CS 229 Project report: Extracting vital signs from video

D.Deriso, N. Banerjee, A. Fallou

December 11, 2013

Introduction

Cardiovascular health is the *sin qua non* of human life. Early detection of cardiovascular disease is of paramount importance in public health. This project aims to develop a method to visualize the perfusion of blood through the skin via pulse oximetry. Pulse oximetry is a technique that exploits the fact that oxygenated and deoxygenated hemoglobin changes the color of red blood cells. The technique maps these changes in rgb color of the visible skin to the invisible presence of oxygenated vs deoxygenated blood in the local vasculature underneath the skin.

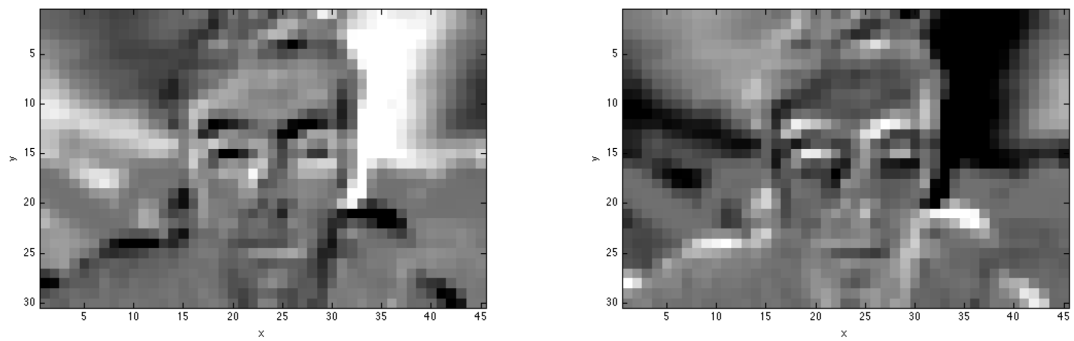
Previous studies have shown that video obtained from an ordinary webcam can be used to visualize perfusion by selectively amplifying temporal frequencies in video [1]. A study by the MIT CSAIL showed that this technique can also be used to infer heart rate from the person being taped. The present project aims to extend this work to detect the relative changes in oxygenated vs deoxygenated blood and reconstruct the pulse oximeter waveform from an ordinary webcam video.

1 Methods

Previous work on Eulerian Video Magnification (EVM), has led to the following processing pipeline, which we later modify for our purpose. The video can undergo different treatments, but the central goal is to amplify spatial or temporal changes that are normally invisible to the naked eye. The process can be summarized as:

- Separate the video into distinct spatial frequency bands by performing a 2D spatial fourier decomposition.
- For each spatial frequency band, blur and downsample several times using Gaussian or Laplacian Pyramids. The first step preserves spatial features (e.g. high frequencies such as edges) through this destructive process.

[1] http://people.csail.mit.edu/mrub/vidmag/



(a) At at a given time

(b) Half a heartbeat period later

Figure 1: Red channel of the EVM output, before recombining with the original video.

- Amplify a pre-selected temporal frequency band.
- Recombine spatial frequencies from step 1, and add the amplified video to the original video.

Figure 1 presents an example for the output of the next-to-last-step.

Strikingly, these two frames showed that the whole video undergoes a periodic color change with a frequency that seems equal to the heartbeat. This did not happen with the data provided in the paper, where in a similar video only the person's face changes color. To extract features relevant to pulse oximetery, we seek out a weighted combination of periodic changes in color space that best reconstruct our training pulse oximetery signal. We therefore undergo the following process:

- Extract each pixels time course from the video.
- For each pixel, separate the colors into R, G, B bands and perform a fourier decomposition where frequency is grouped into n bins.
- Learn weights via linear regression for frequencies within each color band to reproduce the simultaneously recorded pulse oximeter data in the training data set.
- Amplify the temporal frequencies according to the weights of the regression.
- Combine the amplified video with the original video.

2 Preprocessing

Video data

We needed to find a way to convert a video into features. Our feature set initially started out vaguely; we wanted to incorporate the frequency data inherent within each pixel in the video and output frequency data about a pulse oximeter waveform. The three color channel intensity values through time, $I_R(t), I_G(t), I_B(t)$ were our starting points. We converted the video into a four dimensional matrix of pixel location x and y, color channel and time and performed the descontruction mentioned in the previous section.

Pulse Oximeter Data

Pulse oximeter data was collected using an Arduino and a Pulse Oximeter connected to a laptop. A program was written that started recording the pulse ox data and simultaneously started taking pictures every 50-80 ms on a webcam. This gave the pulse ox data corresponding to the image/video data. Thus we had many images taken over small time intervals with a corresponding pulse oximeter values.

Further processing had to be done because the pictures were not taken uniformly in time and so computing a fourier decomposition of the pixels taken from each image would not be correct. An interpolation was needed for images (to create a set of images with uniform time between them) and also for the pulse oximeter data to corroborate with the images.

In order to accurately train our model, we also needed to bin the pulse oximeter values and be able to recreate the signal from the binned values so that they remained true to life.

3 Results and Discussion

Initial results

At first we took 10 bins over a 6Hz interval of the Fourier transform of each pixel. We outputted a binary vector, with a 1 if the binned value exceeded a threshold and a 0 otherwise. The binning involved with our initial results provided a coarse model and was in fact a very poor representation of the data. In particular, for each pixel the binary vector consisted of [1, 0, 0 ..., 0] which is exactly the same distribution we would expect for noise. Thus our weight matrix contained predominantly zeroes, as we were training to a predominantly zero vector.

To make our predictions closer to the ground truth, we increased the bin size, although limitations in our computational power meant that we were restricted to maximum 24 bins over the 6Hz interval. This applied to both the pulse oximeter value and the pixel's Fourier transforms. We also included the phase terms and the pixel locations as further features.

Using the normal equations

Our feature matrix, X, which contained the features for every training example, gave a matrix $X^T X$ with a very high condition number. This meant the normal equations were unusuable and so stochastic gradient descent was used.

Results

Features were extracted from each pixel over time, and included the pixels [i,k] location, the FFT of the pixels time course binned to 4 buckets per frequencies 0-6Hz [fq1, fq2, , fq24], and phase [ph1, ph2, , ph24]. Linear least squares regression models were trained on the following combination of features:

Pixel location	Phase	Frequency	MSE	Max cross correlation
✓	✓	✓	0.00884	0.0936
✓	✓	×	0.114	0.0943
✓	×	✓	0.00903	0.0936
✓	×	×	0.116	0.0943
×	✓	✓	0.0115	0.0936
×	✓	×	0.748	0.0967
×	×	✓	0.0117	0.0936

Table 1: Results of different feature selections with corresponding mean-squared error between predicted and input signal and maximum cross correlation

This table REFERENCE HERE shows that

4 Conclusions

- We built a program to enable training of the data: the program could record a series of images and pulse oximeter readings for those images.
- Using a binned Fourier Transform, we could efficiently reduce the video data to a finite dimensional feature space and perform regression on it.
- Increasing the variety of features included in the video reduced our mean squared error in the linear regression, however the cross correlations were little affected.
- Plenty of further work can be done with this project on a more realistic time-scale. These are outlined below.

5 Further Work

Given the time contraint on this project, we feel we have made great progress. However in order for the project to be considered successful, we have identified some areas where our methodology was non-ideal (due to time constraints or computational power).

- The main limitation on our method is the autoencoding process used for discretizing the FT of the pulse oximeter signal. As can be seen in the graph INSERT GRAPH AND REFERENCE IT if we tried to reconstruct the original pulse ox signal with our compressed version, the compression is a lossy one. Given this lossiness in our compression
- Another area for improvement

References