



ECE 480 Design Team 4

Hot Strip Mill Center Line Tracking

Facilitator

Dr. Selin Aviyente

Team Members

Bryan Blancke

Mark Heller

Jeremy Martin

Daniel Kim

Sponsor

ArcelorMittal

Project Liaison

John M Dujmovich

Nick Korzow

William J Sammon

DATE

12/4/13



Executive Summary

The finishing section of ArcelorMittal's steel hot strip mill passes a length of heated steel through seven high pressure rollers to flatten the steel into a long strip. Several issues, such as inconsistent temperatures and pressures, can cause the steel length to curve off the intended path. This bending can cause the steel to impact the guard rails on either side of the rollers, damaging equipment and stopping production. Currently, the curvature of the steel is monitored by an operator in the hot mill's control room. When the operator sees curvature in the strip, corrections are made and changes can be made to improve future strips. Steam, distance from the strip, and a low viewing angle make monitoring the shape of the strip difficult. Team four's design project entails a vision solution to display a clearer visual of the shape of the strip to the operator. Giving the operator a clear, live feed of the strip will allow for more accurate adjustments and will help to prevent production halts. Data obtained from the system could be used in control systems to produce straighter steel and reduce failures in the hot strip mill.

Acknowledgements

Team 4 owes many thanks to the individuals that provided us with assistance throughout the semester. We would like to give a special thanks to the following people.

Dr. Selin Aviyente - As our team's facilitator, Dr. Aviyente met with us weekly to provide guidance on the project and give constructive criticism on how to improve our presentations or reports, as well as give suggestions on project implementation.

Roxanne Peacock - As the team's purchasing consultant, Roxanne provided support to the team whenever items needed to be ordered. She was efficient and very helpful, giving the team updates on ordered parts, allowing the team to remain organized and on schedule.

Greg Motter - As the team's business application advisor, Greg provided his informative insight on a wide variety of business management strategies that were employed in our project.

Greg Mudler - Lab supervisor and technician. Greg provided supplies and also gave us insight on ideas to make our project better. He was a tremendous help by supplying us with the materials to design the teams test demonstration bench.

Derek Molloy - Derek is a professor from Dublin University who has created a website in support of the BeagleBone Black as a computer vision and image processing device. His instructional videos and resources were extremely useful in the development of this project. Our most sincere thanks goes to Derek for providing such amazing resources.

James Hendrickson - A special thanks goes to James Hendrickson from ArcelorMittal for reaching out to the team and offering his support. James met with the team at MSU and provided valuable voice of customer feedback which helped the team decide on the issues that needed to be solved with this project.

ArcelorMittal - Thank you for sponsoring our design team and providing this project.

TABLE OF CONTENTS

Chapter 1 - Introduction and Background.....	5
Chapter 2 - Exploring the Solution Space and Selecting a Specific Approach.....	9
Chapter 3 - Technical Description of Work Performed.....	15
Chapter 4 – Testing and Results.....	20
Chapter 5 – Final Cost and Conclusion.....	25
Appendix 1 – Technical Roles, Responsibilities, and Work Accomplished.....	26
Appendix 2 – References.....	30

Chapter 1 - Introduction and Background

The purpose of this project is to measure the centerline of a steel strip in the hot mill rolling process. Additionally, the project will provide a visual for operators to see what is happening in the hot strip mill from a bird's-eye view. The project will inform operators if there is a problem in the hot strip mill and provide a video in order to reflect on why things went wrong. This will aid in improving future operations to avoid repeating the previous error.

The reason for this project is to monitor and control the curvature of a steel strip. In the steel manufacturing process, after all raw materials have been melted down and turned into a slab of steel, it must be sent through a hot strip mill. In a hot strip mill, the slab width is reduced from 6-10 inches thickness into a strip anywhere from .06 to .5 inch thickness. This strip of steel is rolled into a large coil and shipped to customers or other steel processing plants within the company. All defects that occur here will cause problems within future finishing processes, in addition to delivering a lower quality product to customers. The issue at hand is known as

Figure 1 : Thick Steel Slabs before processing



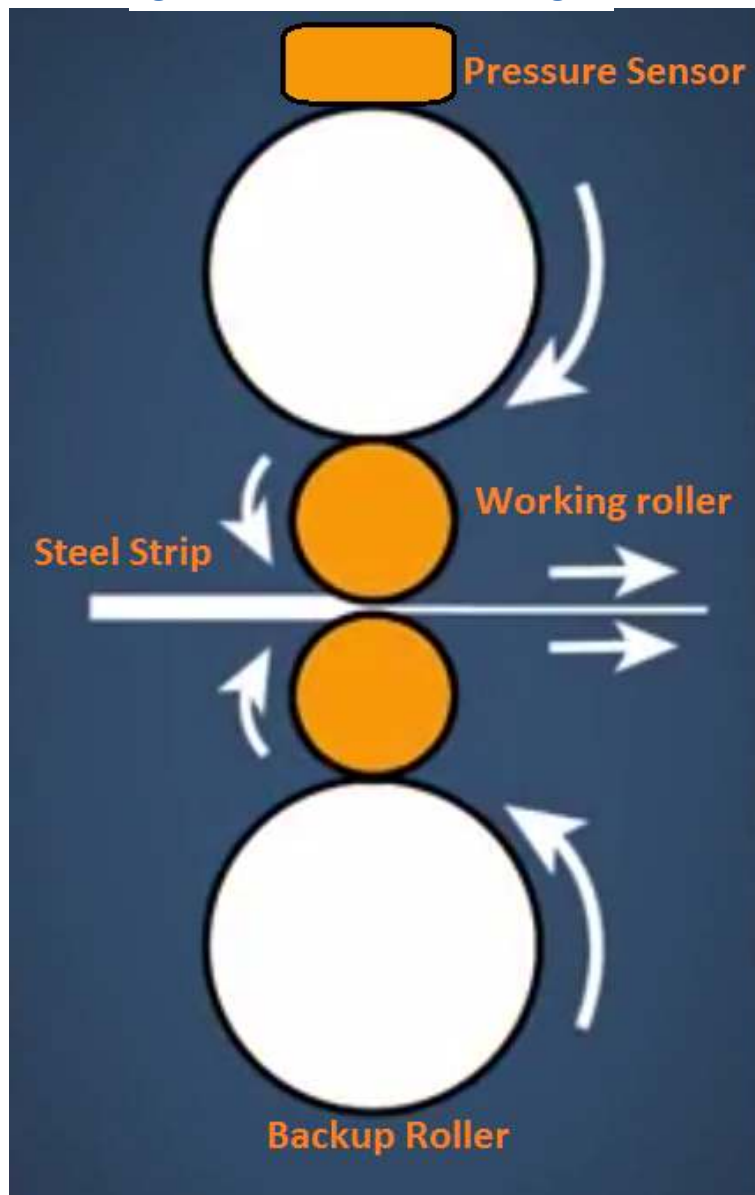
Figure 2 : Steel coils, finish product after hot mill process



cambering in the steel industry. Cambering is the action of steel curving off its intended path. It is very difficult to keep the steel running straight through the mill during the hot milling process. It is critical to keep cambering under a reasonable threshold during the hot strip mill process. Cambering can be caused by many discrepancies in the hot mill process, such as differences in temperature throughout the steel and uneven pressure being applied from the rollers. Due to these factors, a uniform press is not always obtained. If camber becomes too extreme, the steel can cause a wreck in the mill. A wreck occurs when the steel runs off the line and causes production to be halted. The last line of defense for preventing a wreck is a guard rail. Ideally, it is better to avoid contact with the guard rail altogether. There is a chance that when the steel contacts the guard rails, it will curl and fold onto itself. This causes another issue called cobbling. Cobbling occurs when the steel is not of uniform thickness. As the cobbled steel strip passes through the subsequent rollers, it damages the rollers and they can no longer perform a uniform press. Consequently, when there is a wreck, the rollers are no longer usable. This causes production to halt. The cost of downtime is approximately \$12,000 per hour. Therefore, it is critical to prevent production shutdown. Skilled staff can change rollers as quickly as seven minutes, though it is an expensive problem that should be avoided at all costs. A wreck from the steel running off the mill past the guard rails can produce a downtime in production for hours. When there is a wreck, cranes are used to clear the steel from the mill so that production on a

new strip can resume. The aim of the centerline tracking project is to monitor cambering in order to reduce the number of wrecks and minimize contact of steel strips with the guard rail. This will reduce wear and tear on the hot strill mill.

Figure 3 : Hot Mill Roller Housing



Today, Arcelormittal has no visual of the hot strip mill. The mill operator's only visual is from the operating room, which is far from the rollers and does not provide not a clear view of what is happening in the milling process. Operators can use controls to make manual adjustments to the system. The most effective method of camber control is from the rollers themselves. The rollers have sensors on each side, and by measuring the differences in the pressure across the rollers, the control system makes adjustments to keep the steel strip running as straight as it can. With this method, the height of the gap between rollers is being changed based on the measurements of the pressure sensors to control cambering. This method of centerline tracking is limited because the pressure sensors are not sensitive enough for fine tune adjustments. Wrecks can occur sporadically, but on average there is about one wreck per week at ArcelorMittal's Burns Harbor facility.

There are other methods of tracking the centerline of a strip of steel in the hot mill. All-in-one products made for centerline tracking are available from various companies, but are tremendously expensive. A product called the hotCAM is available from EMG Automation GmbH. The hotCAM is a high speed camera that tracks the centerline of a steel strip. This product is an excellent solution. A typical setup includes stands for the mounting hotCAMs between every pair of rollers. The hotCAM will take infrared video of the steel strips. To deal with the harsh conditions of the steel mill, the hotCAM has 3 connections for power and air pressure, compressed air cooling, and an optional water cooling connection. The hotCAM can handle temperatures up to 100 degrees Celsius. The hotCAM accurately records the position, width, and camber of the strip. The performance and accuracy of this camera for a solution to camber is unmatched. ArcelorMittal's hot strip mill would require 11 hotCAMs, with each hotCAM costing \$130,000. The total cost of this solution for ArcelorMittal would be \$1,430,000 excluding the cost of the mounting equipment and installation. This is a far too expensive solution.

Figure 4 : EGM hotCAM

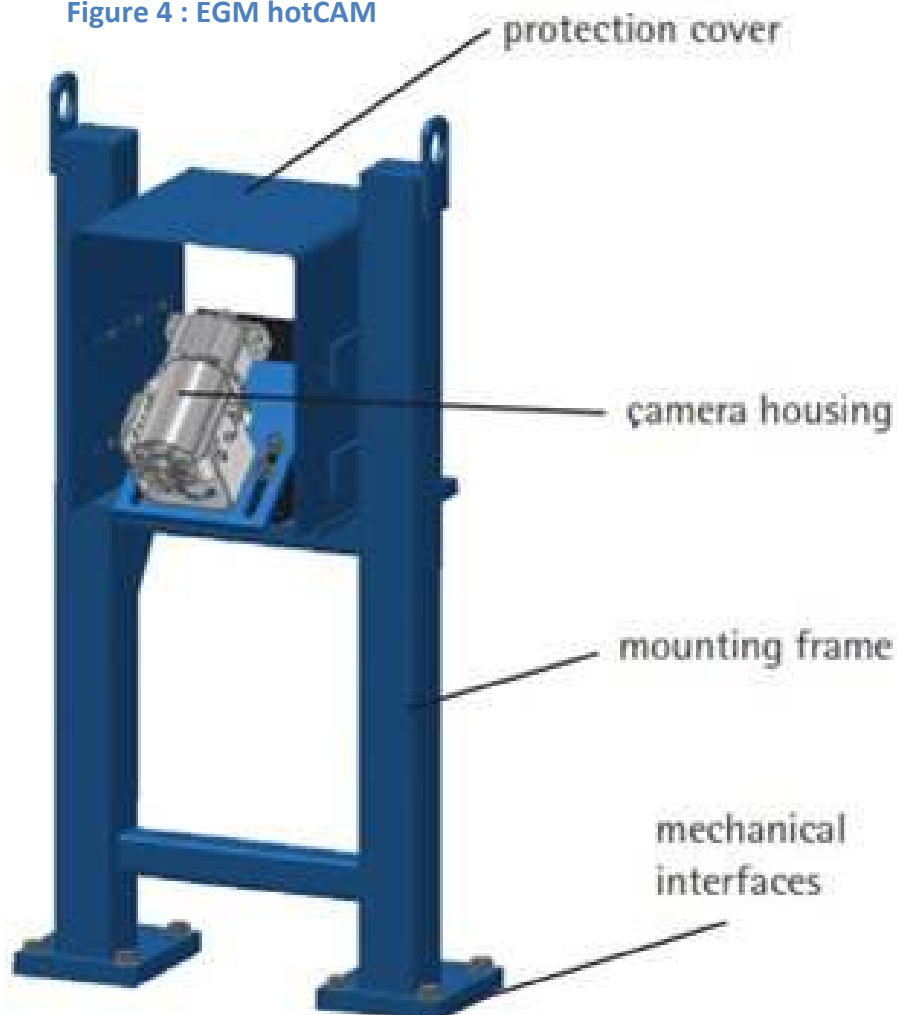
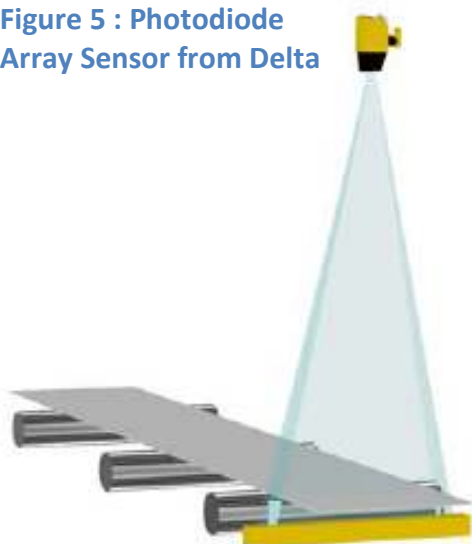


Figure 5 : Photodiode Array Sensor from Delta



Centreline measurement

Another current method for centerline measurement and control is available in the form of a similar product offered by Delta. A photodiode array measurement sensor is a product offered by delta with applications in the steel industry. The sensor consists of a lens, build around a linear silicon photodiode array and processing electronics. The array of 512 diodes detects light in the visible and near infrared spectrum. Each diode is responsible for detections of a signal pixel, which gives the images from the product a 512 pixel resolution. The image of the scanned area is projected onto the diode array by a thin lens. The information from the diode array is processed by a microprocessor which converts the position or size of different dark or luminous areas to an analog output of 0 to 10 volts. When a pixel of light area is detected, the diode is turned on and its individual voltage contributes to a part of the voltage

measurement between 0 to 10 volts. Various models of the product are available for a single function which includes center position, edge position, dimensions, and sum of dimensions. This sensor is able to tolerate extreme environmental conditions. This product is also not a feasible solution due to its high expense.

The final design will incorporate a high-resolution camera integrated within a microcomputer. The microcomputer will be able to detect the edges of the steel and calculate a centerline. The centerline data will be stored and compared with future data. When the change in centerline crosses a threshold, an operator is alerted of the error and will be able make adjustments accordingly. The software will be user-friendly and compatible with what the operators use during their typical monitoring of the process.

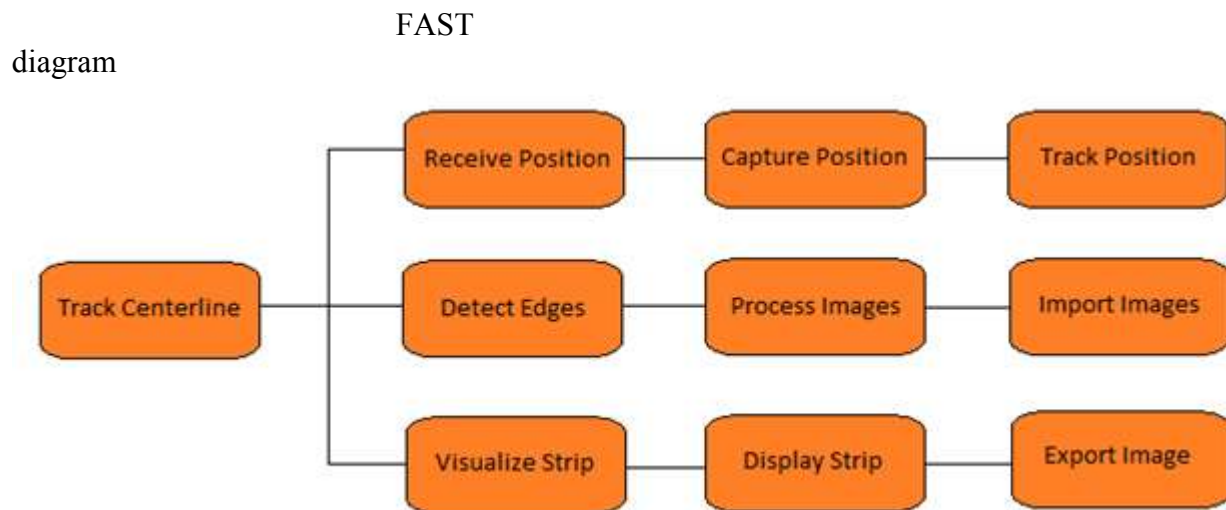
The team's design solution revolves around two key hardware components, a microcomputer and a high-definition camera. Secondly, the project relied on image processing C++ codes in combination with support from OpenCV (open-source computer vision) libraries. The reason our design will be so successful is due to the cost, ease of use and transportability. With all of our design pieces well underneath the five-hundred dollar budget, this solution is the most reasonable. Data is calculated and sent as an output through a single executable program, providing real-time calculations of a centerline without the need for large industrial camera systems. One challenge the design will face is environmental factors. When implemented in a steel mill the environmental conditions will be harsh. If placed properly so it is not too close to the steel strip, it will be able to tolerate the high temperatures. Water is sprayed onto steel strips during the milling process to remove oxide layers from forming on the steel. This produces a lot of steam in the steel strip mill. Steam could obstruct the vision of the web cam. It needs to be mounted in a position to minimize its exposure to steam.

The project is a new idea because it utilizes low cost microcomputers which were not being marketed in the past. Handing off the project to ArcelorMittal, we hope that with a simple installation and calibration, our system will work flawlessly. Our final product will be the first building blocks to an entire control system that effectively changes roller pressures in order to straighten cambering without the need of manual intervention. An effective solution will result in the reduction of mill wrecks, additionally resulting in less downtime and mill expenses. Mill operators will no longer have to constantly monitor the steel strip from afar along with their many other responsibilities. It will make the job of the mill operators easier and with less wrecks, many other employees who are affected by the cleanup and repercussions of a wreck will not have to be bothered.

All the code for this project is open source and posted online for anyone to use or modify. This project is creative in its implementation of the code to track the centerline of a steel strip. The code can be used and modified for many other applications. If updates to the hardware need to be made, it would be easy to make slight modifications to the code to update the system. It is also good for its flexibility of the many steel mill environments it may be used in. By changing the thresholds, one can control the amount it will take to yield and error in the centerline tracking. It can also be calibrated to change the thresholds for noise reduce or increase the resolution for systems that need to be more precise.

Chapter 2 - Exploring the Solution Space and Selecting a Specific Approach

In order to fully visualize the scope of the project, a FAST diagram was created; defining all of the functions the design is to entail. The diagram begins with the primary function of the design project: tracking centerline. From the primary function branches a few paths that detail all secondary functions that describe how the project's primary function will be achieved. The FAST diagram is as follows:



The three functions at the ends of the branches on the far right indicate that three major components need to be integrated into one system for our design to be realized. Sub-functions of tracking centerline are:

- Vision capturing device (camera, sensor) to track the position.
- Processing device (laptop, microcontroller) to import and process images.
- Display (monitor, alarm) to output images to for visualization.

For tracking the centerline our fast diagram split in 3 major tasks: visualizing the steel strip, receiving the position, and detecting edges. Strip visualization was done by using the BeagleBone and our capture program which can take video. This video will display the strip and save the video file which accomplishes the bottom chain of the fast diagram. For detecting edges the grabber program takes an image and converts it to grayscale. It uses the grayscale image to make an edge image. After we have an image with the edges the team processes the image to filter out noise. With the processed imported image, the data is used for centerline calculations. This completes the middle chain in the fast diagram. The top chain is to receive position. This is done with the edge detection image after it is filtered for noise. The data for the edge position is in the filtered edge detection image. Extracting this data yields the pixel position of the edges. The two pixel positions of the edge are used to calculate the centerline. Taking multiple images over time is how the team tracks position with time. This is how the team's final project implemented the planned functions of the fast diagram.

Conceptual Design

Design 1: Centerline Tracking Production Monitor

One design solution is to use hotCAM for camber and position measurement. This device is manufactured and sold by an European company called EMG Automation. This device would be mounted above the line in the hot strip mill and is able to record and delivers an image of the camber to the operator's room. The hotCAM could be installed on any of the stands to provide continuous centerline measurement to feed into control systems for corrections. The most critical position for a hotCAM would be at the sixth finishing stand where cambering is most server. It is ideal to have multiple cameras throughout the line since the position of the steel strip changes as it goes through each finishing stand. Once the hotCAM detects the change in position of the steel strip, hotCAM records the image and the data can be used by control systems to make real time fine tune adjustments. The image is then sent to the operator's room where operator will be able to make more coarse adjustment on the rolling force to keep the strip in the middle of the line.

This design is good due to its high accuracy, thermal resistance, and ease of assembly. The downside of the Centerline Tracking Production Monitor is the cost. High-end production monitors are well outside of the team's budget. Additionally, there is little engineering innovation to be done because this system was built to track centerline and there is little left to add to it

Design 2: Laser Strip Edge Detection

In the second design, multiple lasers could be mounted on each side of the line in the finishing stands. When the steel strip begins to camber, different sensors would trip. The microcontroller would then be set to high. Then software interfaced with the microcontroller will output a visual image of the strip, and this is how the operator will know steel strip began to camber. However, there are some downsides with this solution. For instance, the accuracy of the visual representation is highly dependent on the number of lasers installed by each side.

Consequently, this design option will be expensive. Additionally, environmental factors such as debris and steam will interfere with the laser. As a result, the lasers may not even detect when the steel strip begins to camber.

This design is considerably cheaper than the Centerline Tracking Production Monitor; however it includes a significant drop in accuracy. The range of the laser sensors is much less than that of a high-end camera, placing the sensors very close to the hot steel strip. Melting sensors would be difficult to avoid and would require an extensive heat removal process and shielding.

Design 3: BeagleBone Black Microcontroller interfaced with Webcam.

The third design consists of a high-definition webcam connected via USB to a BeagleBone Black microcomputer. The BeagleBone is running an Angstrom distribution of Linux as an operating system. With support from OpenCV, the BeagleBone can properly detect contrasting areas in an image received from the webcam. By checking for the highest density of edges, the strip edges can be located and are ran through a series of algorithms calculate the centerline data. The centerline data is sent back to be stored on the BeagleBone, where it will be anticipating the next location of the centerline. Using the centerline data, we can properly determine if cambering has occurred.

Using AISC's standard for cambering, our product will be able to alert an operator when the centerline has crossed a certain threshold. At this point, the operator must make adjustments to the rollers so that the problem is not replicated in the next strip.

Table 1: Feasibility Matrix (1 = Least Feasible, 5 = Most Feasible)

Design	Centerline Tracking Production Monitor	Laser Strip Edge Detection	BeagleBone Black Controller and Camera
Estimated Cost	0	2	5
Environmental Tolerance	4	2	2
Imaging Resolution	4	1	4
Processing Speed	3	4	3
Averaged Scores	2.75	2.25	3.5 Most Feasible

Looking at our feasibility matrix, all aspects of the hotCAM look like a fantastic choice, with the exception of the price. Within the million-dollar-range, this throws the idea completely out of consideration. The laser edge detection seems feasible at first glance, but due to placement issues and being an inaccurate solution, we looked for a better option. With the microcontroller and webcam solution, we have the ability to display information through a monitor, track the centerline at an accuracy of our choosing, all for an extremely reasonable price.

The final decision to use the microcontroller solution with a web cam was chosen because of its price and expected performance. The team was confident this would work because research sources proved that the BeagleBone has been used for many computer vision applications. With Derek Molloy's website supporting vision projects using the BeagleBone, the team knew it would be able to capture video and get edge detection images. This was a major part of the project and this solution could take care of that mostly on its own without additional code or set up. It is a new approach to the problem because it is using new and affordable hardware which has great performance. The performance to dollar ratio is much better than more expensive solutions. The BeagleBone Black is a new device which released late 2012. More and more applications of the BeagleBone Black have been documented on the internet, and the team decided we would add to that list because this great device has many capabilities for do it yourself technical projects.

Our webcam will be able to take pictures at a resolution of 1080p and a speed of 30 frames per second. If the team only uses the BeagleBone microcomputer to process image data, it would cause a serious bottleneck to the system. The team was concerned that it may not be able to process the data fast enough. Instead, much of the processing is being done in the

circuitry of the camera which frees up processing power on the BeagleBone. That is why the camera gave us confidence that it would work well with the BeagleBone. We chose BeagleBone over other microcomputers because of the processing speed, along with ease of setup. The group considered adding a digital signal processing chip (DPS) to increase the processing power of the project. The team was not sure how to integrate this and decided to test it without a DPS and that it could be added later if needed. The team ended up going without the DPS because it was not necessary.

Budget: Initial Estimate

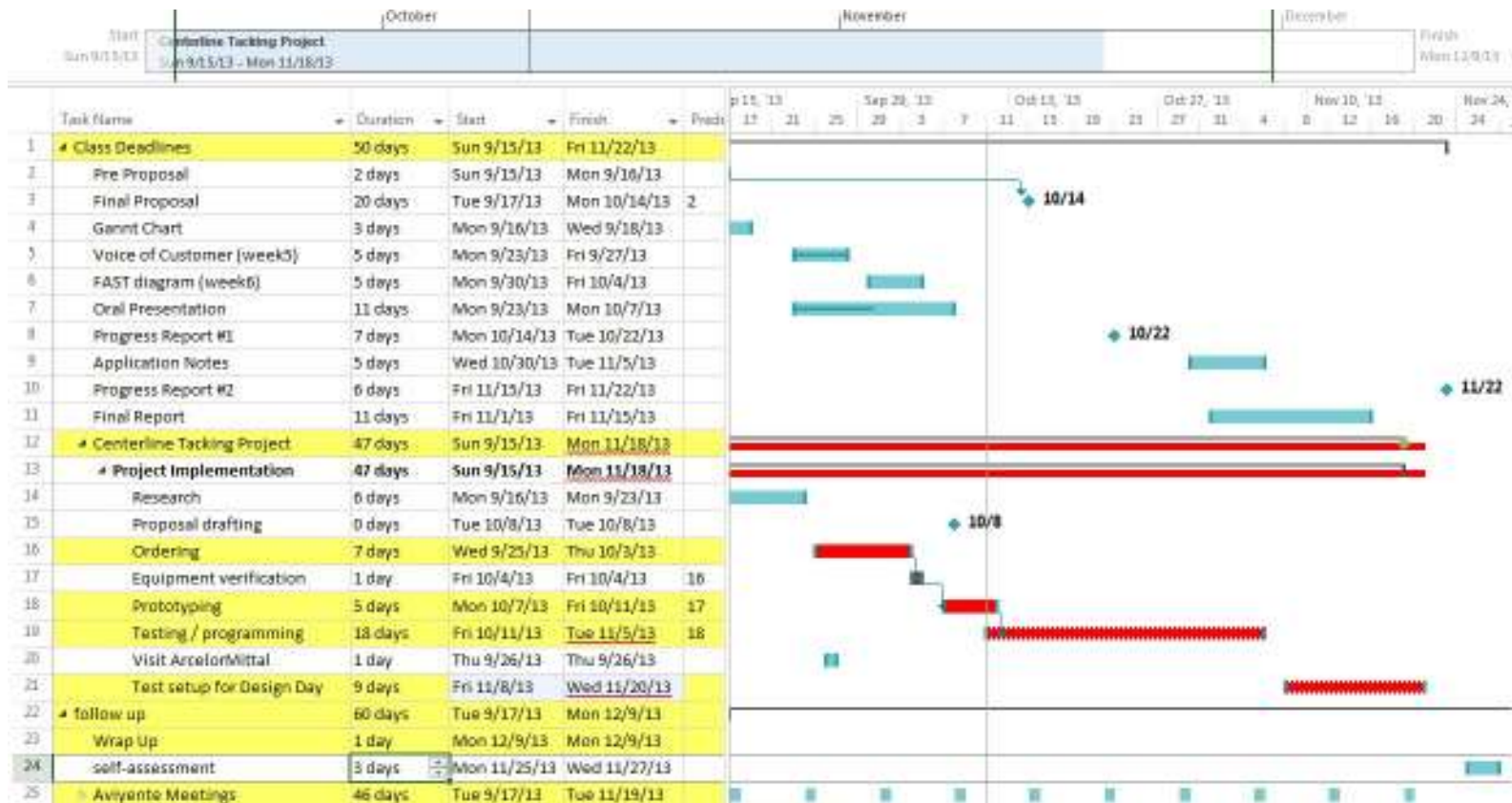
The team has been funded with \$500. The majority of the budget will be spent on purchasing a camera and the BeagleBone black microcontroller. Initially, purchasing a state of the art camera was being considered, but that will go over team's budget. Consequently, it is more feasible to purchase a cheaper, yet high quality camera. Peripheral hardware such as cables and power supplies will not put significant strain on the budget. The cost of building a test setup to demonstrate the project will be around \$50 and only cheap building materials will be required. The estimated cost of the project is \$214, which is well below the budget cap of \$500, and will allow for possible expansion in both materials and hardware capabilities if need be.

Table 2: Budget Estimation

Component	Price
BeagleBone Black Microcomputer	\$45.00
Logitech C920 Camera	\$75.00
HDMI Cable	\$7.00
USB Cable	\$7.00
5VDC 2.5A Power Supply	\$10.00
BeagleBone Enclosure	\$20.00
Demonstration Bench Materials	\$50.00
Total Estimated Project Cost	\$214.00

Project Management Plan

Figure 6 : GANTT Chart



The team's project requires seven critical steps in order to bring the project to its completion. The first step of the project is to conduct the preliminary research required to understand the problem and potential solutions for centerline tracking in the hot strip mill. After researching potential equipment and the software that applies to this project the next step will be to use this information in constructing a proposal and deciding on the optimal design solution.

Once the team concurs on the most feasible design solutions derived from the research phase, the team will proceed with the third step of ordering the equipment. While equipment is being shipped the team will continue to work on the tasks outside of the critical path.

Once the equipment arrives the team will move into the design phase of the project. The fourth step towards completing the hot strip mill center line tracking project is to test the equipment that was ordered. The team needs to confirm the equipment will be able to perform as expected from the research phases of the project. It is important to start this step immediately after obtaining the equipment to ensure it meets the team's needs and will work as expected. If any components do not satisfy the team's expectation it may be necessary to order other equipment. The team may also discover the need for additional equipment during this process. That is why the team must assess the functionality of the hardware as soon as possible. The delay of ordering extra equipment if the first order was not enough will add extra downtime and could significantly affect the completion date of the project.

After verifying the equipment functionality the team will begin prototyping. In this phase of the design the team will interface the equipment with the software. The BeagleBone Black microcontroller and the Logitech camera need to be connected. Once the hardware is

functioning together and the programming interface with the controller is complete, the team will move into the testing and programming step. During the sixth step, the programming and testing the design will be done. This is the bulk of the work required to make the project functional. Therefore, this is the longest phase of the project. This will allow time for coding, debugging and fine tuning of the design. The extra time for the phase is also needed because the team has the least experience in the programming required for this project. The team will need to spend time learning how to use the programming language and to bring software skills up to the technical level required to complete the project.

The final phase of the project will be designing a test bench. The test bench will be used on design day to demonstrate that the project can visually detect the center line of a curved object. Once the project is functioning and the test bench is completed, the team will verify the project performs as needed for design day. This will conclude the centerline tracking project.

The team needed more time than expected for the research and planning phase of the project. The team had a difficult time figuring out the scope of the project and was expected. The team was worried about designing a project that would be suitable for the harsh environmental conditions in the steel strip mill. In the mill there are high temperatures, steam, and the steel strips in the mill move at 30-50MPH. It was decided that the scope of the project would be to design a vision system with centerline tracking at high speeds. The environmental tolerances of temperature and water resistance were not to be worried about for the time being. The team decided on the final solution and ordered equipment. Once the equipment arrived the team verified that it was running. This was to determine if we needed to make any returns. It all worked fine.

The team was not able to move into the design phase right away as planned. Due to other course work the team did not start on time for testing the BeagleBone. This was a problem later on. The team was stuck and unable to get edge detection images. After a lot of trial and error to try and get the BeagleBone to do edge detection images while running Ubuntu the team could not figure out what was wrong. Eventually, the team made so many changes to the Ubuntu environment that it would no longer run. The BeagleBone seemed broken and could no longer boot using Ubuntu. The team needed to start over at this point which set us back. The team then decided to redo the setup of the BeagleBone and try again using a new operating system. Using the default operating system, Angstrom Linux, the team re did all the setup and initialization steps to get the BeagleBone ready for computer vision applications. Once we completed this the team was able to use Derek Molloy tutorials to successfully capture edge images. This was a major step for the team. After this point there was only two weeks left in the course. During this time we were able to develop the code for centerline tracking using the edge detection images as the source of data which the code used to track the centerline.

Given more time to develop the project the team wanted to add sensors to turn the capture program on and off when the steel strip is under the camera. The team planned on doing these using paper weight sensors. When an object moved over the sensor it would trip. Connecting the sensor to one of the GPIO pins on the BeagleBone would switch the logic of the pin high. This can be configured to start the centerline tracking program. It would be an easy modification to the code to let it run until the paperweight sensor turned off. This would effectively run the centerline tracking program while the steel strip is moving under the camera and stops the program once the strip moves on.

Other ideas the group had if given more time on the project we would have been implementing a Digital Signal Processing chip in order to speed up the centerline calculations. This proved to be a daunting task when we were also learning about a newly-released microcomputer. Additionally, our final product could have been designed for higher heat

tolerance. We provided an enclosure for our microcomputer, though extreme heats were not able to be tested in our current environment.

Chapter 3 - Technical Description of Work Performed

Hardware Design Efforts

BeagleBone Black

The BeagleBone Black from Texas Instruments is a microcomputer which handles the image processing software and provides a machine with a user interface. The team selected the BeagleBone Black for its raw power per dollar value. The BeagleBone Black is a new edition to the BeagleBone microcomputer line which released April 2013. Other microcomputers we considered were the Raspberry Pi and Arduino. The Arduino was the beginning of the do-it-yourself microcontroller product line, but had a much lower clock speed and less on-board memory as the other two. The Raspberry Pi has been around for quite some time and there are many available projects that have been

documented on the web. The support for its applications was extensive, but BeagleBone product lines also had a good history of support. The BeagleBone Black was easier to use straight out of the box than the Raspberry Pi, which would reduce the time to start making progress on the project. The BeagleBone Black, with a price tag of \$45 (only \$10 more than a Raspberry Pi), was the best choice for its increased hardware capabilities. The previous BeagleBone products were \$125 and \$145 which did

not justify the extra benefits for the price. The new BeagleBone Black is a serious drop in price with all the functionality. The BeagleBone also comes with a power cord, while the Raspberry Pi does not. This additional cord for the Raspberry Pi is about \$10, bringing the total price of the Raspberry Pi to \$45. In addition, the BeagleBone has a 3-D graphics accelerator. The Raspberry Pi's has a dedicated 3d graphics accelerator which is more powerful than the BeagleBone, but the BeagleBone graphics is good enough. This is the one area that the Raspberry Pi wins over the BeagleBone. As seen in Table 1, the BeagleBone wins in other the categories of processor speed, RAM, memory and I/O capabilities. The BeagleBone has dedicated RAM which is not shared with the GPU and CPU, increasing its speed significantly over the Raspberry Pi. Tests with the Raspberry Pi overclocked to match the BeagleBone proved that even when overclocked, the BeagleBone still runs twice as fast.

Figure 7 BeagleBone Black



Table 3: Hardware comparison

	BeagleBone Black	Raspberry Pi	Arduino	Winner
Price	\$45	\$35	\$29	Arduino
Processor	1GHz TI Sitara AM3359 ARM Cortex A8	700 MHz ARM1176JZFS	16 Mhz ATmega 328	BeagleBone
RAM	512 MB DDR3L @ 400 MHz	512 MB SDRAM @ 400 MHz	32KB	BeagleBone
Storage Space	2G with expandable MicroSD card slot	SD Card Slot	35KB	BeagleBone
GPIO pins	64 usable pins	8 usable pins	14 usable pins	BeagleBone

Logitech C920 Camera

The Logitech C920 is an excellent camera for computer vision applications. It is a high resolution camera that is capable of recording full 1080p HD video. What makes this camera so valuable is that it can encode 1080p in real time using the H264 video compression format. H264 formatting is one of the most commonly used formats for recording and distributing video. The camera has internal circuitry for compressing video to H264 format, which helps take much of the processing burden off of the BeagleBone. The C920 has a high quality Carl Zeiss lens with a 20 step autofocus, which allows for razor sharp images even on close-ups. This camera is very durable in design for a webcam, which will be needed in a steel processing environment. For a top-of-the-line webcam such as this, \$80 is a reasonable price and well within our budget.

By adjusting the resolution of our camera, we can change the speed at which we process images. Within the code for the project, we can define new resolutions by changing pixel height and width. In order to increase the processing time for each image, we are taking image samples at 640p instead of 1080p. This is reasonable to do, considering the large amount of contrast there will be between a glowing piece of steel and a grey background. The same amount of accuracy can be maintained at a lower resolution, while increasing the amount of centerline calculations taking place.

Figure 8 : Logitech C920 Webcam



Demonstration Bench

Demonstration Bench

The design and construction of the test bench was accomplished using parts from previous teams that were left over in the engineering building.

Hardware Implementation/Challenges

There was one major issue that repeatedly occurred throughout the project. That is, the monitor would not detect the micro hdmi signal from Beagle Bone Black microcontroller. As a result, it was time consuming to keep unplugging and plugging the micro hdmi cable until the monitor detected the signal from the microcontroller.

There were rare occasions when the team had to hard reset the microcontroller due to blue screen error. The team was unable to figure out why there were such errors. Fortunately, the frequency of blue screen error decreased significantly later in the project. Otherwise, the team did not encounter any issues of connecting all the devices together. The figure below shows the setup of hardware components.

Figure 9 :

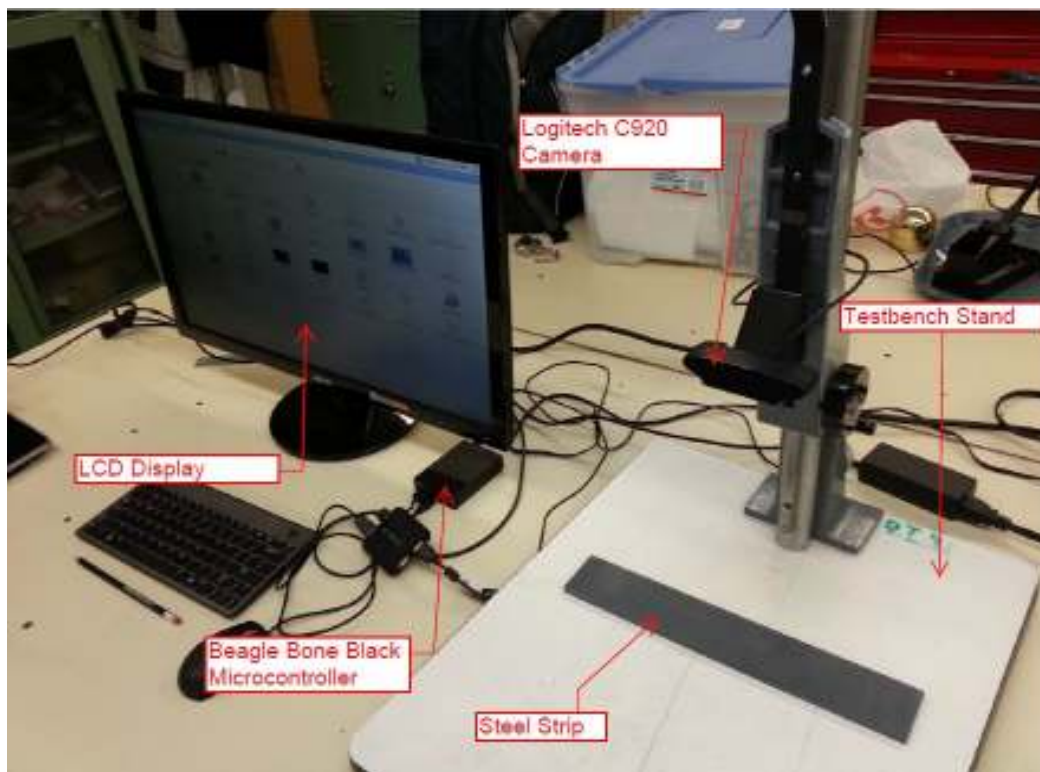
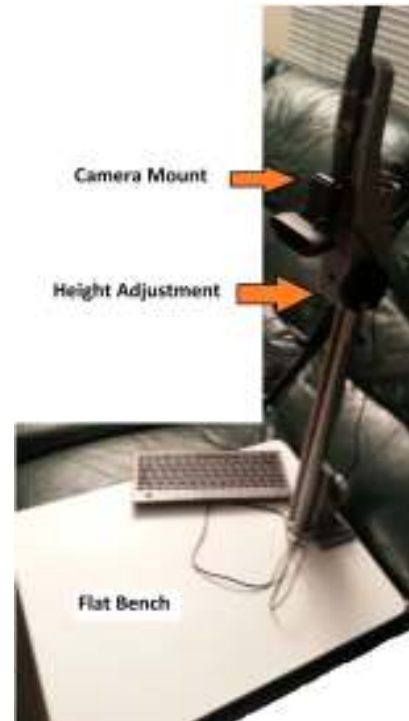


Figure 10 : Setup

Software

The code (boneCV.cpp) created to detect and track the centerline of steel in ArcelorMittal's hot strip mill can be split into three basic sections. The first section focuses on capturing an image of the strip using the Logitech C920 camera and converting the captured image into an edge matrix. The second section analyzes the edge matrix, computes the locations of the edges of the steel strip, and averages the edge locations to find the centerline. The last section of the code outputs computed values and shows the status of various alarms in a user friendly display.

Section 1:

This section of the code uses premade functions in the cv library to identify which port on the BeagleBone is connected to the webcam. It then sets the image size to 640x360 pixels. This image size was chosen because it is big enough to detect a good amount of edge definition and the small size means the code can iterate through images faster, increasing the frame rate. A loop is begun in the code that cycle through all sections of the remaining code: capture, process, and display. The capture time can be changed by the user and is currently set to a high enough number that it is considered infinite. The program uses the capture function in the cv library to obtain a color matrix of the image seen by the camera. The program then uses a premade function to convert the matrix to a grayscale matrix and then implement edge detection of the grayscale image. Edge detection was considered to be too advanced for the group to formulate, so the function Canny was used from the cv library. The Canny function converts the grayscale matrix (values from 0 to 255), into an edge matrix (values 0 or 255), where 255 indicates an edge. The function uses a threshold to determine the level of light contrast required to be considered an edge. The group chose a very high threshold as to eliminate noise; detail is lost in doing this but the edges that are being detected are sharp enough to be acceptable. Next, the left and right most rows of pixels are cut off of the desired edge image. Using a thin, vertical strip allows the program to run faster through iterations. The thin strip also allows the program to view a differential of the strip, allowing the program to track the positions of the edges and not just point out which rows are most dense of edges.

Section 2:

This section of the code begins with a loop that extracts the Mat type elements of the edge matrix into an integer matrix so the values can be manipulated. This loop is then responsible for the creation of a vector that lists the index of each row of the edge matrix and the number of pixels that are considered edges in each row. Another set of loops is used to find the two rows that are most dense with edges. These two rows are then determined to be the edges of the steel strip. Precautions were taken to make sure that each edge is only detected once, as shadows around the edges of the strip can sometimes be detected. It was decided to, after locating the row densest with edges, the five rows above and below this row is ignored during the locating of the second row. Once the locations of the two most edge dense rows are recorded, a block of code potentially swaps the two saved variables to ensure loc1 is the upper row and loc2 is the lower row. Then, a simple averaging line is used to find the center of the row. It was decided that the average of the two edges can be considered the center of the strip.

The group then decided to capture and store the most recent ten values of edge location in order to be able to compare values and find trends in the edge's movement. The vector containing these previous values updates through every iteration (with the first value being the most recent). The program then computes the average of the current values in the history vector. The value in the zero index of the average vector is the average of the current ten history elements. The value in the one index of the average vector is the average of the ten history

elements of the previous iteration. These two average elements have nine of ten values to be identical, so the average matrix does not change abruptly and shows trends in the centerline. As each iteration of the code cycles, the history and average vectors are updated, and previous values move to succeeding indexes.

Section 3:

The final section of the created code focuses on displaying the computed edge values and averages determined in section 2. The display first indicates which image the program is currently processing. The program then displays the current values of the edges and the center that update every iteration of the program. After the current values, the history and average vectors are displayed, with the most recent values located at the top. The bottom section of the display screen shows three separate alarms that trigger when various issues occur. Alarm 1 will trigger when the current centerline value has two values that are drastically different from the previous values. This was implemented to detect when the strip has smaller yet rapid motions. Alarm 2 is raised when the current history value differs over a threshold from the least recent saved history value. Alarm 3 is raised when the current average differs over a threshold from the least recent saved average value. Alarms 2 and 3 both serve to detect general drift over time in the strip and can be used to ensure the strip is in the center of the rollers. Once the alarms are triggered the program must be reset to acknowledge them. Each of the alarm thresholds can be altered in the code depending on the camera's distance from the strip. The display screen updates with each iteration.

Software Implementation/Challenges

The team encountered numerous challenges during implementation of the code. For example, the team was having difficulties with the location of header files and the files needed for pkg-config compilation, so the team decided to abandon the idea of using Ubuntu and write code directly off the BeagleBone operating system.

When the team loaded up the new operating system, the team repeated the process of installing OpenCV and Video4Linux among other required packages. The team utilized a pre made grayscale and edge detection program as well as an image capturing program constructed by electronics professor Derek Molloy of Dublin City University.

The program that Team 4 constructed (boneCV.cpp) goes through several stages of code in order to complete the task of centerline tracking.

The code first uses a pre-constructed function to capture an image from the C920 Logitech Webcam as a matrix of color values. Another pre-constructed function converts the color matrix into a grayscale matrix and then into a binary edge matrix. The edge matrix threshold has been set high so that only drastic changes in gray levels trigger an edge. This reduces noise so that small changes that would produce edges in the grayscale image do not occur, only when the change in grayscale is drastic enough, such as between our grey test stand and our black object. The image that is captured by the camera has a resolution of 640x360 pixels. This resolution was chosen to optimize the frame rate of captured images and also to ensure that the images are detailed enough to gather accurate data.

The team would have preferred to capture a taller and thinner image but the camera did not allow alterations to the capturing aspect ratio. Instead, it was better to use a small strip of our edge matrix for processing in order to speed up the code execution. Only the pixels of a small slice of the cameras view are being used to detect the edges. This reduces noise.

Once the camera had the image of edges, the program could manipulate the matrix. It

took a while to convert values from a MatExpr (openCV variable) to an integer in order for the program to perform operations. The program stored the edge data to a text file and created a matrix giving the value and location of the edge. Afterwards, the program could locate the two highest densities of edges in our image, and assumed those to be the edges of our steel strip. When the team found the row with the highest density, the value and location were both stored in new variables, and the value in the edge matrix would be set to zero. The loop was traversed once again in order to find the second highest edge density and both location and value were stored.

With these two row locations, we determined a centerline by adding the sum of the two row pixel locations and dividing by two to find the centerline.

With the centerline data, the team needed to determine if the data made sense. The program created a blank array and fills it with the current centerline data, along with the previous ten centerline frames. If one or two values suddenly deviate from the previous position, those values can be determined to be “outliers” and will be scrapped. The team has developed a code with three center line checks. If one of the checks fails the program reports that the centerline deviation has exceeded threshold. One check is to see if the centerline has changed drastically from the previous frame. If the change is too great from the previous frame the group determined that the centerline is changing too fast or there is some sort of error. The other two checks compare the centerline and the average centerlines data from the current frame to the values from ten frames ago. If the average center line of the actual centerline value changes too drastically from 10 frames ago the team decided this means the drift of the steel strip is too great and cambering is exceeding threshold. When an error is tripped the program displays a message stating cambering is above the threshold.

Chapter 4 – Testing and Results

Test Bench

It was necessary to determine the range of the camera. The team used a block of steel strip and placed it directly under the camera, and captured the image. From there, the team manually moved the block of steel strip away from the camera slowly, until the range of the camera was determined.

Figure 11 : Centerline Tracking setup with a piece of cambered steel



Figure 12 : Centerline Tracking setup with a straight piece of steel



Testing

Figure 13 : Original Image of a peice of steel on the test bench



Figure 14 : Steel piece converted to an edge image. The extra edges are noise.

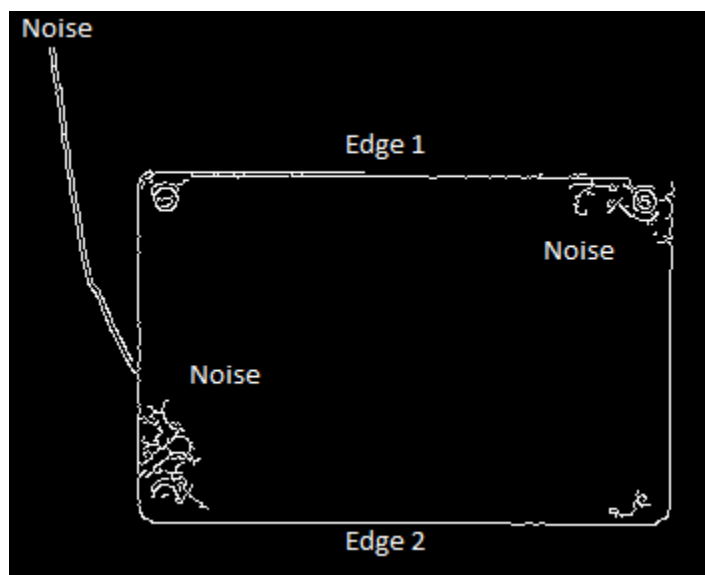


Figure 15 : The image is reduced to a focused area of the edges. This reduces noise because we are only concerned with the edge in this small area for calculating the centerline.



Figure 16 : Centerline Tracking Program display. When cambering exceeds one of the three error check thresholds, that error will be displayed as detected. This test is of a cambering steel strip

```

Captured Image #: 28

Edge1: 155    --Current Values--    Edge2: 234    Center: 194

Index    --Track Centerline--
         Center    Average
-----
1        194        139
2        189        132
3        180        126
4        140        125
5        123        126
6        123        127
7        123        134
8        128        140
9        123        150
10       140        155

Rapid Centerline Shift: DETECTED
Long Term Centerline Drift: NEGLIGIBLE
Long Term Average Drift: DETECTED

```

Figure 17 : Graph of straight steel strip

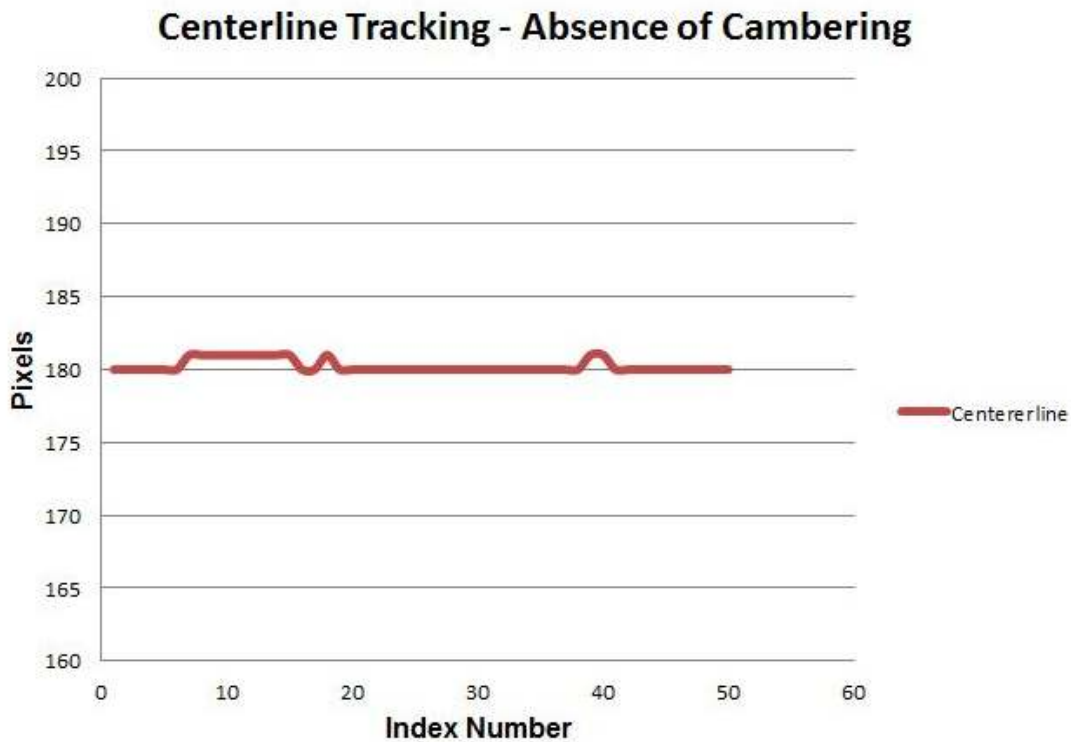
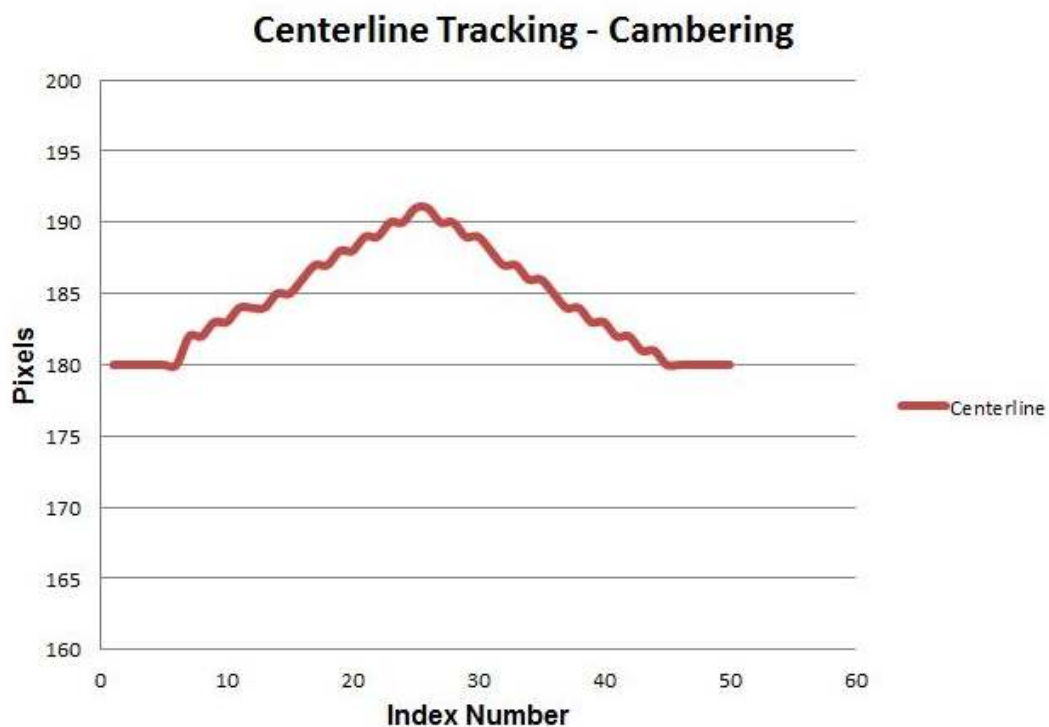


Figure 18 : Graph of cambering steel strip



With the solution for steel strip centerline tracking the team designed there is now a cheap and effective solution for centerline tracking. The setup for the BeagleBone and web camera is minimal which will make it easy to integrate into any steel strip facility.

Chapter 5 - Final Cost, Schedule, Summary, and Conclusions

Budget: Final Cost

Table 4: Budget: Final Cost

Component	Price
BeagleBone Black Microcontroller	\$45.00
Logitech C920 Camera	\$75.00
HDMI Cable	\$7.00
USB Cable	\$7.00
BeagleBone Enclosure	\$20.00
Total Estimated Project Cost	\$154.00

The final budget is shown in above table 4. The original anticipated cost was \$214.00. The team reduced expenditure in the budget by \$60.00; the team used parts from previous teams for the test bench. Otherwise, big portion of the budget was spent on purchasing the BeagleBone Black microcontroller and the Logitech C920 camera. Minor expenditure includes purchasing of the hdmi and usb cables.

Conclusion

The goal of this project was to deliver the first step of a control system to reduce camber wrecks at a reasonable price. By developing a computer vision system using affordable hardware and open source codes, our design is a lightweight, easy-to-use alternative to other industry products that are currently in place and available for purchase. Our group took advantage of the ever-advancing microcomputer market, and applied it in an industrial situation. The new BeagleBone Black paired with a Logitech C920 webcam was an innovative solution to a problem that has plagued the steel industry for years. All of the hardware and software we used up to the edge detection on BeagleBone is readily available for any company that would like to replicate such a system.

An image is fed from the Logitech C920 at a resolution decided by the user in H264 formatting. The image is then fed into the BeagleBone Black, where it is ran through an edge detection program. From this new image, our group took the useful data in order to produce a centerline calculation which will be used in the control systems of various steel mills to reduce cambering.

Our system provides a real-time visual of where the center of a steel strip is at any given point in time. It uses previous values of the centerline to calculate an average, and the short-term or long-term deviation from that point.

The largest success of this project was using low-powered and lost cost equipment to provide a powerful solution.

Appendix 1 – Team Roles, Responsibilities, and Technical contributions



Bryan Blancke - Manager

Bryan was the team manager. His duty as manager including organizing the team and interfacing with the customers. Bryan organized team meetings and established means of communications between the team members. A dedicated phone application called kakaotalk was set up amongst the team to establish group communication with internet or mobile network. This was useful when mobile network was unavailable. The team could still communicate, and all messages were always broadcast to the entire team. Bryan also started a shared folder for documents. The

team worked on reports, presentations, and kept information organized in an online shared folder. This way the team was always up to date on the progress of the other members. As manager, Bryan turned in assignments and kept the team on track towards meeting deadlines. Bryan typically arranged team meetings in the engineering building or at his house, which served as the main headquarters for the teams technical operations.

Bryan's major technical responsibility was to design and build the demonstration bench. This was achieved ahead of schedule and under the expected cost by using parts available from previous teams in the engineering building. Bryan's other major technical contribution were in the design conception and research phase of the project. Utilizing the internet to research applications in the steel industries hot mill strip processes. Using knowledge gained about various systems and hardware available which could be applied to centerline tracking, Bryan brought a lot of research to the table when brainstorming solutions. He was also the main contact with Arcelormittal and performed voice of customer with them to understand the requirements and scope of the project.

Bryan assisted the other team members with the setup and coding of the BeagleBone during the testing and design phase. Bryan's major technical accomplishment during this phase was helping the team when they were stuck and unable to use the grabber program to get images from the camera and convert them to edge detection images. The solution Bryan implemented was changing the operating system from Ubuntu to Angstrom linux and resetting up the BeagleBone for Angstrom. After this the team was able to grab images and convert them into edge images. Development continued from there.



Jeremy Martin - Web Designer

Jeremy initially took the responsibility of setting up OpenCV for use in a Linux environment. All packages were downloaded and installed properly in order to compile and execute image processing software. Having a background in Linux operating systems and image processing, Jeremy spent a great deal of time troubleshooting the problems the group encountered with drivers and compilation on the Ubuntu operating system. He moved header and .pc files (for *pkg-config*) into specific development folders so that our programs could be properly compiled. By accidentally “breaking” Ubuntu after attempting to move a large package, we decided to load the default operating system. We consider this to be a “successful failure” due to the huge amount of progress that came afterwards. The same code was set up on Angstrom for edge detection and the program ran smoothly. After we could properly take images and detect edges, Jeremy wrote the area of the software in which the matrix was sent to an output. From our edge matrix, it could be simplified and manipulated; ensuring only the critical areas would be used. Calculating the highest density area of edges and calculating the centerline was

also coded by Jeremy.

Outside of the BeagleBone, Jeremy completed the entirety of html coding for the team’s website. He also had the idea of using OpenCV for processing images and proposed for the group to use open source programs in order to reach the matrix manipulation step sooner. In the early stages of the project, Jeremy came up with pseudo-code for edge detection that was never implemented due to the speeds of programs available online. Professor Derek Molloy’s website was discovered by Jeremy while researching image processing using the BeagleBone and OpenCV. Derek’s documentation largely contributed to our success and our project may be at a much younger stage if we had not made this discovery.



Mar

k Heller - Document Preparation

Mark initially had the responsibility of flashing the Ubuntu 12.04.3 operating system onto the eMMC of the BeagleBone. The process involved finding a suitable operating system, creating a bootable Micro SD card from the chosen operating system, and installing the operating system on the BeagleBone. After the operating system was installed onto the BeagleBone onboard memory, the now vacant space on the Micro SD card was repartitioned to be used as extra storage by the microcomputer. After the decision to switch to the default operating system this work became largely unused. However, only after it was possible to work in the Ubuntu operating system was it discovered that the switch must be made. It was a crucial step in the development of the project as the demands of the BeagleBone to support image processing was much more thoroughly understood.

During the coding of the team's image processing program, Mark shared the responsibility with Jeremy to process the data received from the camera. Error checking code was written to ensure each edge was only detected once and that noise was reduced as much as possible. Mark also held the responsibility to create the user display screen, an easy to read table of current and past values that update with each iteration of the applied program, as well as optimizing the code to maximize the frame rate of the processed images.

The responsibility of ensuring the quality of deliverables throughout the semester was held by Mark. The task consisted of keeping track of various project deadlines and maintaining a standard of excellence with each report, diagram, and paper to be handed in. The responsibility of preparing the Final Report was not included in Mark's role as the report was large enough to require all team members to contribute in this respect.



Daniel Kim - Presentation Prep.

Daniel's technical role, as stated in the proposal, was purchasing of the equipment, such as the webcam and the BeagleBone black microcontroller. Most importantly, his technical role was focused on testing the microcontroller. However, team's technical roles ended up getting mixed in together. For example, Daniel was tasked with researching and learning about OpenCV and how to implement codes on the microcontroller. While researching about image processing with webcam, our team discovered Derek Molloy, a Senior Lecturer in the School of Electronic Engineering at Dublin City University, on how to interface between our webcam and the microcontroller with OpenCV.

Daniel then researched necessary packages and libraries to install ubuntu on our microcontroller along with other members. Other technical aspects he worked on consisted of helping with writing additional code to track centerline. Daniel also helped with debugging the code as the team added features in the program. For instance, the program team used as a starting point is an OpenCV function. Daniel helped the team figure out how to convert from MAT logic to integer logic type.

Appendix 2 - References

BeagleBone: Video Capture and Image Processing on Embedded Linux using OpenCV. 9 Oct. 2013. <<http://www.youtube.com/watch?v=8QouvYMfmQo>>.

"*Beam Cambering Methods and Cost.*" *Structure*. NCSEA, n.d. Web. 19 Sept. 2013. <<http://www.structuremag.org/article.aspx?articleID=889>>.

EMG hotCAM Camber and position measureme. 2013. youtube. Web. 18 Sept. 2013. <<http://www.youtube.com/watch?v=qjl6jYZJBM>>.

Hot Rolling Mill. youtube. Web. 18 Sept. 2013. <<http://www.youtube.com/watch?v=AuuP8L-WppI>>.

Leonard, Michael. *Makezine*. N.p., 1 Oct. 2013. Web. 25 Nov. 2013. <<http://makezine.com/magazine/how-to-choose-the-right-platform-raspberry-pi-or-BeagleBone-black/>>.

Molloy, D. [DerekMolloyDCU]. (2013, May, 25). *BeagleBone: Video Capture and Image Processing on Embedded Linux using OpenCV* <<http://derekmolloy.ie/BeagleBone/BeagleBone-video-capture-and-image-processing-on-embedded-linux-using-opencv/>>

Molloy, Derek. *GitHub*. N.p., 25 May 2013. Web. 24 Nov. 2013. <<https://github.com/derekmolloy/boneCV/>>.

Process Measurement and Control. Omega, n.d. Web. 18 Sept. 2013. <<http://www.omega.com/pptst/M12C.html>>.

Systems and Sensors for the Steel Industry. Delta, n.d. Web. 18 Sept. 2013. <<http://www.deltasensor.eu/en/catalog/sensors/linear-cameras/tsp.html>>.