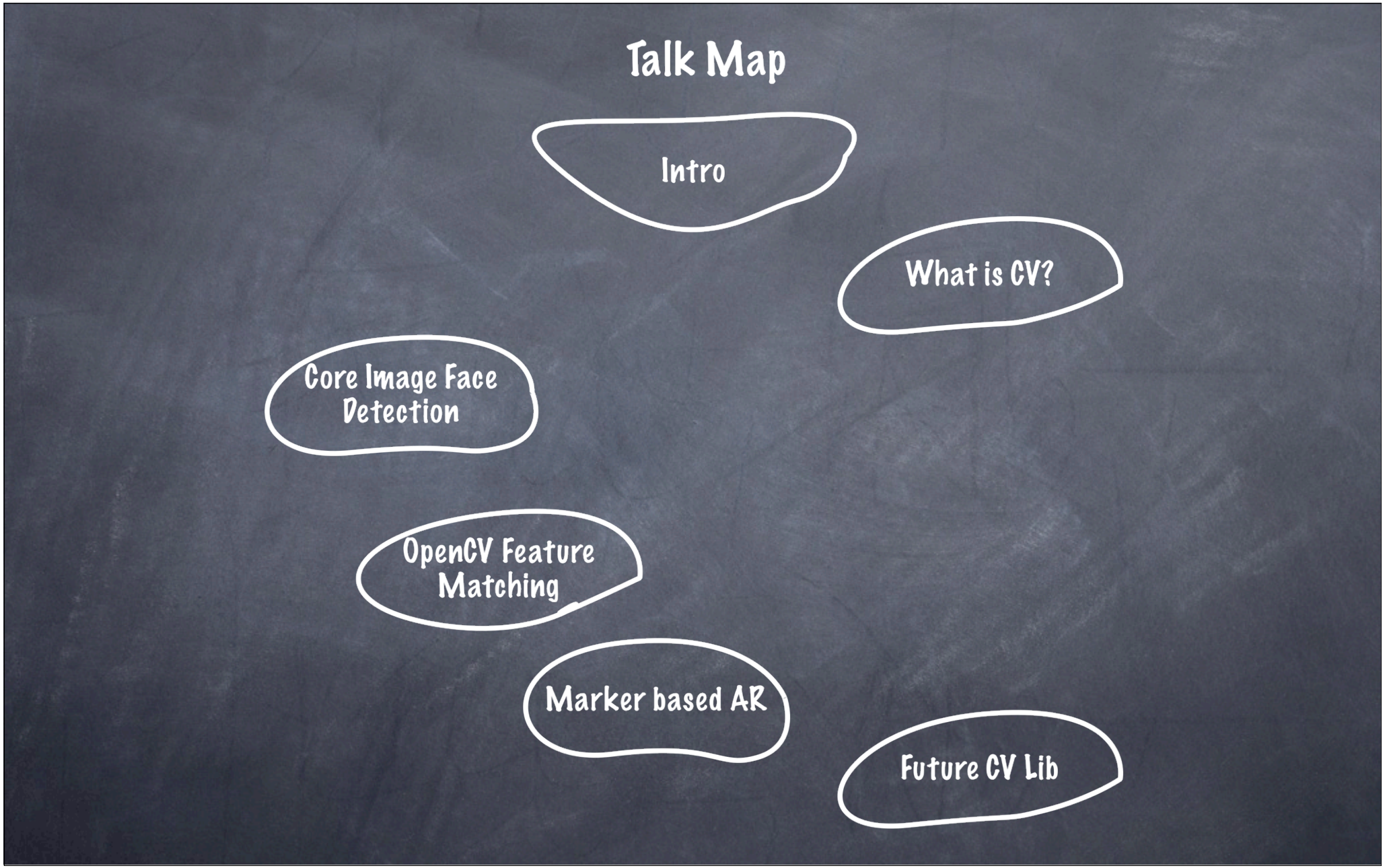
iOS Computer Vision

CocoaConf Chicago 2012

Jonathan Blocksom
Big Nerd Ranch

About Me

- Jonathan Blocksom
  jblocksom@bignerdranch.com
  @jblocksom
- Instructor at Big Nerd Ranch
  Advanced iOS Programming
- 3D Graphics, Computer Vision background
- Worked at neat places


Talk Map
Intro
What is CV?
Core Image Face Detection
OpenCV Feature Matching
Marker based AR
Future CV Lib

What is CV?

Questions about Text

- How many characters / lines?
- How many words / paragraphs?
- Is string in text?
- Difference between doc x and doc y?
- Any misspelled words?
- What's it mean?

Questions about Images

- Statistics about the image?
- Are there any barcodes/lines/blobs/faces/sudoku puzzles in the image?
- Are parts of one image in another image?
- How do you map pixels from one related image to another?
- What is the difference between two images?
- Who or what is in the image?

Text - Image comparison

Existing iOS CV

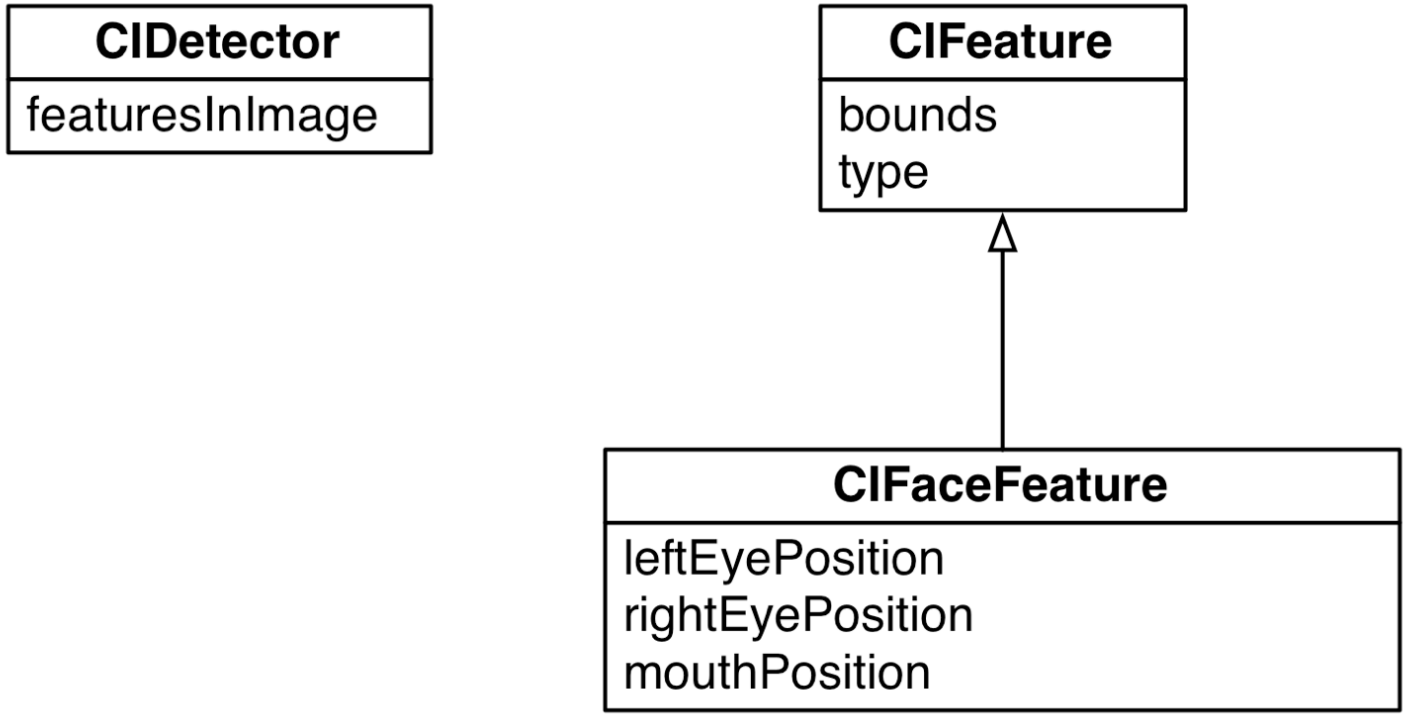
Core Image

- iOS 5
- Image Processing
- GPU Accelerated

- New Image Analysis portion of API

Core Image Face Detection

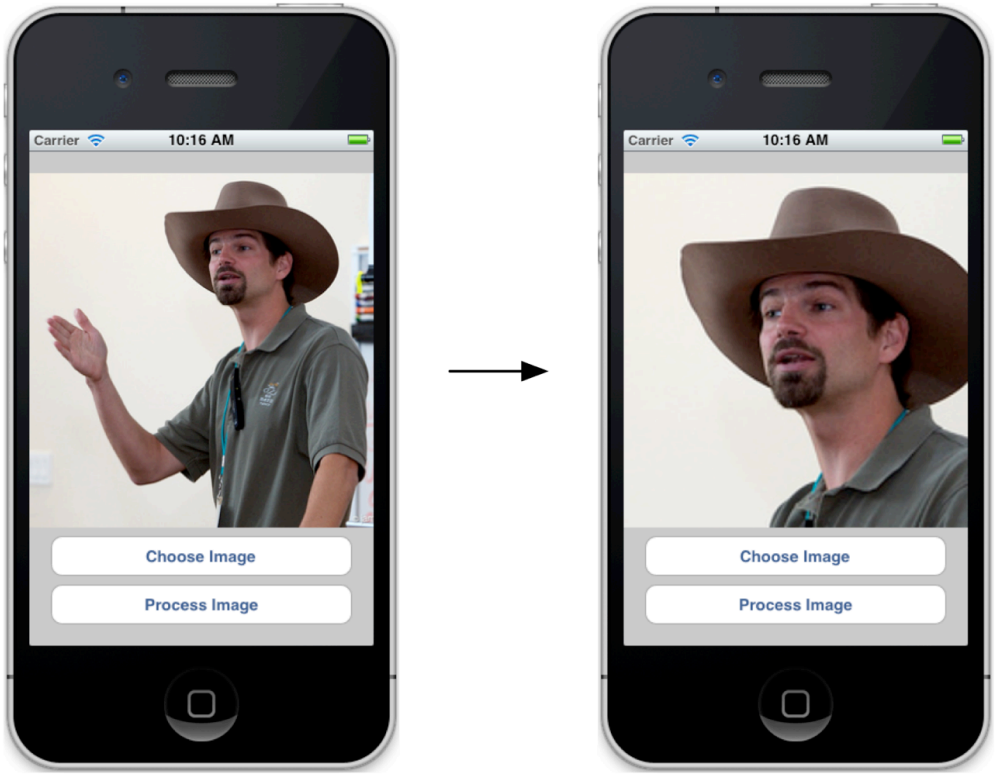
- Face Detection Classes



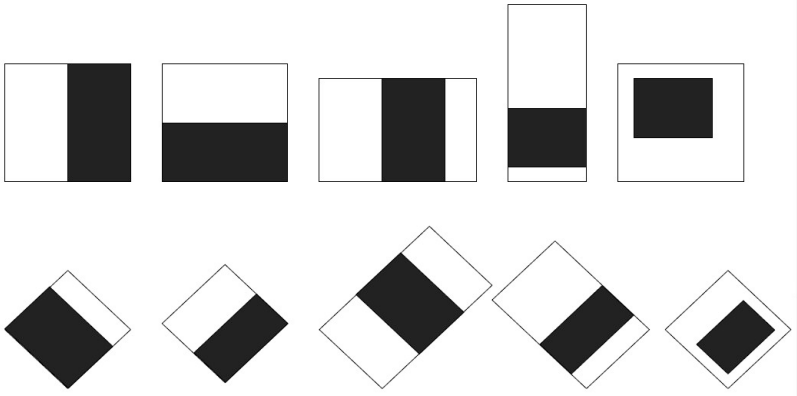
Core Image Face Detection

```
NSDictionary *opts = [NSDictionary dictionaryWithObject:CIDetectorAccuracyHigh
                                                 forKey:CIDetectorAccuracy];

CIDetector *detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:context
                                          options:opts];

NSArray *features = [detector featuresInImage:source];


// Crop result image to face
CIFaceFeature *firstFace = [features objectAtIndex:0];
CIImage *result = [source imageByCroppingToRect:firstFace.bounds];
```

Face Detection Demo

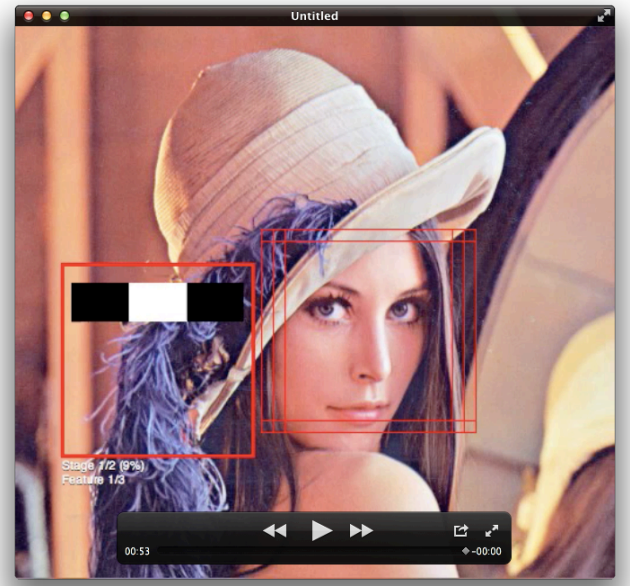


How it Works



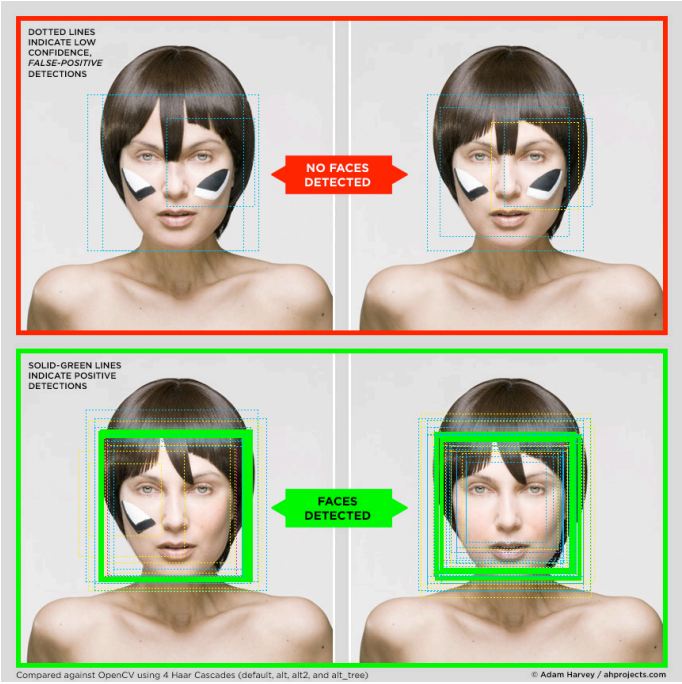
- Classifier run on known images
- Portions of image extracted with “Haar-like Features”
- Compared to results of known values

http://software.intel.com/sites/products/documentation/hpc/ipp/ippi/ippi_ch14/ch14_object_detection_using_haar_like_features.html



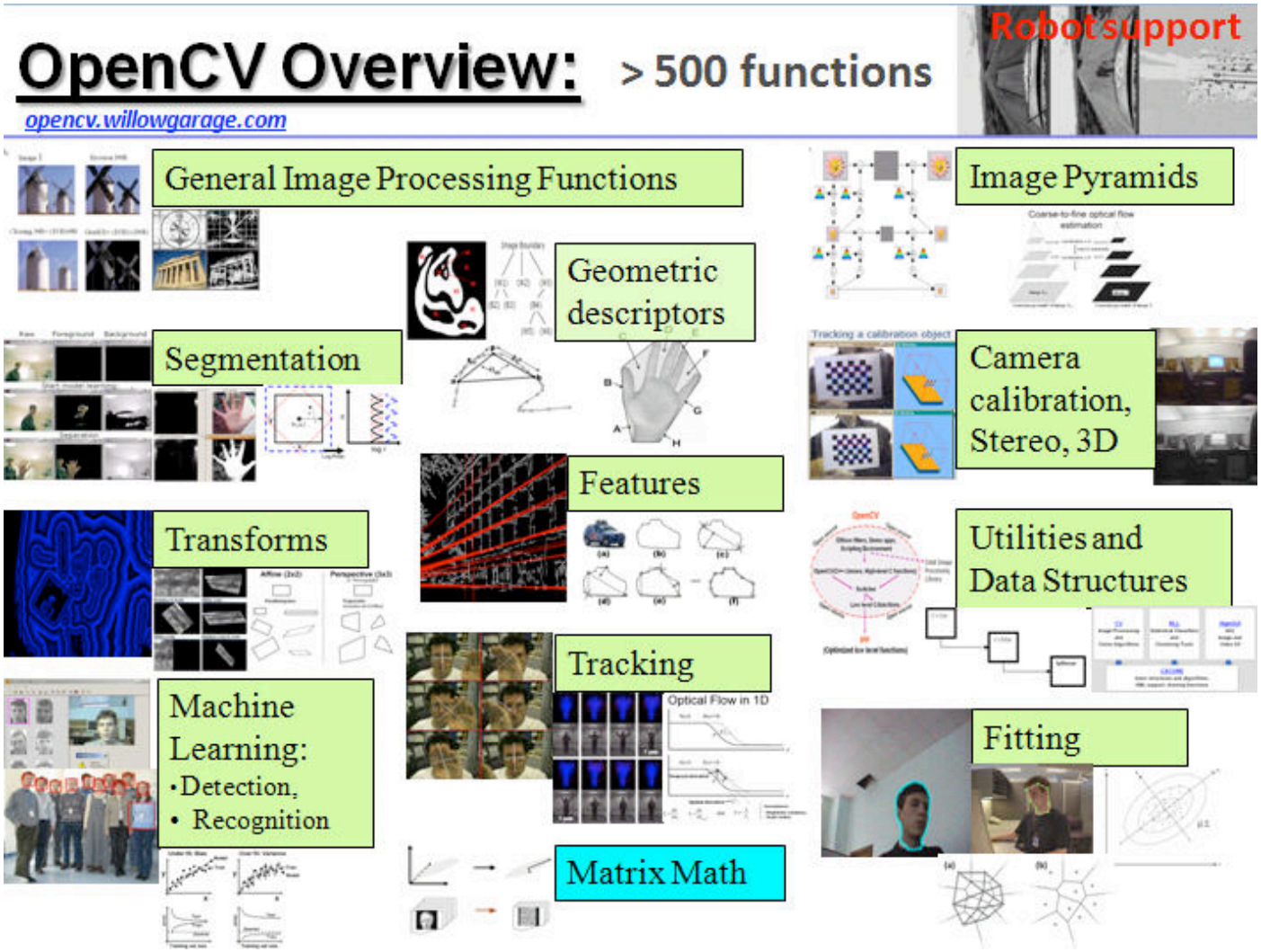
Defeating Face Detection

- CvDazzle project: How to defeat face detection? http://cvdazzle.com



OpenCV

- Open source computer vision library
  Cross Platform (Windows/Linux/Mac/Android)
- Builds for iOS
- C/C++ Based
- “2500 optimized functions”
- Started by Intel, now maintained by Willow Garage
- Current version 2.3



http://opencv.willowgarage.com/wiki/

OpenCV on iOS

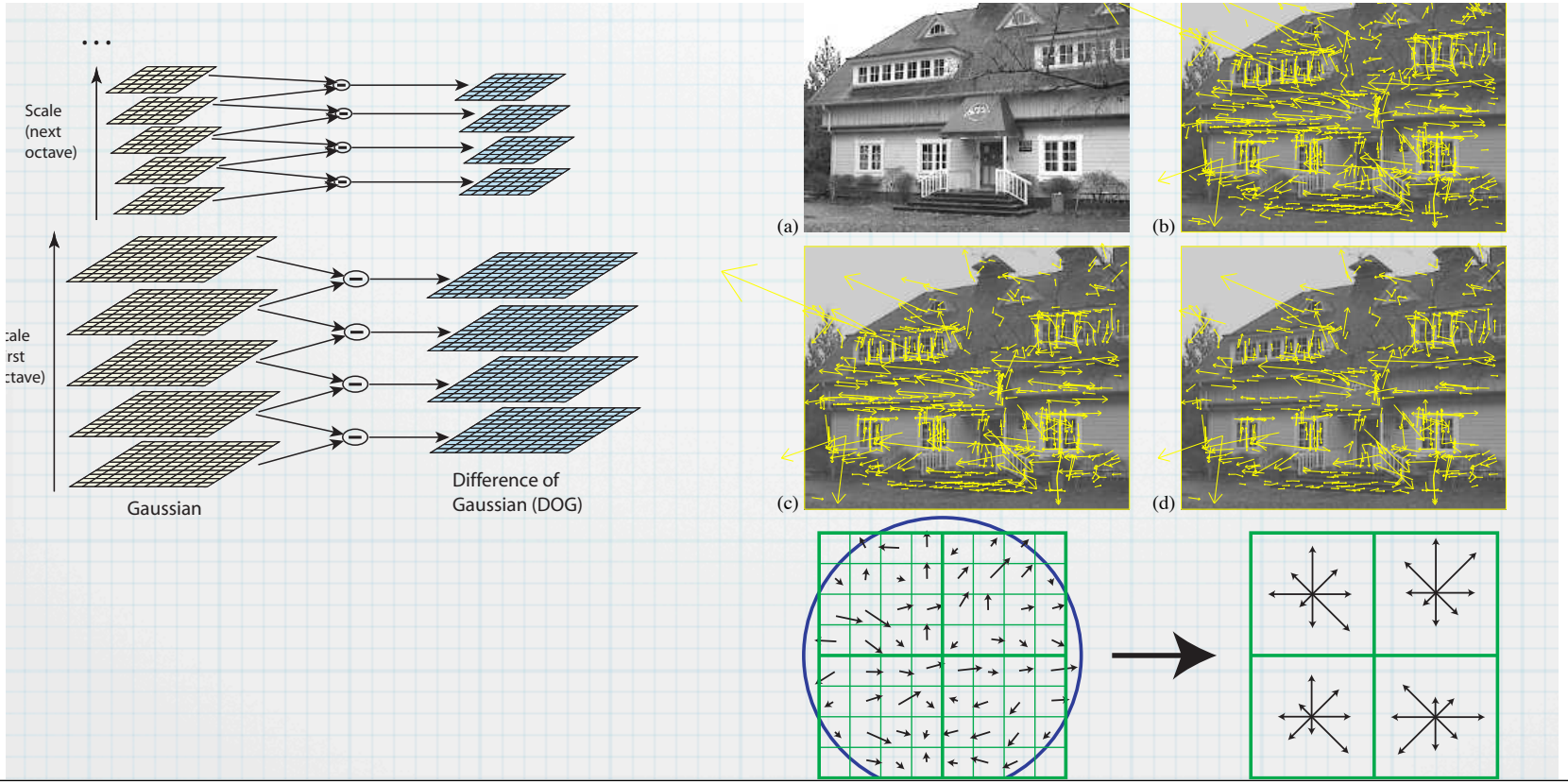
- Build the latest yourself:
  http://computer-vision-talks.com/2011/04/opencv-ios-faq/
- https://github.com/jonmarimba/OpenCV-iOS
  Version 2.2 *This is the one I'm using*
- http://www.eosgarden.com/en/opensource/opencv-ios/overview/
  Version 2.0

Modern CV

- How does image A relate to image B?
- How can we go from a pixel in A to a pixel in B?

Features in Computer Vision

- SURF



Features

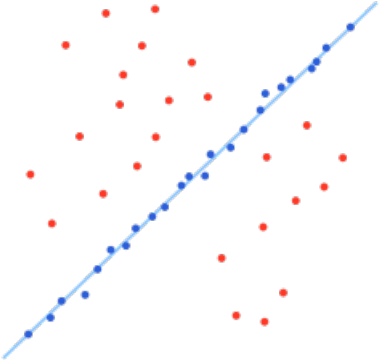
- SIFT / SURF
- Scale Invariant
- Robust to changes in intensity

Feature Matching

- Features are vectors w/ lots of dimensions
- Nearest neighbor search
- Accelerate w/ KD-trees, etc

Robust Transform Estimation

- RANSAC!



What this allows

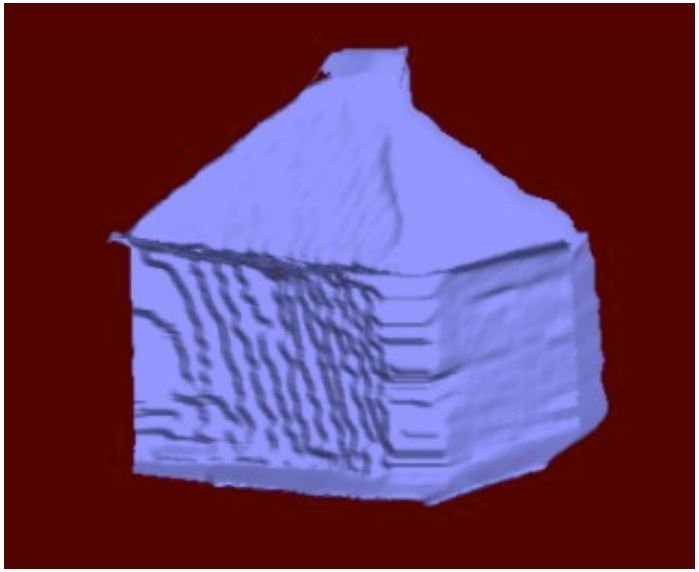
- Stabilization

What this allows

- Mosaics / Panoramas

What this allows

- Georegistration
- Superresolution
- Tracking
- 3D from 2D



Homographys

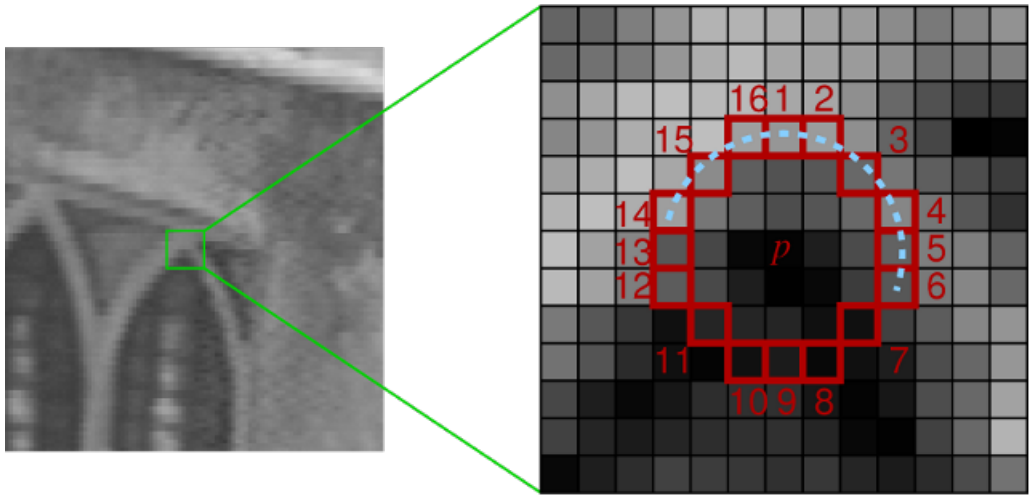
- Fancy name for transform
- CATransform3D, GLKMatrix similar
- Ax = y

Demo

- NerdSURF

FAST Tracking

- Compares brightness of 16 pixels around candidate p
- Accelerated with Machine Learning

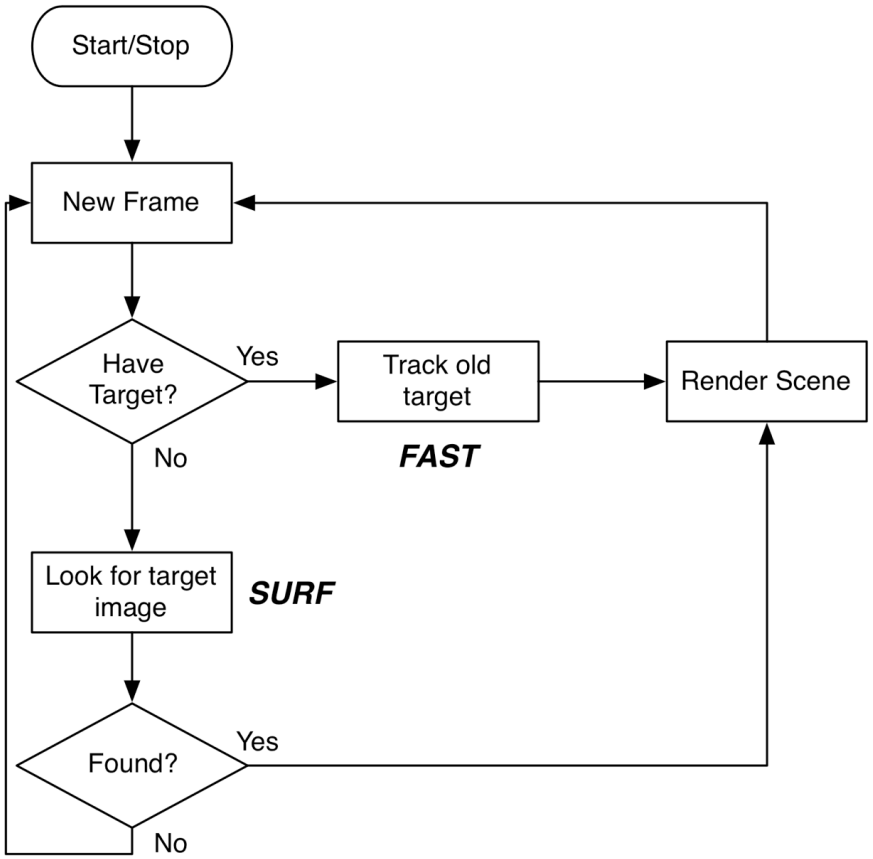


FAST Demo

- iOS Implementation:
  http://www.hatzlaha.co.il/150842/FAST-Corner-V2
- Original algorithm:
  http://www.edwardrosten.com/work/fast.html

Finding + Tracking

- Is an image in the video frame?
- Where did the image go in the new frame?
- What is the 3D transform to the image?
- ... Marker Based Augmented Reality



Marker Based AR Demo

Qualcomm Vuforia SDK


```
- (void)renderFrameQCAR
{
    [self setFramebuffer];

    // Clear colour and depth buffers
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    // Render video background and retrieve tracking state
    QCAR::State state = QCAR::Renderer::getInstance().begin();
    QCAR::Renderer::getInstance().drawVideoBackground();

    for (int i = 0; i < state.getNumActiveTrackables(); ++i) {
        // Get the trackable
        const QCAR::Trackable* trackable = state.getActiveTrackable(i);
        QCAR::Matrix44F modelViewMatrix = QCAR::Tool::convertPose2GLMatrix(trackable->getPose());

        // Choose the texture based on the target name
        int targetIndex = (!strcmp(trackable->getName(), "stones")) ? 0 : 1;

        // Render using the appropriate version of OpenGL ...
    }

    QCAR::Renderer::getInstance().end();
    [self presentFramebuffer];
}
```

Summary

- Face Detection with iOS 5 Core Image
- SURF Feature matching with OpenCV
- Augmented Reality with Qualcomm Vuforia

Beyond OpenCV

- Ugly
- More than we need
- Moving in the wrong direction

```
CIDetector *detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:context
                                          options:opts];
NSArray *features = [detector featuresInImage:source];
```

```
NSString *path = [[NSBundle mainBundle] pathForResource:@"haarcascade_frontalface_default"
                                                 ofType:@"xml"];
CvHaarClassifierCascade *cascade =
        (CvHaarClassifierCascade *)cvLoad([path cStringUsingEncoding:NSASCIIStringEn
                                          NULL, NULL, NULL);
CvSeq* faces = cvHaarDetectObjects(small_image, cascade, storage, 1.2f,
                                   2,
                                   CV_HAAR_DO_CANNY_PRUNING,
                                   cvSize(20, 20));
```

```
NSDictionary *opts = [NSDictionary dictionaryWithObject:CIDetectorAccuracyHigh
                                                 forKey:CIDetectorAccuracy];
CIDetector *detector = [CIDetector detectorOfType:CIDetectorTypeFace
                                          context:context
                                          options:opts];
NSArray *faces = [detector featuresInImage:source];
```

What do we need?

- Objective-C native processing
- Fast -- GPU, Accelerate
- Works with existing APIs
  - UIKit, Core Animation, Core Image, GLKit
- Modern Objective-C Features
  - Properties, Blocks, etc

What do we already have?

- Core Image
  - CIDetector, CIFeature
- Matrices
  - CATransform3D
  - GLKMatrix
- Lots of Points and Vectors

A Call to... Action?

- Open Source?
- Wait for Apple?

- I need a nap