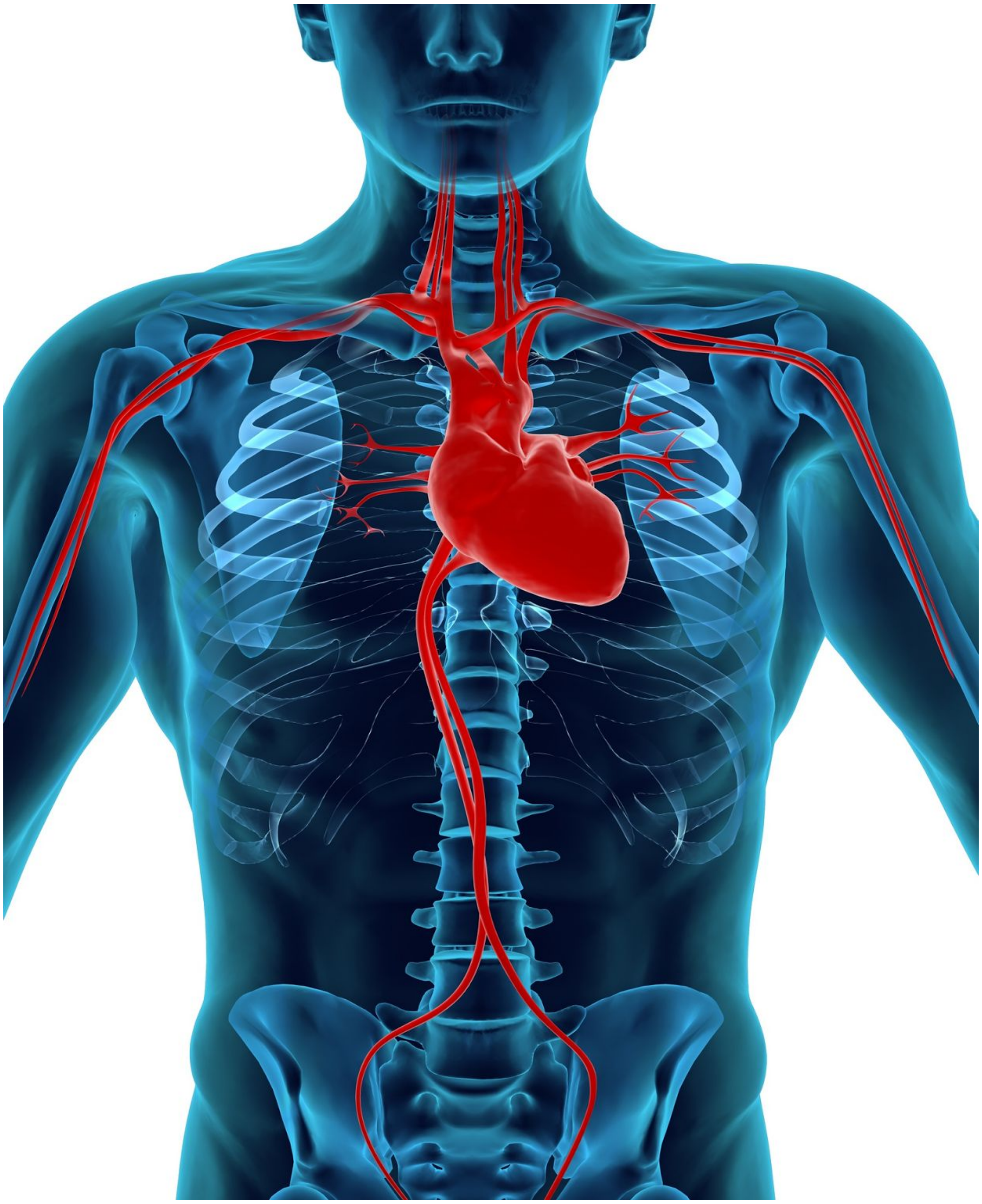

# Eulerian Video Magnification

*Implementing Eulerian Video Magnification to detect subtle changes in motion*

Kiana Alcala & Kathryn Baldauf | Jivko Sinapov, CS378 | Final Project | 5/12/16

# *Table of Contents*

Table of Contents

Abstract

Introduction

Background

Technical Approach

Experiments

Evaluation

Example Demonstration

Conclusion and Future Work

Bibliography

# Abstract

The purpose of this research is to implement Eulerian Video Magnification (EVM). EVM is a technique used to amplify frames within a video to show subtle motions and color changes that the human eye cannot normally see. The first phase of the project involves implementing already existing MATLAB and C++ code and transferring it to a C++ ROS node. Our code works in real time by subscribing to the incoming kinect images, and then applying spatial decomposition as well as temporal filtering to each frame within the captured video stream. The final phase involves the actual amplification of the result, which reveals small motions and color variations. We believe the application of EVM in robots will allow for considerable human-robot interactions not previously found.

# Introduction

While working with the Building Wide Intelligence (BWI) robots over the semester, we noticed the robots didn't have a very reliable way to detect a human in a scene. There are many methods to do this in OpenCV, PCL, and other code bases; however, these methods typically fall incredibly short in consistency and functionality. For example, the HOG person detector in OpenCV will sometimes label segments of people, such as forearms, and cabinets as "people." To eliminate this large variation in accuracy, we theorized that by detecting the heartbeat of a person in a scene the robot could better detect the difference between a human and a false positive. In order to accomplish this, we based our project off of research completed by Wu et al at MIT, *Eulerian Video Magnification For Revealing Subtle Changes in the World* (http://people.csail.mit.edu/mrub/vidmag/).

While Wu et al's focused mainly on the functionality of motion magnification and color amplification, with some insight on how heart rate could be found, we have a series of applications specific to heart rate detection that we intend to implement on the robot after we get it operating.

# Background

### Lagrangian to Eulerian

The researchers at MIT began working with motion magnification in 2005, creating a technique that acted like a microscope for visual motion. In order to do this, they measured visual motions and group pixels together to be modified from a sequence of images from a video. The Lagrangian method consisted of analysis of feature point trajectories and clusters of pixels. Each pixel in the frame is assigned to one of the motion clusters, or layers, which are then stacked on top of each other in the outputting video to describe motion. They are sorted using motion likelihood color likelihood, and spatial connectivity. The final representation is built by using the estimated motions to project each pixel back to a reference frame and ultimately re-render the original video to see the form and characteristics of the magnified motions.

In 2012, research reemerged from an MIT group with some returning members from the 2005 team as well as some new members. This group introduced a new way for revealing subtle changes in the world, which resulted in the Eulerian Video Magnification technique used in this project. The basic approach is to consider the time series of color values at any given pixel and amplify variation in a given temporal frequency band of interest. Therefore, in order to detect the heartbeat of a human, we could specifically select plausible human heart rate temporal frequencies. Unlike the Lagrangian approach, the Eulerian processing does not actually track motion but rather is dependent on the video pyramids and temporal processing that produce the magnification.

# Technical Approach

To implement EVM on the robots, we based our code on previous versions by various Computer Scientists on github as well as the official code released by Wu et al from their work. The general structure of the computation was a follows: read an input video file or video sequence, create an image pyramid based on the end goal, pass each level of the pyramid through a temporal filter to acquire desired frequencies, amplify each level of the filtered pyramid, and finally combine the resulting image on top of the original. Since our goal involved the use of the robots in real time as opposed to just any computer, it was necessary to create a ros node that subscribed to the kinect sensor's images. Every time the node received data from the kinect, the image was copied into an OpenCV matrix. This matrix was then passed through the general structure mentioned above. We found that there were many different ways the general structure was computed, some used temporal IIR filters, some used Butterworth filters, and even some just used a bandpass filter. However, every implementation used Laplacian pyramids for motion magnification and Gaussian pyramids for color amplification.

### Motion Magnification

Motion Magnification is particularly important for revealing subtle movements within a scene. In order to do this, a Laplacian pyramid is computed, as we just want what's most important in a scene without worrying about color. This is done by taking each image and downsizing it then taking the difference between the original image and the current one. This is particularly useful because it just shows the moving edges of important objects within a scene. Next, this pyramid is passed through a temporal IIR filter. This filter is used to enhance the signal of a particular range of frequencies, specifically those within the normal range of a heart rate. After each level of the pyramid is filtered, the image is amplified to exaggerate the filtered image. Finally, the pyramid is collapsed and the result is placed over the original. This process continues while the roscore of the robot is functioning and waits after each image for an new image to compute, ensuring no exceptions or unpredicted results. The problem with this method is that the small motions of a vein pumping blood is usually contained within a larger motion, such as the swaying of the head or wrist. We hope to address this issue in the future.

**Color Amplification**

As blood flows through the body, the skin changes color slightly with each “pump.” Thus, color magnification seemed to be the most intuitive way to exaggerate a heart rate, as most stationary objects do not fluctuate in color, but may slightly move as a consequence of the environment. First, a Gaussian Pyramid is created with each image received from the kinect sensor, since we want to keep in tact the colors. The image is broken down into various levels. At each level, the resolution is decreased and the image is blurred. This allows us to visualize the “overall” colors of an image, necessary in seeing the pumping of blood. If instead we simply picked a point on the face to use, we might end up picking an eye or hair, as each person’s appearance is different. Next, we pass each level of the pyramid through a bandpass filter. This allows us to retain just the frequencies within a specified range, in our case, the typical coloring skin. Next, we amplify the colors of each level of the pyramid and collapse it. Finally, The resulting image is placed on top of the original. Similar to the motion magnification method, we continue this process for each image received while the roscore of the robot is on.

# Experiments

Originally our goal was to detect heart rate, and so we proposed to experiment by using a Fitbit to monitor a participant's heart rate whilst our program also gathered a heart rate, and then compare both to determine accuracy. We still wish to get to this point, but since we were limited on time and ran into several complications, we had to reconstruct what our experiments looked like.

**Method of Experimenting**

Since we were not able to get the heart rate monitor aspect of our program working, we focused on getting each method of detecting subtle changes working at full functionality, as this is the first crucial step in heartbeat detection. Thus, our experiments developed into simply testing whether or not these two features were working correctly. We recorded ros bags in real time with the kinect camera to test more efficiently, comparing the computed output to the original video.

*Experiment Results*

After a period of trial and error, we were able to get our motion detection code working. We were seeing the same results that the MIT researchers saw, where the amplified frames were displaying the subtle changes in motion. We tested the program on both a wrist and a person. When we tested the wrist, we were able to see the subtle pumping of the veins amplified, which makes sense since that area of pixels would have a higher frequency of movement due to the heart beats. In comparison, when we tested the program on a human's face, we were able to see the subtle motions that a head actually makes, whereas to the human eye, it appears to be still. Unfortunately, we were not able to get the color magnification woking before the deadline. We got it to produce something, but it was not the same as the results that were output by the researchers at MIT. We theorize the reason this part of our project doesn't work as intended is that we need to limit the area of reduction to just the face.

## Evaluation

We came into this project with a lot of ideas, especially considering all the applications that could be created using Eulerian Video Magnification. Detecting heart rate definitely was a strong interest, and we felt that if we could get the robot to complete this function properly, it could further be used to identify humans. The robot could then be used in a symbiotic relationship with a human, and have a better understanding of who and when to ask for help. Medically, it could determine if a human is in distress, and alert someone else.

The software written by MIT was written in MATLAB, so we started out at a disadvantage in terms of interpreting and understanding the code presented. We were able to do some searching and find several github programs that involved Eulerian Video Magnification in C++. Between the both of these, we were successfully able to adopt and alter existing code, turning it into a C++ ROS node. From then on, we used OpenCV and the internet to debug, and ultimately got the motion amplification to work.

Although we were not able to get the robot to detect color variations correctly, we have some ideas on how we could get it function properly if we were to continue our research. To start, when we ran our color node on the video, it was taking in all of the pixels from the entire frame when what we really want is for the color variation to be accounted for only on the human's face. In order to do this, we could utilize already written facial recognition code so that the color variation is only applied to the pixels in the face, rather than taking in the background pixels as well. Another variable that we considered was the lighting in the lab, which could have affected the outputted video. The lights were fairly bright, which is evident in the videos we recorded, so possibly having a differently lit setting could help. Also, we did not take into account skin color, which is another variable that would affect the program's output.

# Example Demonstration

Motion: Wrist - https://www.youtube.com/watch?v=Lv7f95ICWSA

Motion: Kiana - https://www.youtube.com/watch?v=AHkIhCLyKeg

Color: Kathryn - https://www.youtube.com/watch?v=DBtb5zUugdY

Code on Github - https://github.com/katiewasnothere/bwi_pulse_detector

# Conclusion and Future Work

Overall, we went into this project ambitious, with a lot of ideas on utilizing the Eulerian Video Magnification technique. Even condensing all of our objectives into a succinct research topic was tough. Although we weren't able to finish every goal that we set up, transferring over a combination of MATLAB and C++ code was an accomplishment in itself. We were able to perform the motion amplification in real time on the robot and make great progress on the color amplification.

In the future, we hope to finish the color variation aspect of our research and use both to visualize the subtle changes that we cannot normally see. We then have several applications we wish to teach the robot. Using the frequencies, we aim to allow the robot to detect a human's heart rate, which would then allow for the robot to verify whether or not the subject is a human or not. In the planning stage, we sought to use a Fitbit to compare someone's actual heart rate with the reading that the robot gave. In addition, we wanted to check if the robot would be able to determine if it was viewing a human based on either sending in input of a human or a picture of a human. We theorized that if the robot was not able to read a heartbeat, then it could assume that the subject of the video was not a human and vice versa.

Furthermore, once all the code runs how it should, we plan to run an experiment in which we place participants in different situations in which heart rates would be varied. We would give the robot a threshold so that it would be able to determine which heart rates are "dangerous", and from there could find a human with a normal heart rate to alert. We want the robot and human to have as best of a symbiotic relationship as possible.

# Bibliography


We would like to thank the authors of the following code:

https://github.com/wzpan/QtEVM

https://github.com/thearn/webcam-pulse-detector

https://github.com/kgram007/Eulerian-Motion-Magnification