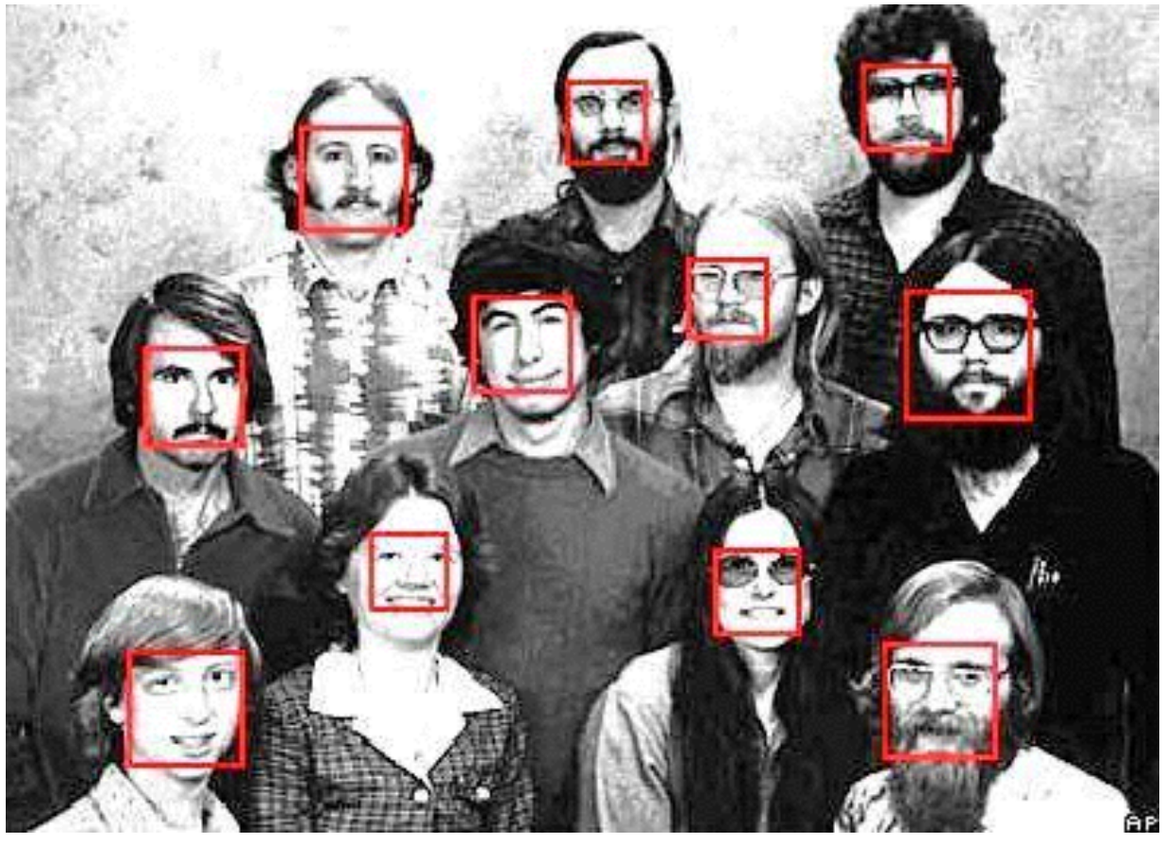
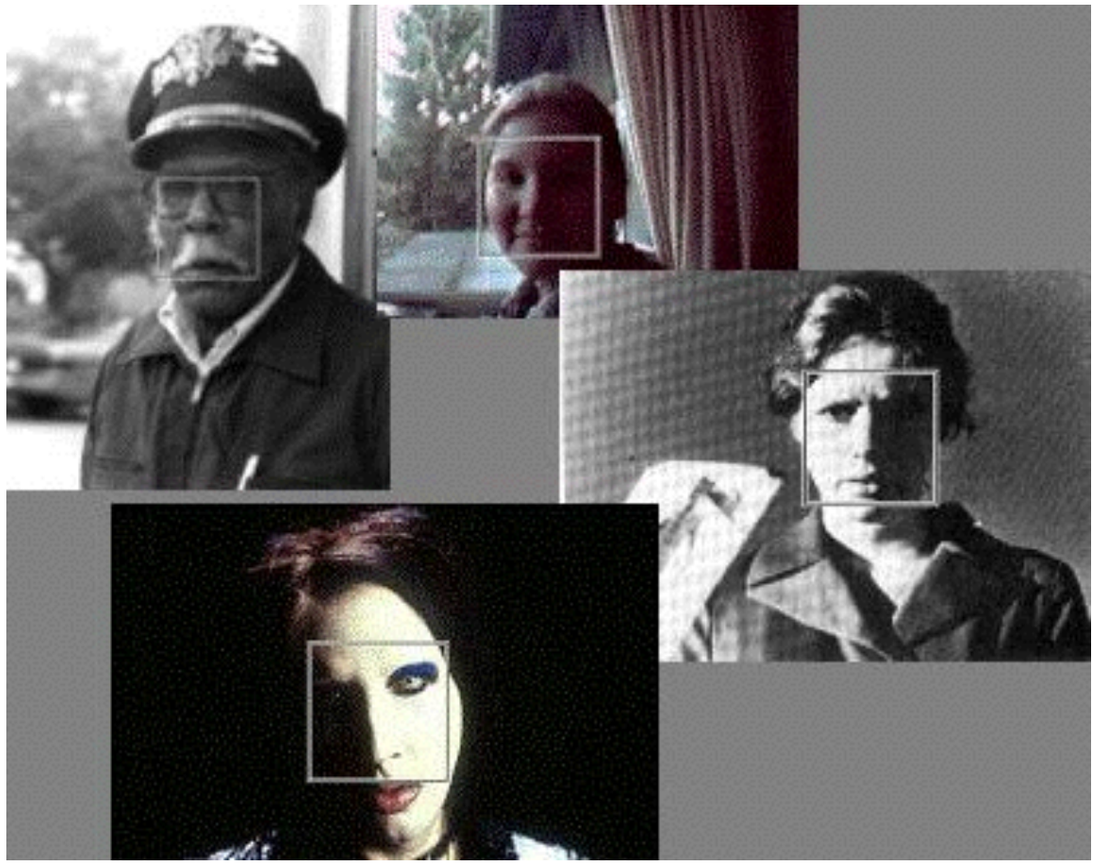
OpenCV CrashCourse

Feb 5th 2017

Face detection





Many slides adapted from P. Viola

Face detection

- Basic idea: slide a window across image and evaluate a face model at every location

Window-based models
Building an object model

Given the representation, train a binary classifier

→

Face/non-face Classifier

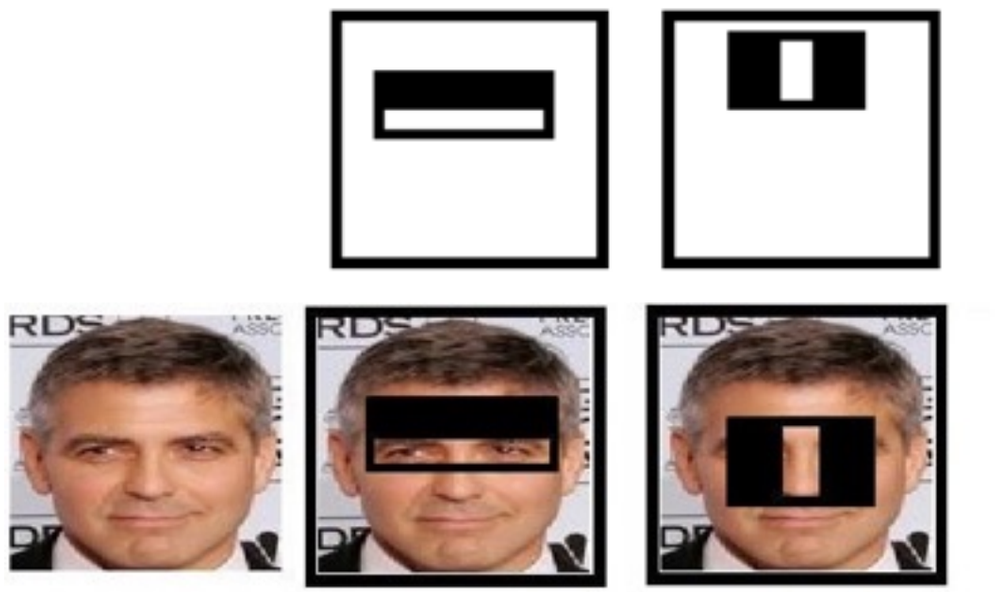
↓

Yes, face.

No, not face.

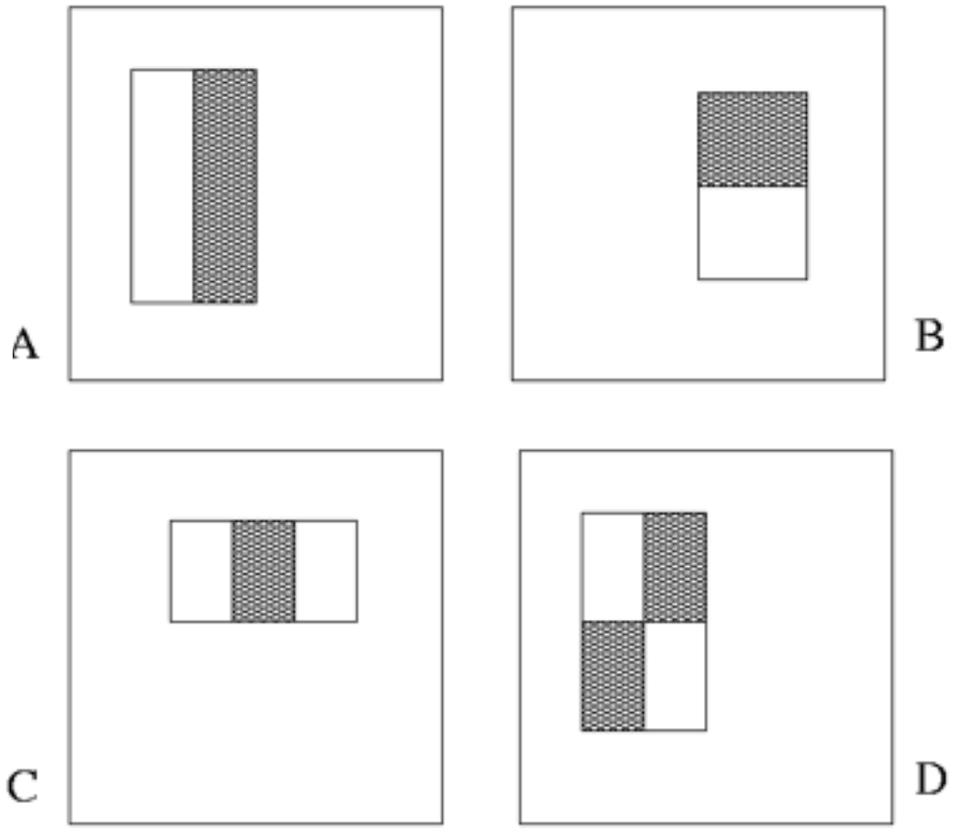
Haar-features (2)

- Capture the face symmetry



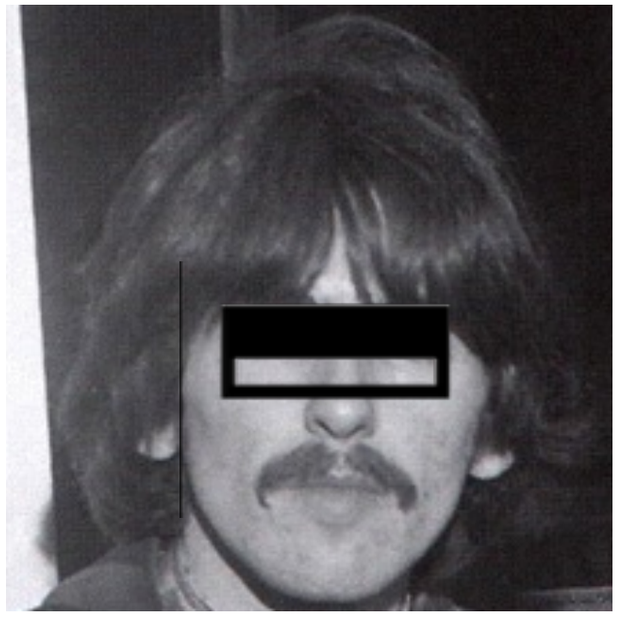
Hear Features

"Rectangle filters"

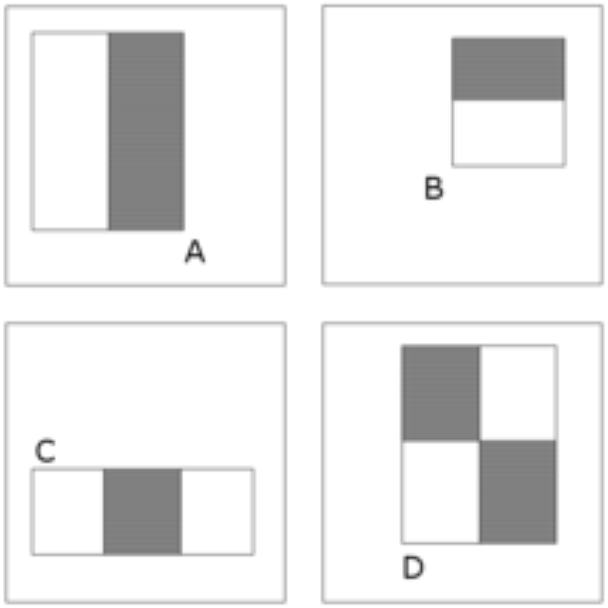


Value =

∑ (pixels in white area) –
∑ (pixels in black area)

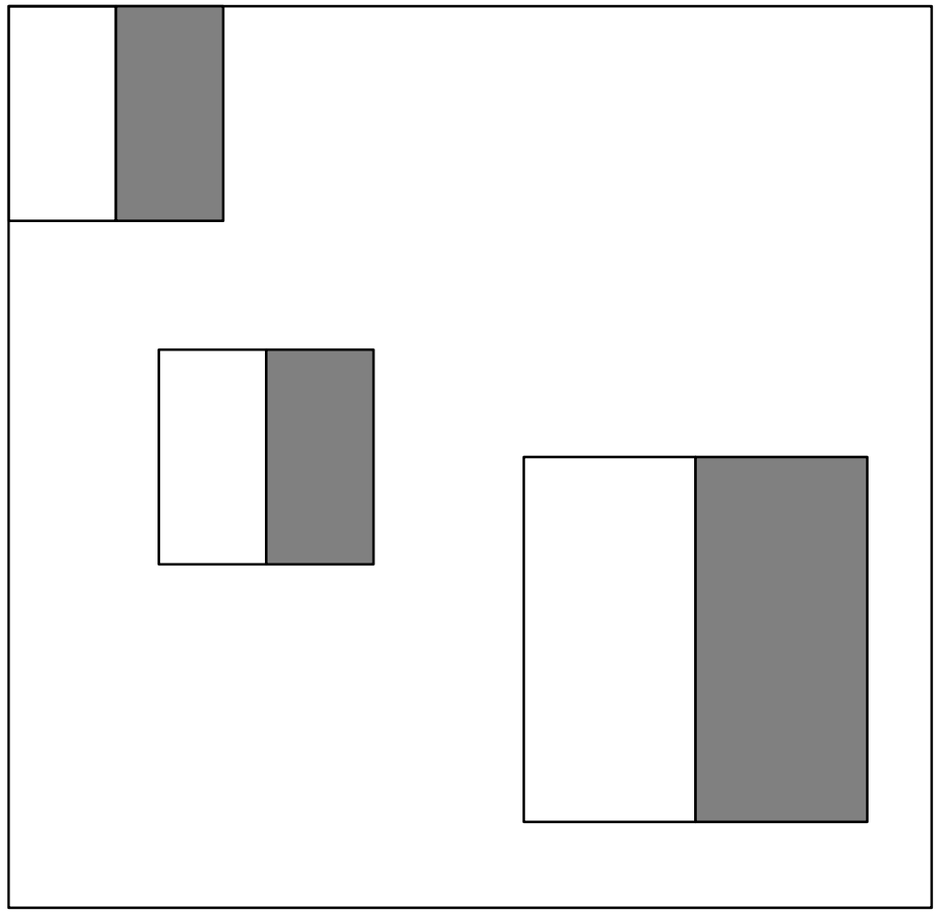


Haar-features (3)



Four types of haar features

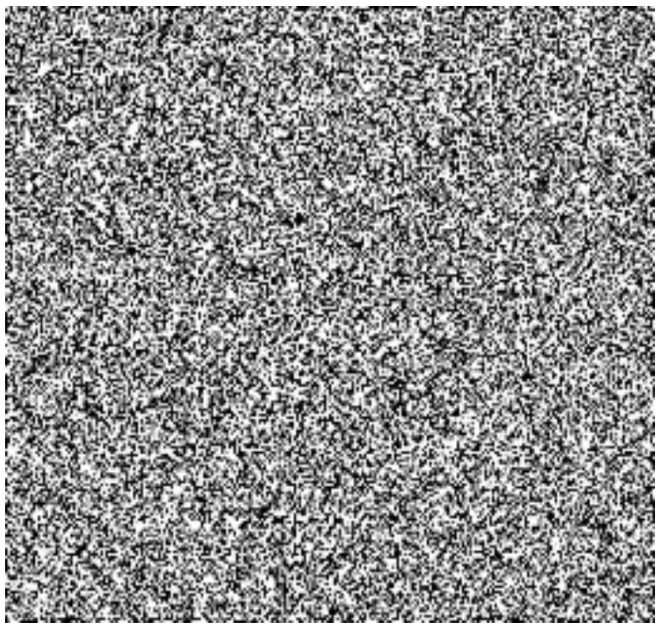
Type A



A 24x24 detection window

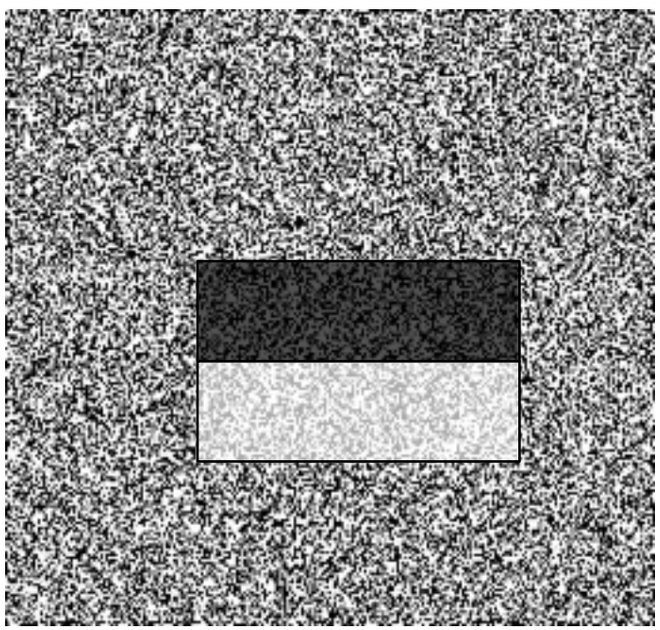
Can be extracted at any location with any scale!

Example



Source

Result

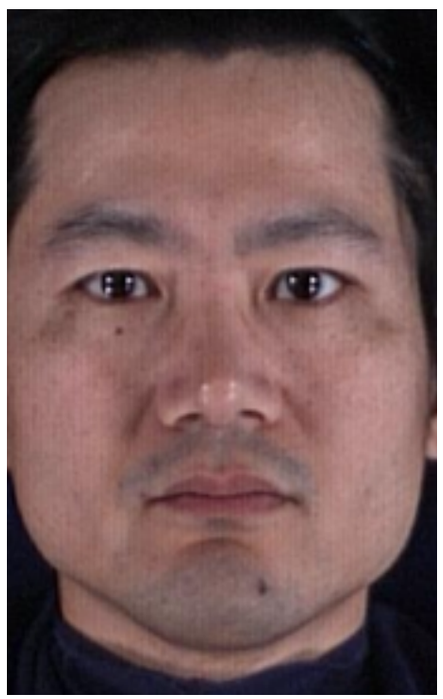


Challenges

view point variation

Michelangelo 1475-1564

illumination



occlusion

Magritte, 1957

scale

Slide credit: Fei-Fei Li

Challenges

deformation

Xu, Beihong 1943

background clutter

Klimt, 1913

Slide credit: Fei-Fei Li

Algorithm

- 1. Load image
- 2. Load detector (Haar Cascade)
- 3. Scan multiple patches of image at different scales to detect which ones are faces
- 4. Visualize faces by drawing a bounding box

Code

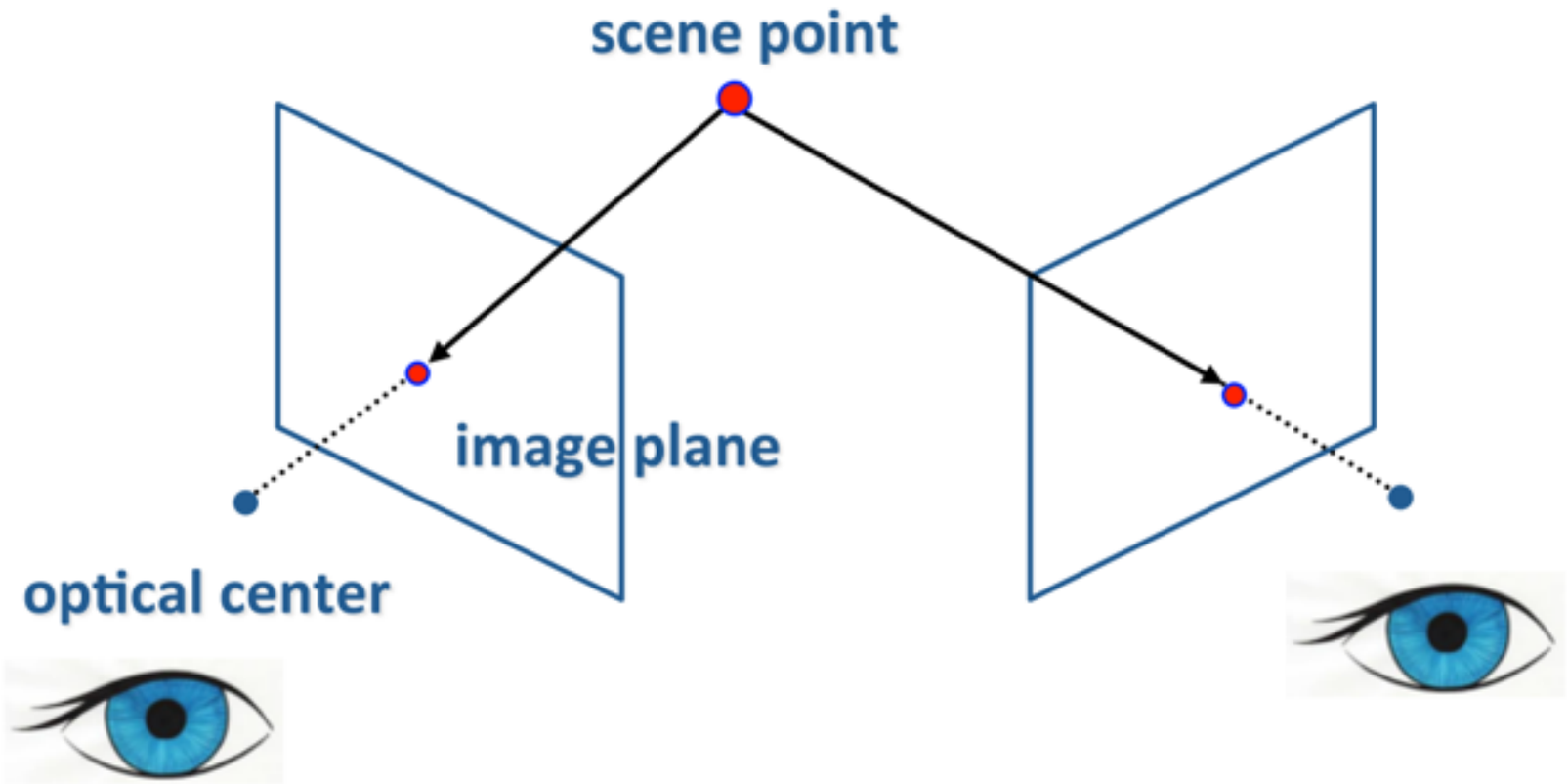
- Python 2.7
- OpenCV 2.4
- Clone repo: https://git.io/vDC81
- All code will be pushed to repo after this

3D reconstruction

- Idea: Build a 3D representation of an object using images
- How many images?
- Any specific requirements on the images?

Multi-view geometry

Two-view (stereo)

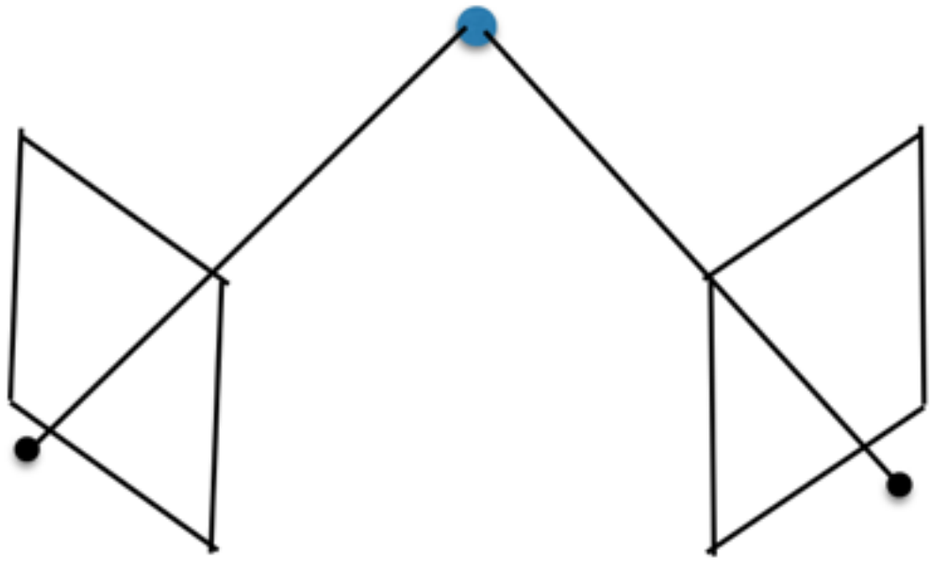


Given 2 corresponding points in 2 cameras, see where the cast rays meet

Much of basics can be derived from this picture

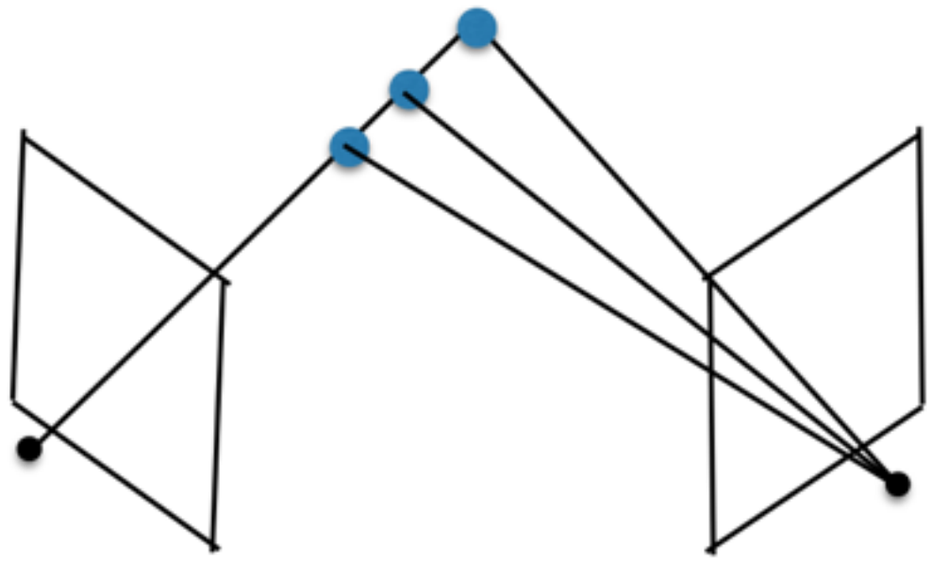
Goal: let's get the geometric intuition before the match

Questions



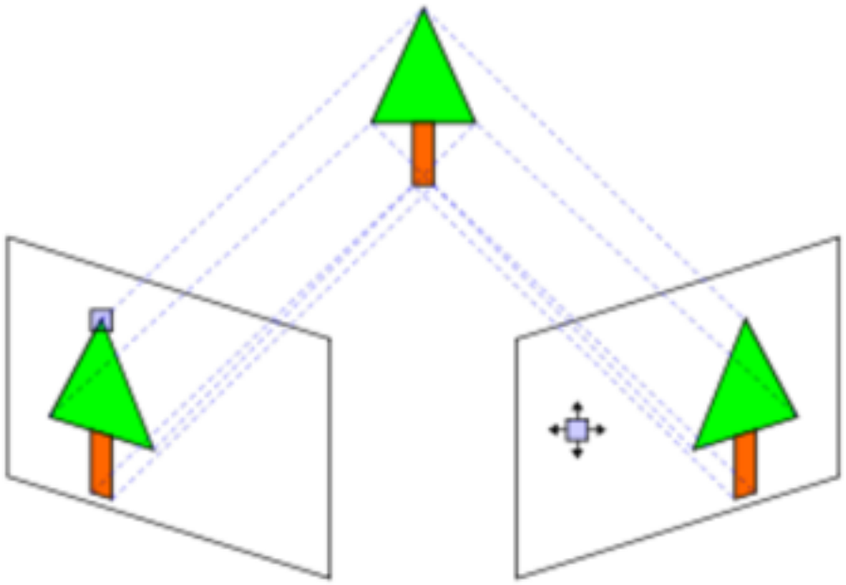
Given a point in left view, what is the set of points it could project to in right view?

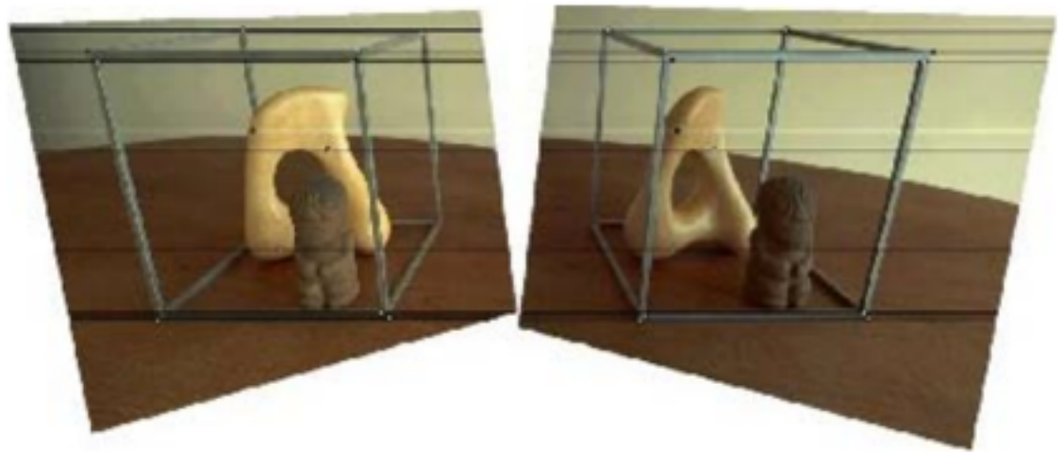
Questions



Given a point in left view, what is the set of points it could project to in right view?

Implies that *for known camera geometry,* we need search for corresponds only over 1D line





Stereo Pair

Rectified Stereo Pair

Question: do the epipolar lines depend on scene structure, cameras, or both?

Epipolar geometry is purely determined by camera extrinsics and camera instrinics

Steps

- 1. Load left and right images
- 2. Calculate depth using both images (disparity map)
- 3. Create 3D point cloud
- 4. Visualize