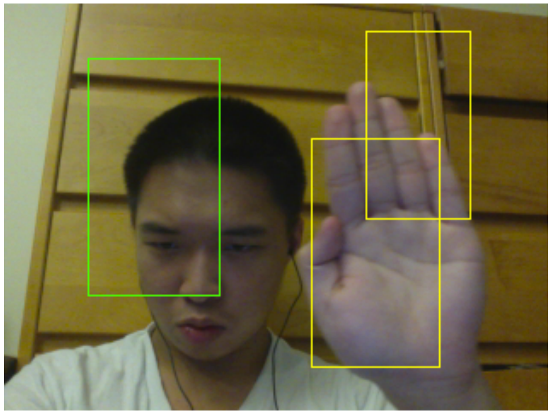
Pipeline for Improving Accuracy of Hand Tracking

By Kenny Yu

Motivation: Implement Kinect-like hand gesture detection capabilities using a commodity webcam (e.g. control Google Maps with your hands)

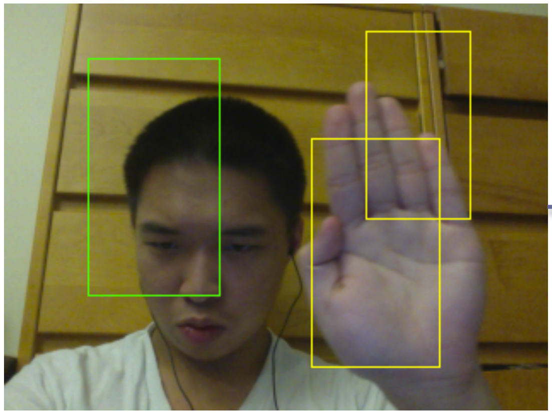
Problem: Given a poorly trained Haar cascade classifier for hand gestures, how can we improve the accuracy of the tracking?

My Solution: Build a Pipeline.

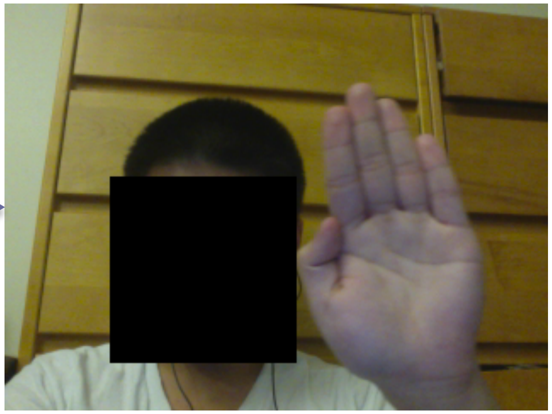


1. Original Haar Cascade

My Solution: Build a Pipeline.

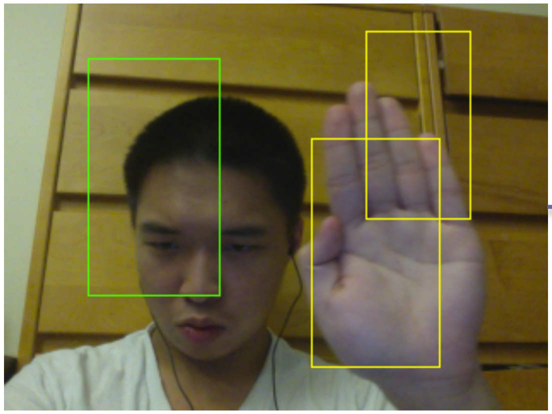


1. Original Haar Cascade

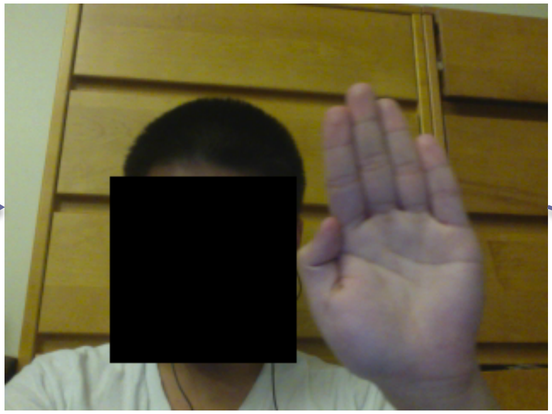


2. Remove Faces

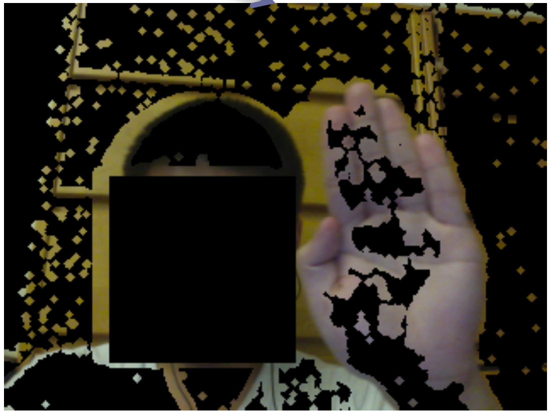
My Solution: Build a Pipeline.



1. Original Haar Cascade

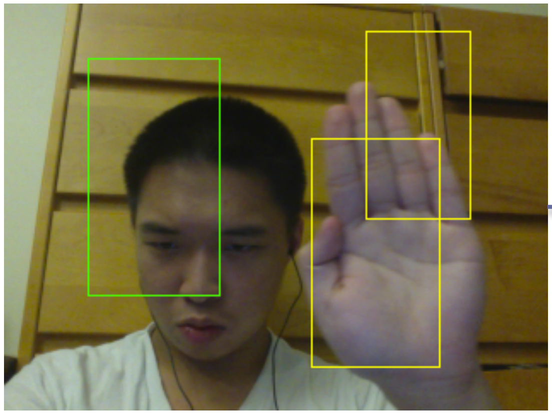


2. Remove Faces

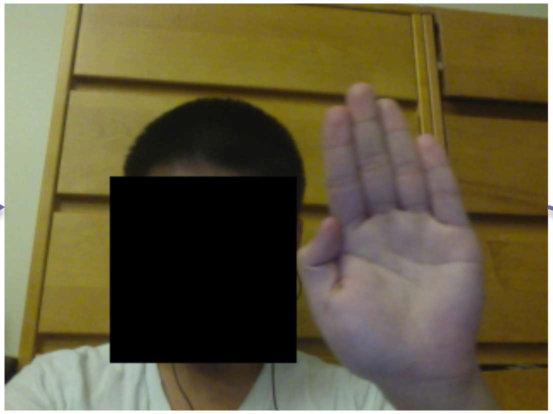


3. Background Subtraction

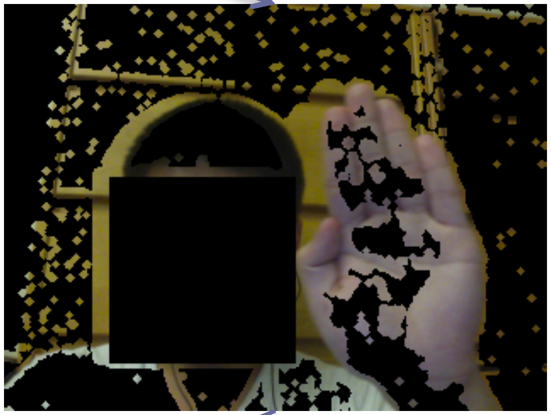
My Solution: Build a Pipeline.



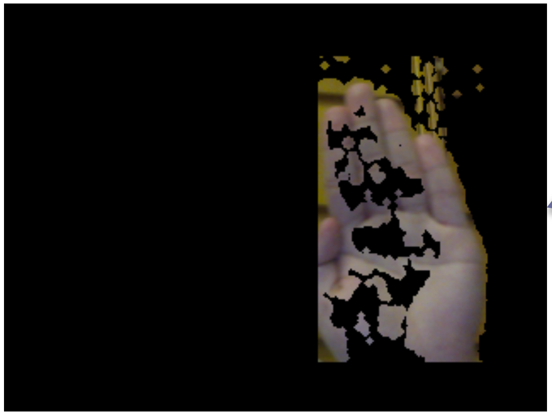
1. Original Haar Cascade



2. Remove Faces

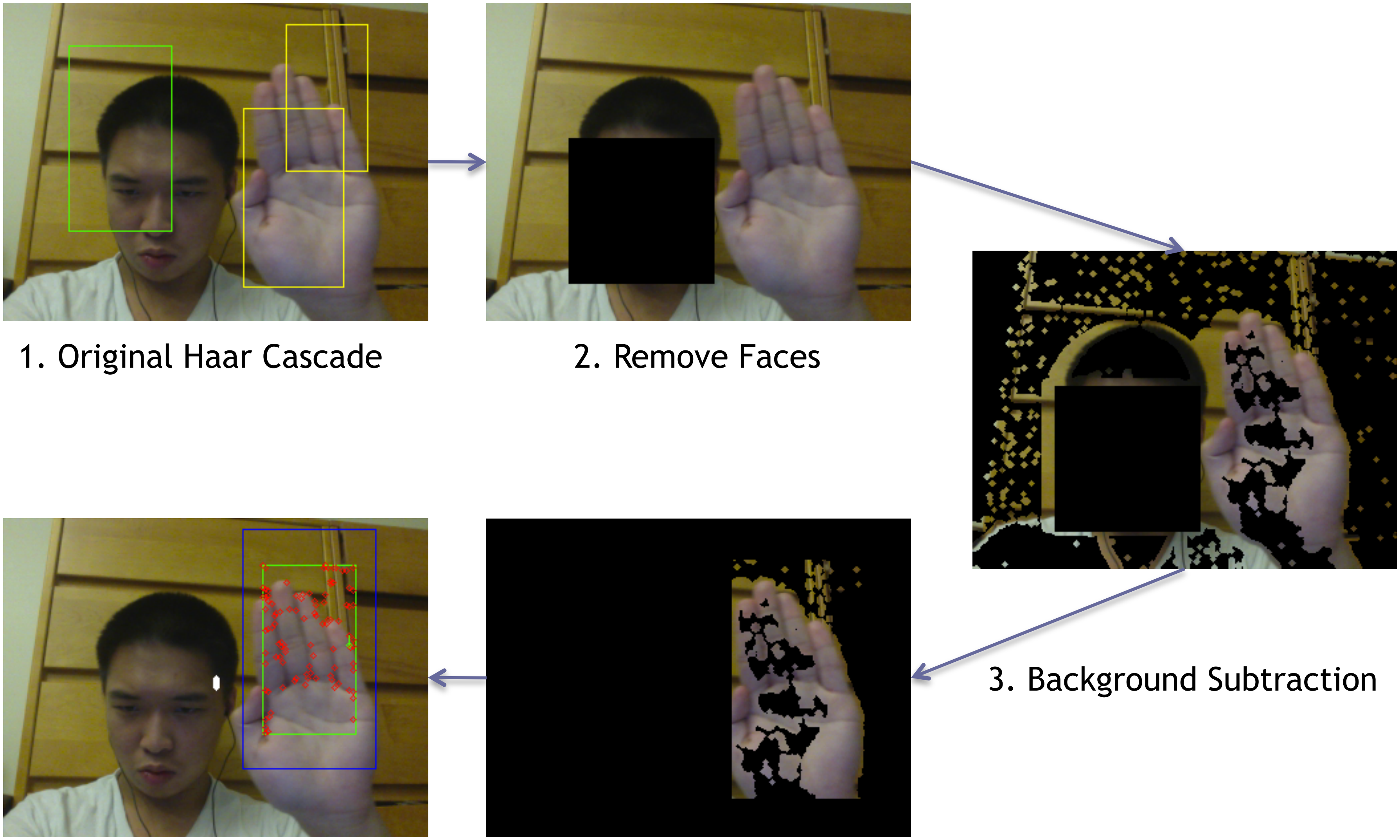


3. Background Subtraction



4. Kalman Filter Prediction

My Solution: Build a Pipeline.



1. Original Haar Cascade

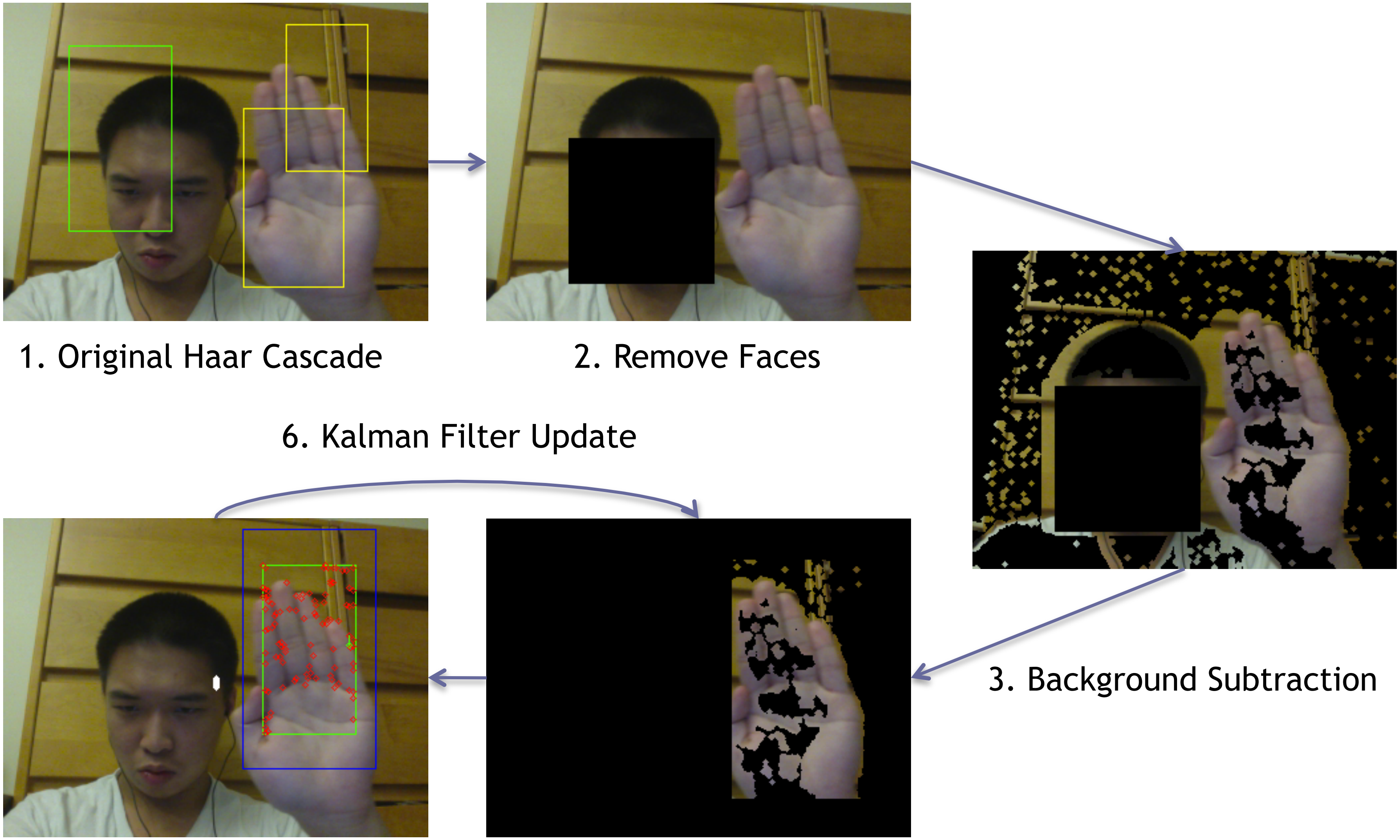
2. Remove Faces

3. Background Subtraction

4. Kalman Filter Prediction

5. Hand Detection; Optical Flow using Lucas-Kanade

My Solution: Build a Pipeline.



1. Original Haar Cascade

2. Remove Faces

3. Background Subtraction

4. Kalman Filter Prediction

5. Hand Detection; Optical Flow using Lucas-Kanade

6. Kalman Filter Update

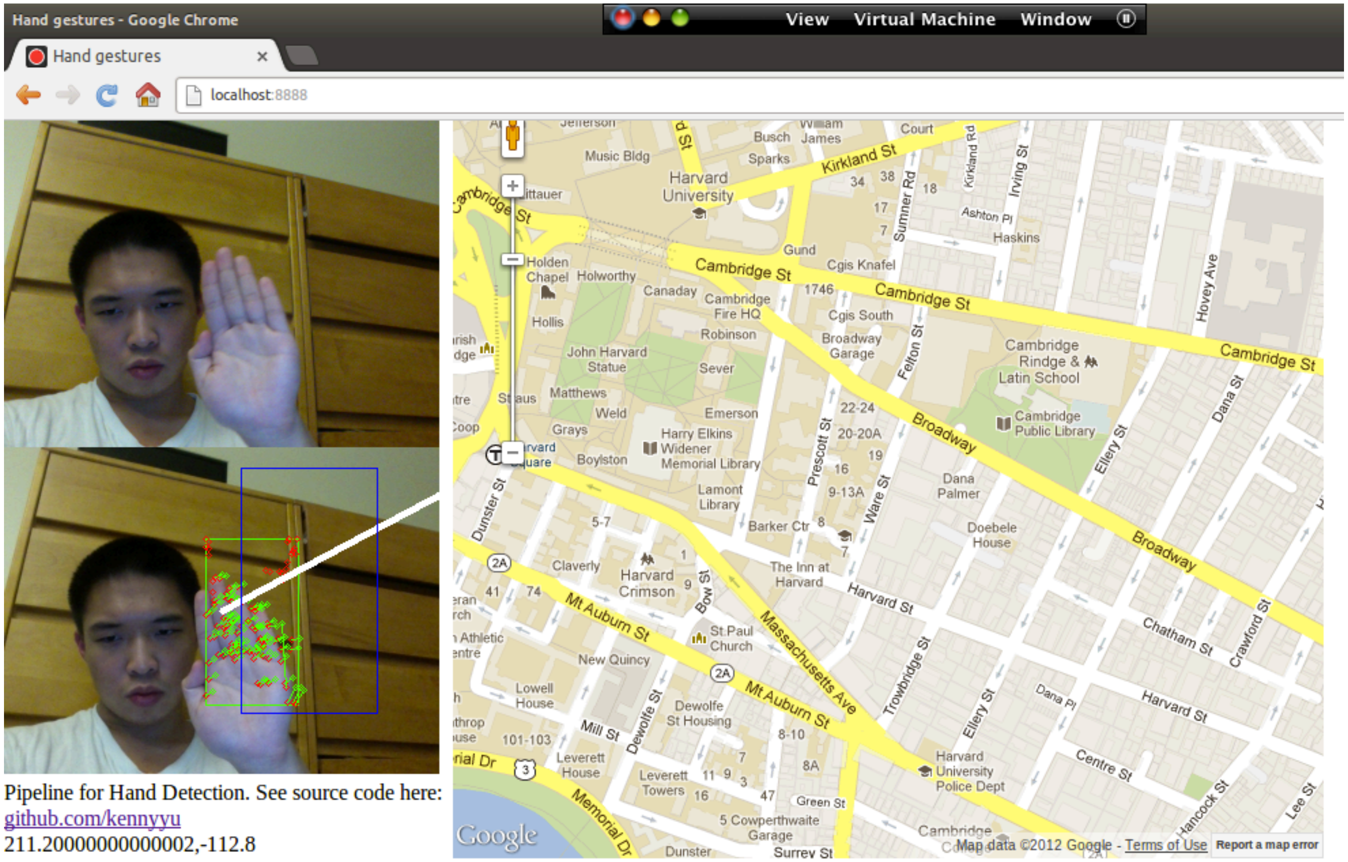
Result: Control Google Maps With Your Hand

Built using:

- OpenCV
- Websockets
- Chrome WebRTC (access to webcam)

Notes:

- Fewer false positives than applying Haar cascade alone
- Sensitive to moving objects in background
- Sensitive to non solid background



Pipeline for Hand Detection. See source code here: github.com/kennyyu
211.20000000000002,-112.8

Need Chrome and webcam to run demo!

Code: http://github.com/kennyyu/cs283-project
Live Demo (you can try it out!): http://goo.gl/pcNxG