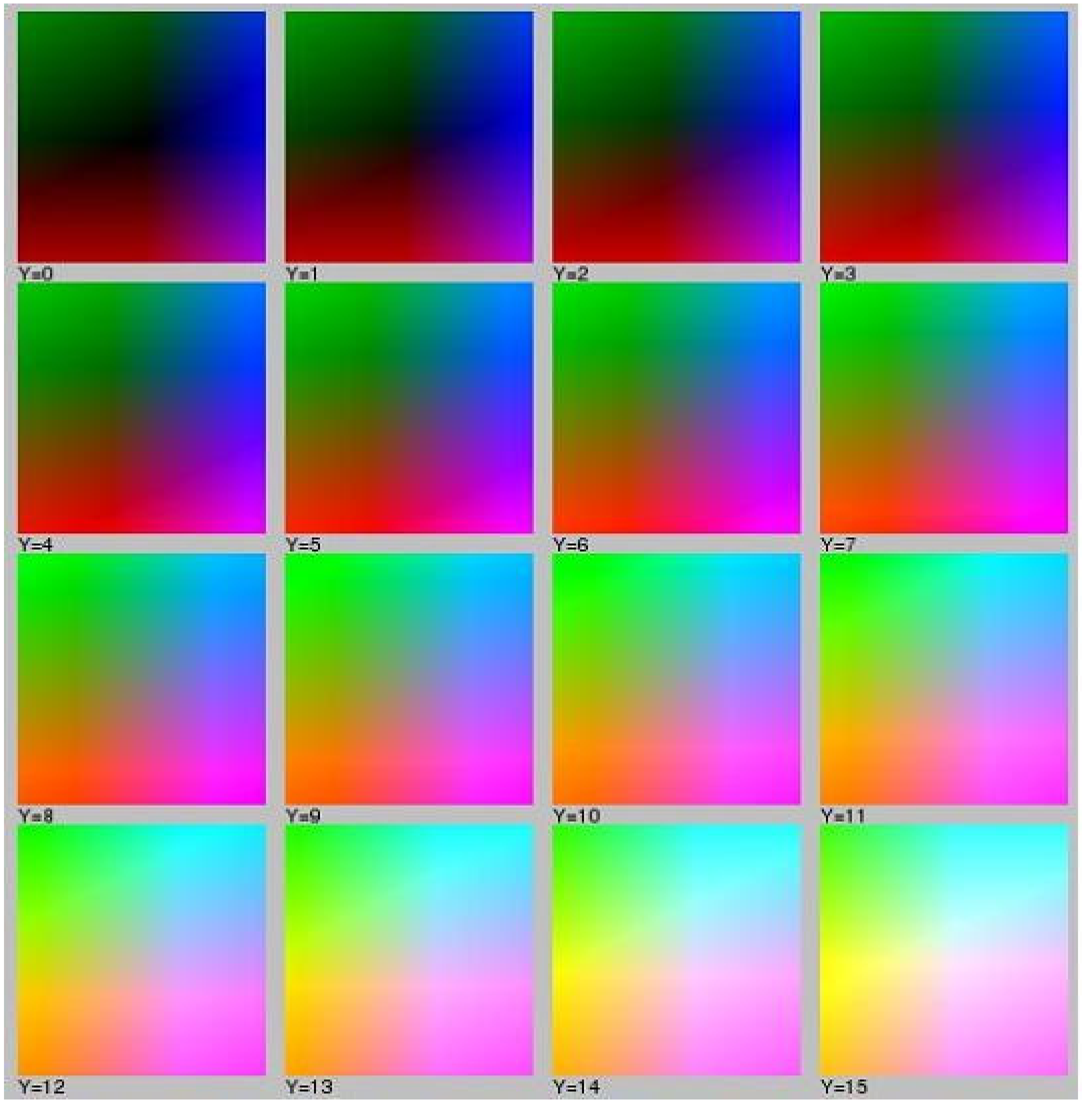
Color Spaces - YUV

- The AIBO camera provides images in YUV (or YCrCb) color space
    - Y – Luminance (brightness)
    - U/Cb – Blueness (Blue vs. Green)
    - V/Cr – Redness (Red vs. Green)
- Technically, YUV and YCrCb are slightly different, but this does not matter for our purposes
    - We will refer to the AIBO color space as YUV

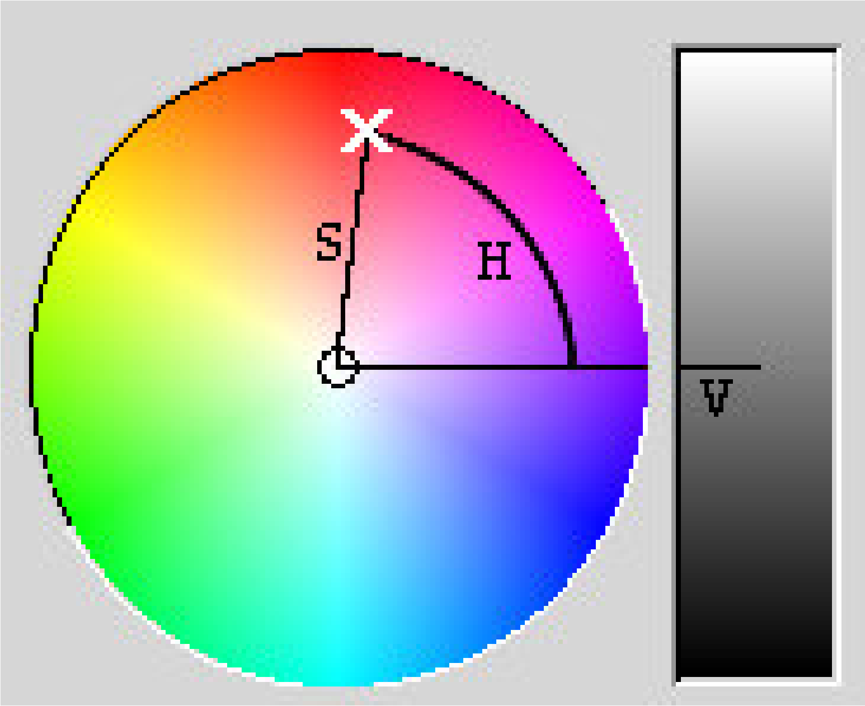
Color Spaces – YUV

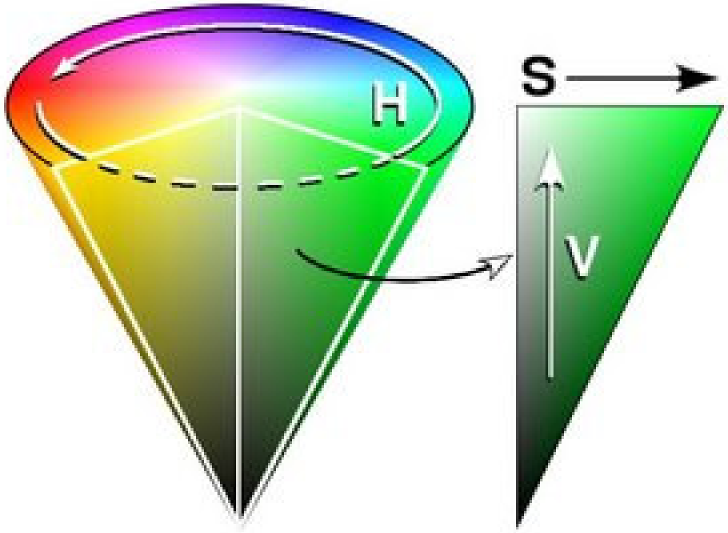


Color Spaces – YUV



Color Spaces – HSV





www.wordiq.com/definition/HSV_color_space

Color Spaces - Discussion

- RGB
  - Handled by most capture cards
  - Used by computer monitors
  - Not easily separable channels
- 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

Image RGB

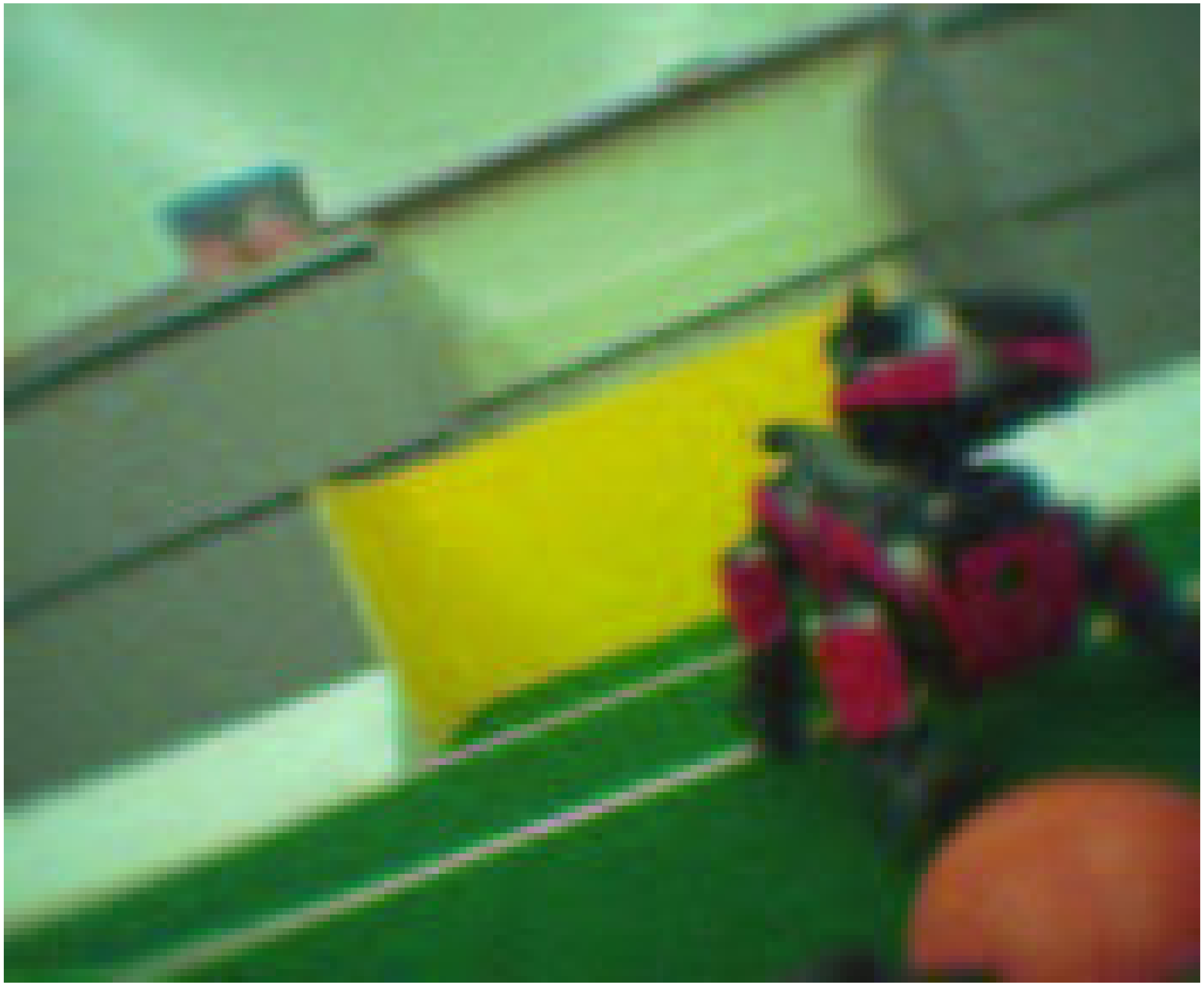
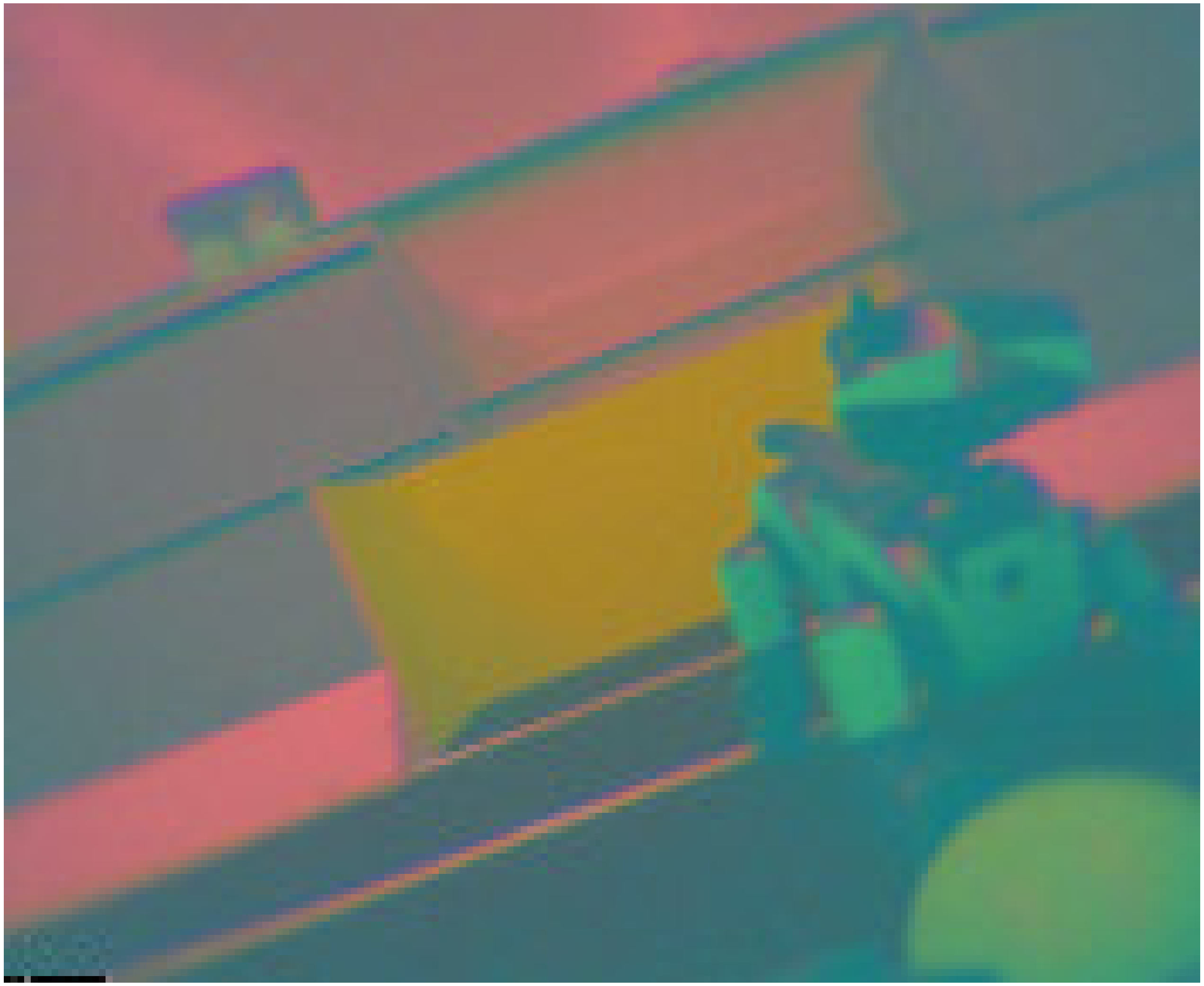
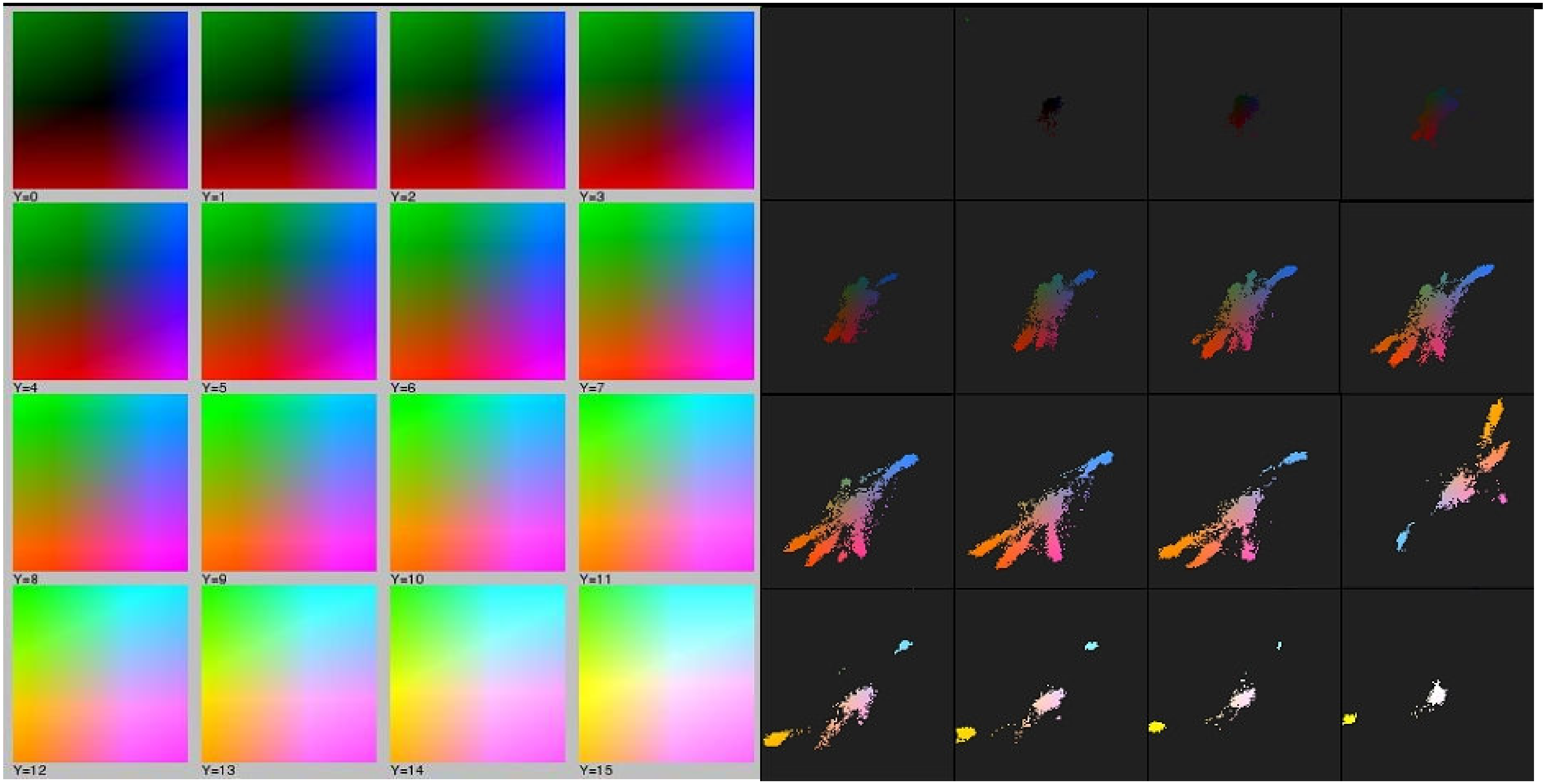


Image Raw



R=Y
G=U
B=V

YUV Histogram



Vision Overview

- CMRoboBits vision is divided into two parts
- Low level
  - Handles bottom-up processing of image
  - Provides *summaries* of image features
- High level
  - Performs top-down processing of image
  - Uses *object models* to filter low-level vision data
    - Identifies objects
    - Returns properties for those objects

Low-Level Vision Overview

- 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*

SCS Carnegie Mellon

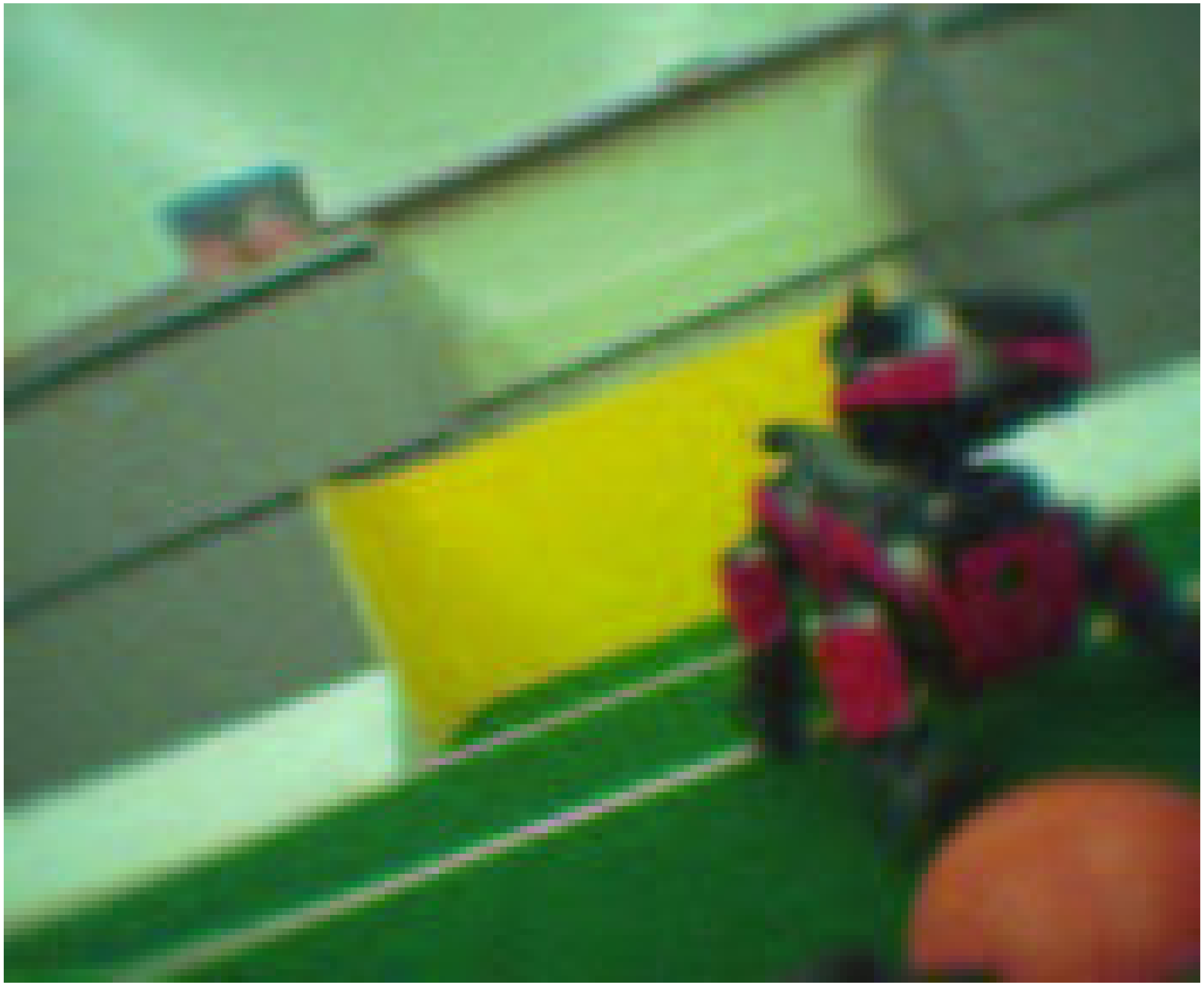
Color Segmentation

- 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 *C*

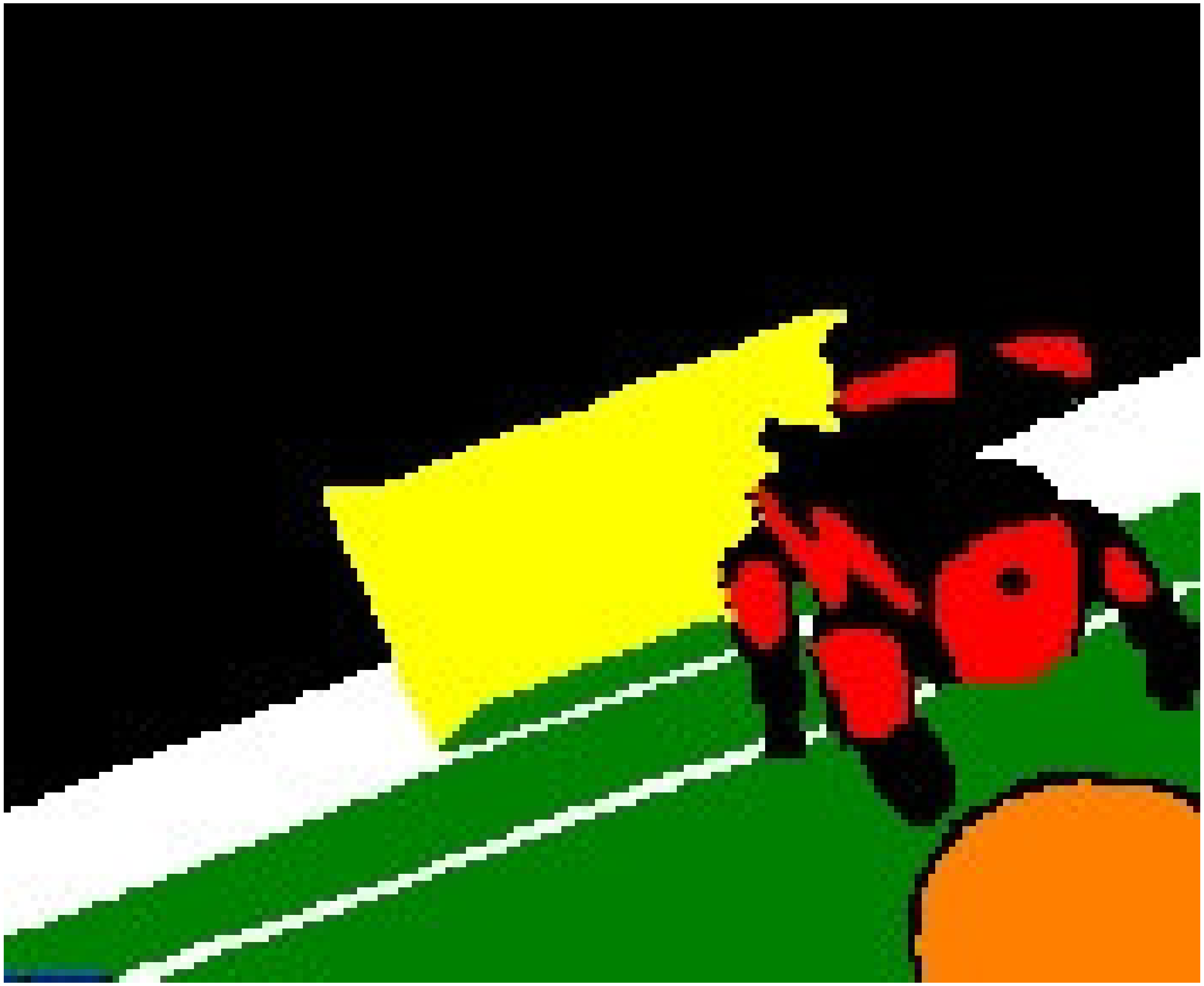
$$F : y, u, v \rightarrow c \in C$$

  - C includes ball, carpet, 2 goal colors, 1 additional marker color, 2 robot colors, walls/lines and unknown
- Reduces the amount of information per pixel roughly by 1.8M
  - Instead of a space of 256^3 values, we only have 9 values!

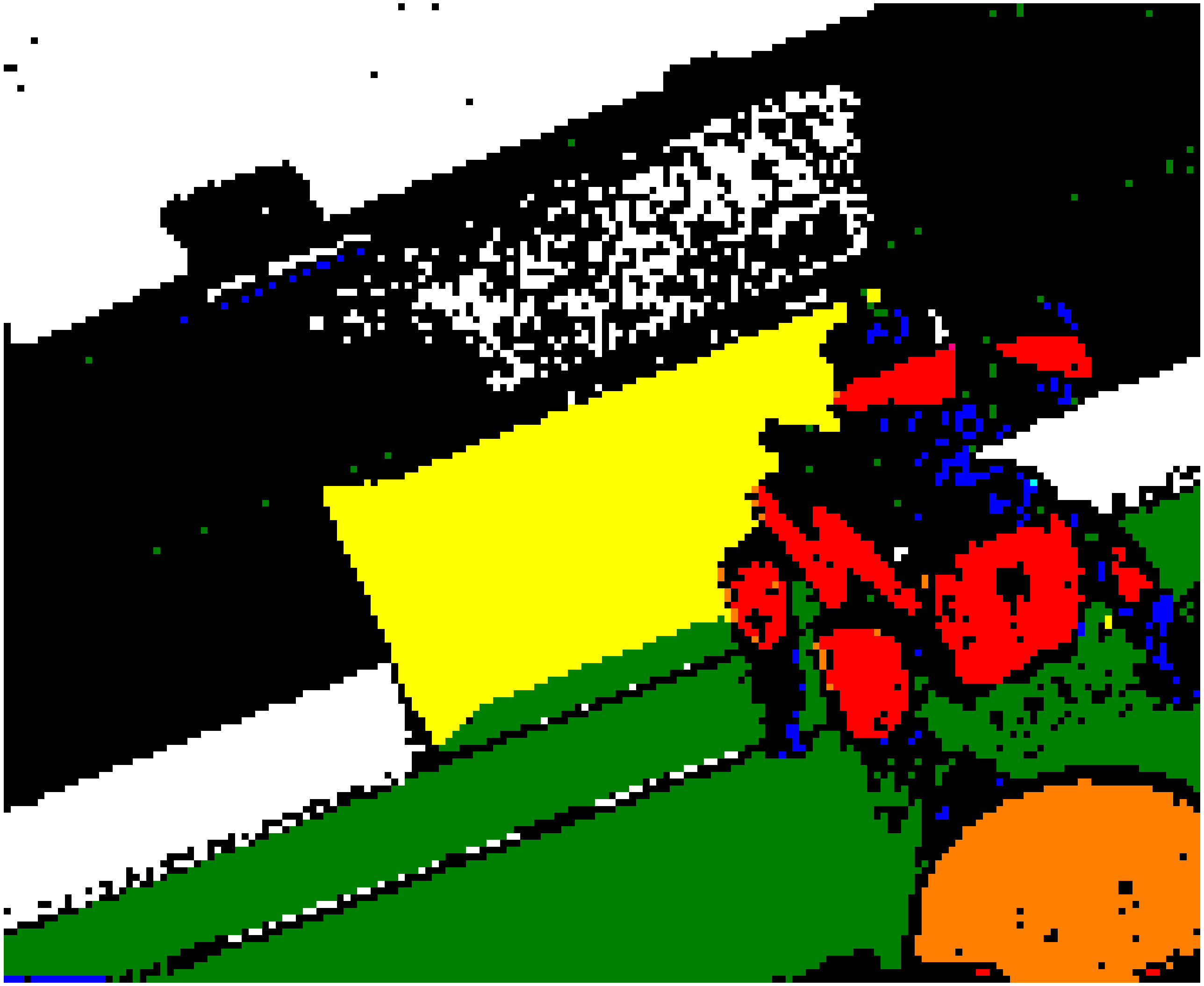
Before Segmentation



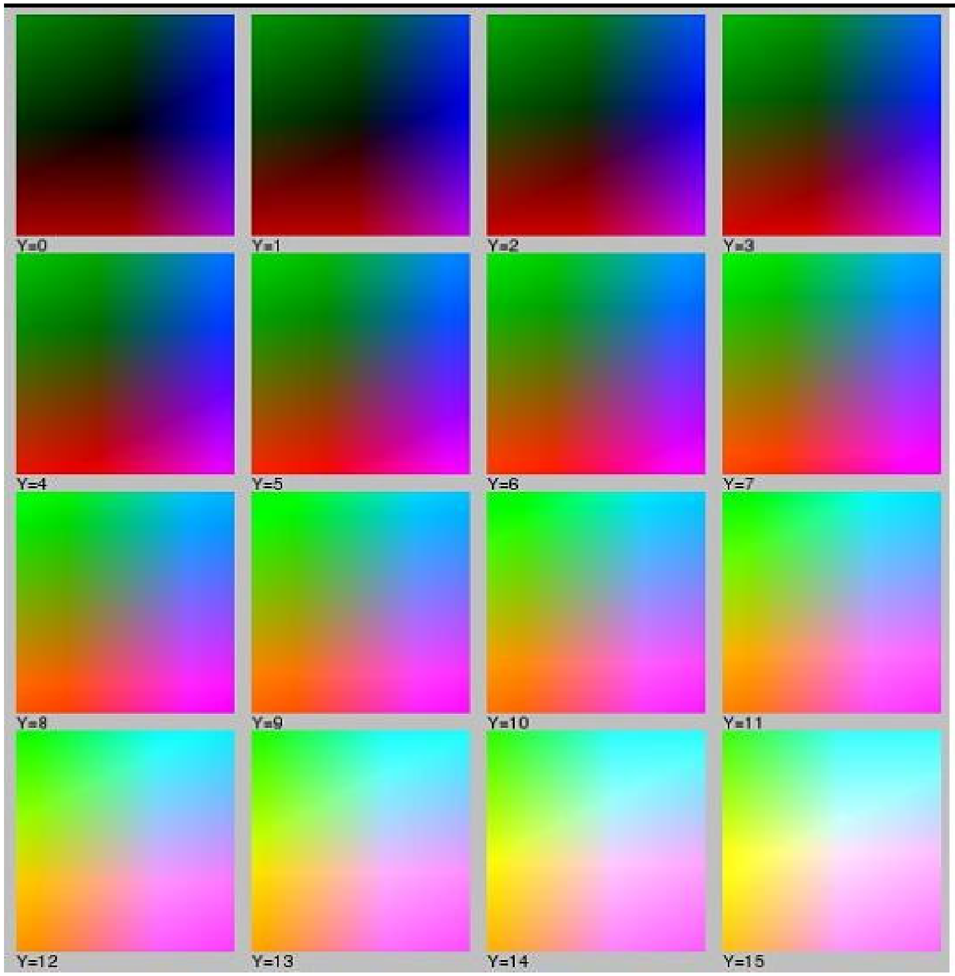
Ideal Segmentation

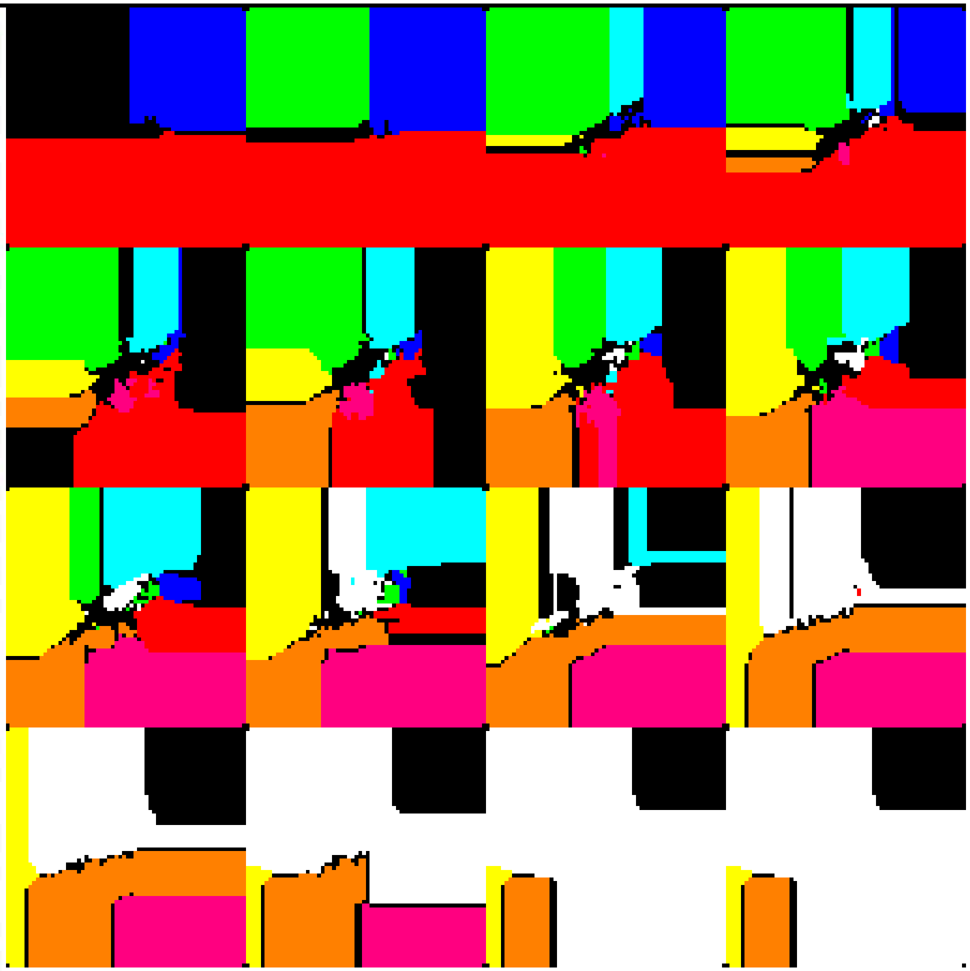


Result of Segmentation



Color Class Thresholds





Potential Problems with Color Segmentation