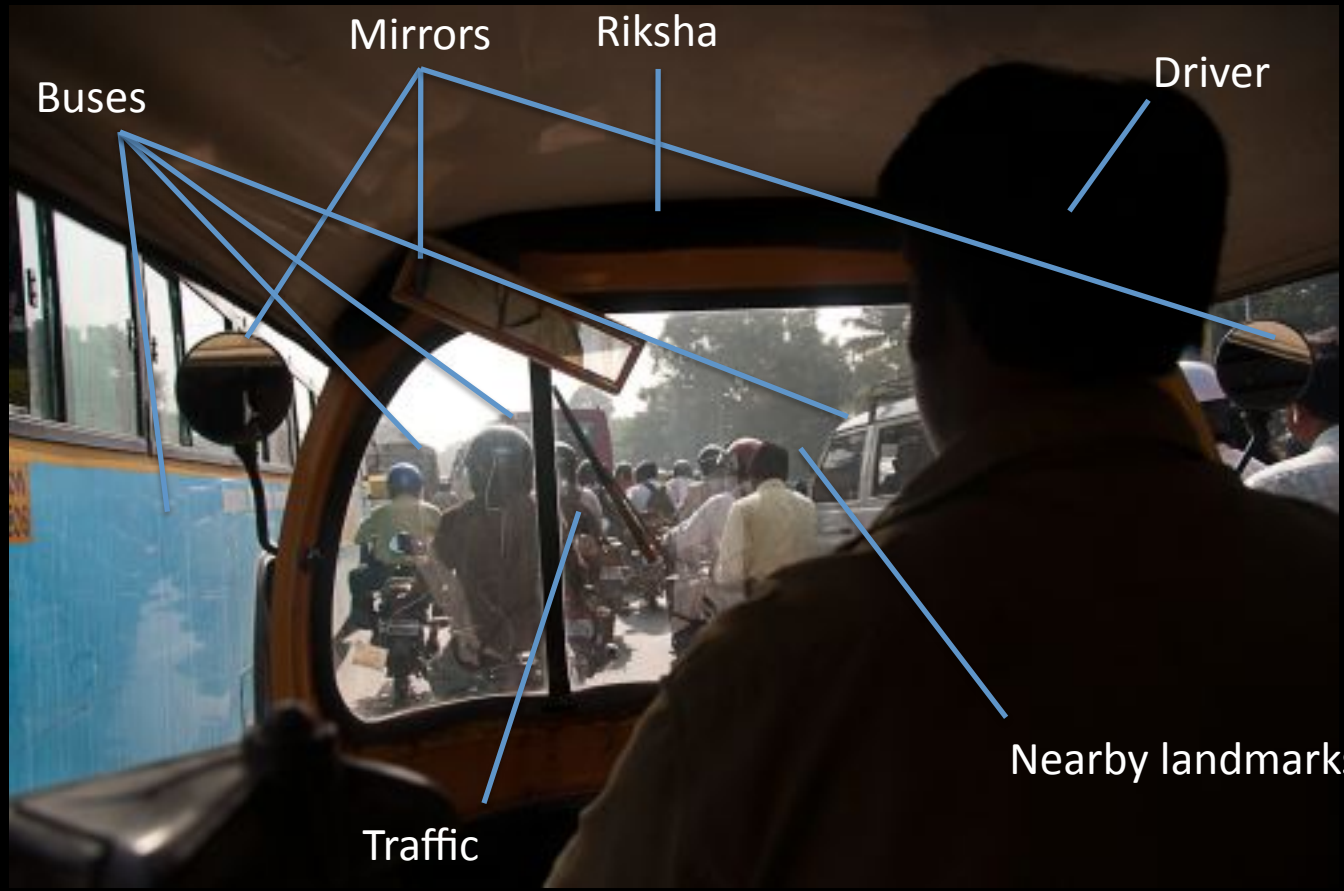
Workshops in Creative Computing 2: Computer Vision Module

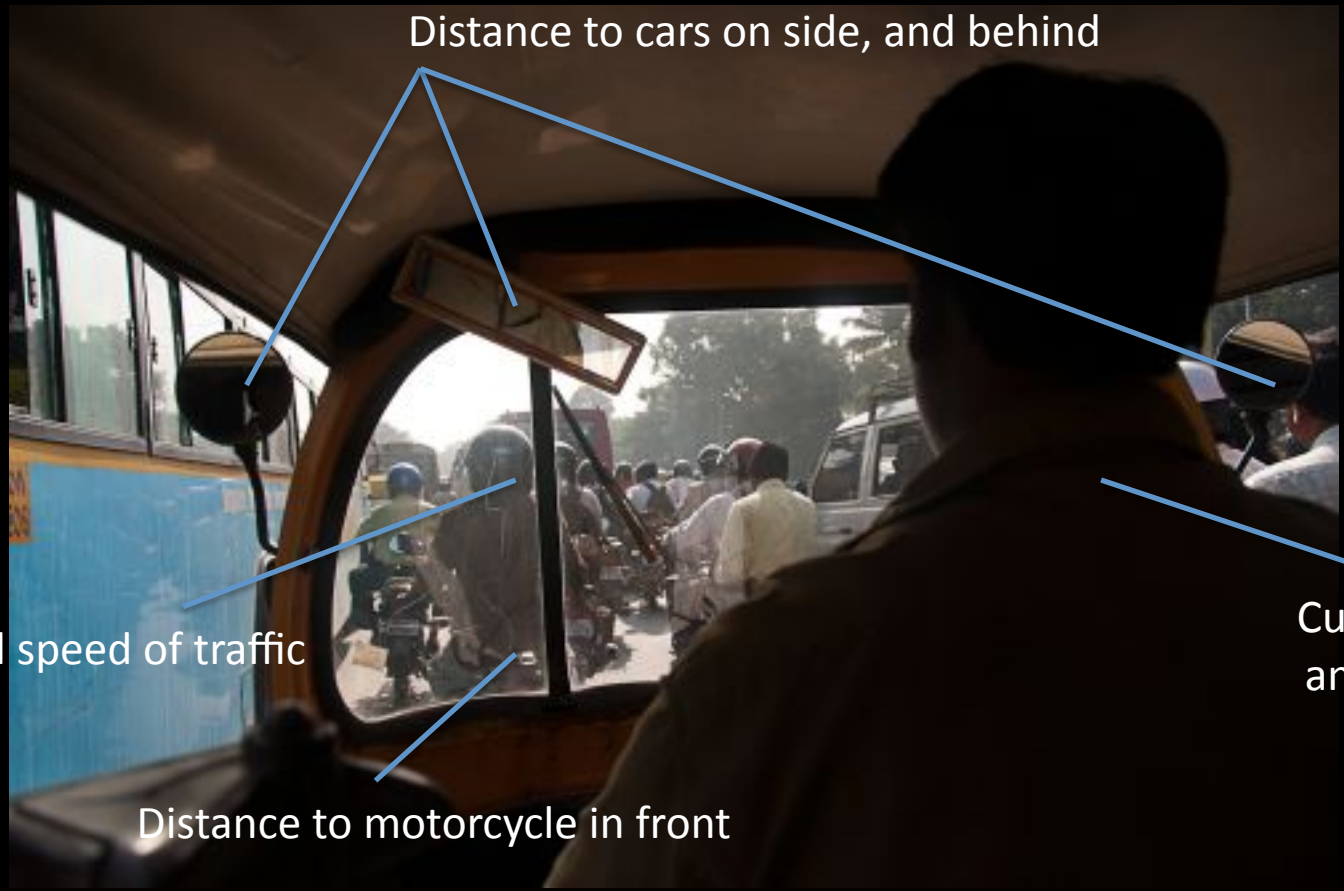
Lecture 1: Introduction to Computer Vision

Wednesday Feb 20, 2013

Parag K Mital

Assignment 1: Solve computer vision.


Mirrors
Riksha
Buses
Driver
Nearby landmarks
Traffic


Distance to cars on side, and behind
Overall speed of traffic
Current speed and direction
Distance to motorcycle in front

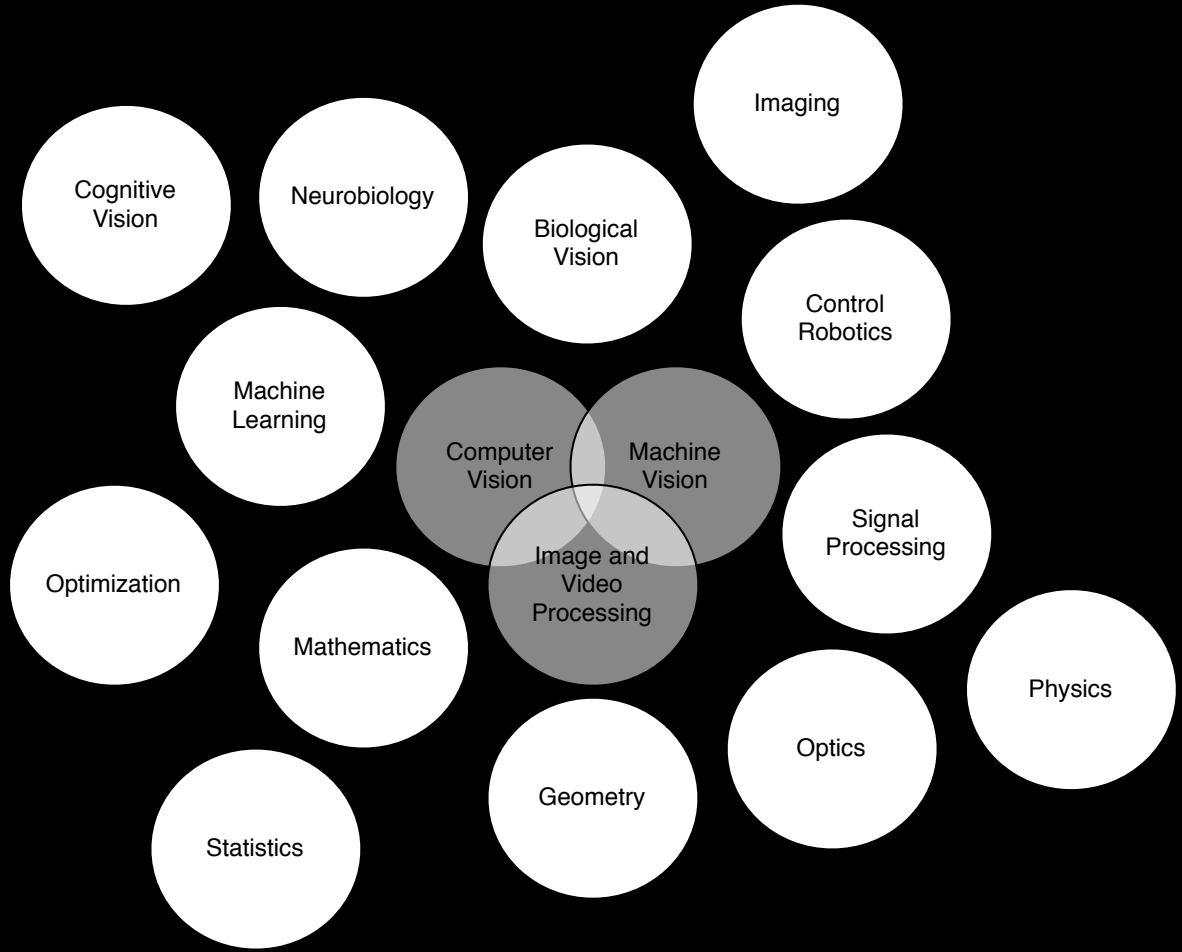
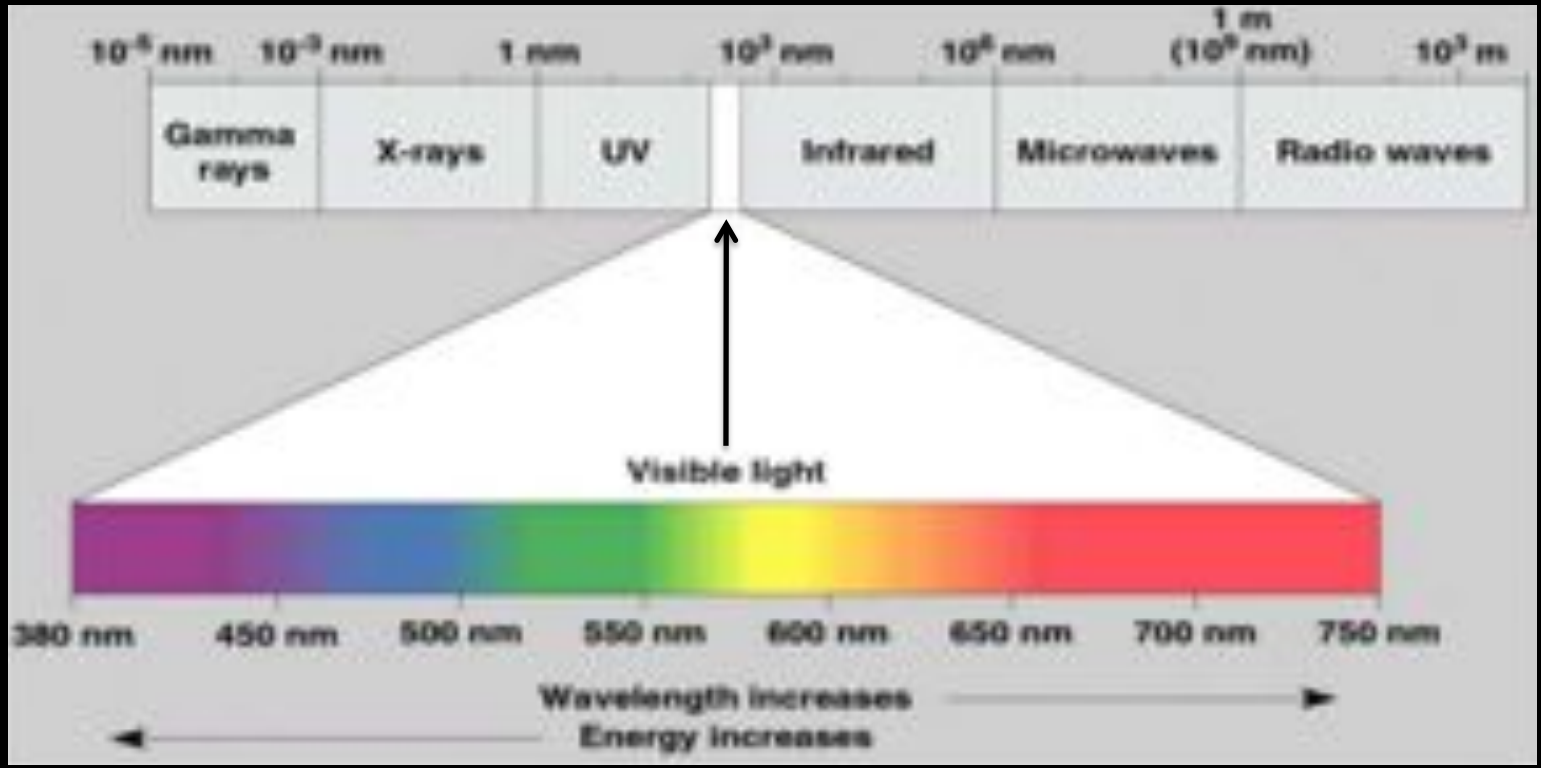
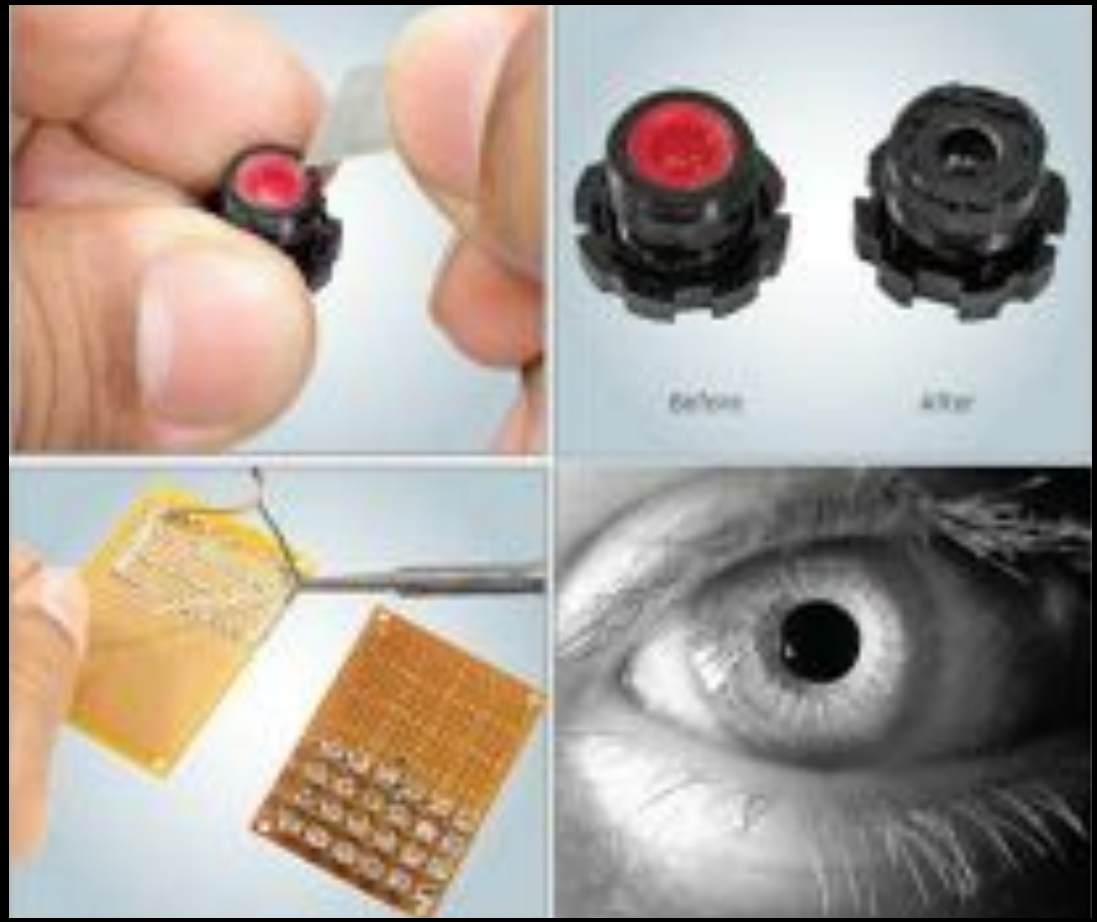

Imaging
Cognitive Vision
Neurobiology
Biological Vision
Control Robotics
Machine Learning
Computer Vision
Machine Vision
Signal Processing
Optimization
Image and Video Processing
Mathematics
Physics
Optics
Geometry
Statistics

Image Acquisition

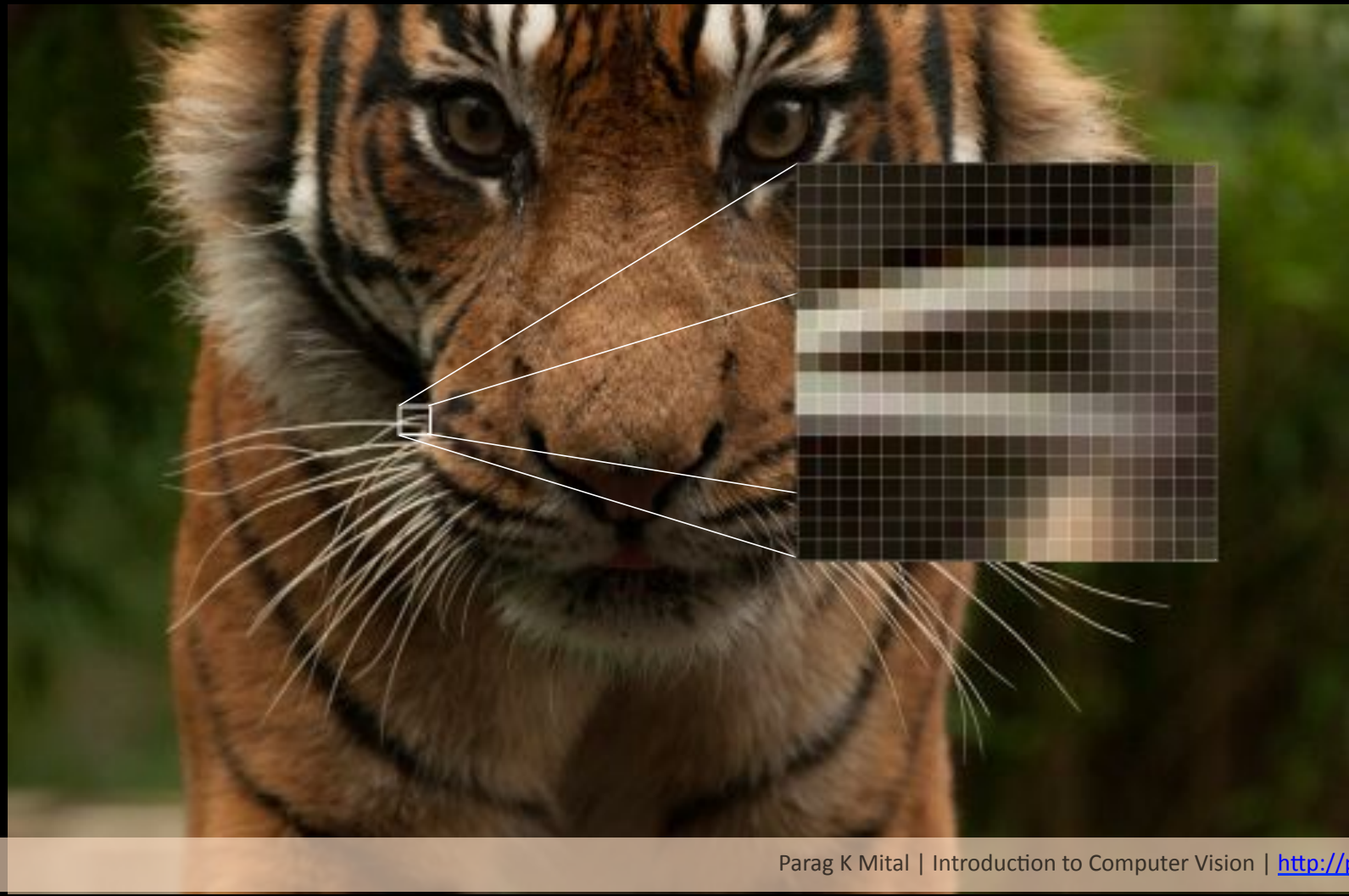
Depth	**LIDAR; Kinect; Structured Light; Stereo Vision**
Infrared/ Thermal/ Spectral	**Satellite; Heat detection**
Light Field	**Synthetic Camera Control**

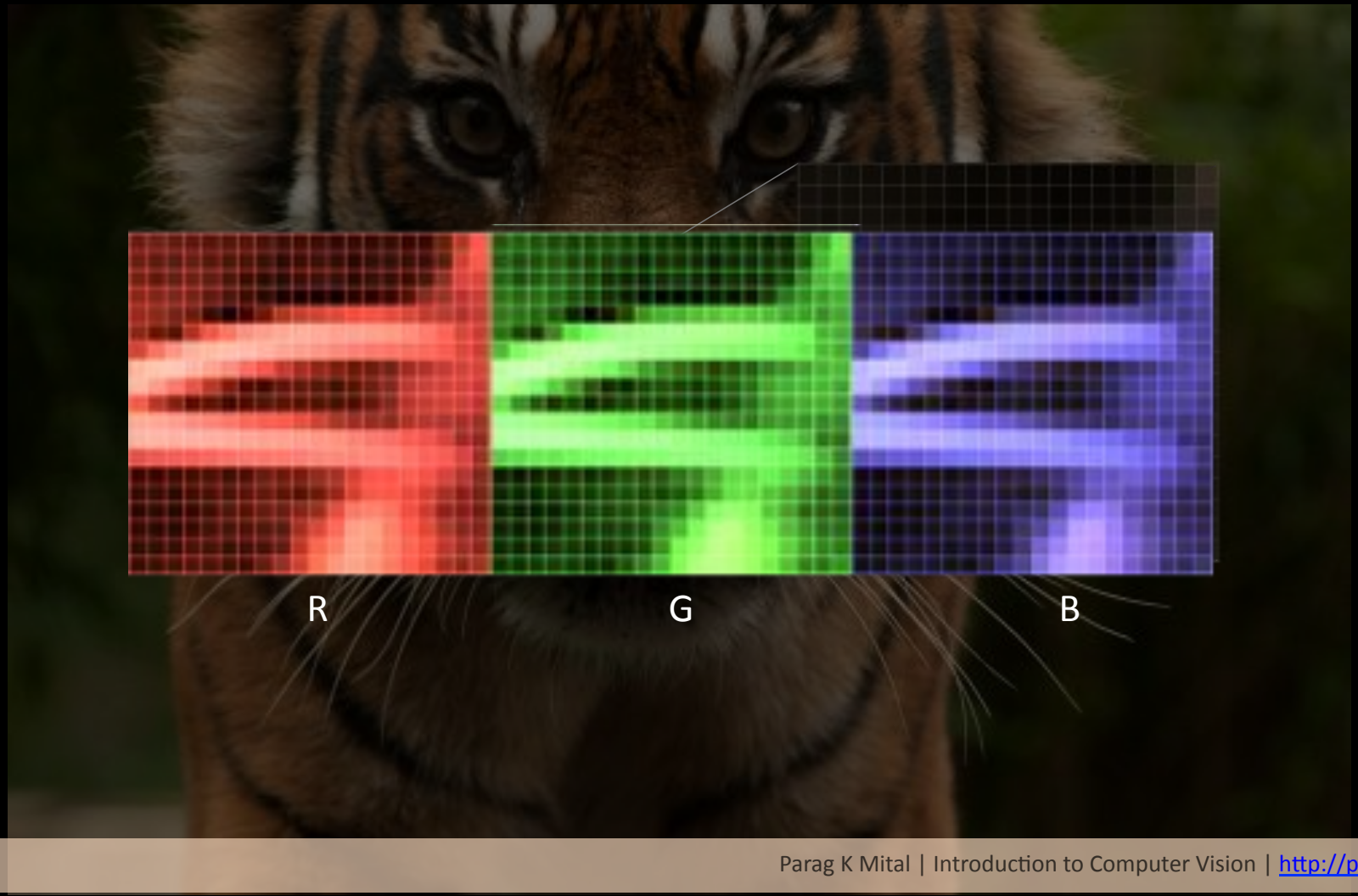

10^{-5} nm
10^{-3} nm
1 nm
10^{3} nm
10^{6} nm
1 m (10^{9} nm)
10^{3} m
Gamma rays
X-rays
UV
Infrared
Microwaves
Radio waves
Visible light
380 nm
450 nm
500 nm
550 nm
600 nm
650 nm
700 nm
750 nm
Wavelength increases
Energy increases

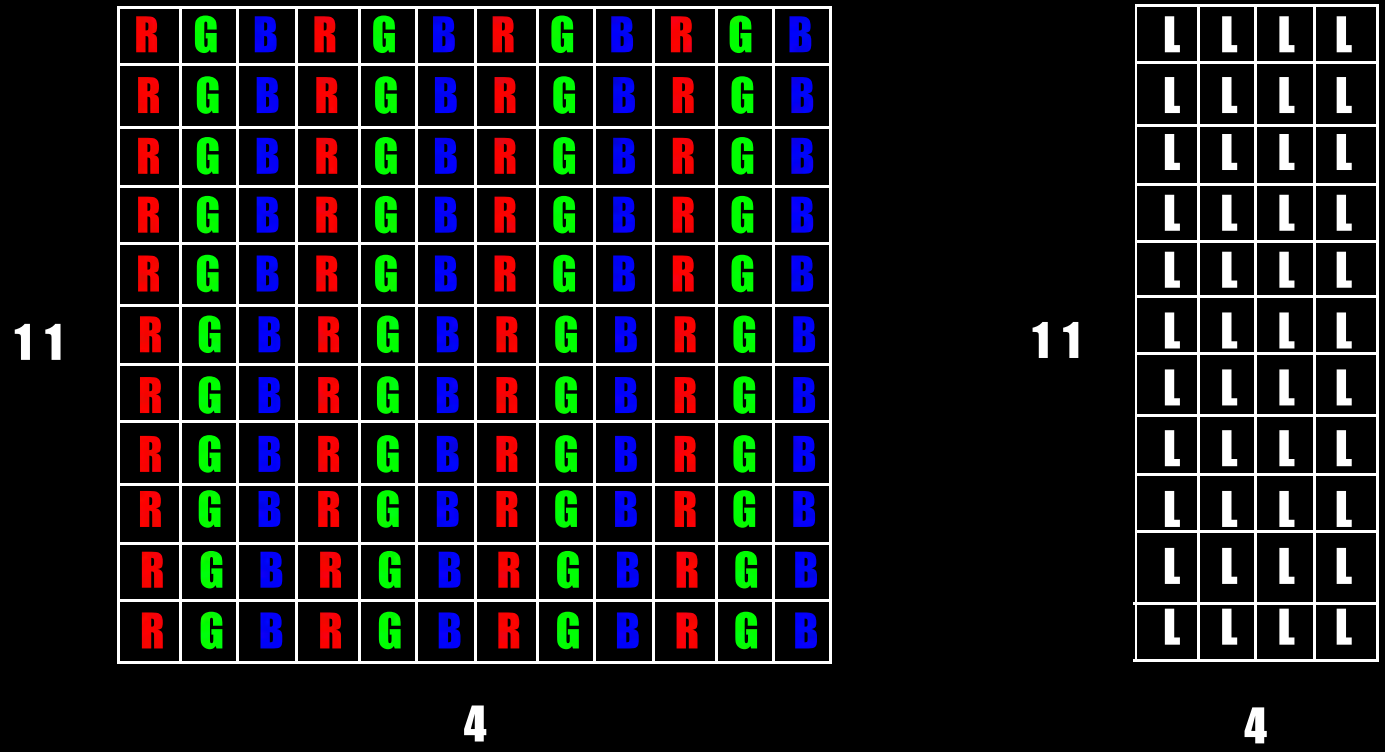

Before
After

Pixel Descriptions

RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
L*a*b*	Perceptually uniform color distribution (CIE1976)
CMYK	Cyan, Magenta, Yellow, Black
Grayscale	Single channel of a color image; luminance




R
G
B


11
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
R G B R G B R G B R G B
4
11
L L L L
L L L L
L L L L
L L L L
L L L L
L L L L
L L L L
L L L L
L L L L
L L L L
4

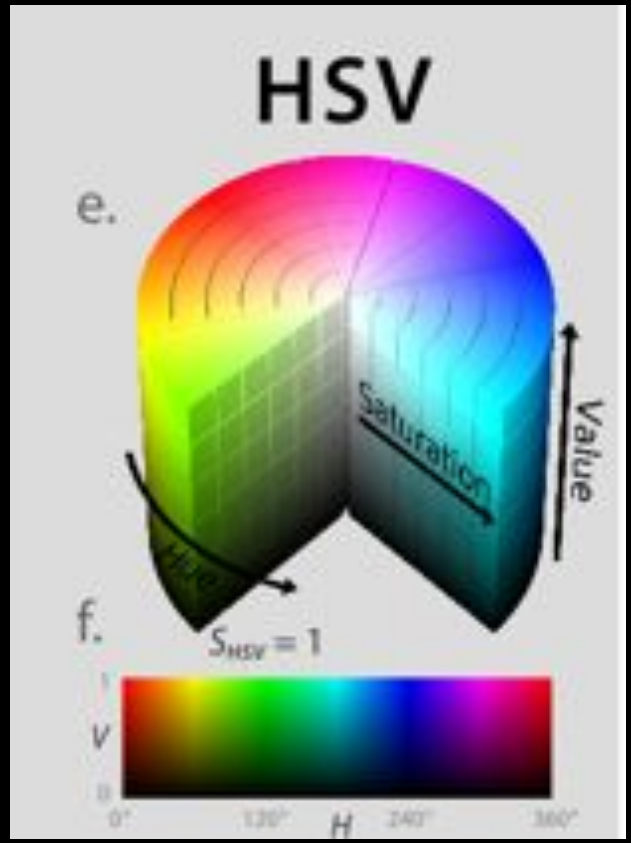
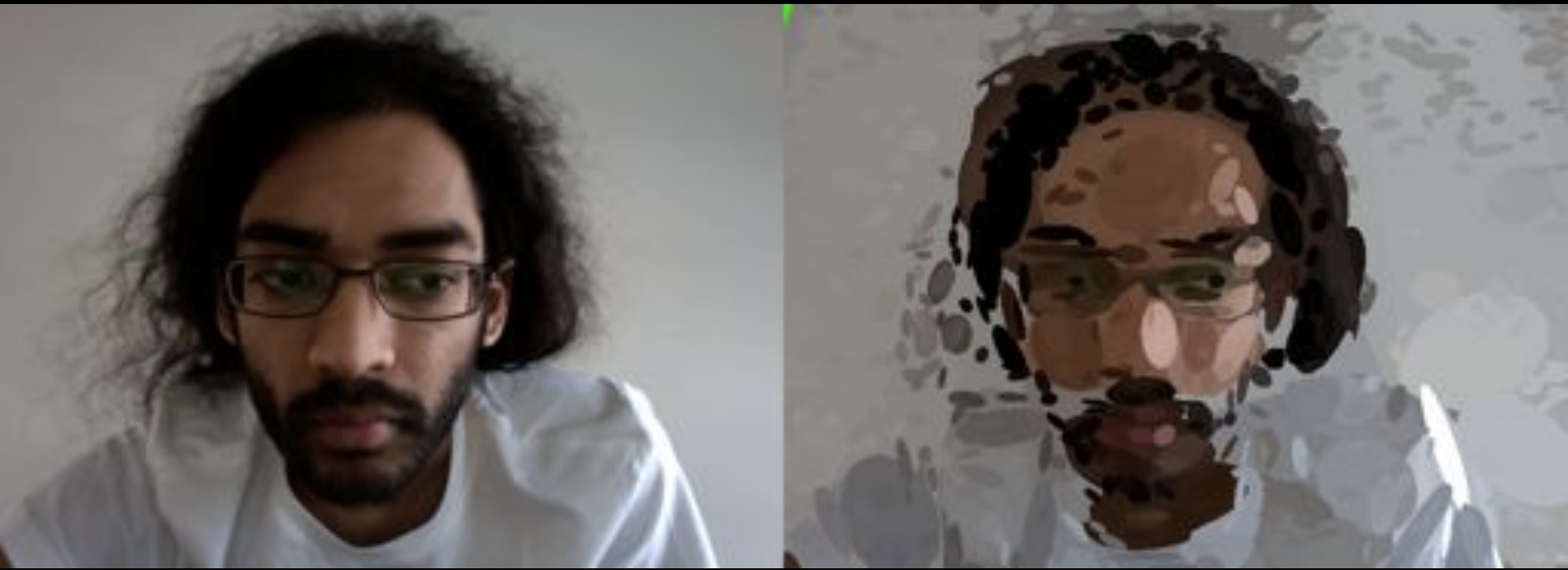
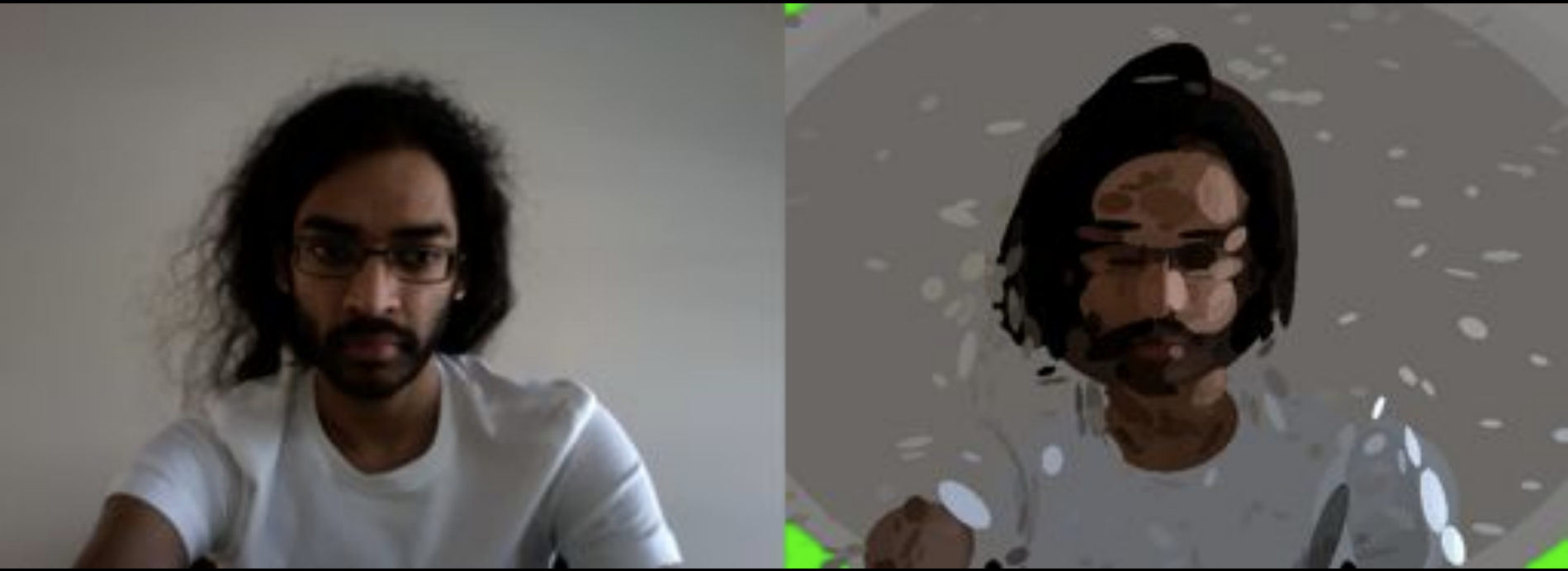
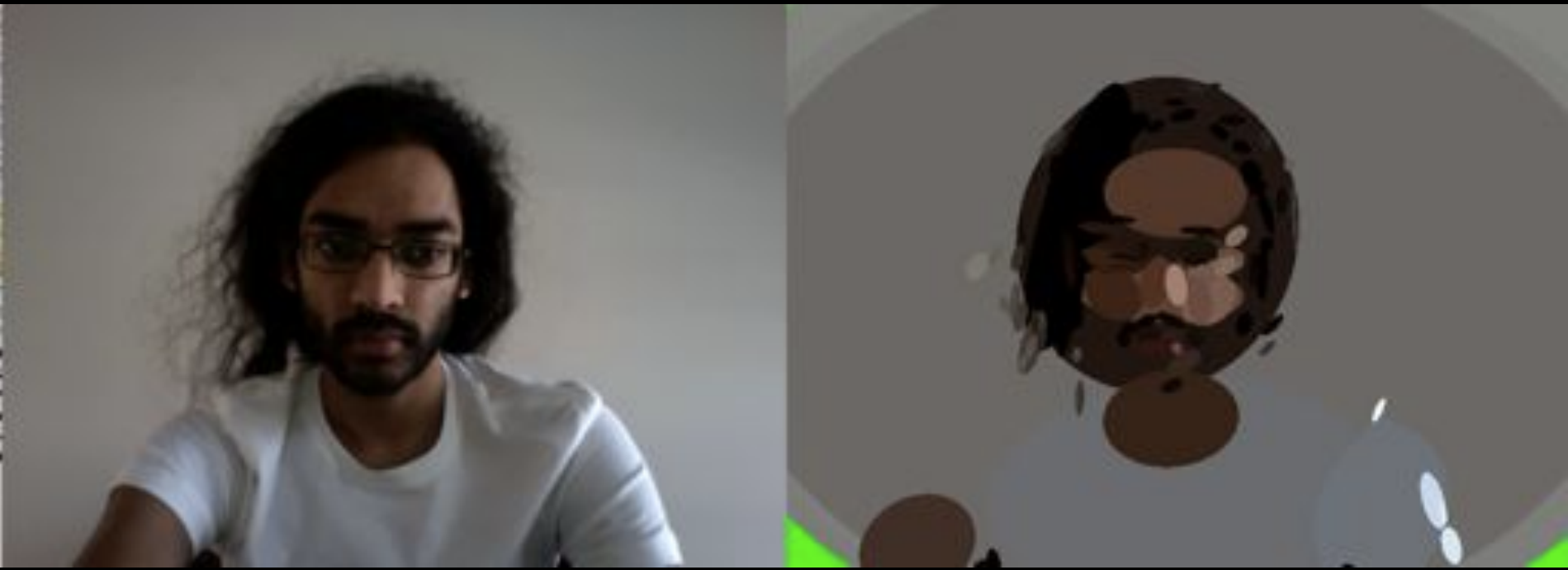

HSV
e.
Saturation
Value
Hue
f.
$S_{HSV} = 1$
V
H

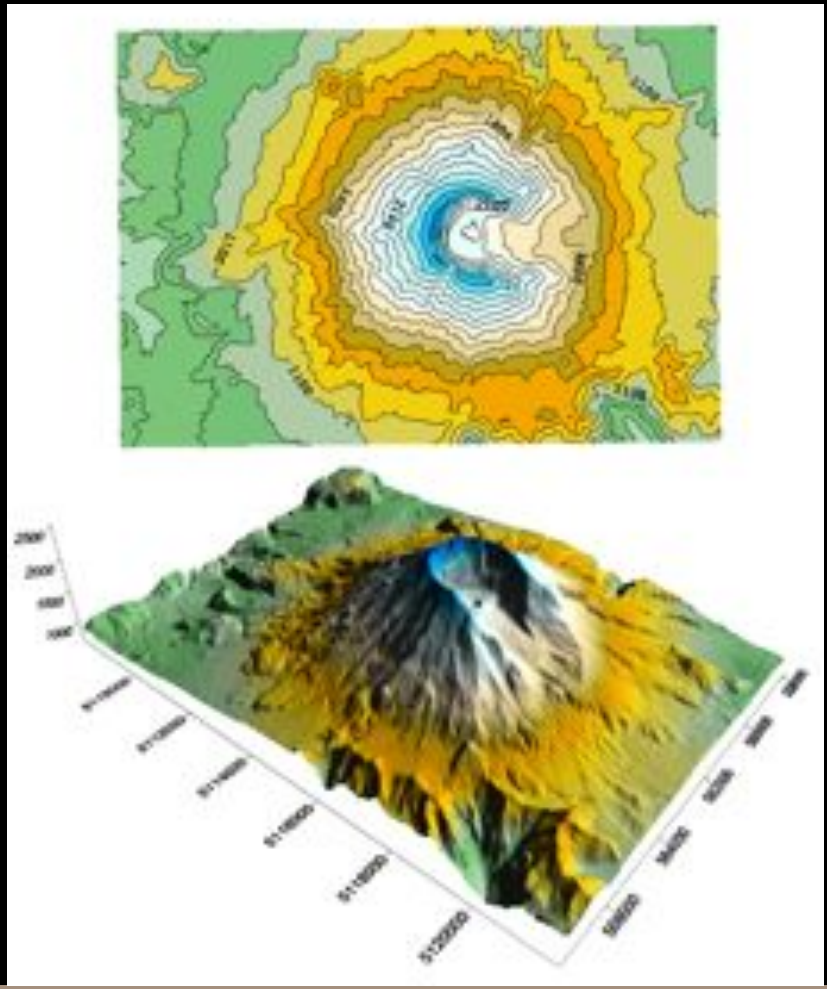
Image Descriptions

Pixels	Luminance; Color-spaces; Depth; Heat
Feature Points	SIFT; SURF; Harris Corners; HOG; FAST
Edges/Lists	Sobel; Canny; Hysteresis; Connected Components; Shape Models
Blobs/Regions	Mean-Shift; MSER; Watershed; Graph-Cuts; Background Subtraction; Appearance Models
Maps	Geodesics; Topography; Density









OpenCV

Insane library for manipulating image/video data
Machine Learning
Regression
Classification
GUI
Image/Video Acquisition (talks to many types of hardware)
File I/O, Management
Feature Tracking
Object Detection
Pattern Recognition
…

OpenCV

Major overhaul from 1.1 to 2.0 from C style to C++ style
Older containers called IplImage and CvMat
Newer ones called Mat

Can convert between all types seamlessly without moving any memory

ofxOpenCV

Provides very basic interface to using OpenCV
Handles all OpenGL/Texture drawing stuff
Some simple interfaces for commonly used functions
Some extended interfaces for contour tracking and haar finder tracking
Can convert to other basic types in openFrameworks,
    e.g. `ofTexture`, `ofImage`, etc…
Access pixels using `getPixels()`
Can use OpenCV without using ofxOpenCV

ofxOpenCV

Very easy to get the OpenCV container from an ofxOpenCV one.

e.g.:

```
// first declare the object
ofxCvGrayscaleImage myImage;
// then allocate some memory for it in setup()
myImage.allocate(320, 240);
// then if you ever want to access the
// internal OpenCV structure,
IplImage *cvImage = myImage.getCvImage();
```

OpenCV IplImage *

Using an IplImage *, you can easily access the pixel's contents in memory.

```
IplImage *cvImage = colorImg.getCvImage();
for (int i = 0; i < cvImage->height; i++) {
     unsigned char *pix =
     (unsigned char *)cvImage->imageData +
     i * cvImage->widthStep;
     for (int j = 0; j < cvImage->width; j++) {
          pix[3 * j + 0] = rand() % 255;
          pix[3 * j + 1] = rand() % 255;
          pix[3 * j + 2] = rand() % 255;
     }
}
```

OpenCV Mat C++

Can also use the new OpenCV 2.1+ C++ style container, Mat:

```
IplImage *cvImage = colorImg.getCvImage();
cv::Mat newCvImage(cvImage);
```

Let's try finding the mean value of this image:

```
cv::Scalar val = cv::mean(newCvImage);
double firstDimensionsMean = val[0];
```

Next Week: Tracking

Blob Tracking
Contour Tracking,
Face Tracking,
Generic Object Tracking