Color Amplification

By

Jose Carrillo

Cynthia Barajas

Advisor: Tina Smilkstein

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2015

Table of Contents

List of Figures and Tables

Abstract

The goal of this project is to design and implement a hardware version of MIT's and Cal Poly's color amplification system. MIT created a software based system named the "Eulerian Video Magnification" that can reveal small variations that cannot be seen with the naked eye by processing the input sequence of the video. The process consists of using a Gaussian pyramid with a Gaussian kernel in order to do spatial decomposition. This process continues until there is one pixel left which then gets amplified and added back onto each pixel of the original frame for reconstruction. The reconstructed signal can then be output onto a screen where the change in color of a person's face as blood circulates can be visualized.

The purpose of creating a hardware version is to increase the processing speed by parallel processing. It could eventually be made small enough to fit in a watch where the user will be able to check the blood flow on the go. This is important for people who may have blood circulation problems due to diabetes, kidney failure, and such [6]. It would allow them to recognize the blood circulation problem early on before it causes major problems.

Introduction and Background

Measuring the pulse rate gives important information about your health. Any change from your normal heart rate can indicate a medical condition [5]. A fast pulse may signal an infection, dehydration, etc. and in emergency situations can help determine if the patient's heart is pumping. For people incorporating healthier lifestyles, a pulse measurement can reveal information about a person's fitness level and health. In addition, newborn babies have a tendency of having irregular breathing patterns as well as Sudden Infant Death Syndrome (SIDS) [1][2]. Parents tend to worry about a newborn baby's health thus their stress level rises which is unhealthy for both the parent and baby.

Measuring a person's pulse and monitoring a newborn baby's breathing are one of many applications that can be realized with a real time hardware implementation using MIT's Eulerian Video Magnification algorithm [8]. Pursuing research on this noninvasive system can help monitor a person's heart rate with ease in real time and gather accurate and precise data to determine health. Because it operates in real time it can optimize the response time for a health problem thereby potentially saving a person's life.

Some of the current products and our solution to measure a person's pulse and monitor a newborn baby's breathing are explained below:

One product that measures a person's pulse is the Garmin FR70 wristwatch as seen in Figure 1. The device requires the user to wear a chest-strap heart rate monitor which then connects wirelessly to the watch [3]. It allows continuous heart rate tracking and is able to sync with other compatible devices.

Figure 1: Garmin FR70 Wristwatch

A product that measures a newborn baby’s breathing is the Snuza Hero which is a portable device that clips onto a baby's diaper that monitors and ensures that normal movement is maintained [1]. The device will alert the parent/guardian if the baby's movements are weak. If no abdominal movement is detected for a period of 15 seconds the device will vibrate gently which will be enough to arouse the baby. After three vibrations with no abdominal movement then the device will alert the parent/guardian. The Snuza hero trio is the product shown in Figure 2 shows the Snuza Hero with a wireless camera and a video monitor that can track a baby's movement, display room temperature, and has live video.

Figure 2: Snuza Hero Device

MIT’s system requirements were to reveal hidden information (temporal variations) from videos that are near impossible to detect with the naked eye by processing and amplifying specific color variations. The method MIT implemented was called the Eulerian Video Magnification where they took video sequence as an input and applied spatial decomposition followed by temporal filtering to the frames. The filtered signal is then amplified which reveals the hidden information where they were able to visualize the flow of blood in the face as well as small motions. According to MIT arterial problems can be recognized with facial blood flow. This could be a quicker and easier way to detect certain blood flow problems throughout the body.

There is also another software based system that was made by previous Cal Poly students. The pixels are down sampled based on the Gaussian Kernel and Gaussian Pyramid. The grouped pixels continue to be down sampled until one pixel is left which will then pass through the FIR filter and be amplified to the respective color. The final image will show the amplified color. For example, if a video of a person's face is taken, then the software will magnify those pixels and the outcome is being able to see the skin turn red as blood circulates.

Our Color Amplification hardware version will be implemented using a camera as the input to an FPGA where an algorithm will be implemented for the spatial decomposition. The FPGA will also contain the filtering, amplification, and reconstruction processes. All of these processes will be done in parallel which will increase the processing speed compared to the sequential process done in the software version. We also predict that it will do so without compromising precision and accuracy of the output with minimal lag time compared to the models stated.

Cal Poly's algorithm uses a similar technique to MIT's when it comes to separating the colors to red, green, and blue before the Gaussian Kernel is implemented, then the final value is filtered and amplified at the end of the process. The difference between the two methods is the filtering which is not implemented on each processing step and is only used on the final iteration. This saves time when processing the data and trades accuracy and precision of the output value for time. The advantage of this is that the algorithm developed by the Cal Poly students could be run in real time whereas the original MIT algorithm required the dumping of a video into a file, running the algorithm on that file, then playing the resulting file to see results. The problem with the tradeoff will depend on the significance it has at the result of the output which gives up details of changes that occur in only a small area of the frame. In addition, the type of filter that was used was also changed: Cal Poly changed the Infinite Impulse Response (IIR) Filter that was used in the MIT used to a Finite Impulse Response (FIR) Filter in order to widen the bandwidth. This was done because of the range of heart rates that were monitored which can vary from person to person and that the FIR filter was able to handle a wider frequency range. The advantage of putting the algorithm into hardware is that there is no need for a tradeoff. The hardware is fast enough to calculate and filter all levels of the Gaussian

pyramid and therefore give better results than the simplified algorithm used by the previous Cal Poly group.

In comparing our device to products that are already out in the market as mentioned previously it can be seen that our product would allow us to obtain data using non-invasive products. The Garmin FR70 wristwatch uses a chest wrap which may be uncomfortable for users and the Snuza Hero needs to be clipped to the baby's diaper. Our device would be perfect for newborn babies, because they have to sleep on their backs for the first year in order to prevent SIDS [4]. Our device can capture the heartbeat of a baby without disturbing the baby.

I. Functional Decomposition

Figure 3 and Figure 4 demonstrate the level zero and level one block diagrams for the Color Amplification device, respectively. The level zero diagram includes the two main inputs and the output of the color amplification device.

Table 1, Table 2, Table 3, **Error! Reference source not found.**, and Table 5 describe the inputs and outputs of each block in the level one block diagram (Figure 4).

Table 1 has the inputs and outputs of the memory: the input is the signal coming from the camera that will be processed by the FPGA, the outputs send the same data, but one to the Gaussian Pyramid for processing and one to the reconstruction in order to put the video back together. Table 2 has the inputs and outputs of the Gaussian pyramid: the input signal is coming from the memory block in order to do averaging on the pixels while it outputs the different levels. **Error! Reference source not found.** has the inputs and outputs for the FIR filter: the nput comes from Gaussian pyramid where the values will be filtered, and the output is any value between 60 to 120 Hz. Table 4 has the inputs and outputs for the amplification section: the input comes from the filter and the values that go through will be amplified and sent to the output. Table 5 has the inputs and outputs of the reconstruction section: the two inputs coming in are from the memory and the amplification sections because they will be processed together in order to get the video output.

Figure 3: Level Zero Block Diagram

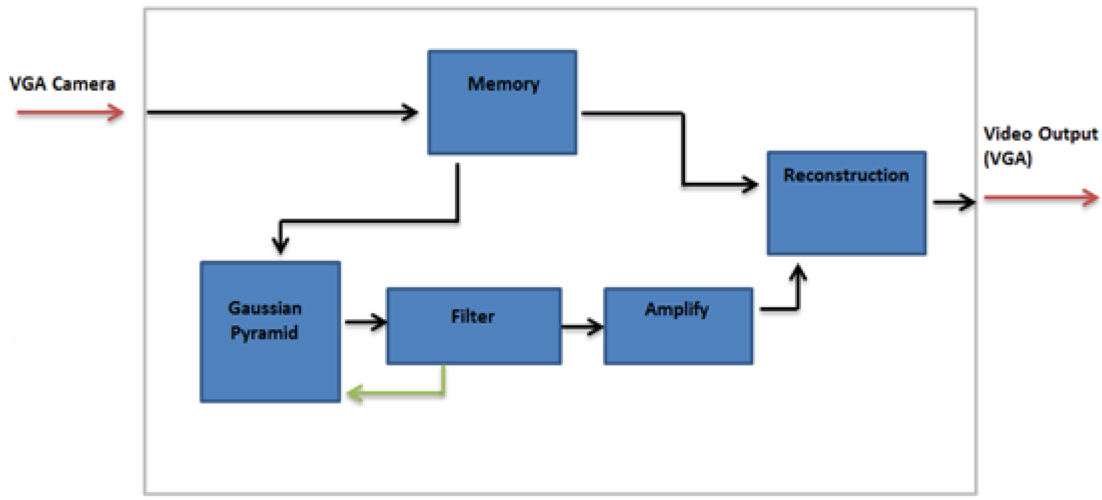

Figure 4: Level One Block Diagram

Table 1: I/O's for Memory

Input/ Output	Number of Bits	Type	Description
Input	8	Video Data	The VGA signal from the camera is processed by the FPGA.
Output	13068	Video Data	The frame that is saved is then sent to the Gaussian pyramid block.
Output	13068	Video Data	The frame is sent to the reconstruction block.

Table 2: I/O's for the Gaussian Pyramid

Input/ Output	Number of Bits	Type	Description
Input	13068	Video data	The frame that was saved in the memory block is passed to the Gaussian pyramid block.
Output	13068	Video data	Outputs all of the Gaussian pyramid's level's frame to the filter block.

Table 3: I/O's for the FIR Filter

Input/ Output	Number of Bits	Type	Description
Input	13068	video data	The input is all the levels of the Gaussian Pyramid
Output	13068	pixel	Output all of the filtered pixels for each level

Table 4: I/O's for Amplification

Input/ Output	Number of Bits	Type	Description
Input	12	video pixel	The FIR filters the pixel.
Output	12	Amplified pixel	The filtered pixel is amplified by constant.

Table 5: I/O's for Reconstruction

Input/ Output	Number of Bits	Type	Description
Input	13068	video data	The original frame that was saved in the memory block is passed on to the reconstruction block.
Output	13068	Video output	The filtered pixel is added to each of the original frame's pixel and the output is sent to an external monitor via the VGA port of the FPGA

II. Design and implementation

A. Camera:

The camera being used is a VGA OV7670 CMOS camera module with the capability of outputting a 640x480 frame size as shown in Figure 5. Figure 6 demonstrates the inputs/outputs of the camera. The SIO_D pin is the serial interface data I/O; the SOI_C pin is the serial interface clock input; the RESET pin can clear all registers and resets them to their default values; the XCLK pin is the system clock input; the D[7:0] pin is the output data; the VSYNC pin is the vertical sync output; the HREF is the horizontal output; and the PCLK is the pixel clock output.

Figure 5: OV7670 CMOS Camera

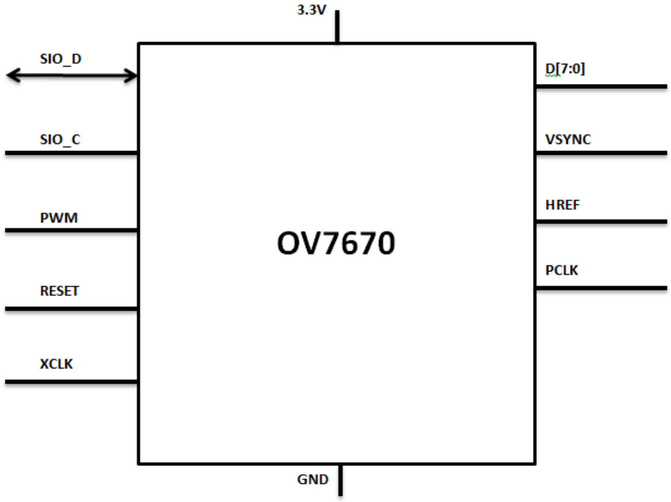

Figure 6: OV7670 CMOS Camera Black Box Diagram

The camera outputs one pixel at a time that is 8 bits in length (4 red bits, 2 blue bits, and 2 green bits), but the program that is used converts each pixel into 12 bits. It does this in order to make each colored pixel into 4 bits. Each pixel is then sent to a module called “Row.”

B. Row and Memory:

The Row module starts to store each pixel in a single variable until a whole row (33 pixels) is filled. Once a whole row is filled, it will trigger a value that prompts what memory it will be stored in. This is due to having 3 memories with a width of 396 bits (one row which contains 33 pixels and each pixel is 12 bits long) and a depth of 11 (33 rows and 3 memories); therefore, each row is stored in a different memory for faster access, see Figure 7. For example, the first row is stored in address 0 of the first memory; the second row is

stored in address 0 of the second memory, and so forth. This speeds the process of accessing three rows at one time instead of waiting three clock cycles to have all the data.

To read the memory, three multiplexers (muxes) are used because there are three memories to choose from as well as what address from that memory. There were obstacles when attempting to create this module because of the components required. There was an obstacle with finding a way to send the signal for what address to read from the specified memory because some of the addresses were repeated from one reading to the next as seen in Table 6. From each set of three rows are accessed in parallel where the “Row” and “Address” are integers, and the “Mux_Sel” is in binary (2 bits).

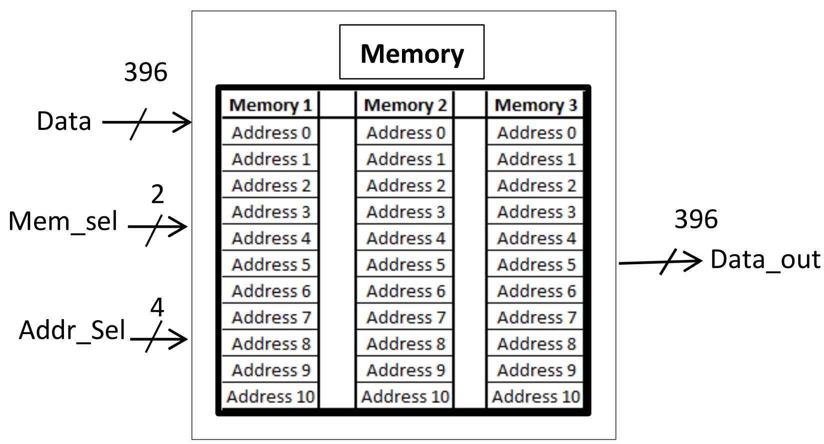

Figure 7: Memory Block Diagram

Table 6: Key for Memory Access

Row	Address	Mux_Sel
0	0	00
1	0	01
2	0	10
2	0	10
3	1	00
4	1	01
4	1	01
5	1	10
6	2	00
6	2	00
7	2	01
8	2	10
8	2	10
9	3	00
10	3	01
10	3	01
11	3	10
12	4	00
12	4	00
13	4	01
14	4	10
14	4	10
15	5	00
16	5	01
16	5	01
17	5	10
18	6	00
18	6	00
19	6	01
20	6	10
20	6	10
21	7	00
22	7	01
22	7	01
23	7	10
24	8	00
24	8	00
25	8	01
26	8	10
26	8	10
27	9	00
28	9	01
28	9	01
29	9	10
30	10	00
30	10	00
31	10	01
32	10	10

C. Gaussian Pyramid and Gaussian Kernel:

The Gaussian Pyramid is an image processing method which uses a series of weighted down images using the Gaussian kernel. Figure 8 shows the Gaussian Pyramid technique of blurring which is shown by Level n, the original image. As the Gaussian Pyramid levels decreases from Level n to Level n-1 each of the pixels in the lower level will contain a local average that surrounds the specific pixel. At each level the image decreases by half, for example an image that is 32 by 32 pixels will decrease to 16 by 16 pixels.

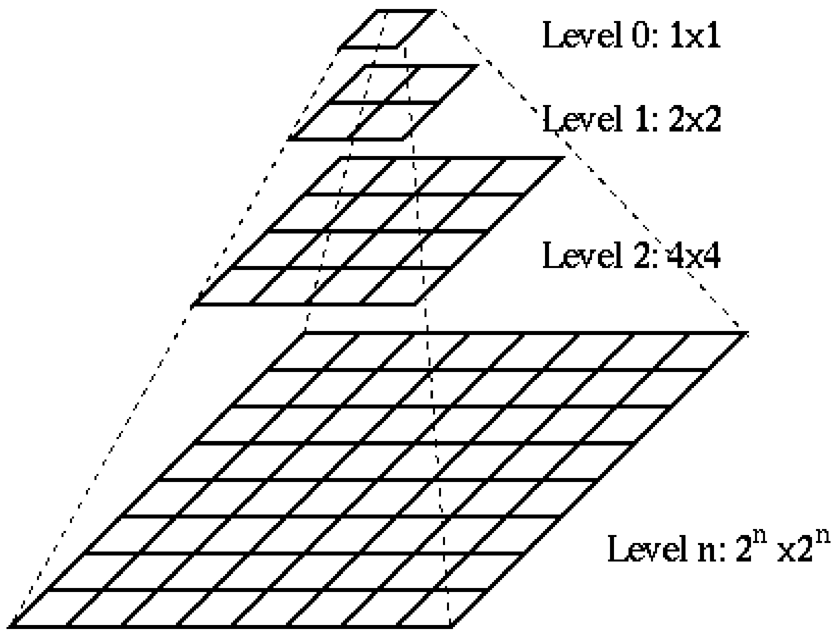

Figure 8: Gaussian Pyramid Scaling down process.[9]

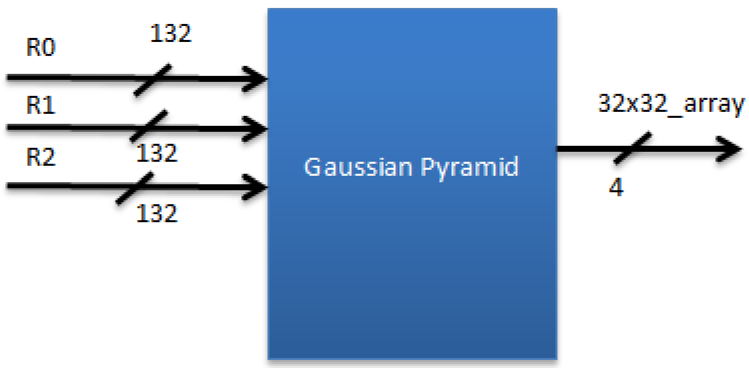

Figure 9: Gaussian Pyramid Black Box Diagram

From the memory, 3 rows are accessed in parallel in order to average the pixels with the use of the Gaussian Pyramid and the Gaussian Kernel which is shown in Figure 9. The Gaussian kernel is an odd matrix that is used to get the average of an area of interest which consists of 9 values that surround the pixel. For example, the averaging for a 3x3 frame is shown below:

$$3x3\ Frame = \begin{bmatrix} X_{01} & X_{02} & X_{03} \\ X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \end{bmatrix}$$

$$Gaussian\ Kernel = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

In order to reduce the 3x3 frame down to one pixel, averaging is done on the surrounding pixels as shown below.

3x3 Frame

$$\begin{bmatrix} X_{01} & X_{02} & X_{03} \\ X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \end{bmatrix} \rightarrow \begin{bmatrix} \cancel{X_{01}} & \cancel{X_{02}} & \cancel{X_{03}} \\ \cancel{X_{11}} & X_{12} & \cancel{X_{13}} \\ \cancel{X_{21}} & \cancel{X_{22}} & \cancel{X_{23}} \end{bmatrix} = \bar{X}_{12}$$

The original frame is first passed to the Gaussian Pyramid. The original frame gets multiplied by placing the Gaussian Kernel on top of the original frame. For example, the pixel X_{12} is multiplied by 4 from the Gaussian Kernel and the surrounded pixels are also multiplied by their corresponding numbers and added together. Then the resulting number is divided by the number of pixels being added. After the pixels are averaged the frame is reduced by half by deleting every other row and column. This process is done for every pixel until only one pixel is left. The calculations for the Gaussian Pyramid are shown below:

$$\overline{X_{12}} = \frac{4X_{12} + 2[X_{02} + X_{11} + X_{22} + X_{13}] + 1[X_{01} + X_{03} + X_{21} + X_{23}]}{16}$$

The averaging is done on each set of 3 rows that is accessed and then the original frame and the subsequent levels are sent to the reconstruction module.

1. Modular implementation of the Gaussian Pyramid:

Figure 10 shows a module that is an implementation of the Gaussian Pyramid which averages two pixels. The input is a 5x3 frame which can be seen in Figure 11. In order to keep debugging at a minimum a module was built that can calculate the averages of two pixels at a time. In

addition to keeping debugging at a minimum the module provides a simpler solution to averaging every pixel referencing one module.

In order to preserve the most information without losing the information in the pixels the implementation shown in Figure 11 was used over taking the Gaussian pyramid of a 3x3 frame and using the mirrored values of the first row and first column.

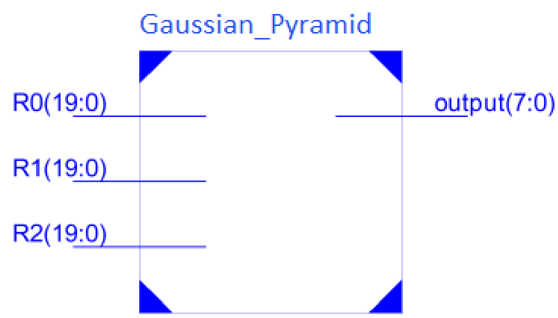

Figure 10: Gaussian Pyramid module.

Figure 11 shows the Implementation of the block diagram shown in Figure 10. In order to get the average of each pixel the first three columns are used to calculate the average of the pixel X11 and the last three columns are used to calculate the average of pixel X13.

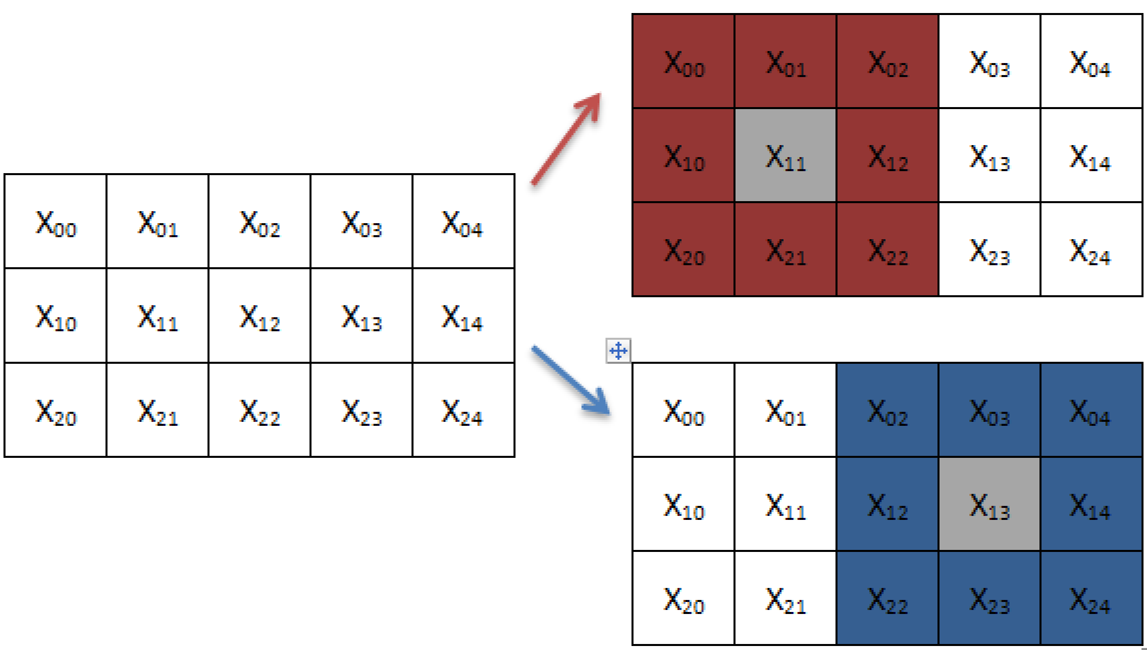

Figure 11: Gaussian Pyramid VHDL implementation diagram

The Gaussian Pyramid module is used to calculate all of the pixel averages for each level. A matlab script file was made in order to generate the port maps of the Gaussian Pyramid module. Figure 12 shows the output of the script file that was made.

```
GP2_1_1: Gaussian_Pyramid
port map(
 R0 => R0(131 downto 112),
  R1 => R1(131 downto 112),
  R2 => R2(131 downto 112),
  Output_frame => m2_1_1
);
GP2_1_2: Gaussian_Pyramid
port map(
 R0 => R0(115 downto 96),
  R1 => R1(115 downto 96),
  R2 => R2(115 downto 96),
  Output_frame => m2_1_2
);
```

Figure 12: VHDL GP implementation

Reconstruction:

For the reconstruction process, each of the pixels in all the levels excluding the original frame will get multiplied by a constant and then added back to the original frame. As seen in Figure 14, the 1x1 frame value will be added into each pixel of the original frame. From the 2x2 frame each pixel is added one fourth of the original frame original pixel. This is done to get a value that will amplify the average color values. The process is done until every level is added to the original frame. The reconstruction process is also shown in Figure 13. Each level of the Gaussian Pyramid represents an average of a certain region with level 0 representing the whole frame.

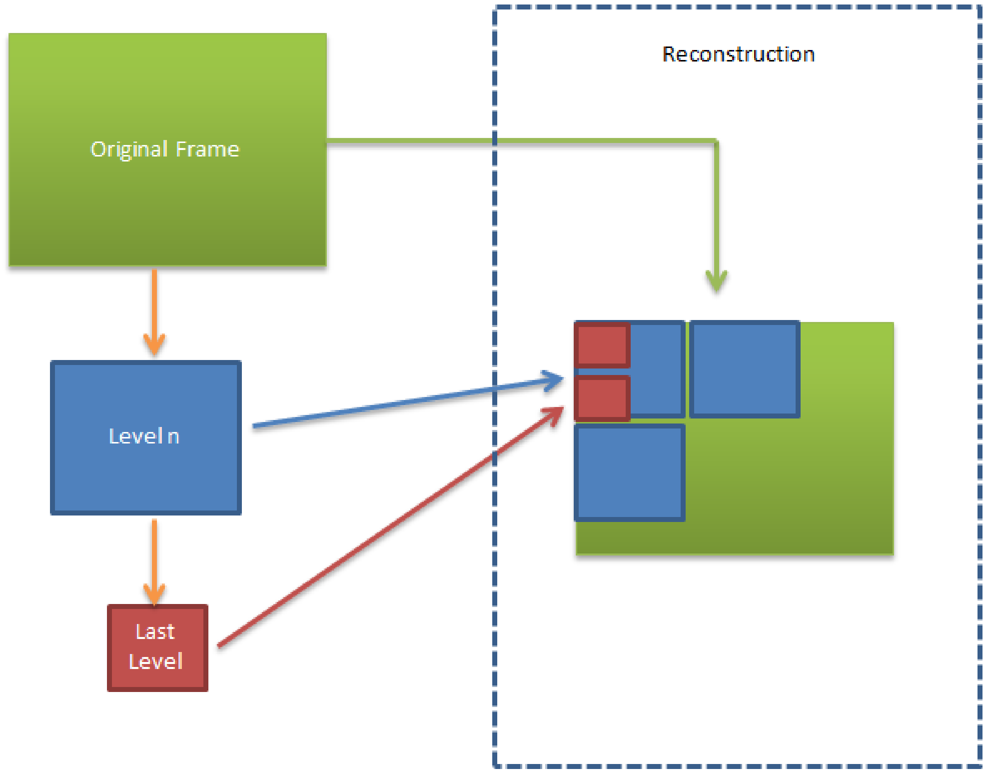

Figure 13: The block diagram of the reconstruction process.

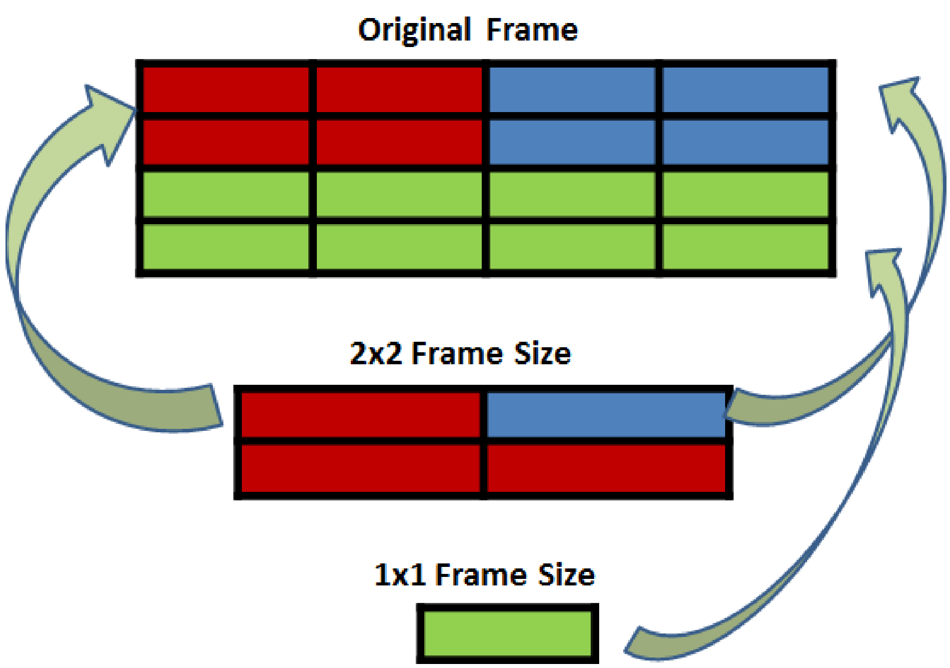

Figure 14: Lower 3 Levels of the Reconstruction Process

III. Testing

A. Gaussian Pyramid:

Figure 15 shows the simulation of the Gaussian Pyramid module using ISE Design Suite's Isim. For the initial testing of the Gaussian Pyramid module all of the inputs where set to '1' or logic high. The output shows an 8 bit hexadecimal value of 0xff or 255 which is the expected value.

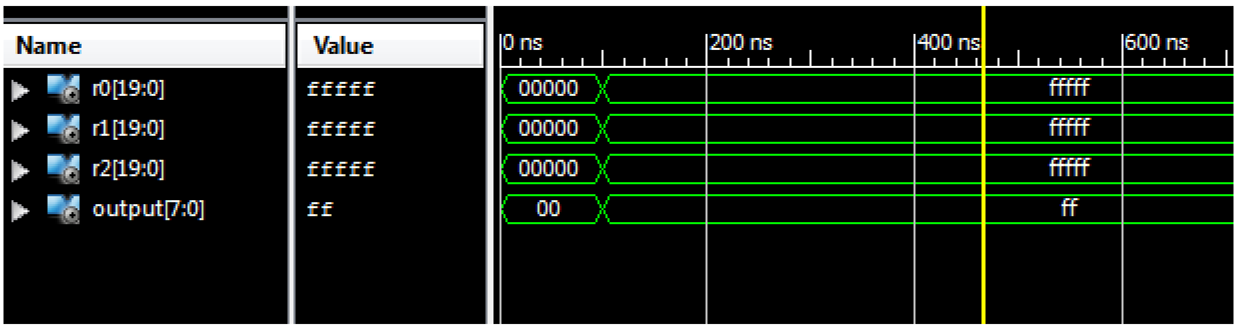

Figure 15: Gaussian Pyramid module simulation.

Figure 16 shows the values of the inputs which are set to either logic high or logic low. The initial test plan to use either logic high or logic low proved useful in determining if the output was correct. Figure 17 shows the values of the inputs and outputs of the software iSim.

```
-- insert stimulus here
   R0<= (others => '1');
   R1<= (others => '1');
   R2<= (others => '1');
   R3<= (others => '0');
   R4<= (others => '0');
   R5<= (others => '0');
   R6<= (others => '1');
   R7<= (others => '1');
   R8<= (others => '1');
   R9<= (others => '0');
   R10<= (others => '0');
   R11<= (others => '0');
   R12<= (others => '1');
   R13<= (others => '1');
   R14<= (others => '1');
   R15<= (others => '0');
   R16<= (others => '0');
   R17<= (others => '0');
   R18<= (others => '1');
   R19<= (others => '1');
   R20<= (others => '1');
   R21<= (others => '0');
   R22<= (others => '0');
   R23<= (others => '0');
   R24<= (others => '1');
   R25<= (others => '1');
   R26<= (others => '1');
   R27<= (others => '0');
   R28<= (others => '0');
   R29<= (others => '0');
   R30<= (others => '1');
   R31<= (others => '1');
   R32<= (others => '1');
   wait for 100ns;
```

Figure 16: Simulation inputs for the Complete Gaussian Pyramid.

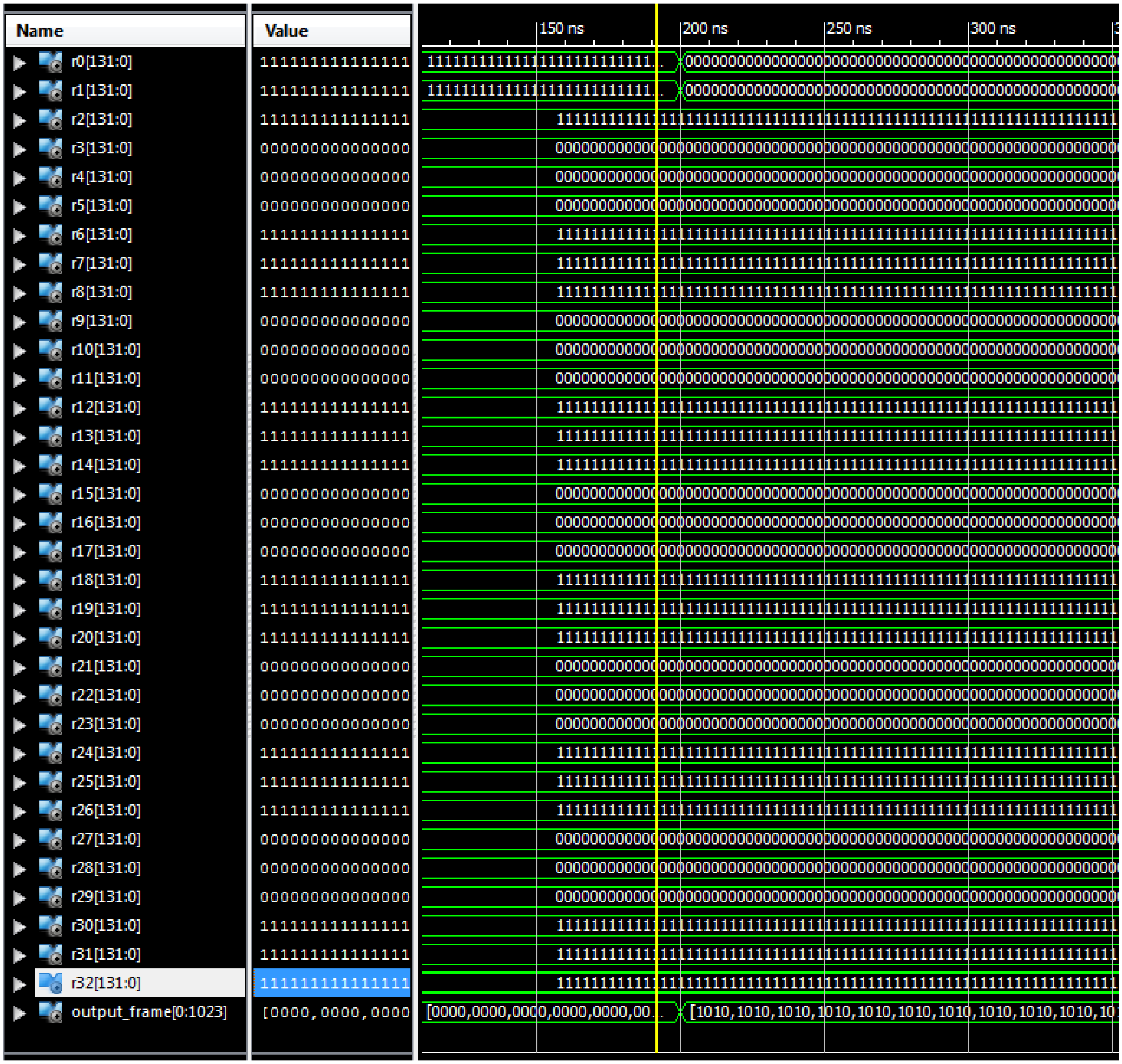

Figure 17: Complete Gaussian Pyramid combined with the reconstruction Simulation.

IV. Conclusion

The color amplification system will be able to detect small variations that cannot be seen with the naked eye. In this case the system did not work since the required blocks were not completed in time for testing. In addition, for the system to be able to monitor blood flow or a heartbeat a filter would need to be implemented before the reconstruction section. The filter would filter out the values that are above or below a 60 Hz to 120 Hz since the heart beats within that range. With the filter implemented the overall system has many uses including variation in color or small movements, heart rate monitor, and blood flow monitor. The

algorithm that is presented by MIT cannot run in real time as the filtering that MIT uses is an IIR filter which has a feedback loop. The algorithm that is presented in this report can run in real time after all of the separate components are built and a smaller frame is used than the frame that first Cal Poly group implemented. Some of the tradeoffs for the Color Amplification project were the FPGA resources that were used and the camera's frames per second. To be able to run the code in real time the frame being processed needed to be reduced from 64x64 to 32x32 pixels.

V. References

[1] *Baby Breathing Monitor*, blog; http://www.baby-breathing-monitor.com/baby-breathing-monitors/.

[2] Chang, Louis MD. "Your Newborn Baby's Breathing Noises," 01 August 2012; http://www.webmd.com/parenting/baby/your-newborn-babys-breathing-noises

[3] "Garmin FR70", 2014, https://buy.garmin.com/en-US/US/into-sports/running/fr70/prod104012.html

[4] Newborn Baby sleeping on stomach, 2014. http://www.livestrong.com/article/213256-can-a-newborn-baby-sleep-on-her-stomach/

[5] "Pulse: MedlinePlus Medical Encyclopedia." *U.S National Library of Medicine*. U.S. National Library of Medicine, 9 Oct. 2014. Web. 20 Oct. 2014.

[6] RAMS Therapy Centre. "Circulatory Disorders"; http://www.ramsni.com/information_links/Circulatory%20Disorders/Circulatory%20Disorders%20info.html

[7] R. Ford and C. Coulston, *Design for Electrical and Computer Engineers*, McGraw-Hill, 2007, p. 37.

[8] Rubinstein, Michael. *Analysis and Visualization of Temporal Variations in Video*. Diss. Massachusetts Institute of Technology, 2014. Boston: Massachusetts Institute of Technology, 2014. Print.

[9] Gaussian Pyramid. "Gaussian-Laplacian Pyramid Image Coding"; http://fourier.eng.hmc.edu/e161/lectures/canny/node3.html

VI. APPENDIX D: ANALYSIS OF SENIOR PROJECT DESIGN

Project Title: Color Amplification

Student's Name: Jose Carrillo and Cynthia Barajas

Student's Signature:

Advisor's Name: Professor Tina Smilkstein **Advisor's Initials:** **Date:**

I. **Summary of Functional Requirements**
   a. The Color Amplification project will be able to process a video without compromising precision and accuracy of the output with minimal lag time compared to models started.

II. **Primary Constraints**
   a. One of the biggest challenges associated with the implementation of this project is finding the hardware that will allow us to do the process in parallel. We have to make sure we have enough memory to store the data needed as well as enough RAM for temporary storage. Another challenge would be finding a pattern for the process in order to create an algorithm that will allow us to do the parallel processing.

III. **Economic**
   a. If this project was funded it could lead to more designing and implementation being done in order to create a smaller and faster version. This system has the potential to eventually be implemented in a small device such as a watch. This would create jobs for jobs for people (i.e. engineers, manufacturers, etc.). In addition it would benefit people with blood circulation problems by being able to detect a problem before it becomes severe.

Initially, the company or research firm who would take on the device would have to invest in upgrading the system to make it smaller and faster. They would also have to test it as they decrease its size to make sure it still works accurately. Once they have the product they want, they will have to invest in manufacturing it and mass producing it. Improving and mass producing the system is where the costs and benefits accrue through the devices life cycle.

Another area where costs would accrue is in advertising and marketing the device. Table 7 shows the estimated cost for the parts and labor. The parts are based off on the components that we will be using for the project while the labor is based off of approximate internship pay since we are still undergraduates. In using Ford and Coulston's equation we can find the estimated cost of the project [7]. The optimistic cost total (T_a) is about $3190.36. The most realistic cost (T_m) is about $4690.36 with the labor pay at $30.00/hour. The pessimistic cost (T_b) is about $5440.36 with the labor pay at $35.00/hour.

$$Cost = \frac{T_a + 4T_m + T_b}{6}$$

Table 7: Color Amplification Estimated Cost

Item	Quantity	Company	Cost
OV7670 300KP VGA Camera Module	1	DX	$7.77
Nexys 4 Artix-7 FPGA Board	1	Digilent	$159.00
Pmod – 16 Mbit Serial Flash ROM	1	Digilent	$23.59
Labor	150 hours		$3000.00 ($20/hr)
		Total	$3190.36

IV. **If manufactured on a commercial basis:**

a. Estimated number of devices sold per year: 5000

b. Estimated manufacturing cost for each device: $220

c. Estimated purchase price for each device: $250

d. Estimated profit per year : $150k
e. Estimated cost for user to operate device, per unit time: $0.1/Hour

V. **Environmental**

a. The environmental impact of the color amplification device will come from the materials used which are the camera, FPGA, wiring, and VGA cable. In the future the cameras will be better and cheaper thus the way that the cameras are disposed is important. If the camera is disposed in the landfill it can eventually release harmful toxic chemicals to the environment.

VI. **Manufacturability**

a. This product will need a lot of funding to create a device that will eventually be manufactured. In the future the board will be much smaller than it currently is because we are using an FPGA thus making the device easier to manufacture. Also when we implement the device using a pcb board, an outside company will be used.

VII. **Sustainability**

a. Once the device is completed there will be room for improvement before it's considered a device that will be manufactured. Some of the improvements are increasing performance and speed since the current components are not the newest. The challenges that would come with upgrading the device is updating the code according to the components that are being upgraded. For example, if the camera is upgraded to a higher pixel resolution then the size of the memory will have to be upgraded so it has the capability of storing the initial input frame.

VIII. **Ethical**

a. Ethical implications that arise with respect to the design and manufacturing of the device are being aware of any errors that may arise. If the algorithm created does not process the entire frame appropriately and it is noticed then it should

be properly addressed and corrected. In the case of manufacturing, if the device is tested and an error occurs somewhere along the process then it should not be ignored because it could cause a bigger problem later on. Some positive ethical implementations are being able to help people detect problems such as blood circulation or monitoring a newborn's breathing before the problem becomes severe. This device has the potential to save lives.

IX. **Health and Safety**

a. One of the main health and safety concerns is that the device will alert a person of any conditions that are problematic (i.e. irregular blood circulation or a newborn baby's breathing becoming irregular), but the person has to take action to prevent major problems. This is something that needs to be made clear to the user.

X. **Social and Political**

a. This device impacts anyone that may have problems that can be detected through temporal variation or anyone who may use the device to monitor a newborn or a person with a disability. The device would indirectly impact the medical field because more patients will go to the doctor to get diagnosed for a specific condition and get treated as the conditions are found. In general, it will impact society as a whole due to having the ability to detect problems that could get severe over time and cause life threatening issues.

XI. **Development**

a. During the initial process we had to research a chip with enough dedicated multipliers (DSP slices) in order to process the frame in parallel to achieve real-time video processing. By looking and comparing different FPGA development kits we found one that was reasonable in price and had enough of dedicated multipliers.