CISC 6525

Perception (Computer Vision)

Chapter 24

VM For Class

Download the virtual machine for Oracle virtualbox

http://erdos.dsm.fordham.edu/~lyons/ROSIndigo64Bits.ova

Google team drive: CISC 6525 Fall 2018

File: RosIndigo64Bits.ova

This is an Ubuntu 14.04 VM with some special software installed.

This has the ROS (Robot operating System), OpenCV (Computer Vision) and FF (a high performance symbolic planner) installed.

Outline

- Perception generally
- Image formation
- Early vision
- 2D → 3D
- Object recognition

Slides from R&N Chapter 24 or DML unless otherwise attributed

The Problem

Stimulus (percept) S, World W

$$S = g(W)$$

E.g., $g =$ "graphics." Can we do vision as inverse graphics?

$$W = g^{-1}(S)$$

Problem: massive ambiguity!

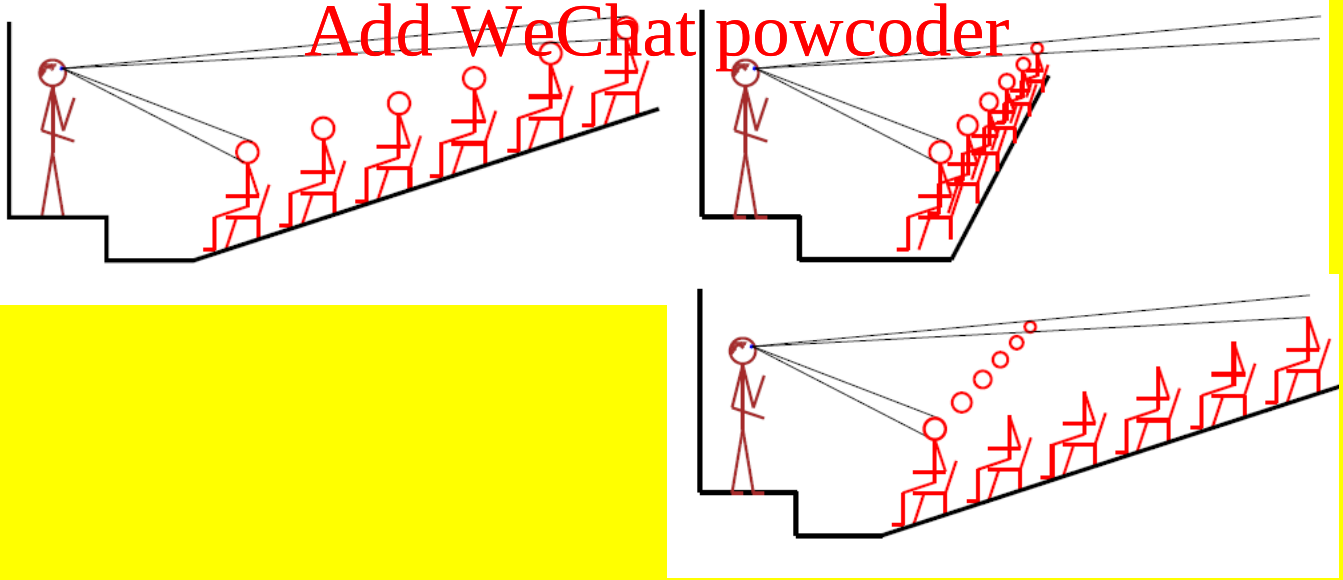
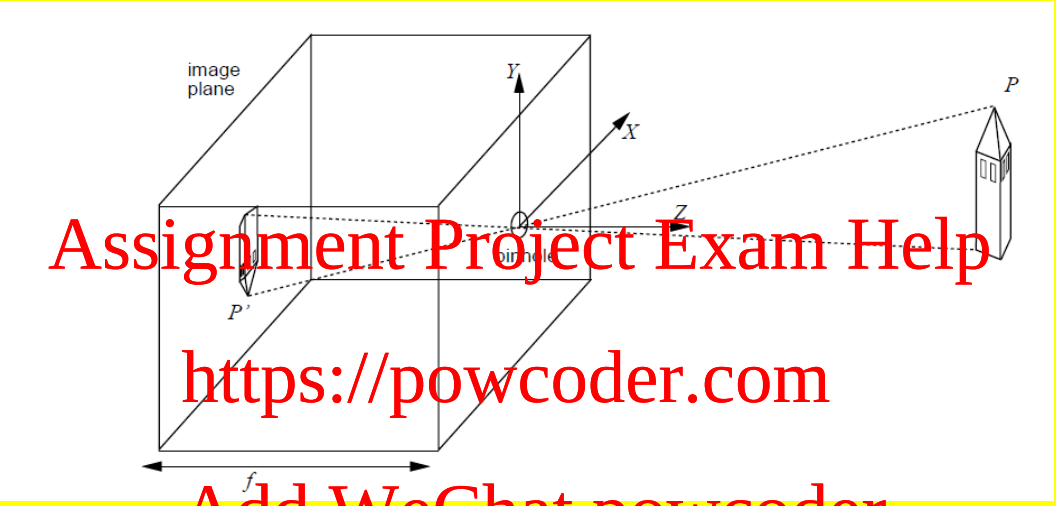


Image Formation

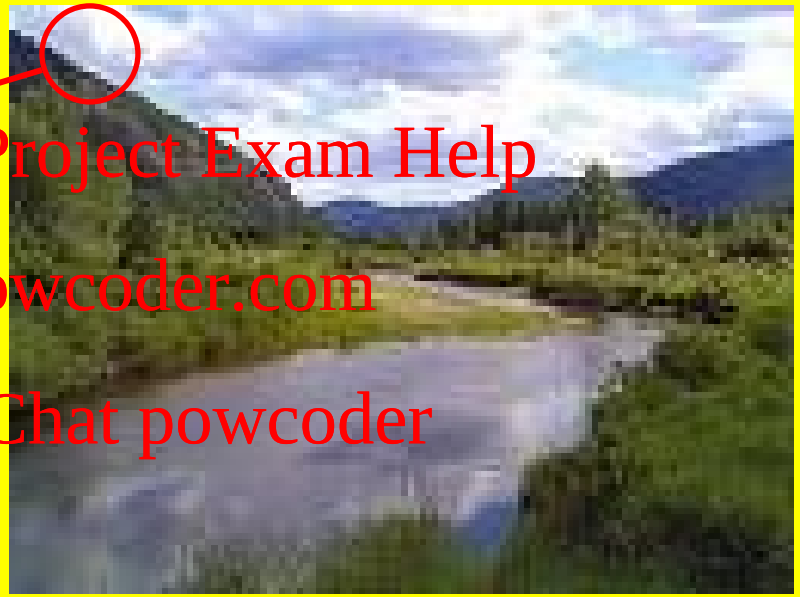


P is a point in the scene, with coordinates (X, Y, Z)
P' is its image on the image plane, with coordinates (x, y z)

$$x = \frac{-fX}{Z} \qquad y = \frac{-fY}{Z}$$

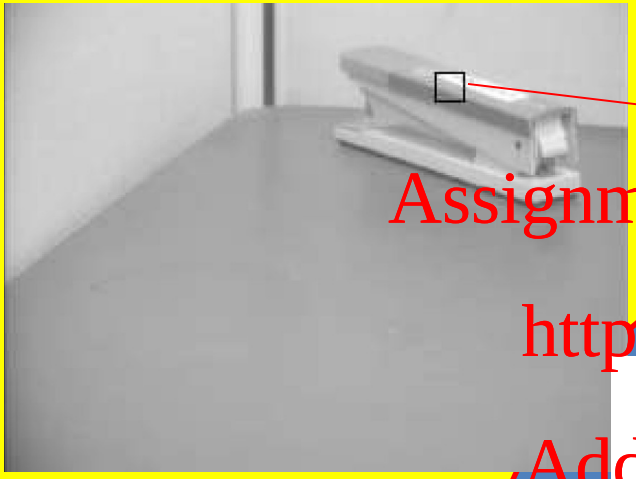
by similar triangles. Scale/distance is indeterminate!

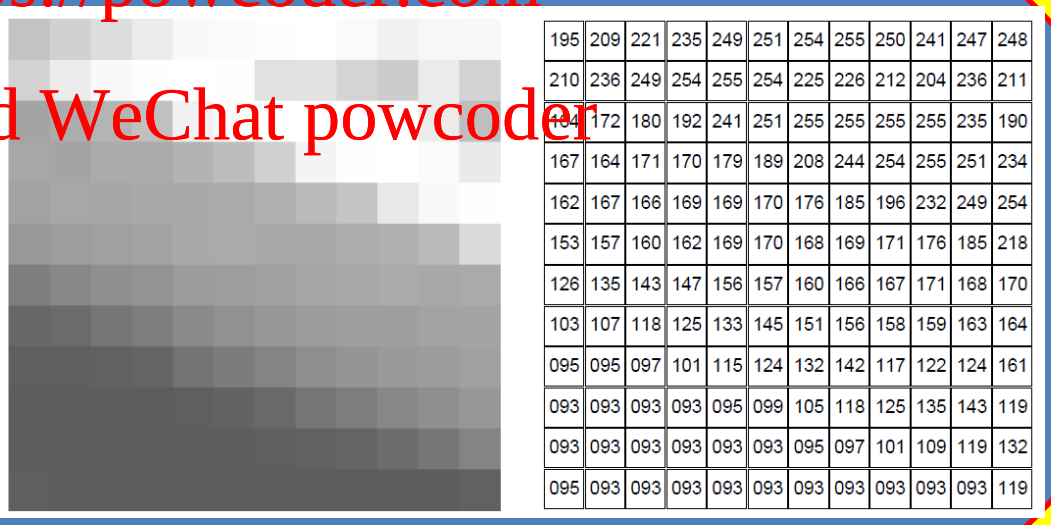
Images



Individual values are called **pixels** for picture elements.

Images





195	209	221	235	249	251	254	255	250	241	247	248
210	236	249	254	255	254	225	226	212	204	236	211
164	172	180	192	241	251	255	255	255	255	235	190
167	164	171	170	179	189	208	244	254	255	251	234
162	167	166	169	169	170	176	185	196	232	249	254
153	157	160	162	169	170	168	169	171	176	185	218
126	135	143	147	156	157	160	166	167	171	168	170
103	107	118	125	133	145	151	156	158	159	163	164
095	095	097	101	115	124	132	142	117	122	124	161
093	093	093	093	095	099	105	118	125	135	143	119
093	093	093	093	093	093	095	097	101	109	119	132
095	093	093	093	093	093	093	093	093	093	093	119

Images & Video

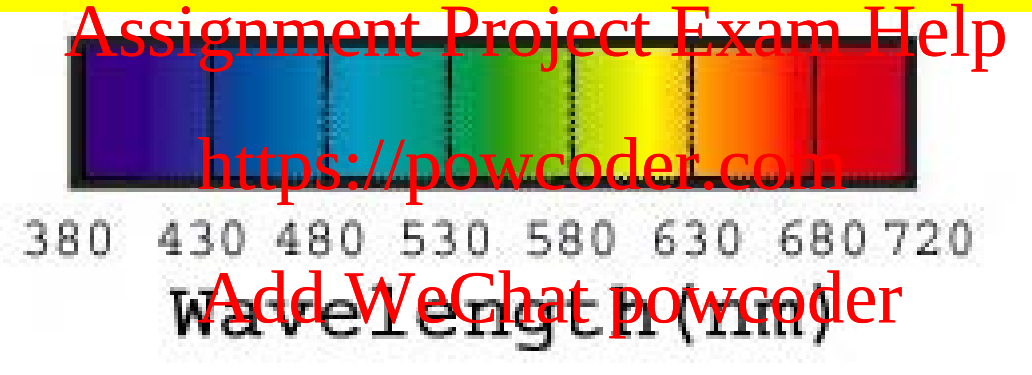
- *I(x, y, t)* is the intensity at *(x, y)* at time *t*

- CCD camera 4,000,000 pixels, 4Mpixel; human eyes 240,000,000; 240 Mpixels

- i.e., ~5 terabits/sec at 20hz = 20fps

What is color

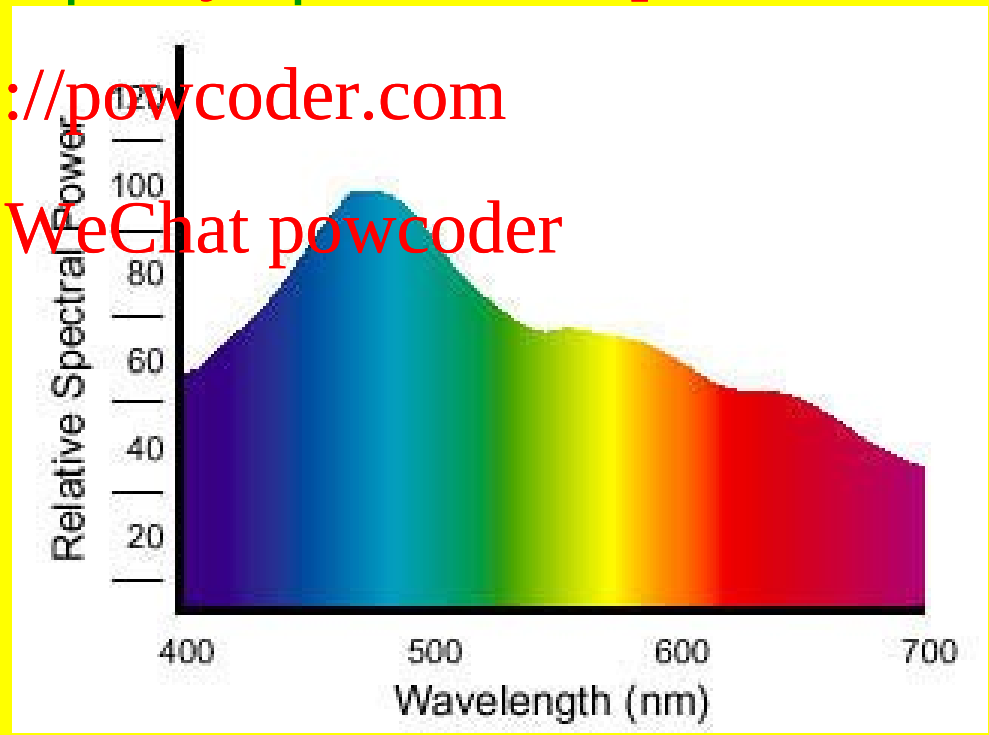
- Color is related to the wavelength of light



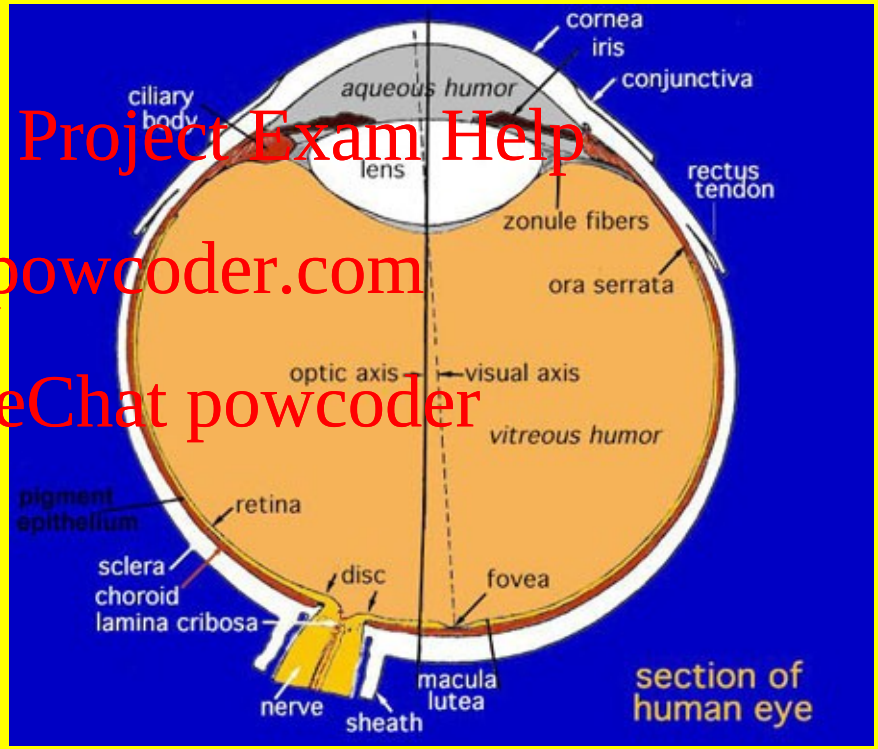
- The shorter wavelengths are perceived as blue and longer as red with green in between.

What is daylight

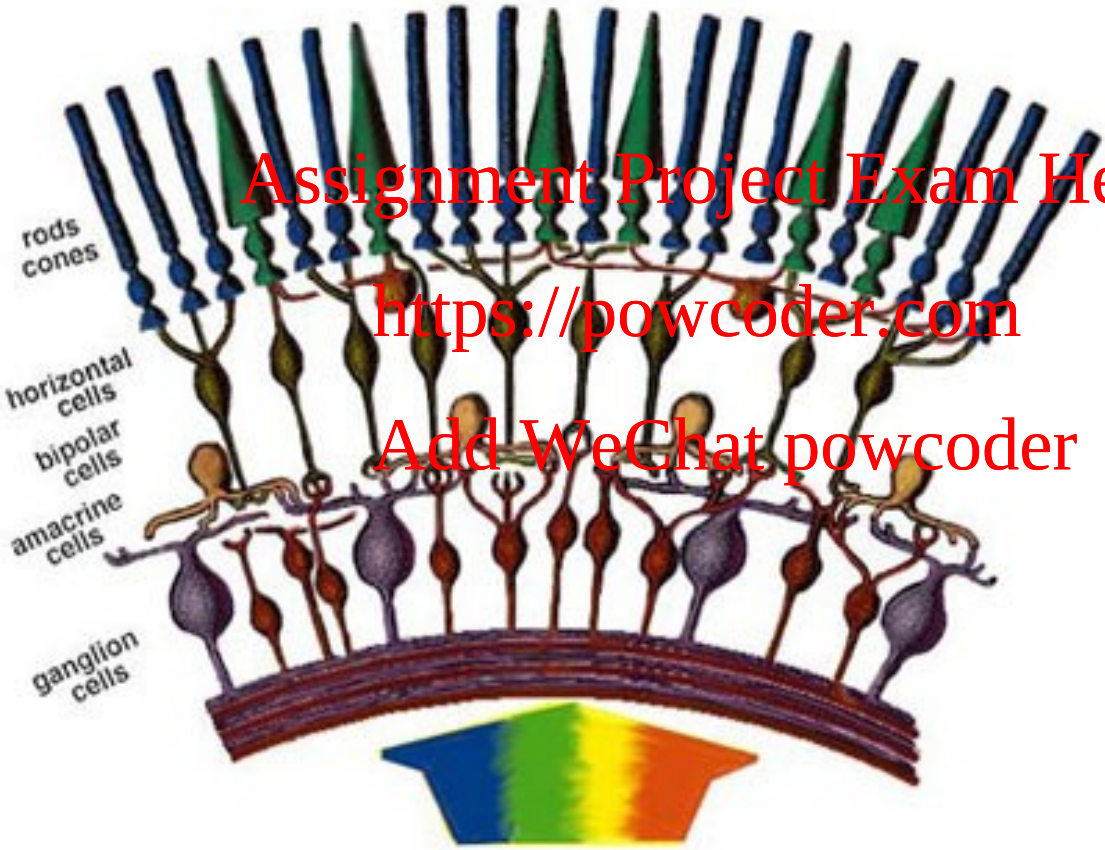
The intensity of light of each frequency that falls on the earth during day can be represented by the spectral power distribution graph.



From a subjective viewpoint

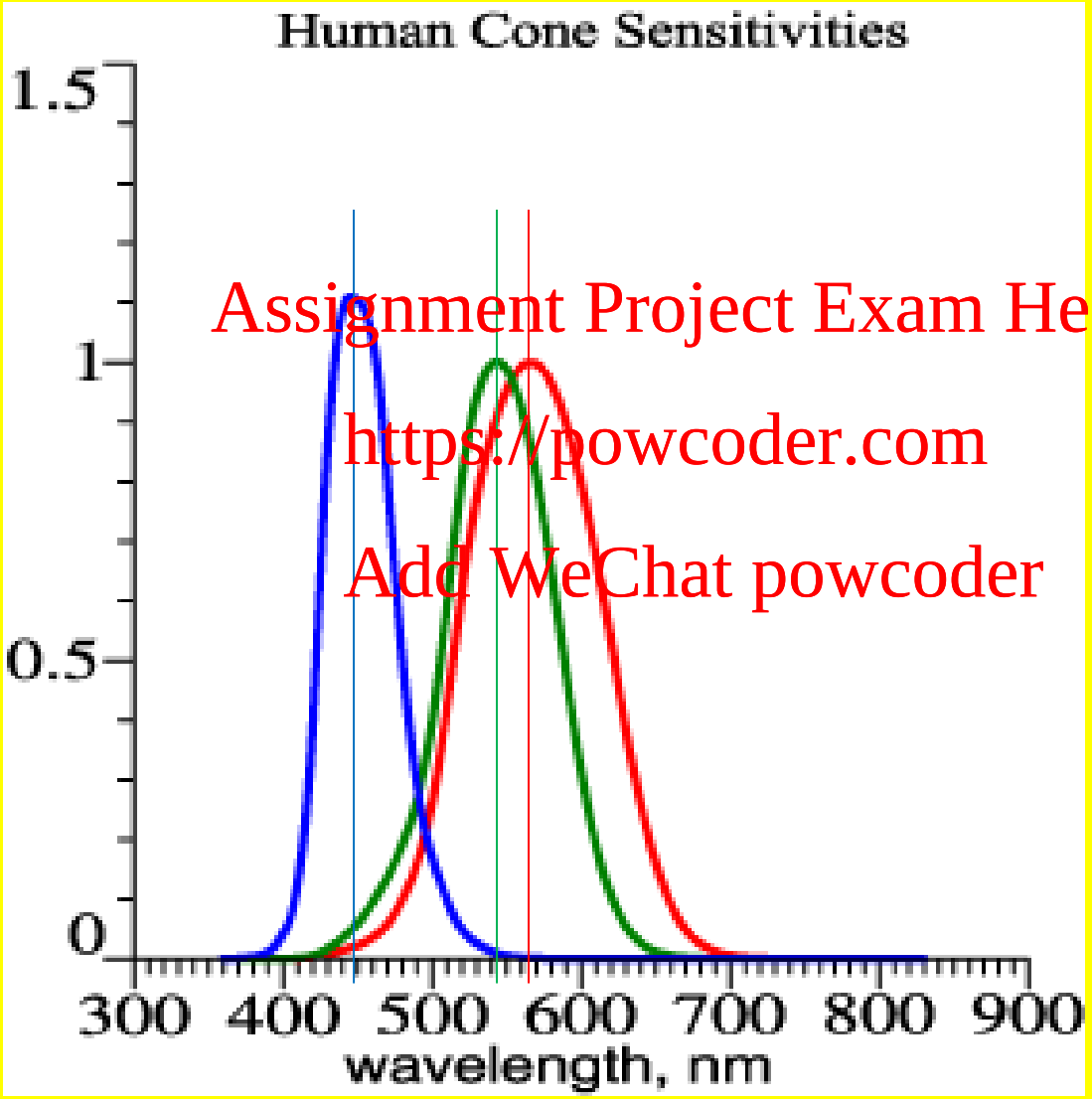


The Retina



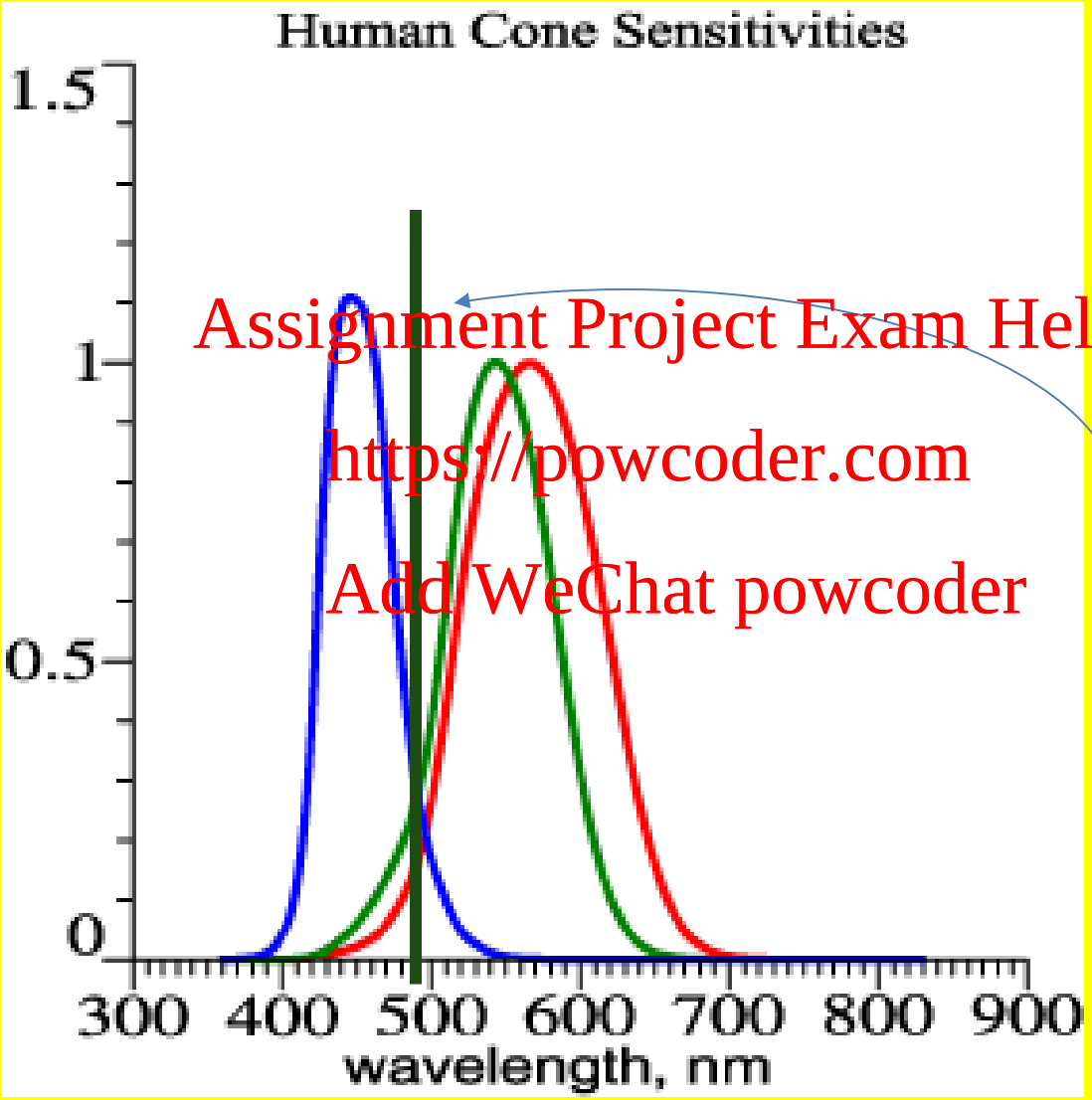
- Rods sense 'light intensity';
- Cones sense 'color'.
- Each cone has one of three pigments: red, green, or blue.

Color sensitivity of the 3 cones



The closer the wavelength to the target wavelength for that cone, the more active the cone cell becomes

How do we see all those colors!



Depending on how 'activated' each of the types of cones are, We see a different **Color** = wavelength of light

E.g.:

10% Blue
30% red
60% Green

= approx. light
Of 500nm

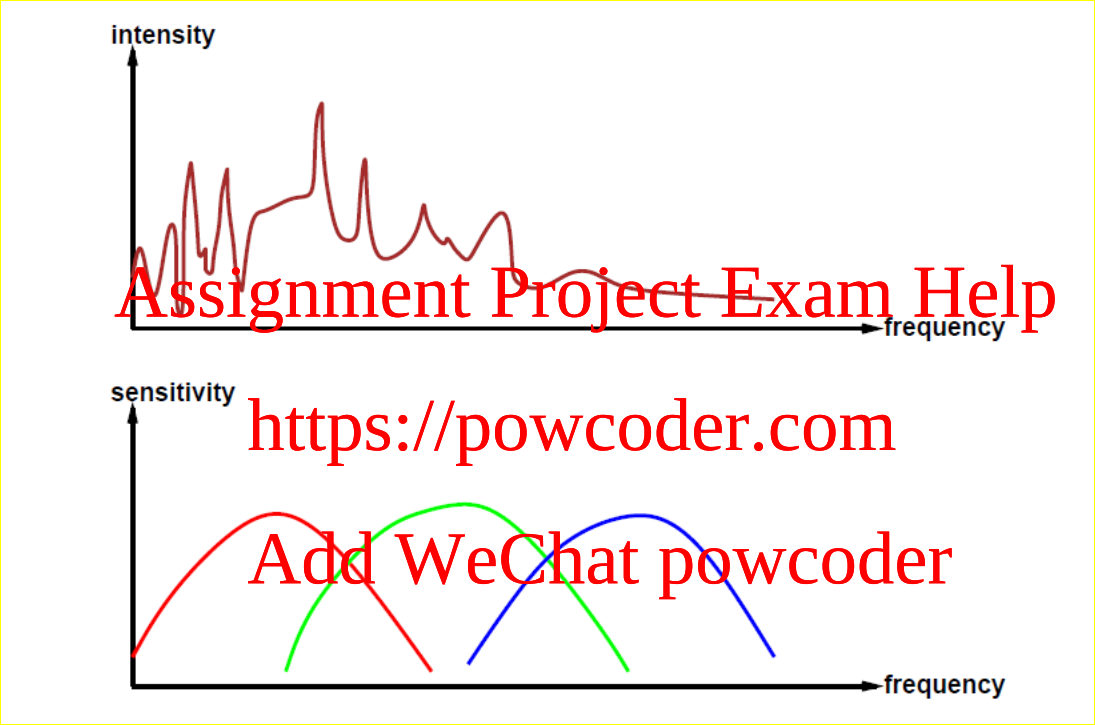
The Tristimulus Theory

- This is the theory that any color can be specified by giving just three values.

- We call Red, Green and Blue, the additive primary colors.

- We can define a given color by saying how much red, green and blue light we need to add to get that color

Color - Summary



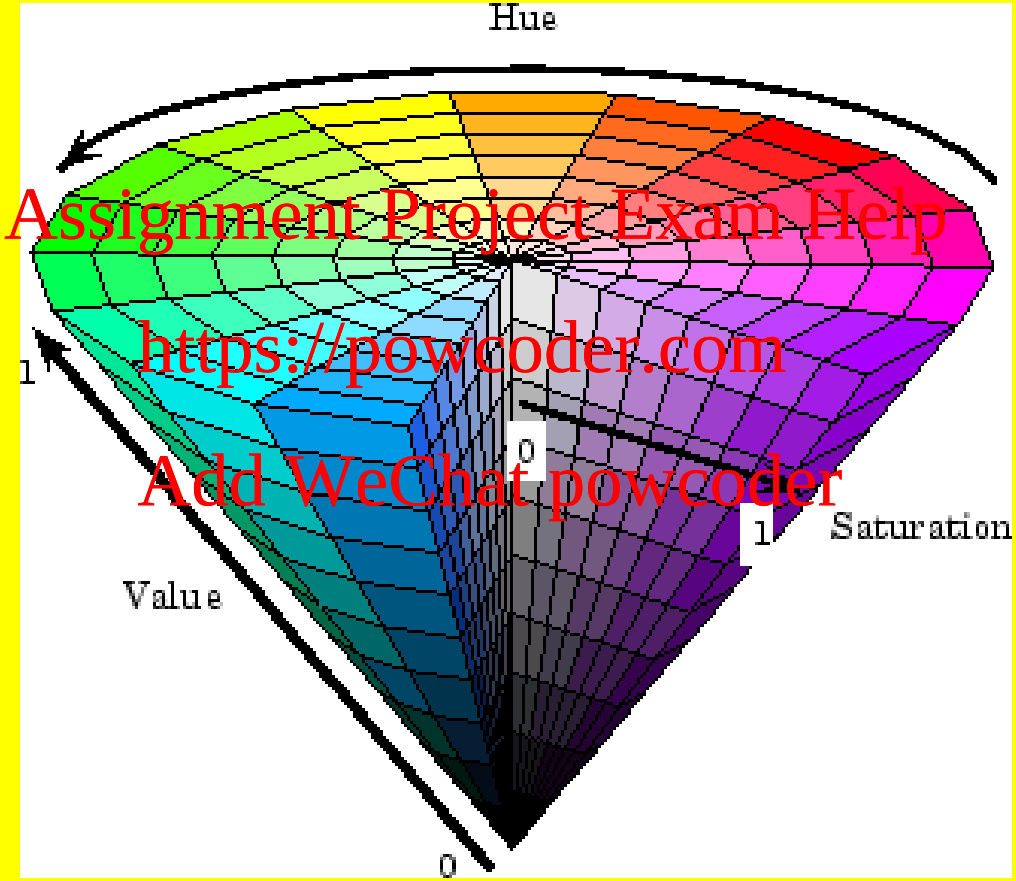
- Intensity varies with frequency – infinite dimensional signal

- Human eye has three types of color-sensitive cells;
  each integrates the signal => 3-element vector intensity

HSV

- Alternative way of specifing color
- *Hue* (roughly, dominant wavelength)
- *Saturation* (purity)
- *Value* (brightness)
- Model HSV as a ‘cylinder’: *H* angle, *S* distance from axis, *V* distance along axis
- Basis of popular style of *color picker*

HSV Color Cone



Why is it
not a cylinder?

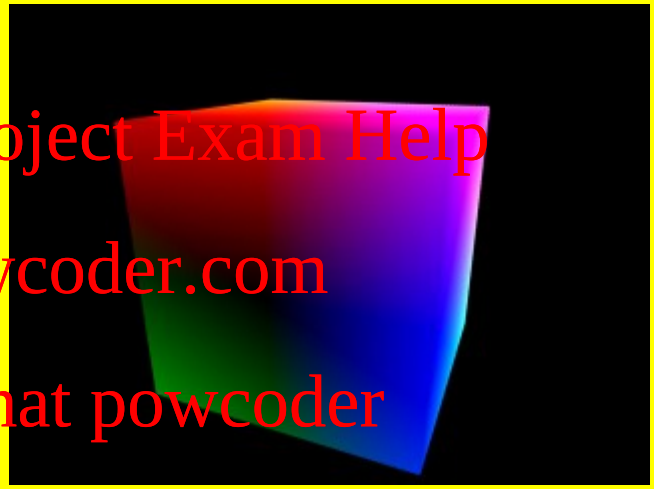
YUV

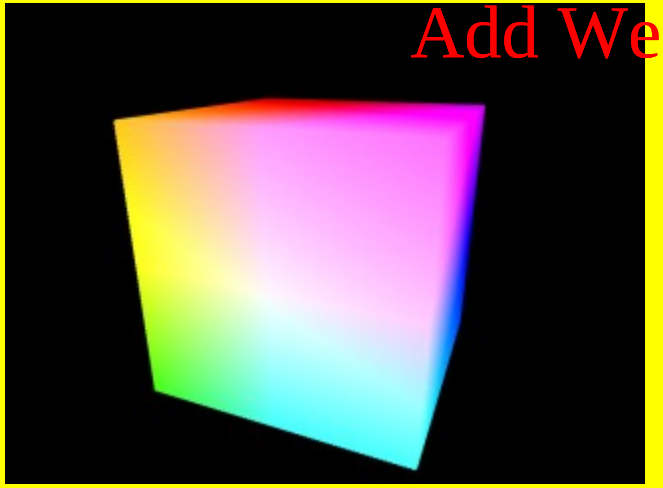
- However, Y not simply related to *R*, *G* and *B* because eye is more sensitive to some colors

$$Y = R * .299000 + G * .587000 + B * .114000$$
$$U = R * -.168736 + G * -.331264 + B * .500000 + 128$$
$$V = R * .500000 + G * -.418688 + B * -.081312 + 128$$

- Digital TV uses $Y'C_BC_R$ not YUV (different weights).

YUV Color Cube
two perspectives



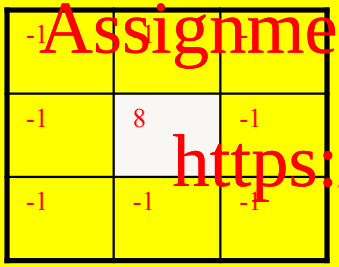


Pixel Group Processing

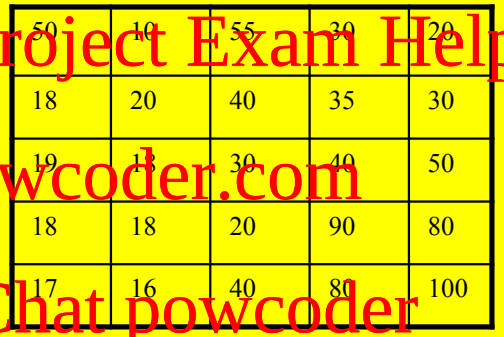
- Compute new value for pixel from its old value and the values of surrounding pixels
  - *Filtering operations*
- Compute weighted average of pixel values
  - Array of weights known -- *convolution mask*
  - Pixels used in convolution -- *convolution kernel*
- Computationally intensive process

Pixel processing

Convolution kernel



Image



Kernel applied left to right,
top to bottom

$$E_\theta(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_\theta(u,v) I(x+u, y+v)\, du\, dv$$

Blurring

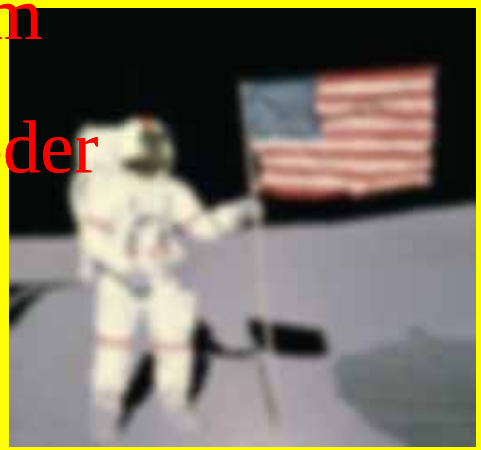
- Classic simple blur
  - Convolution mask with equal weights
    - Unnatural effect
  - Gaussian blur
    - Convolution mask with coefficients falling off gradually (Gaussian bell curve)
      - More gentle, can set amount and radius

Gaussian Blur Filter

No blur

small radius

larger radius



Sharpening

- Low frequency filter
  - 3x3 convolution mask coefficients all equal to -1, except centre = 9
  - Produces harsh edges
- Unsharp masking
  - Copy image, apply Gaussian blur to copy, subtract it from original
    - Enhances image features

Sharpening Filter

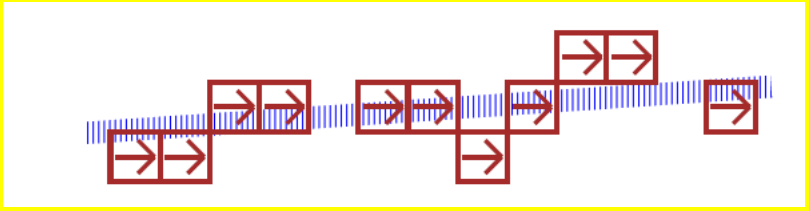
Edge Detection

Convolve image with spatially oriented filters (possibly multi-scale)

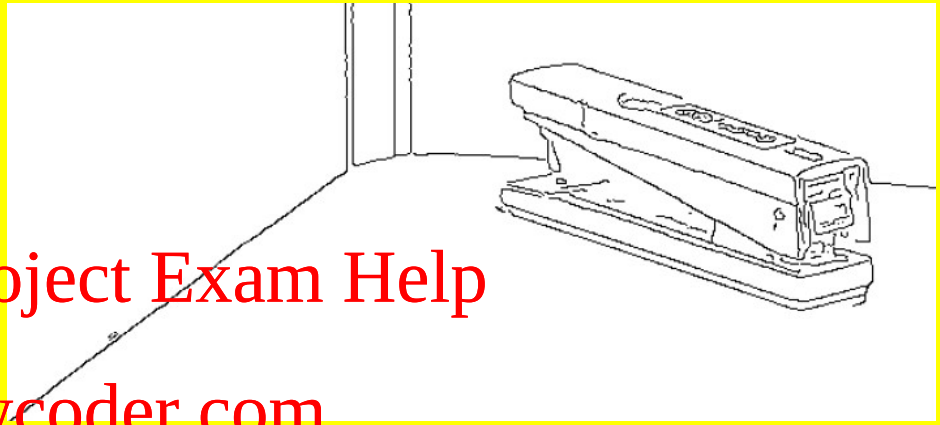
$$E_\theta(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_\theta(u,v) I(x+u, y+v)\, du\, dv$$

Label above-threshold pixels with edge orientation

Infer “clean" line segments by combining edge pixels with same orientation



Edge Detection



Edges in image come from $I(x,y,t)$ discontinuities in scene.

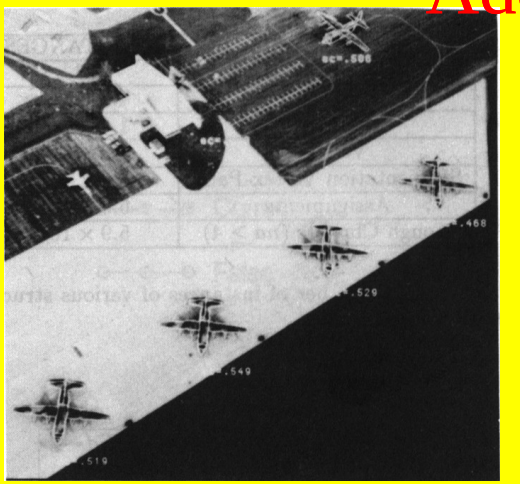
These can be due to:

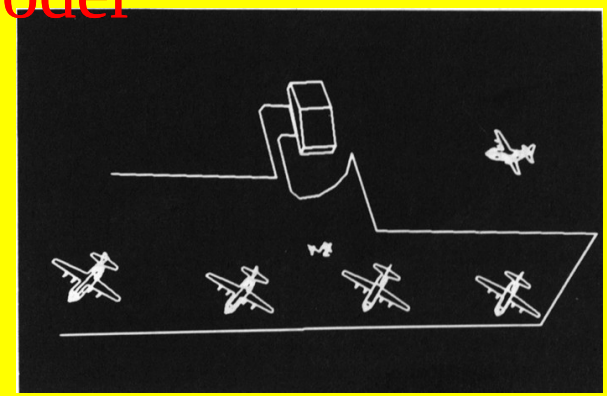
1) depth
2) surface orientation
3) reflectance (surface markings)
4) illumination (shadows, etc.)

Laplacian Edges

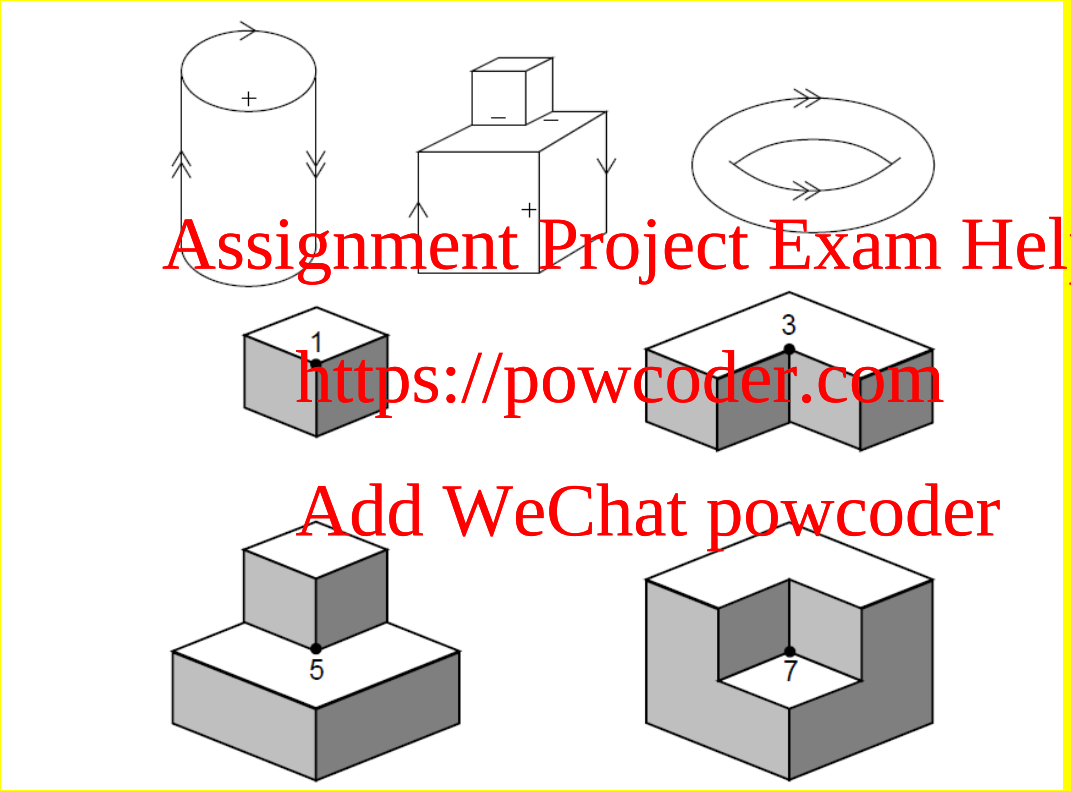
It calculates the Laplacian of the image given by the relation, $\Delta src = \frac{\partial^2 src}{\partial x^2} + \frac{\partial^2 src}{\partial y^2}$ where each derivative is found using Sobel derivatives. If `ksize = 1`, then following kernel is used for filtering:

$$kernel = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



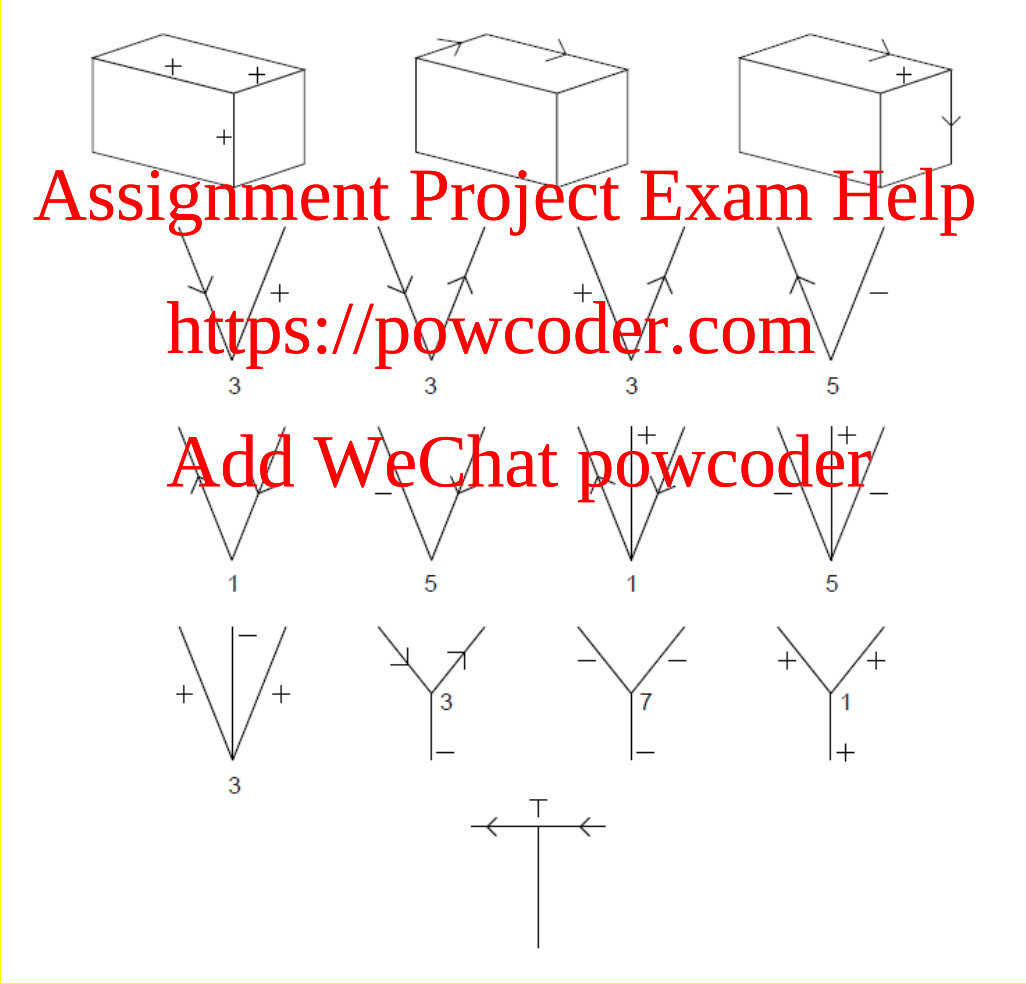


Reconstructing based on edges

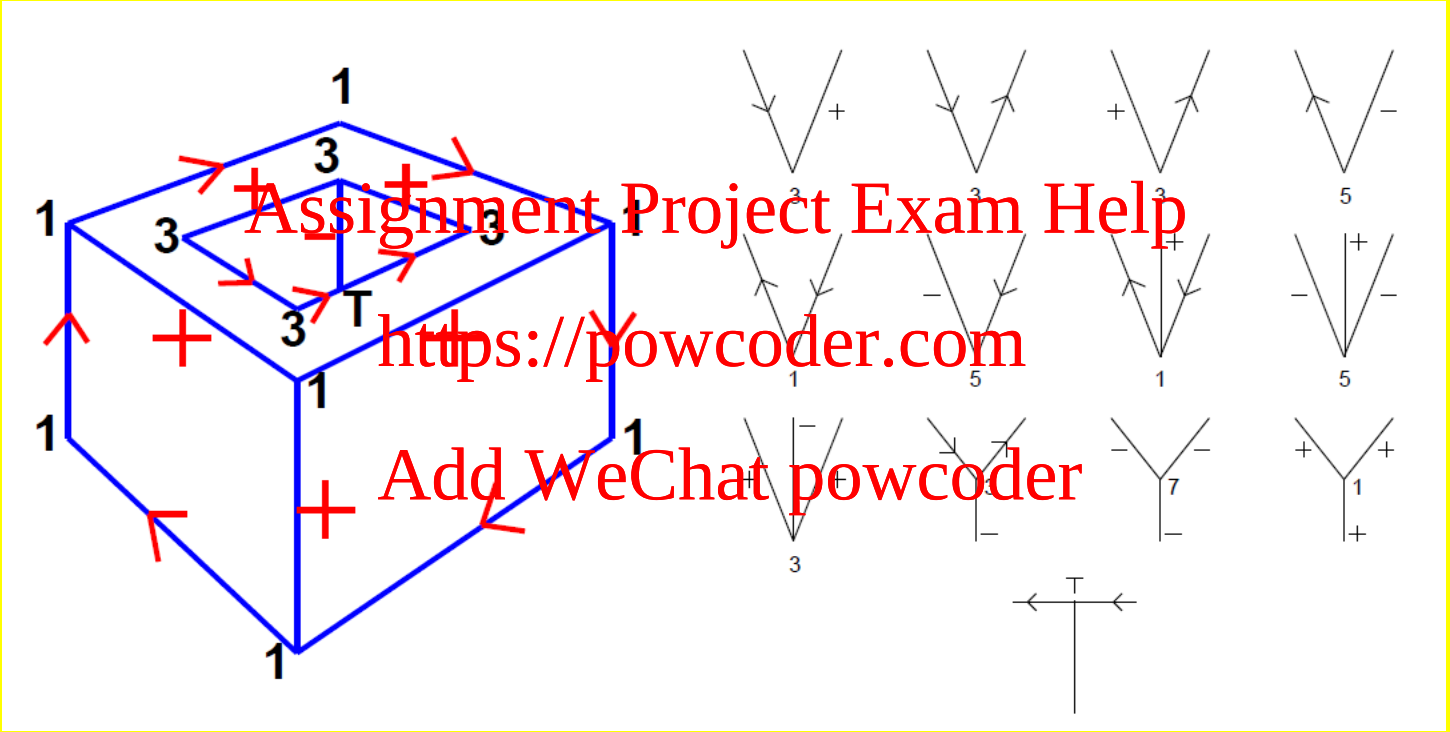


Solid polygons with trihedral edges

Trihedral Edges



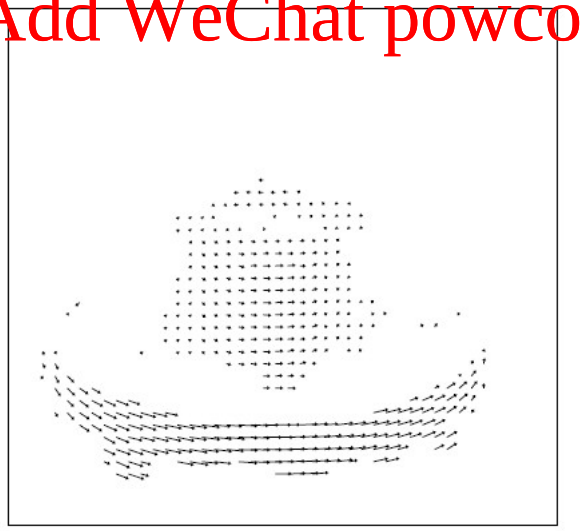
Vertex/Edge Labeling Example



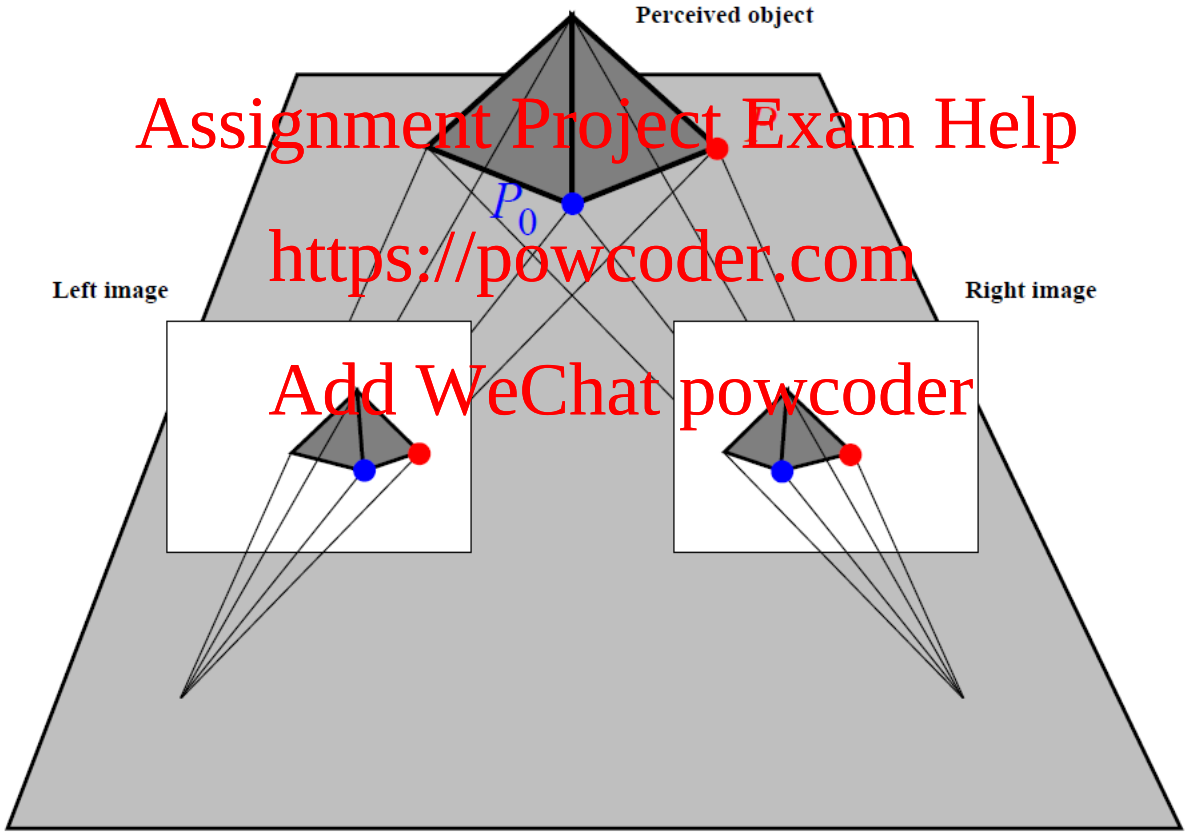
Cues from Prior Knowledge ("Shape from X")

Shape from	Assumes
motion	rigid bodies, continuous motion
stereo	solid, contiguous, non-repeating bodies
texture	uniform texture
shading	uniform reflectance
contour	minimum curvature

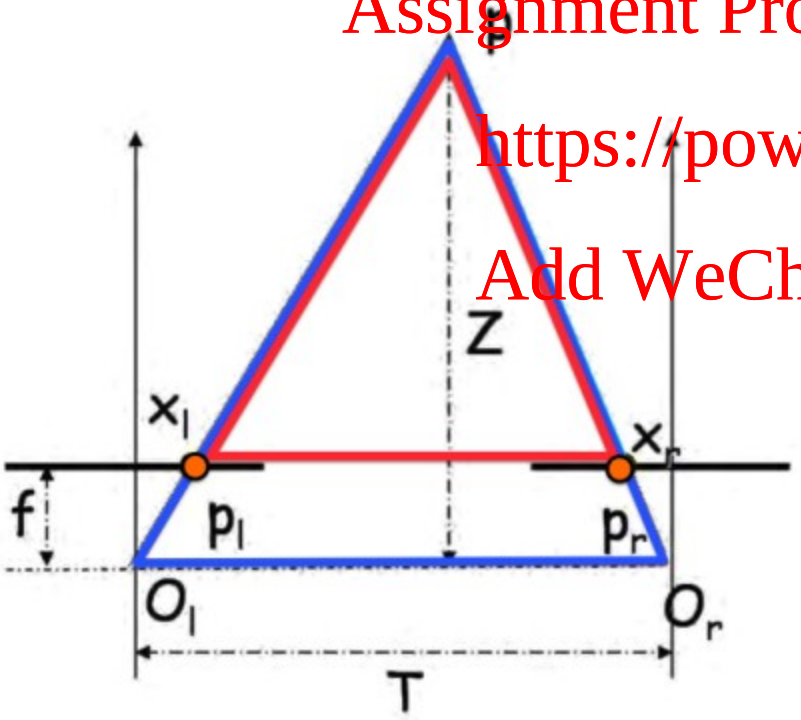
Shape from Motion



Stereo



Stereo Depth Calculation



Similar triangles:

$$\frac{T}{Z} = \frac{T + x_r - x_l}{Z - f}$$

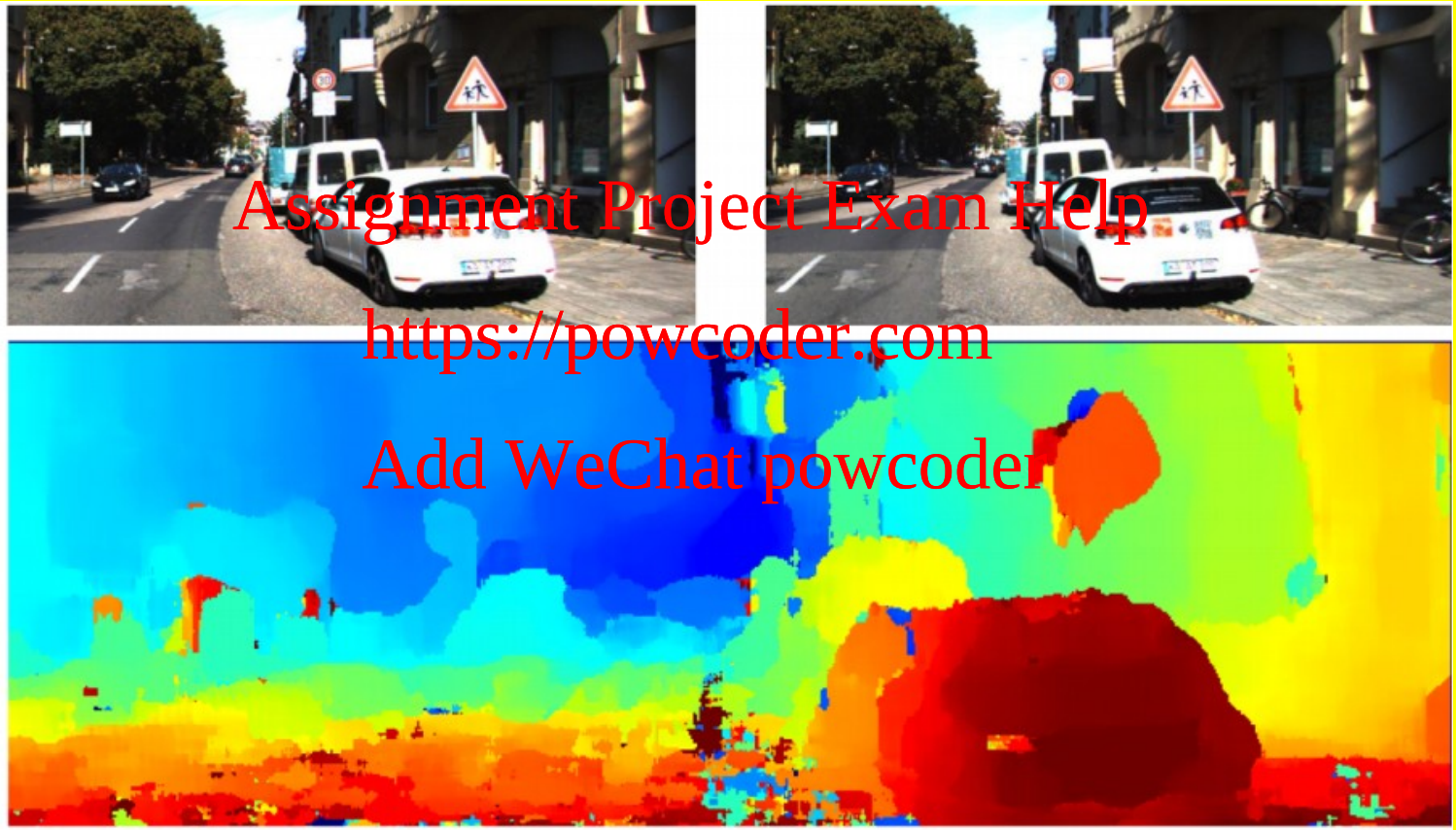
$$Z = \frac{f \cdot T}{x_l - x_r}$$

baseline

focal length

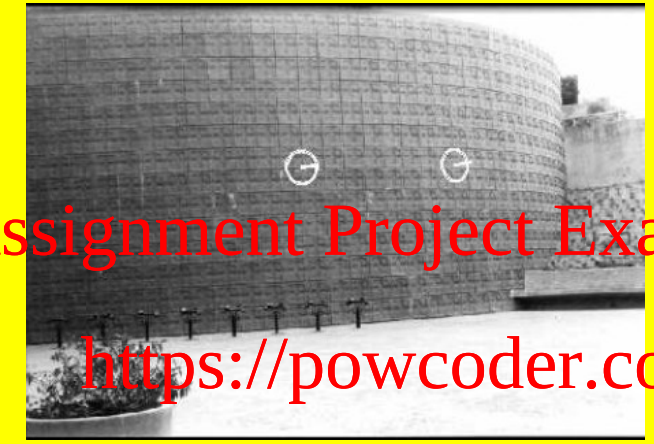
disparity

Example Stereo Disparity



Result: **Disparity map**
(red values large disp., blue small disp.)

Shape from Texture



Idea: assume actual texture is uniform, compute surface shape that would produce this distortion

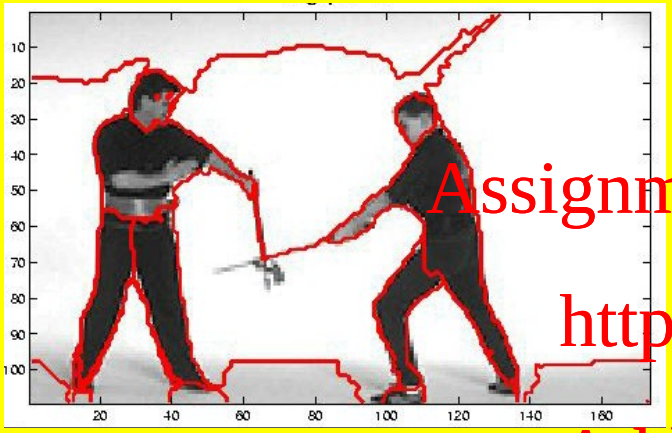
Similar idea works for shading – assume uniform reflectance, etc.

But inter-reflections give nonlocal computation of perceived intensity
=> hollows seem shallower than they really are

Shape from Optical Flow

Optical flow describes the direction and speed of motion of features in the image.

Segmentation of Images



- Which image components “belong together”?
- Belong together=lie on the same object
- Cues
  - similar color
  - similar texture
  - not separated by contour
  - form a suggestive shape when assembled

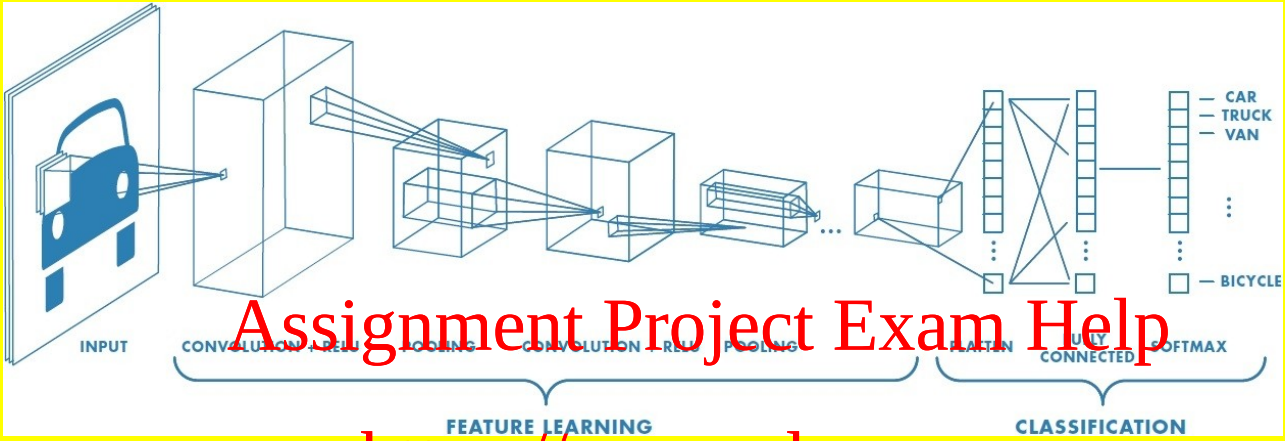
Object Recognition

- **Simple idea:**
  - extract 3-D shapes from image
  - match against “shape library"
- **Problems:**
  - extracting curved surfaces from image
  - representing shape of extracted object
  - representing shape and variability of library object classes
  - improper segmentation, occlusion
  - unknown illumination, shadows, markings, noise, complexity, etc.
- **Approaches:**
  - index into library by measuring invariant properties of objects
  - alignment of image feature with projected library object feature
  - match image against multiple stored views (aspects) of library object
  - machine learning methods based on image statistics

ImageNet

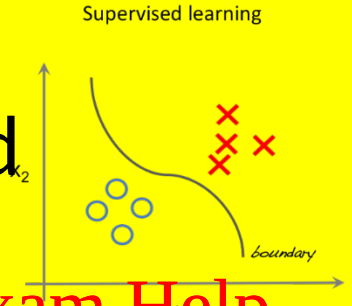
- 2012, 1.3 million hand labelled images
- 1000 classes (e.g., 120 dog classes)

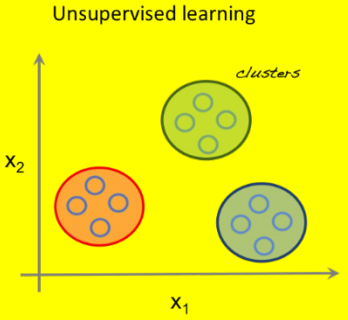
Deep Learning for Image Classification



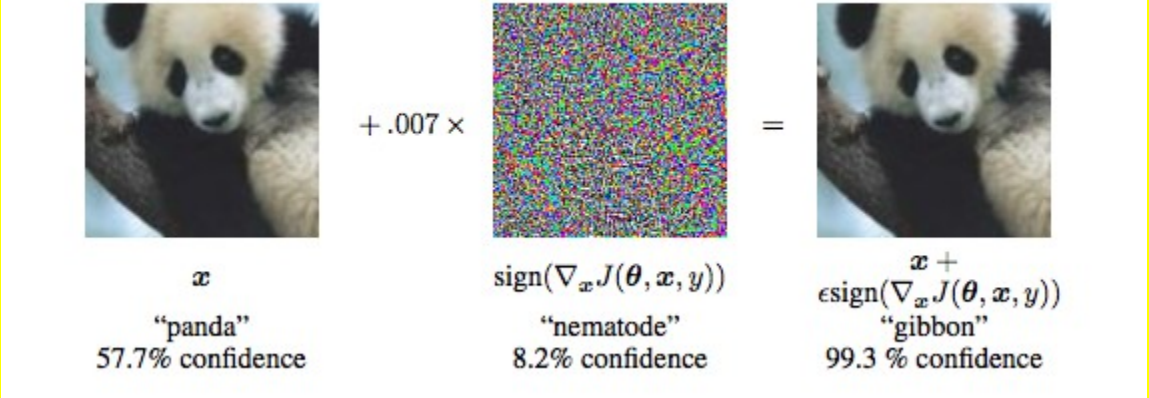
- Regular NN don’t scale up to image size well
- AlexNet 2012, 50% red. in ImageNet error rate.
- ResNet 2015, performs exceeds human

Deep Issues





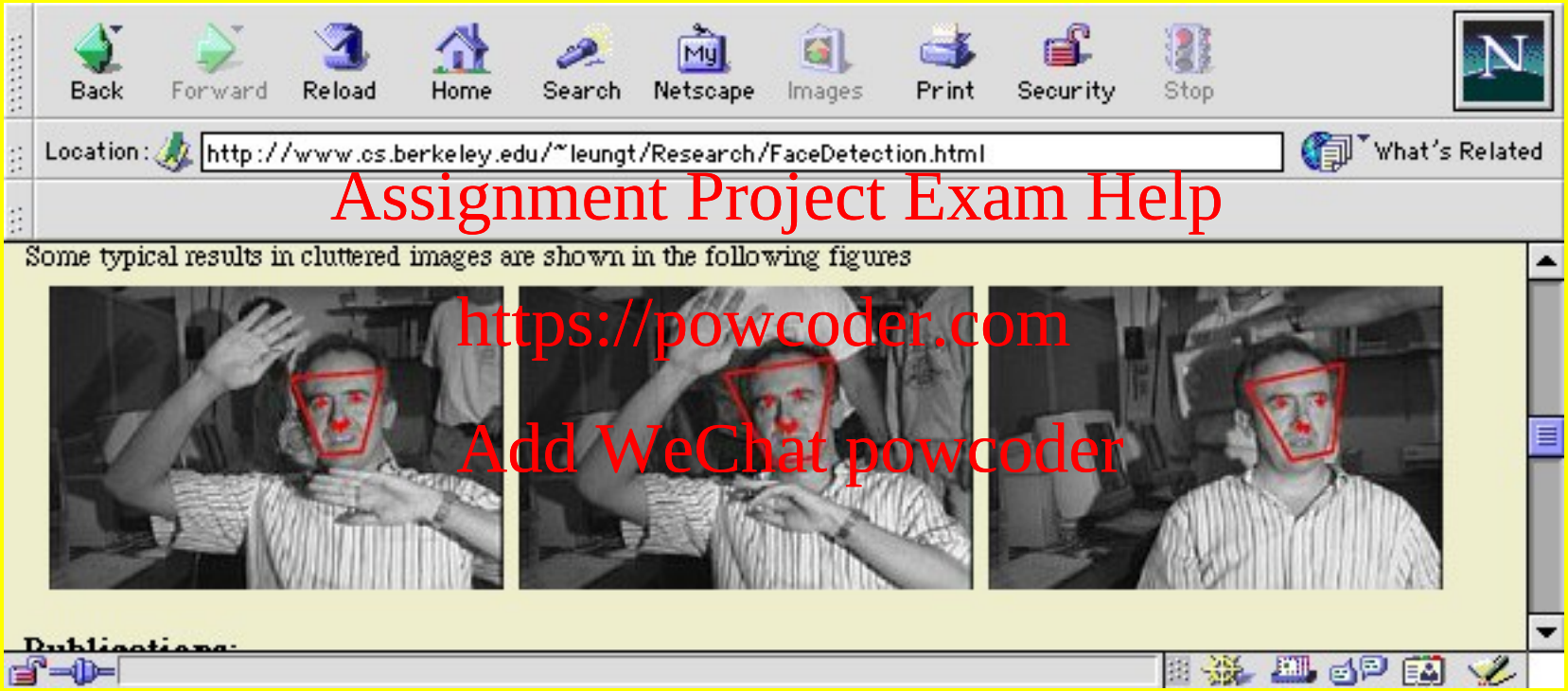
- Supervised vs. Unsupervised
- Transfer training
- Computational requirements, GPUs
- Adversarial images:



Matching templates

- Some objects are 2D patterns
  - e.g. faces
- Find faces by
  - finding eyes, nose, mouth
  - finding assembly of the three that has the “right” relations
- Build an explicit pattern matcher
  - discount changes in illumination by using a parametric model
  - changes in background are hard
  - changes in pose are hard

http://www.ri.cmu.edu/projects/project_271.html



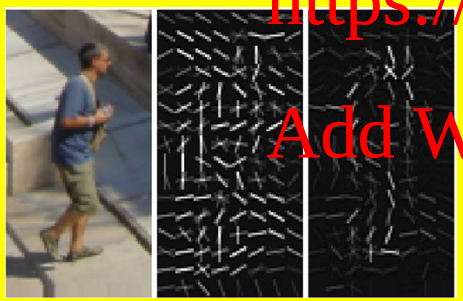
http://www.ri.cmu.edu/projects/project_320.html

People

- Skin is characteristic; clothing hard to segment
  - hence, people wearing little clothing
- Finding body segments:
  - finding skin-like (color, texture) regions that have nearly straight, nearly parallel boundaries
- Grouping process constructed by hand, tuned by hand using small dataset.
- When a sufficiently large group is found, assert a person is present

Action recognition from still images

- Description of the human pose
  - Silhouette description [Sullivan & Carlsson, 2002]
  - Histogram of gradients (HOG) [Dalal & Triggs 2005]



  - Human body part layout

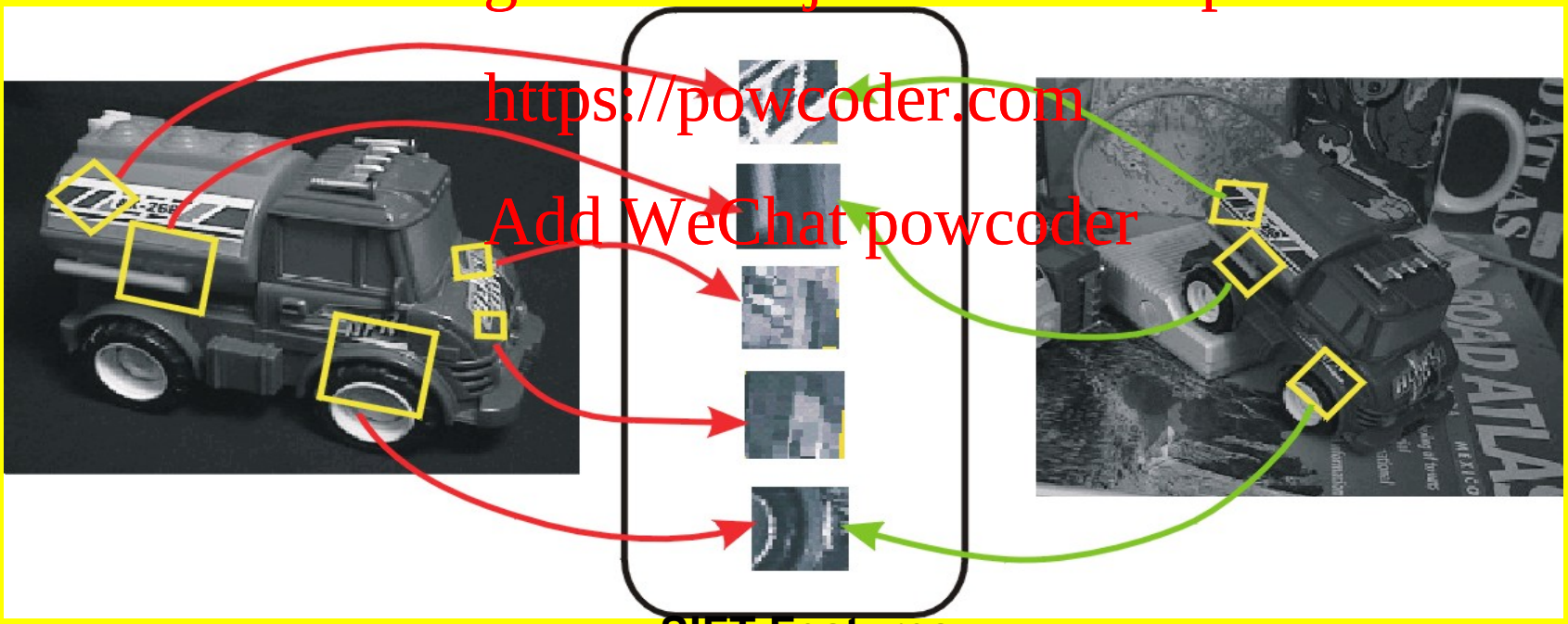
    [Felzenszwalb & Huttenlocher, 2000]

Tracking

- Extract a set of features from the image
- Use a model to predict next position and refine using next image
- Model:
  - simple dynamic models (second order dynamics)
  - kinematic models
  - etc.
- Face tracking and eye tracking now work rather well

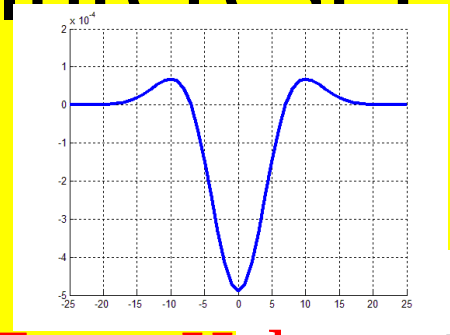
SIFT Features (Lowe 1999)

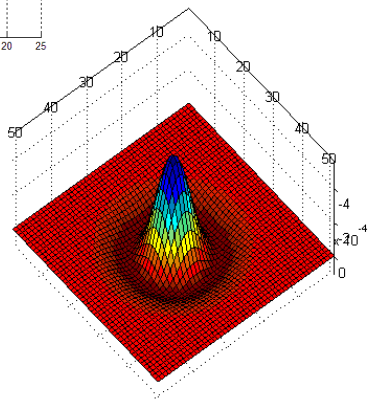
- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters

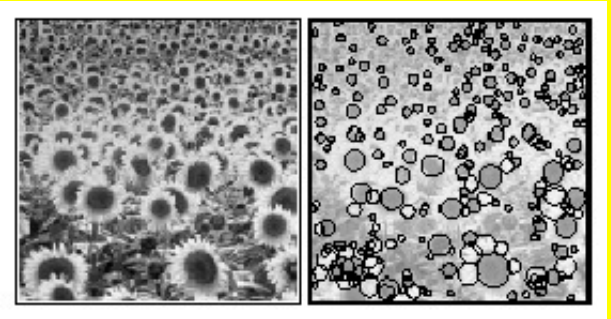


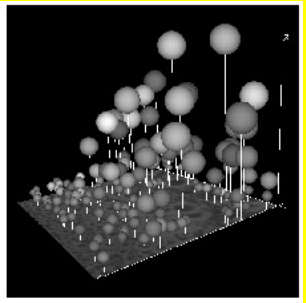
Lowe's Scale-space Interest Points

- **Laplacian of Gaussian** kernel
  - Scale normalised (x by scale^2)
  - Proposed by Lindeberg
- Scale-space detection
  - Find local maxima across scale/space
  - A good "blob" detector







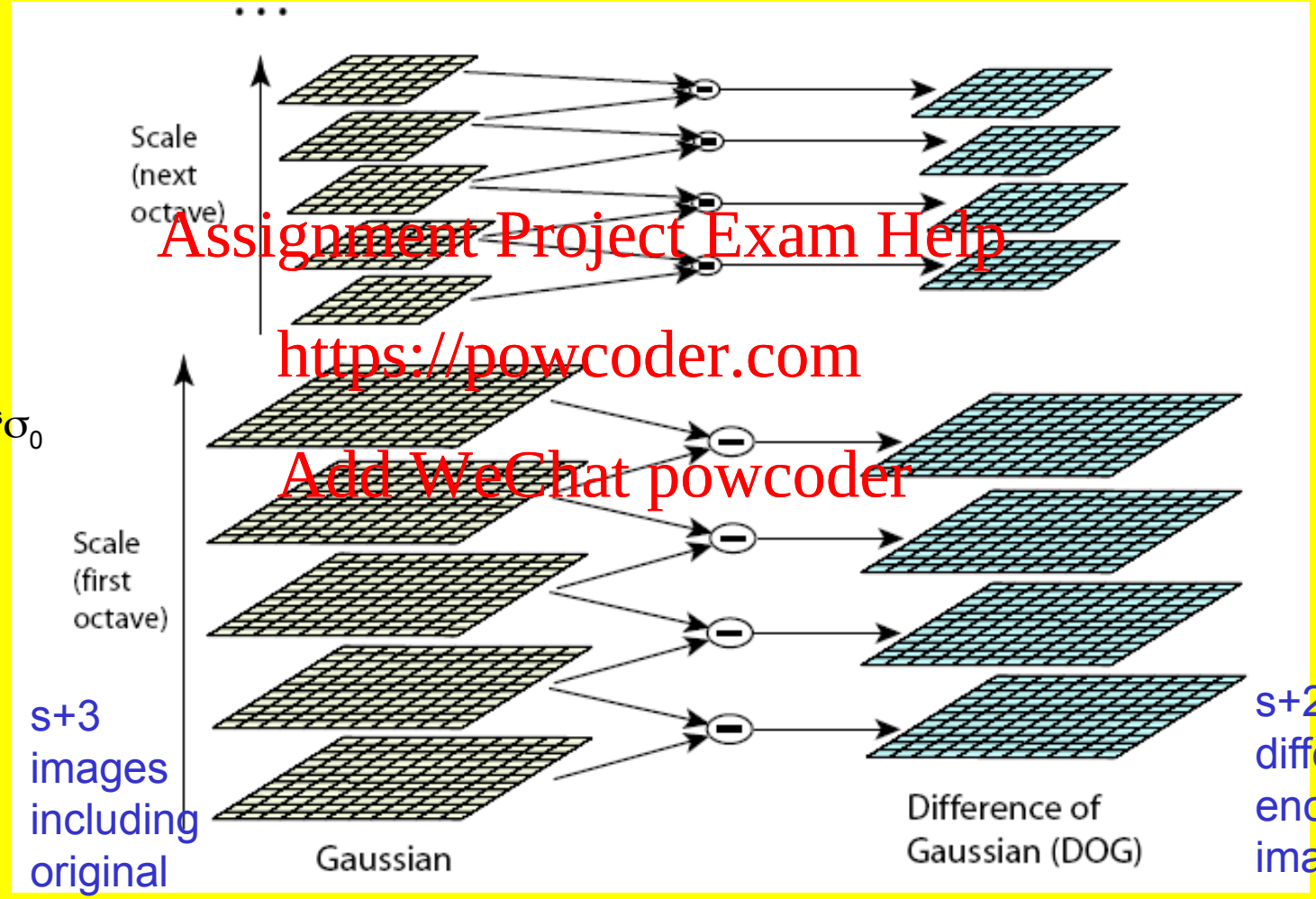


$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{x^2+y^2}{\sigma^2}}$$

$$\nabla^2 G(x, y, \sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

[T. Lindeberg IJCV 1998]

Lowe's Pyramid Scheme



s+2 filters

$\sigma_{s+1}=2^{(s+1)/s}\sigma_0$

.

.

$\sigma_i=2^{i/s}\sigma_0$

.

.

$\sigma_2=2^{2/s}\sigma_0$

$\sigma_1=2^{1/s}\sigma_0$

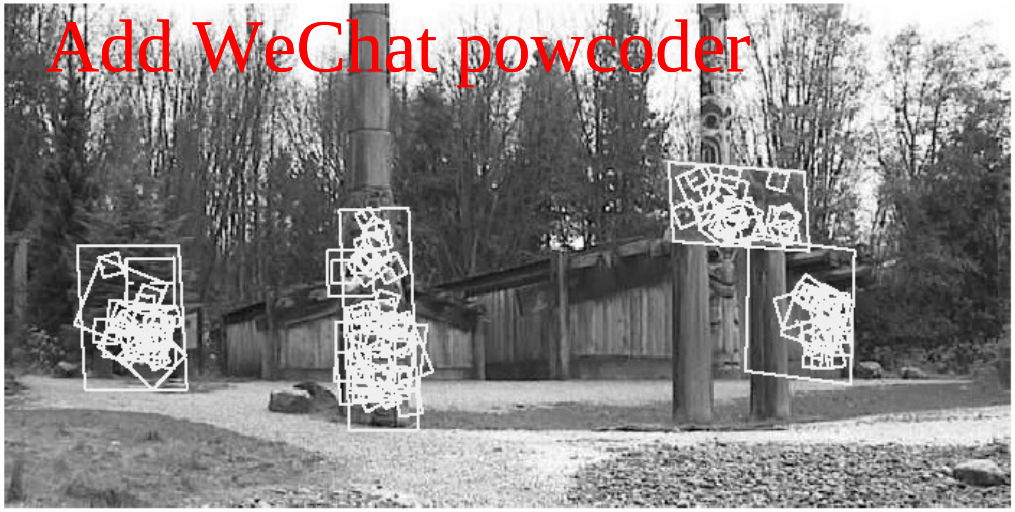
σ_0

s+3 images including original

s+2 differ-ence images

The parameter **s** determines the number of images per octave.

Using SIFT for Matching “Objects”



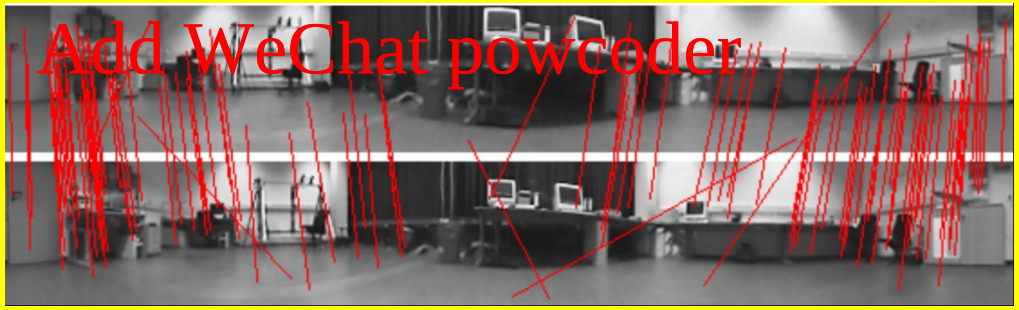
SIFT for Navigation

- Homing in Scale Space (HiSS)
  [Churchill & Vardy 2008]
  - Uses SIFT Feature matching
  - Add scale information to improve homing performance

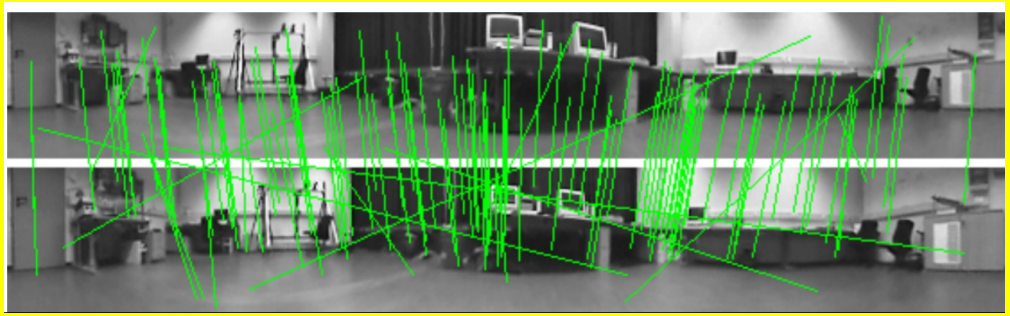
Home image

-

Current image



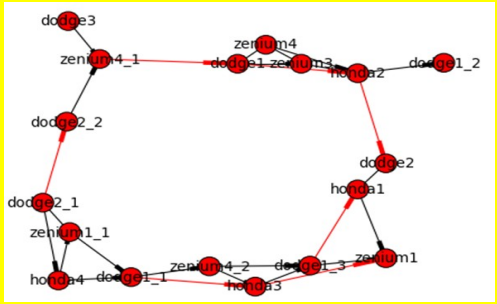
+

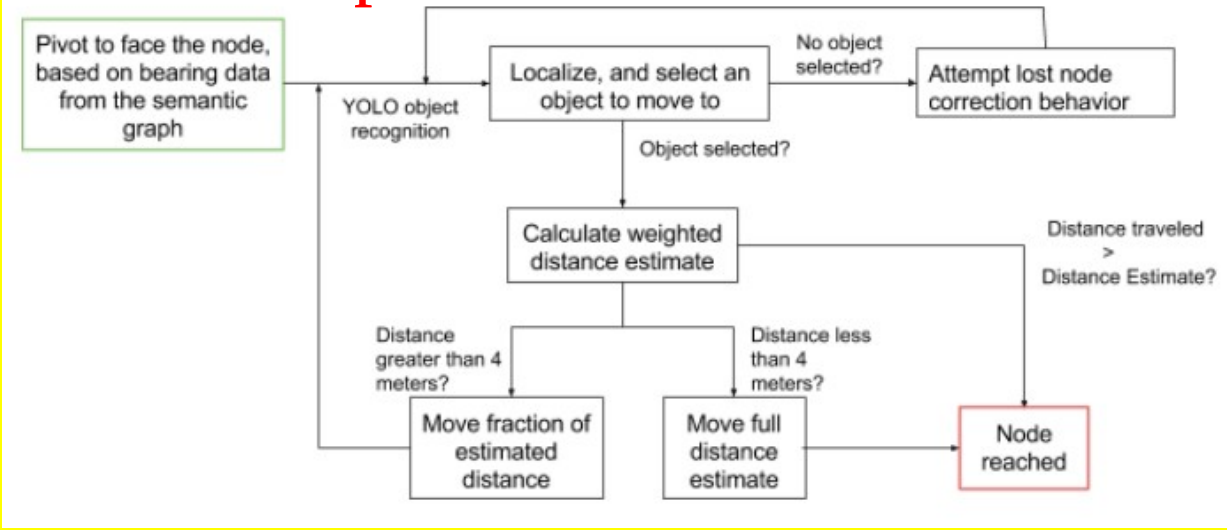


Semantic Navigation (Hulbert 2018)

ROS/Gazebo 3D simulation of a large Suburban scene

Yolo: Extremely fast (155 fps) object recognition using special CNN

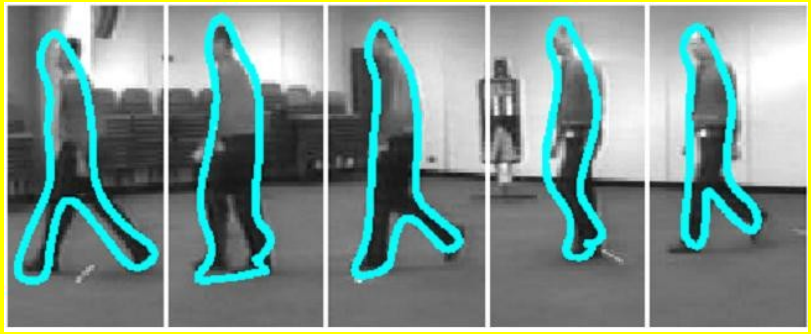




Action recognition in videos

Motion history image
[Bobick & Davis, 2001]

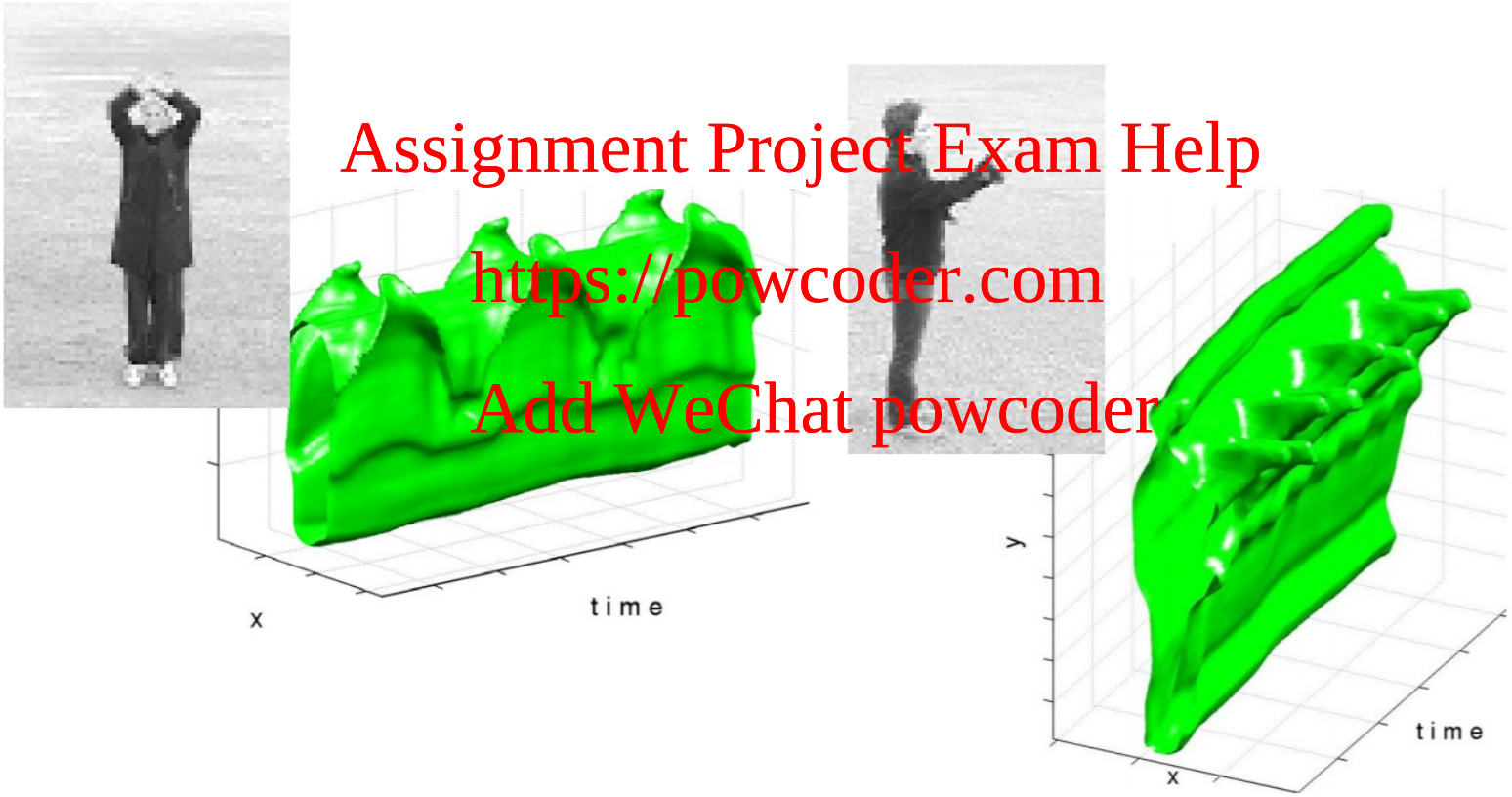
Spatial motion descriptor
[Efros et al. ICCV 2003]



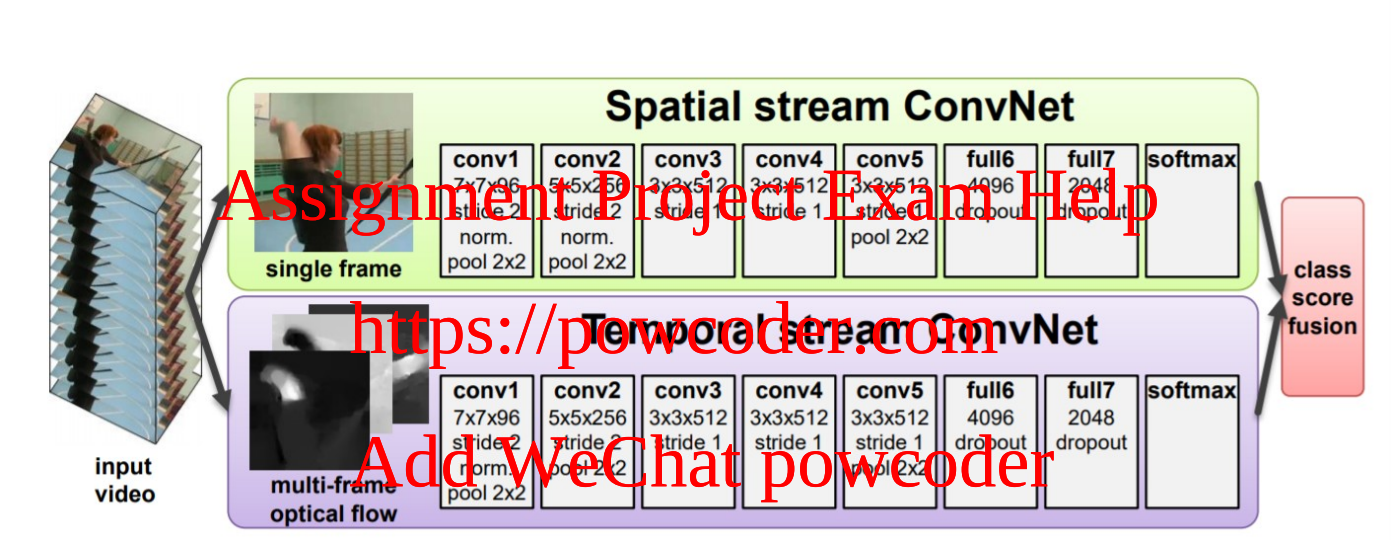
Learning dynamic prior
[Blake et al. 1998]

Sign language recognition
[Zisserman et al. 2009]

Action Recognition:
Action = Space Time Object



CNNs & Activity Recognition



Karen Simonyan & Andrew Zisserman, NIPS 2014