ECE 495 Project 4 (October 21, 2015)
Camera as a Sensor

Team W8: Bode, Bode, Bode, Bode Plotting Everywhere

Ryan Barker
Nicholas Beulick
Peter Gallagher
Rishabh Agarwal
Willie Brown

I. Customer Requirements

The purpose of project 4 was to design a camera sensing system that could determine the color and position of colored stickers on a board. The customer required that a camera sensing system be made and mounted so that the camera could determine sticker colors as well as sticker positions on a stickered game board. The sensing system had to be able to perform with and without out camera filters. When taking an image in to MATLAB background subtraction must be completed. Upon determining all sticker colors and sticker positions the system will output the results to the user.

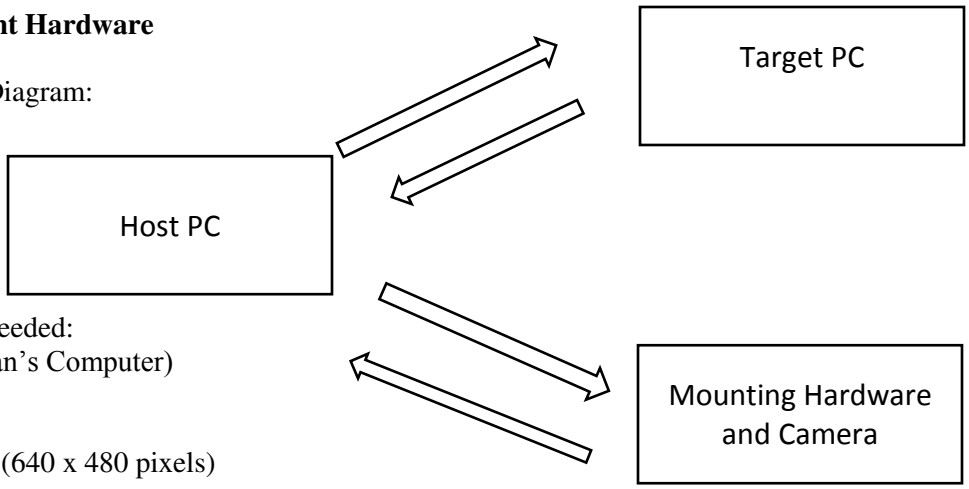
II. Customer and Engineering Requirements

Customer Requirements	Engineering Requirements
Design must determine sticker color	Create a MATLAB algorithm that detects sticker color and returns text to user
Design must determine sticker position	Create MATLAB algorithm that detects sticker position as an angle relative to the center of the board
Design must perform with filters	Test sensing system with filters attached
Camera must be mounted	Create Velcro mount on system
The design must have MATLAB fully integrated	Write code using MATLAB and interface with Simulink blocks
Algorithm must perform background subtraction	Integrate binary background subtraction via MATLAB BITXOR function
The design must be capable of holding filters in front of the camera	Make a Styrofoam mount that can hold filters over the camera
The design must work reliably without filters	Ensure a 99.5% success rate without filters over a series of at least 100 tests

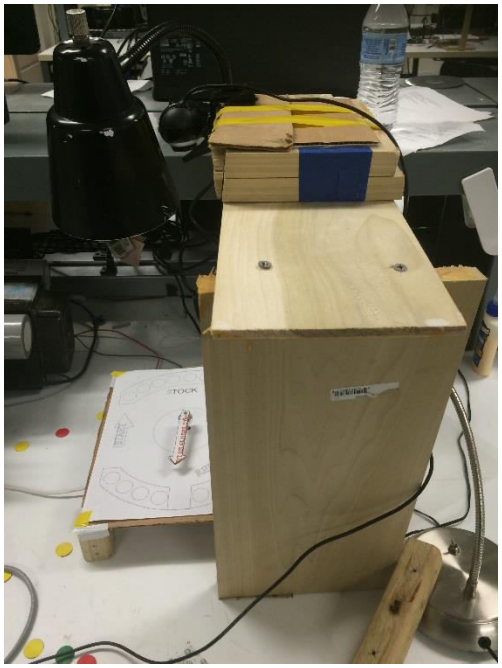
Figure 1. Camera Sensing System Requirements.

III. Document Hardware

Connection Diagram:



Equipment Needed:
Host PC (Ryan's Computer)
Target PC
Wood Mount
Web Camera (640 x 480 pixels)



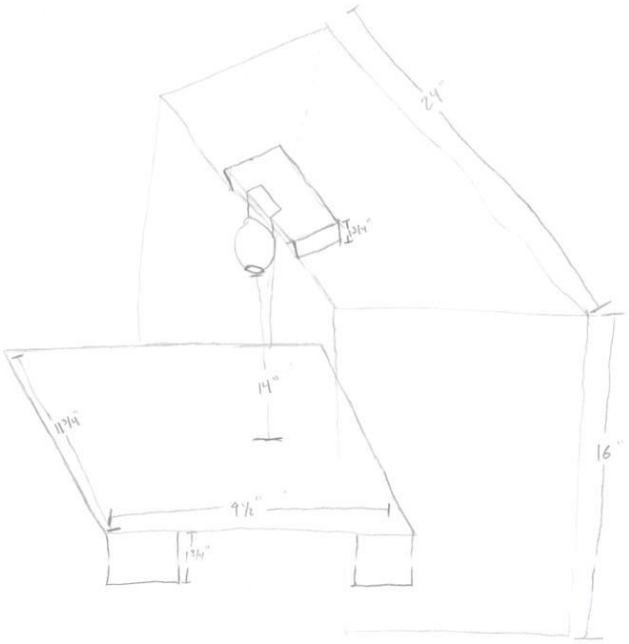


Figure 2. Hardware Design, Robot Sketch, and Connection Diagram.

Calculation of Resolution:
(640 pixels x 480 pixels)/(8.5 in x 11 in) = (X pixels)/(0.75in x 0.75in) → 1848 pix/area of sticker
Since there are 1848 pixels per sticker, this project is a reasonable task for our camera configuration.

IV. Document Software

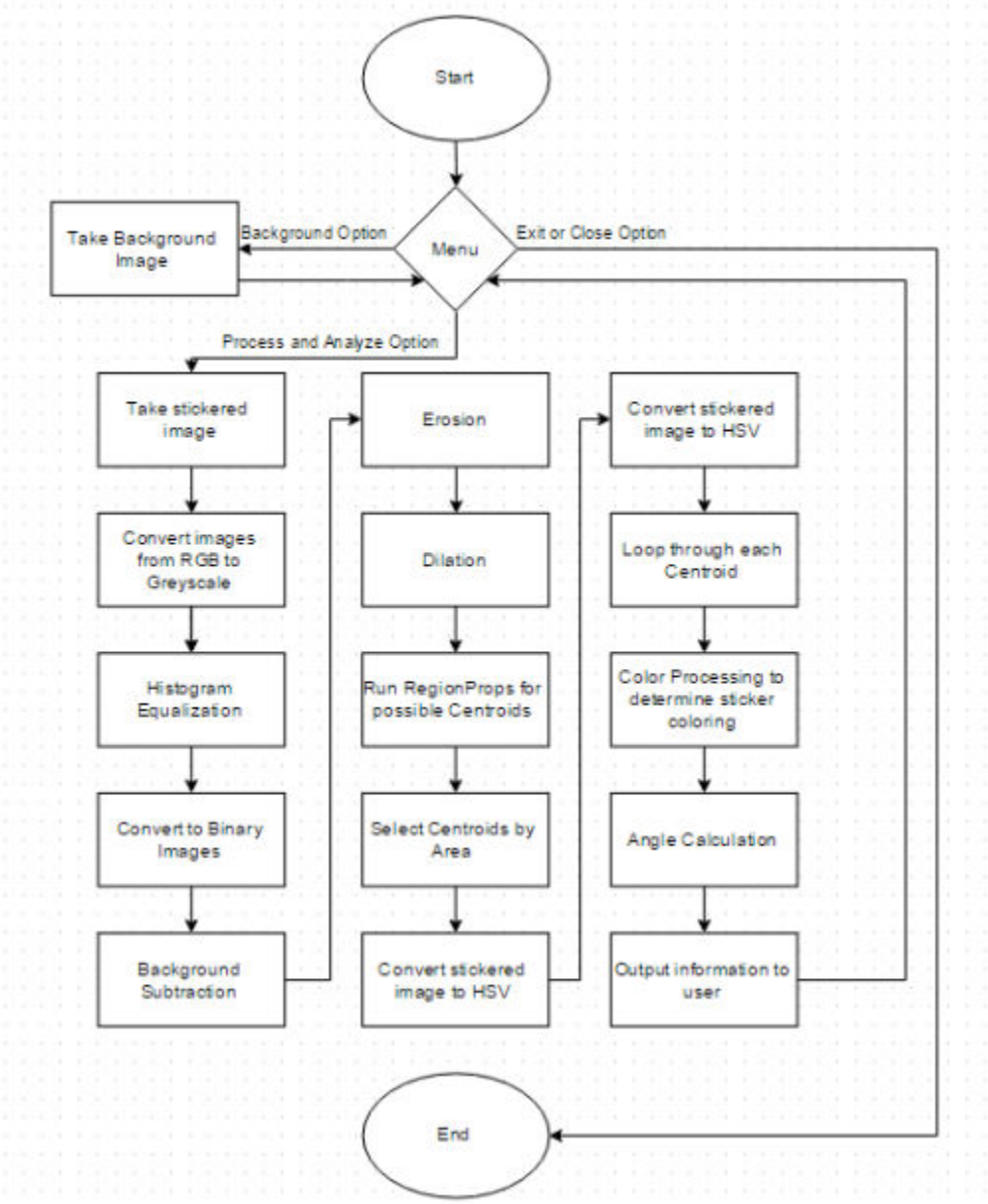


Figure 3. Software Flow Diagram.

Figure 3 shows the flow chart for the overall image processing software algorithm implemented. As seen in Figure 3, upon execution, the program would prompt the user with a menu containing three options. From here, the user could choose to take a background image, take a stickered image and process the images, or close and exit the program. If the user chose the first option, the software would engage the camera, take a background picture, and overwrite the current background picture if one existed. If the user chose the second option, the software would first check if a background image was stored in memory. If one did not exist, it would prompt the user and return to the main menu. Otherwise, it would engage the camera, save the picture as the image with stickers, and run an image processing algorithm. At a high level, the image processing implemented scanned the board for Avery 457472 stickers and reported the centroid, color, and angle of each sticker it found as formatted output.

The camera used as a sensor for taking photos was an HP Pro Webcam. All interfacing with the HP camera was done via built in MATLAB functions. VideoInput() was used to establish a connection to the camera and GetSnapshot() was used to take background and stickered image photos. Each time these were taken, they were shown to the user via the figure and ImShow() commands.

The menu operation was implemented via a standard structure of a WHILE loop containing an IF statement. The looping structure re-prompted the user with the menu and reset all necessary variables to continue execution at the bottom of each iteration. At the beginning of the program, the variable holding the background image was initialized to zero. This allowed it to check if image processing was possible when menu option two was selected in the looping structure: If the background variable still equaled zero, there was no background image in memory and image processing could not occur.

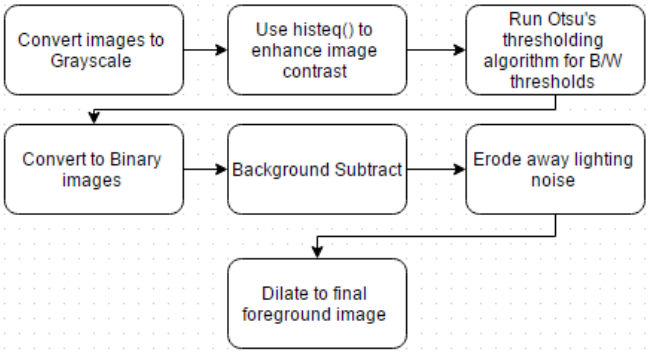


Figure 4. Image Processing Flow Diagram.

Figure 4 shows the flow of the image processing code implemented. Note this is a subset of the "Process and Analyze" option code from Figure 3 with more detail. When image processing starts, the first focus of the program is to eliminate as much lighting noise as possible. To do this, it first converts the input images to grayscale images via the RGB2Gray() function to get each pixel's contents solely in terms of intensity. At this point, histogram equalization is run, which normalizes all of the pixel intensities in each image and enhances the contrast in each. In the correct circumstances, this removes the majority of lighting effects and enhances yellow stickers so they show up in the final foreground image, but it also enhances the effects of shadowing on the board. After over 500 tests, the group determined that Otsu's thresholding algorithm returned more stable binary thresholds than any manual threshold they used, so at this point, the graythresh() function is run to generate these numbers. Once binary images are obtained with the Im2BW() function, the bitxor() function is used to quickly subtract the images and produce a difference image. Any noise from the subtraction is eroded away with the ImErode() function and the remaining sticker detections are restored to their original size with the ImDilate() function.

The above image processing algorithm was easy and intuitive to implement, but its main downfall is that it is extremely light sensitive. The group frequently had to retune thresholding settings to different lighting situations and had to use two external light sources to get the algorithm to produce consistent results. Retuning was often long, painful, and involved using ImShow() to print the image(s) at each step of the processing algorithm to change values or lighting one by one. On demo day, the group realized there was a much better way to implement image processing in the future by doing background subtraction in color and combining image processing with HSV color processing to produce color-specific foreground images. The group plans to use this design for the image processing in their final project.

The remainder of code depended heavily on output from the RegionProps() function. RegionProps() was first queried for all possible centroids in the image, and then its output was

parsed with a FOR loop. During the parse, an object was only counted as a sticker if its area was greater than 200 square pixels. These objects were concatenated to an initially empty array. After the first parse, the final centroid array was parsed to perform color processing and angle calculation for each centroid at once.

To perform color processing, the program looked at an HSV version of the stickered image at each centroid location, averaged each centroid's hue with twenty five pixels around it, and compared the averages to thresholds to determine the color of each sticker. The thresholds were determined with fifty test runs of each sticker in all possible well locations. HSV was used over RGB because it is significantly more light invariant, and because all color information is stored in the "Hue" representation member.

To perform angle calculation, the absolute X and Y axes distance from the center pixel of the 640x480 images (320, 240) was calculated. A coordinate system for the board was set up with zero degrees at the rectangular notch on each board and positive angles going counterclockwise. The distance values were given to an inverse tangent function, which returned the correct angle for each centroid in quadrant one. The X and Y boundaries of each quadrant were determined and used to add either 0, 90, 180, or 270 degrees to each of these values to obtain the real angle.

gameState Member	Meaning
info	Stored information about the camera being used.
erodeSE	Stored structured element used for erosion.
dilateSE	Stored structured element used for dilation.
centroids	Stored centroids of each sticker found.
wellColor	Stored color of each sticker found as an integer.
wellLoc	Stored angle of each sticker found in degrees.
key	Decoded wellColor.
background	Stored background image.
current	Stored stickered image.
backG	Stored background in grayscale.
currG	Stored stickered in grayscale.
backE	Stored histeq() enhanced background.
currE	Stored histeq() enhanced stickered image.
backThresh	Stored background threshold from Otsu's algorithm.
currThresh	Stored stickered threshold from Otsu's algorithm.
backBW	Stored binary background image.
currBW	Stored binary stickered image.
differenceIm	Stored background subtracted image.
diffErode	Stored eroded difference image.
foreground	Stored final foreground image.

Figure 5. gameState Variable Structure.

Figure 5 shows the structure of the variable running all of the code above; gameState. The last thing the program did was output each sticker it detected as formatted output with a unique number identifier. It would simultaneously also display the foreground image with these numbers at each centroid location, making it easy to match each sticker in software to its position on the board.

V. Does the Design meet Requirements?

The physical customer requirements for the design were to build a camera based sensing system to determine the color and position of the colored stickers on given game board. The software requirements stated that design had to be able to store these results in a data structure called gameState and use them for motor control in project five. Further, it needed to take a background image, an image with stickers, and perform image processing on both images to determine the relevant information. All software operations in the design had to be able to run more than once through either a menu or GUI until the user terminated the program.

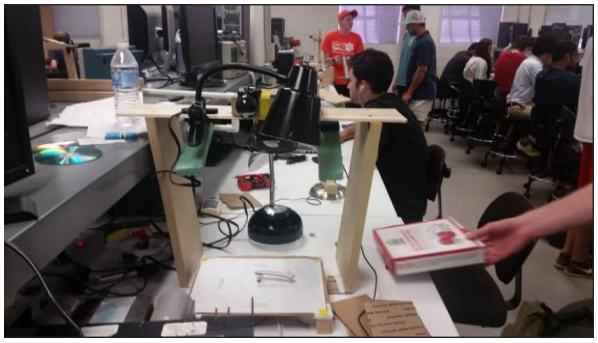


Figure 6. Design photo.

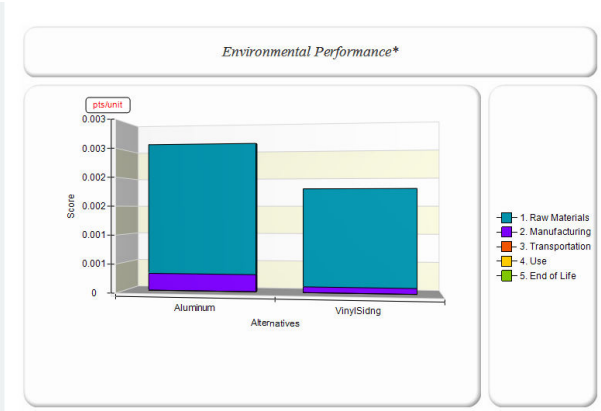
To fulfill the design requirements, we made the system shown in Figure 6. The system is comprised of a pi-shaped stand with Velcro camera mounts and a cardboard stand with wooden legs holding the motor for project 5. It interfaced with a MATLAB program called project4.m to perform image processing, and output a structure called gameState to the MATLAB workspace. This structure stored the color of each sticker as an integer in a member called wellColor and contained a key for decoding the color in a member called key. To report positioning, it stored the x and y centroid components of each detected sticker in a member called centroids and the angle of each detected sticker in a member called wellLoc. The coordinate system set up for the angles started at the rectangular notch in the board with positive angles in a counterclockwise direction.

The software for the design meet requirements. Upon execution, a user friendly interface menu appeared which gave the user options to take a background image, take a stickered image and process it, or quit execution. Each of these buttons performed their desired operation and were capable of performing their desired operation more than once. The program's execution did not stop until the user quit it or interrupted its execution.

After fine tuning thresholds in the program, during testing with stickers and no filter paper, our program found 95 green stickers in 100 trails, 97 blue stickers in 100 trials, 99 red stickers in 100 trials, and 90 yellow stickers in 100 trials. As these numbers prove, the system was very reliable without filter paper, but the one caveat was that it was extremely sensitive to light. The lamps used had to be positioned in exactly the right positions or the program would not work. This was partially due to the shape of the stand casting inconsistent shadowing over the game board and the algorithm performing background subtraction in black and white instead of in color. We now realize that utilizing background subtraction in color in combination with color processing will result in a far less light sensitive algorithm, and plan to use that for our final project.

During testing with filter paper, the design still detected each color sticker with about the same success rates as without the paper, but it also picked up a significant portion of extra lighting noise and counted it as stickers. This was because of the prior discussed light sensitivity and the sheer amount of light we had to introduce to the design to compensate for it. Again, our group plans to use a far less light sensitive algorithm in the final project.

VI. Life Cycle Analysis

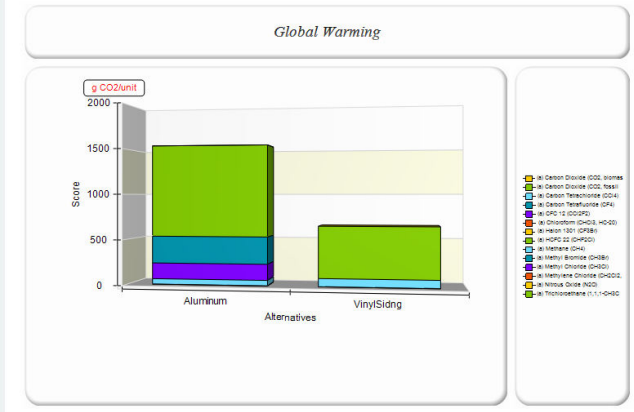


Note: Lower values are better

Category	Aluminum	VinylSidng
1. Raw Materials	0.0023	0.0017
2. Manufacturing	0.0003	0.0001
3. Transportation	0.0000	0.0000
4. Use	0.0000	0.0000
5. End of Life	0.0000	0.0000
Sum	0.0026	0.0018

Figure 7. BEES Aluminum and Vinyl Siding Environmental Performance Data.

Figure 7 reveals that the most significant environmental impact from the choice of siding materials comes from the Raw Materials and Manufactoring categories. Most of the weight comes from the Raw Materials category. This chart reveals that vinyl siding has the lowest impact on the environment.



Note: Lower values are better

Category	Aluminum	VinylSiding
(a) Carbon Dioxide (CO2, biomas	0.0016	0.1934
(a) Carbon Dioxide (CO2, fossil	1004.4803	565.1632
(a) Carbon Tetrachloride (CCl4)	0.0032	0.0301
(a) Carbon Tetrafluoride (CF4)	297.9217	0.0062
(a) CFC 12 (CCl2F2)	175.4912	0.0001
(a) Chloroform (CHCl3, HC-20)	0.0000	0.0000
(a) Halon 1301 (CF3Br)	0.0000	0.0005
(a) HCFC 22 (CHF2Cl)	0.0000	0.0010
(a) Methane (CH4)	58.5673	86.2281
(a) Methyl Bromide (CH3Br)	0.0000	0.0000
(a) Methyl Chloride (CH3Cl)	0.0000	0.0000
(a) Methylene Chloride (CH2Cl2,	0.0004	0.0001
(a) Nitrous Oxide (N2O)	1.5330	2.5951
(a) Trichloroethane (1,1,1-CH3C	0.0000	0.0000
Sum	1537.9987	654.2178

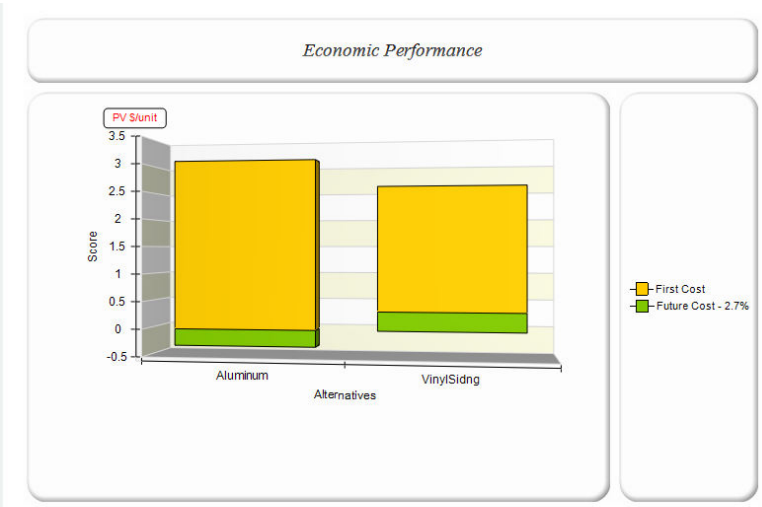
Figure 8. BEES Aluminum and Vinyl Siding Global Warming Data.

Figure 8 reveals that aluminum has a much wider and more diverse emissions footprint, while Vinyl Siding only has significant Carbon Dioxide fossil and Carbon Tetrachloride impact. If the decision were made solely based on this graph, vinyl siding would be used.

Category	Aluminum	VinylSidng
Feedstock Energy	0.4661	4.1403
Fuel Energy	12.1329	8.1830
Sum	12.5990	12.3233

Figure 9. BEES Aluminum and Vinyl Siding Embodied Energy by Fuel Usage Data.

Figure 9 reveals both materials use roughly the same amount of energy per unit. It also shows that vinyl siding relies more on feedstock energy while aluminum relies more on fuel energy. The choice of best material from this graph is not immediately clear and would depend on the application.



Category	Aluminum	VinylSidng
First Cost	3.05	2.22
Future Cost - 2.7%	-0.30	0.34
Sum	2.75	2.56

Figure 10. Aluminum and Vinyl Siding Economic Performance Data.

Figure 10 reveals vinyl siding is less expensive per unit.

Overall, the group believes vinyl siding would be the best choice for the packaging. Vinyl siding less impact on the environment while also having a lower global warming emissions footprint and being more economically effective. A similar analysis could be done on the entire project to reduce the robot's environmental impact.

VII. Employee Training Program

Engineer Employee Solving Technical Problems		
Activity	Benefit	Cost
IEEE Membership	Networking opportunities along with having access and exposure to new technology and updates in current technology	$400
Present at Clemson IEEE Seminar	Will allow employees to get in touch with current students, will make them refresh their technical knowledge by putting presentation in terms students can understand, and will be put on the spot to answer questions as they arise	Travel & Food ($50) + Cost of Missing Work, 1 day ($220) = $270
IEEE Annual Meeting	This conference will help the employees to become familiar with new technology and the most recent electronic developments. This conference will also focus on the application of these new fields to industry. This conference would also encourage employees to stay more in touch with IEEE.	Conference Fee ($675) + Hotel Fee ($325) + Travel to Texas and back ($300) + 2 days missed work ($440) = $1740
Engineering Professionals Leadership Instutite	A special part of this conference is that the managers are encouraged to invite early career professionals to attend. Employees will learn how to improve their leadership skill set along with other professional engineers.	Conference Fee ($100) +Travel to GA, food, hotel ($180)+ Cost of missing work, 2 days ($420) = $700

Figure 11. Engineering Employee Solving Technical Problems Training Program.

Engineer Employee Managing Technical Projects		
Activity	Benefit	Cost
IEEE Membership	Networking opportunities along with having access and exposure to new technology and updates in current technology	$400
Professional Engineering Conference	Will improve leadership skills by attending classes such as "Project Management" & "Strategic Planning"	Travel to Wisconsin and back ($700) + Price of Conference ($200) + Cost of Missing Work, 1 day ($220) = $1120
Represent Company at Clemson Job Fair	Will require the employee to gain broader knowledge of company in order to present at career fair and answer students' questions	Travel & Food ($50) + Cost of Missing Work, 1 day ($220) = $270
IEEE Annual Meeting	This conference will help the employees to become familiar with new technology and the most recent electronic developments. This conference will also focus on the application of these new fields to industry. An additional benefit for the manager would be that it emphasizes networking with peers and professional development.	Conference Fee ($675) + Hotel Fee ($325) + Travel to Texas and back ($300) + 2 days missed work ($440) = $1740
Engineering Professionals Leadership Institute	A big part of this conference is that the manager would invite the employees to attend. When the manager invites his team members, it will bring the group closer together. At the conference itself the manager will be able to practice leadership with other managers from the Southeast Region.	Conference Fee ($100) +Travel to GA, food, hotel ($180)+ Cost of missing work, 2 days ($420) = $700

Figure 12. Engineering Employee Managing Technical Project Training Program.

VIII. References

[1] A. Kapadia. ECE 4950 - Integrated System Design [Online]. Available: http://people.clemson.edu/~akapadi/ece4950_references.html. Cited: 10/18/2015.

ECE 4950 Integrated System Design I
Project 4
ECE495 - Project 4: Camera as a Sensor

Group Name and Members: _____

Score	Pts		ABET Outcomes
	5	General Report Format - Professional Looking Document/Preparation (whole document) a) Fonts, margins (11pt, times new roman, single spaced. 1" margins on all sides). b) Spelling and grammar are correct c) Layout of pictures – all figures need captions and must be referenced in text d) References. Use IEEE reference format. ***e) All report components are included in your website*** **Page 1: Title, Group Name, Group Members, and Date** **Customer Requirements** Description of what the Customer wants for this project.	g
	5	**Page 2: Customer Requirements and mapping to Engineering Requirements** In the context of just Project 4, developing a camera sensing system (not the full final project), make a two column table that contains a column for Customer Requirements (what are the functions of the sensing system?) and the resulting Engineering Requirements. Each row should contain a specific customer requirement and the resulting engineering requirement. One customer requirement may generate multiple engineering requirements. For example, the customer will want an "accurate" system, the Engineering Requirement could be 99.5% detection success.	c
	20	The following should be a narrative report that describes your design decisions and final design, e.g., don't just have a flowchart without text that explains it. **Page 3: Document Hardware** (1 page) • Connection diagrams • Equipment • Mounting hardware sketches and photograph. Calculation of resolution - pixels per square on board. Is the camera an appropriate sensor? **Page 4-6: Document Software** (3 pages) • Flow charts, state diagrams, data structures, etc. that describe how the software is implemented. Do not include a copy of the source code. **Page 7: Does the design meet the Requirements**? (1 page) • Evaluate your system in regard to achieving the Requirements • Include statistics such as "found green sticker 80 times in 100 trials	c
	10	**Pages 8-9 Life Cycle Analysis** (2 pages) You are proposing a design that consumes resources. Follow the "Life Cycle Assessment (LCA) Exercise" for the shipping box for your project to examine the life cycle for this one part of your design. **Be sure to interpret the results of the computer program**. Complete this section of the report by saying that a similar analysis could be done on the entire project to reduce environmental impact.	h
	50	• Laboratory demonstration of your prototype (evaluated by instructor and TAs See "ECE 495 Project 4 Evaluations" for scoring) • Scenario 1: Five stickers (arbitrary color and location) will be placed on the board with no film over camera. • Scenario 2: Same stickers moved to other locations on the board. One filter sheet will be inserted in the mount and values of recorded variables (color and position) will be noted. **There is no opportunity to acquire a new background image after the filter is added**. • Scenario 3: Repeat with two sheets.	c