Development of a strain microsensor using Moire Fringes

DESIGN DOCUMENT

Team 9

Client: Dr.Long Que

Adviser: Dr.Long Que

Christian Tanberg: Project Manager

Parvaraj Bhatt: Design Engineer

Ki Jun Shin: Research Lead

Matthew Thies: Test Engineer

Team Email: sddec20-09@iastate.edu

Team Website:http://sddec20-09.sd.ece.iastate.edu/team.html

Revised: November 15, 2020 - Final

Executive Summary

Development Standards & Practices Used

Standards

- Following all IEEE safety procedures
- Weekly meetings
- Bi-weekly reports

Software

- MATLAB

Summary of Requirements

- Visible with the naked eye
- Measure compressive strain
- Use of Moire Fringes
- Create a MATLAB simulation

Applicable Courses from Iowa State University Curriculum

EE 224 and 324 - Signals and Systems I and II

EE 531 - Micro and Nano Systems and Devices

EE 532 - Microelectronics and Fabrication Techniques

EE 538 - Optoelectronics Devices and Applications

New Skills/Knowledge acquired that was not taught in courses

Through the research and design of this project, we have learned many new skills that we haven't covered in classes taught at Iowa State. One critical skill we learned was how to prototype new designs. We did this through planning and drawing moire fringe patterns we plan to model in MATLAB. We also learned how to work with a client to reach the desired outcome. Finally, we learned what moire fringe patterns are and how they can be used for real world problems such as determining the strain on a structure.

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

List of Figures

List of Tables

1 Introduction

1.1 Acknowledgement

Our team would like to thank our advisor/client, Dr. Long Que, for providing us with help and guidance throughout this project.

1.2 Problem and Project Statement

Due to an increased use of large structures, such as bridges and buildings, there is a need to develop new ways for determining the amount of strain a structure is under. This is because these structures are heavily used and are important to the infrastructure of our country. An unexpected failure of these structures would be catastrophic, resulting in a loss of life and money. It is also important for the method to measure strain to be cheap and danger-free.

A possible solution was developing a strain visualization microsensor using moire fringes. By using moire fringes we would be able to tackle both the cheap and danger-free issues. The moire fringe pattern would be visible and readable by the naked eye. If the structure is in a remote area or placed where a person cannot approach without protection, then the moire fringe pattern can be read by taking a picture. This could be done with a drone. As a result, both the cheap and danger-free issues are solved.

To accomplish this solution, we plan on writing MATLAB code that will create a moire fringe pattern. It will be able to replicate a real life example of a moire fringe filter which will measure strain.

1.3 Operational Environment

Our final product will be used in locations that can be either accessible or inaccessible to people. If the location is accessible, people would be able to approach the moire fringe pattern to observe the strain level without any possibility of danger. An example is the base of a building or bridge. Inaccessible location could be anywhere that is dangerous for anyone to approach. An example would be the base of a bridge over a deep canyon. In this scenario, the user could utilise a drone to view the moire fringe pattern. The moire fringe pattern should be able withstand extreme conditions, such as heat, cold, wind, dust and other forces of nature.

1.4 Requirements

Functional requirements

- It should be able to detect the smallest change in strain
- It should show the current level of strain

Non-functional requirements

- The moire fringe pattern, which will be letters, must be large enough to be visible to the naked eye
- It must be accessible to a person or a drone
- The moire fringe pattern must be cheap

1.5 Intended Users and Uses

Our product would be used by anyone that requires a structure or object that needs strain to be measured. An example would be a local maintenance department for structures in their district. Another user will be our advisor/client, Dr.Long Que.

1.6 Assumptions and Limitations

Assumptions

- Cost to build this product should be relatively cheap
- Durable, but not permanent
- Strain given as single characters (numbers, letters or symbols etc.)
- To observe the strain level, either person will view it live or a picture could be taken

Limitations

- Strain levels should be single characters to keep simplicity
- Only way to see it would be either in person or with a picture
- Each moire fringe pattern design is unique to the structure
- View strain in levels, not actual values

1.7 Expected End Product and Deliverables

The deliverable to our client is a MATLAB simulation of how the moire fringe pattern would work in person. The MATLAB simulation shows two gratings (patterns), a number grating and a line grating, which should be visible to the naked eye. The number grating is on the bottom and the line grating is on the top. The bottom one is still and immovable, while the top one slides over it. We selected an image that would represent four different levels of strain. We delivered the deliverable on November 16, 2020.

2. Specifications and Analysis

2.1 Proposed Approach

At the start of last semester we started reading many articles given to us by our advisor detailing different ways we could go about building a moire fringe sensor to detect strain on an object. The model we first selected to use composed of a sticker would be attached to structures and would display the strain on the structure. The moire fringe sticker would also be readable by a microsensor. This microsensor will take in the nonelectrical quantity of the moire fringe filter and convert it into an electrical signal. The strain that is read off the filter could either be tensile or compressive strain. The moire fringe sticker will only display images when strain is applied to the structure.

To accomplish this task we started sketching the two patterns that would need to be made to display the image. Through discussion we decided that for the bottom layer of the image we would use numbers that indicate the level of strain on the structure. From here we needed to select the pattern for the top pattern that would allow us to displace each of the levels individually. For this we decided to go with a rectangular pattern that would distort all other levels but the one that was selected.

We continued on this approach through the rest of the semester, but weren't able to start building the moire fringe pattern due to the switch to an online format. At this point we anticipated, we would be back the next semester in person to build the moire fringe pattern, so we spent the rest of the semester further designing the pattern. Unfortunately, we were forced to continue working online the next semester causing our deliverable t0 be switched from a physical model to a MATLAB model. Thankfully most of our designs we had made from the previous semester were able to be applied to the model we were going to build in MATLAB. At this point our advisor gave us a sample program to use that would make simulating the moire fringe pattern easier. With this new code we planned to take an image with the numbers 1-4 that would represent the different levels of strain and use that as the bottom layer that will be attached to the structure. From here we would use the code given to us to overlay a rectangular pattern over the image. We would then take an all white image as the top image and overlay the same rectangular pattern over it with a slightly different pitch. This will allow us to show only one level as opposed to all the levels at once. Unfortunately the code given to us wasn't very user friendly making it very difficult to implement our plans. This led us to make our own MATLAB code.

We took the same approach as above, but instead of using a rectangular pattern we decided it would be easier to use a simple horizontal pattern. After completing this,we discovered the horizontal pattern wouldn't give us the desired moire fringe effect. This was because we planned to slide the two images together in a horizontal fashion this meant to get the moire fringe effect we needed to switch the pattern to a vertical line pattern.

To fix this pattern we flipped the pattern, but it didn't quite work the way we wanted it would display all the levels of strain as opposed to just the one we wanted. After talking to our advisor we discovered our pitch was off. To fix this we recalculated the pitch and were able to create the moire fringe pattern that met the deliverable.

2.2 Design Analysis

Over the last two semesters we have had three different design approaches for the geometrical pattern used to make the moire fringe pattern.

The first design we planned to use was a rectangular pattern that would make small rectangles throughout the image at a defined pitch to distort the image until a second filter is applied to reveal a certain image on the original pattern. In the end we decided this pattern was more complicated than a simple line filter that has the same effect.

This led us to make a horizontal line pattern that occurred every four pixels through the image we selected. This was a random pitch that we tried first just so we could get a better understanding of how the moire fringe effect works. This method showed promise compared to the rectangular pattern, until we went to the testing phase. Since we planned to shift the patterns together in a horizontal fashion to make the moire fringe effect we realized that horizontal lines would not work and we tried vertical lines instead.

With the switch to vertical lines we were close to the getting moire fringe result we wanted, the only issue was we couldn't individually select each level. Everytime we shifted the top filter it would highlight all the levels, not just one. To fix this we talked to our advisor and found out we needed to manipulate the pitch of the pattern. With this switch we were successful in making the moire fringe effect.

2.3 Development Process

Requirements

- Moire fringe pattern must be visible to the naked eye
- Moire fringe pattern must be cheap
- Moire fringe pattern will indicate strain on a structure
- Must be accessible to a person or drone (object that can take a picture)

Design

- Research how moire fringes work
- Create two gratings (number and line grating)

Testing

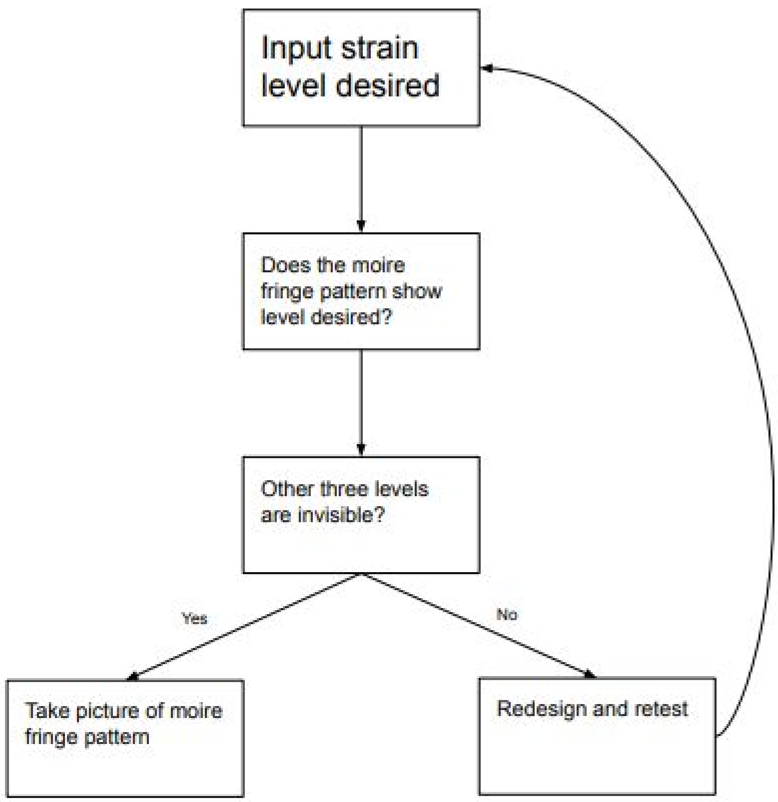
- Take the gratings from design and superimpose (overlay) to get the desired outcome
- Test each strain to see if the other three levels are invisible

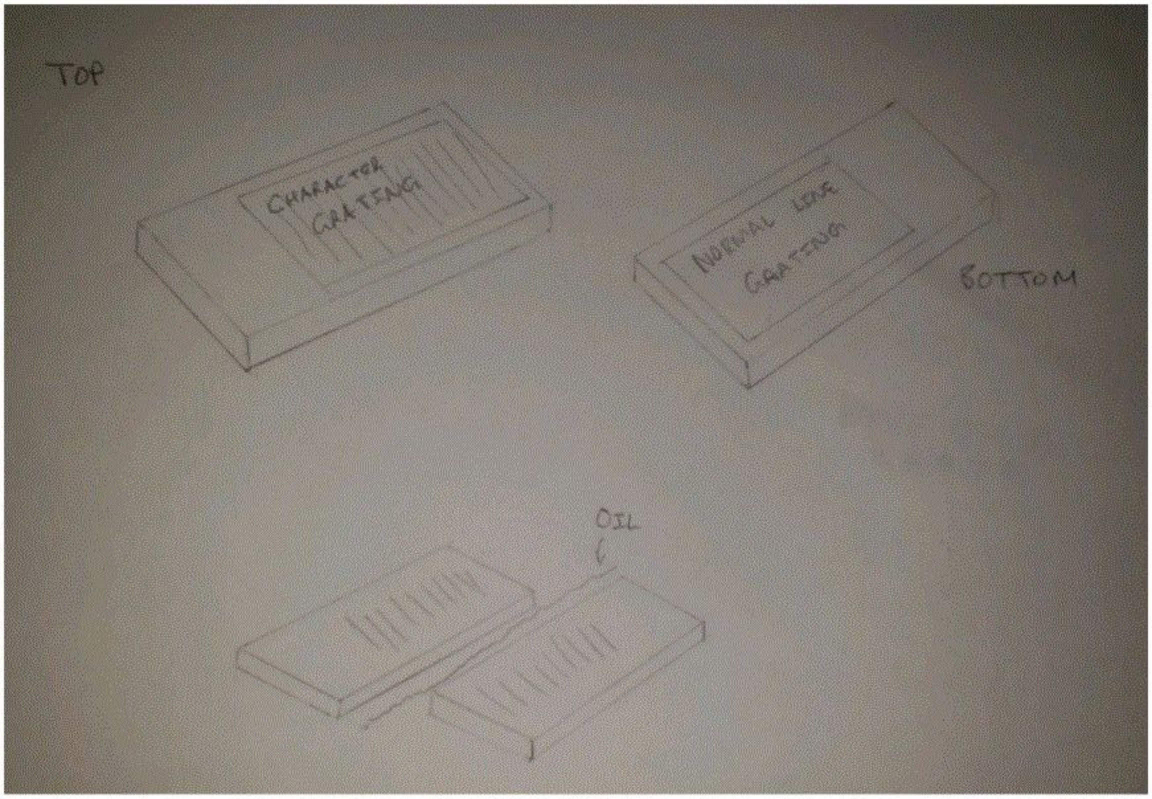
Verification

- Does the final image show levels of strain?
- If they don't match we need to retest or redesign

Our group is using the waterfall development process. This means we are taking the steps of our project one at a time to each step depending on the one previous to it. The first step is look at the requirements of the project and start to do research over the ways we can meet the requirements. Through the research we can start to design how we will build the moire fringe pattern. When we finish designing these components we will start to test them. If the tests are successful we have completed our project, but if the tests are unsuccessful we will need to go back into the process and redesign and retest until we get the desired result.

2.4 Conceptual Sketch





First prototype drawing of the moire fringe pattern we planned to build before we switched to an online simulation.

3. Statement of Work

3.1 Previous Work And Literature

In our research of moire fringes we have found several examples of similar techniques using moire fringes to test the strain on structures. Our technical advisor, Dr. Long Que, gave us an article explaining the basics of moire fringe patterns. The article discussed how moire fringe stickers could be implemented to display different patterns when structures experienced strain. It then would use a camera with a telescope lens connected to a computer to get a measurement of the strain. Our project differs from this because instead of using a camera with a telescope lens connected to a computer to read the strain we plan to make the sticker reable with the naked eye. This would allow the user to either go out and record the strain without the need of a camera. A drone with a camera could also be used to record the strain on the sticker if the location is in a remote area.

3.2 Technology Considerations

The primary strengths of our project is the capability to see the strain with the naked eye. This will allow the user to determine the strain without any access to electrical equipment. This will make the process of finding the strain easier and less expensive than previous methods. Another advantage of our product is the ability to find the strain from a distance with a drone. This will help the user because if the structure is located in a remote or hard to reach location, then instead of having to find a way to reach the area they can fly a drone in and take a picture of the strain. One possible weakness of our product is instead of a machine reading the strain the stickers will display a number that correlates to the amount strain on the structure. This will lead to a higher degree of error due to the visible numbers being an estimate rather than an exact number. One way we can limit this error is through adding symbols or smaller increments between numbers.

3.3 Task Decomposition

Last semester we separated our project into two sections. Two people will be working with the microsensor and two other members will work with the moire fringes. The group working on the microsensor will need to determine how we test the moire fringe pattern and how we plan to implement the microsensor. The group that will work with the moire fringe pattern will need to design and build the filter. This will include picking how they want to display the strain as well as any redesigns that need to be made. Due to the change to online format we have kept our groups the same; we just assigned each group a slightly different task. The group that was working on the microsensor will now work on the MATLAB code to build and simulate the moire fringe pattern. The other group will now focus on the reports and presentations. We choose this format because only two of us have experience with MATLAB, so it will be easier for them to work on the code.

3.4 Possible Risks And Risk Management

There are several possible details that could potentially slow our progress. One of those is the recent outbreak break of COVID-19 which has caused our group to work from home. This will prevent us from meeting in person, which will hurt communication. We also will not be able to meet with our advisor, Dr. Long Que. This is a big hindrance as communication is slowed down and significantly more difficult. Another way it hurts our progress is that without access to labs we will be unable to build or test or project, which has led to a change to a MATLAB simulation of the moire fringe pattern. This has led to several new challenges. One challenge we faced was learning new functions that allow us to superimpose the two filters on top of each other. We also needed to experiment with different geometrical designs to get the right moire fringe effect.

3.5 Project Proposed Milestones and Evaluation Criteria

Some key milestones that we have achieved last semester

- We brainstormed different ideas to build this project
- Met with Dr. Long Que to determine that we were going in the right direction
- Wrote the first version of the design document
- A plan on testing the sensor for next semester
- A final sketch or drawing for when we build it next semester
- Possible testing method

Some key milestones that we have achieved this semester

- A new design to demonstrate the moire fringe effect
- Creating a number (bottom) grating and line (top) grating
- Successful testing of the moire fringe pattern going through all levels of strain
- Completion of the final presentation and report

3.6 Project Tracking Procedures

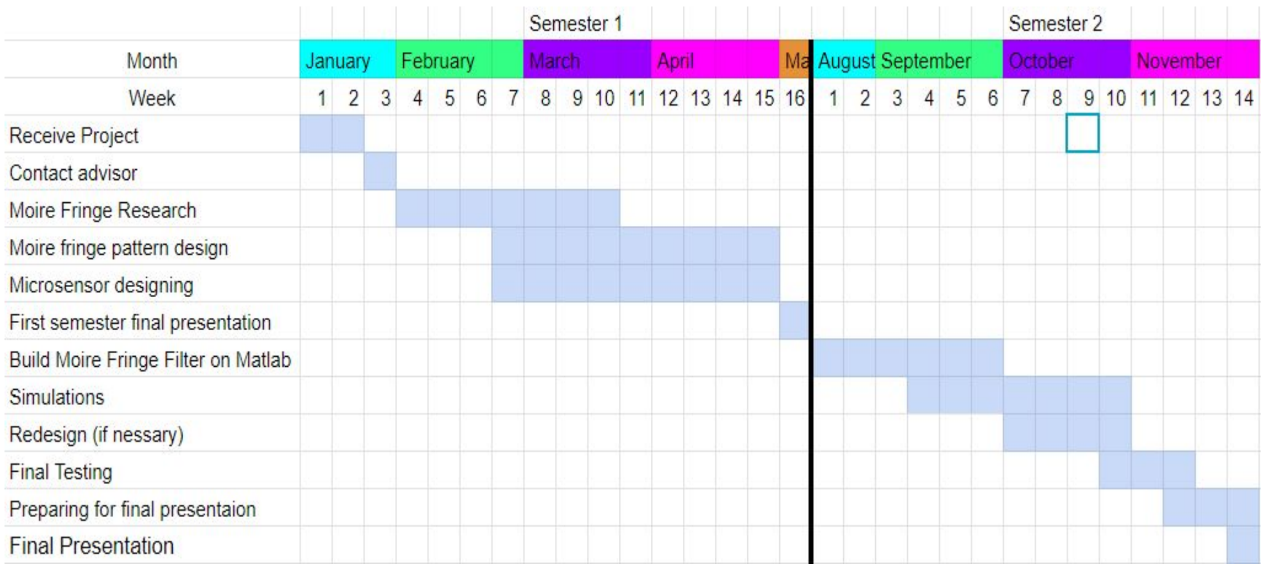
As of now we have been using a progress document with all the required tasks and when they are supposed to be finished by. This has helped us keep on track and to make sure everyone knows what they must complete by what time. Our group has decided to use this method for this semester as well as last semester to make sure we are all focused. We also schedule weekly meetings to give updates on what we have done during the week and if we need help to keep up.

3.7 Expected Results and Validation

The desired outcome of our project is to create a moire fringe pattern that is capable of indicating the strain a structure is under using only the naked eye. To confirm that our code works at a high level we will run a for loop that will iterate through all the pixels in the superimposed image by sliding the top grating over the bottom grating. This will cause the image to go through every level of strain that the pattern is capable of representing. After running the for loop we will identify the points in the loop where each level is clearly defined. This will allow us to know the displacement the object is experiencing at each level.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline



This is our updated gantt chart that represents when we completed each milestone of our project. This gantt needed to be changed several times throughout this semester due to us falling behind and the switch to an online format. The biggest change was the shift to online, this was because instead of actually building the product, we now had to do an online simulation instead.

We started out the year on track working on the moire fringe code, but we fell behind due to the decision to move from the code our advisor gave us to one we made ourselves. This caused us to add a few more weeks for the development of the code. This cut into our testing of the code, but we were still able to meet our deadline due to the fact we built in some time for redesign.

4.2 Feasibility Assessment

This project is feasible to complete according to the gantt chart listed above. There were several foreseen challenges we experienced throughout the project. Some of these challenges include: not being able to meet up in person due to the covid pandemic, issues making the MATLAB code due to our limited knowledge, redesigns of the geometric pattern so we can get the desired moire fringe effect due to our limited understanding of how the moire fringe effect happens.

4.3 Personnel Effort Requirements

Name/Task	Hours Spent	Description:
Modeling Moire Fringe pattern:		
Christian Tanberg	20	Worked with Parvaraj on the image processing of the moire fringe pattern in MATLAB, as well as the creation of the code to upload the images and any redesigns of the moire fringe pattern.
Parvaraj Bhatt	20	Worked with Christian on developing the code we used to make the geometrical patterns for the moire fringe model, including the prototyping of different patterns to get the best result.
Ki Jun Shin	15	Worked with Matthew Thies on the design of the images and patterns: symbols and numbers.
Matthew Thies	15	Worked with Jun on moire fringe design patterns using MATLAB and also worked to create the base gratings.
Testing Moire Fringe pattern:		
Christian Tanberg	20	Worked with Parvaraj on figuring out how to overlap the two moire fringe patterns to create the moire fringe effect as the two patterns are slid on top of each other.
Parvaraj Bhatt	20	Worked with Christian on making the MATLAB function that takes the two images we edited and superimposing them together to create the moire fringe effect.
Ki Jun Shin	15	Worked with Matthew to create the testing plan for the moire fringe pattern.
Matthew Thies	15	Worked with Jun to create the testing plan for the moire fringe pattern.
Documentation:		
Christian Tanberg	10	Worked with the team on various PIRM presentations as well as bi-weekly reports, final reports and posters.

Parvaraj Bhatt	10	Worked with the team on various PIRM presentations as well as bi-weekly reports, final reports and posters.
Ki Jun Shin	15	Worked with the team on various PIRM presentations as well as bi-weekly reports, final reports and posters.
Matthew Thies	15	Worked with the team on various PIRM presentations as well as bi-weekly reports, final reports and posters.

4.4 Other Resource Requirement

We received two resources to help us through this project. The first resource we received is our advisor gave us an example code to help make the moire fringe pattern. The second resource we received was the licence to use MATLAB from the university.

4.5 Financial Requirements

Originally we had a budget of $500 to build the moire fringe pattern, but due to the switch to an online simulation instead, we have no financial resources to conduct the project.

5. Testing and Implementation

5.1 Interface Specifications

As the deliverable changed to a simulation due to COVID, all the testing in our project is through MATLAB, as there aren't any physical elements. The goal is to simulate a moire fringe effect by creating two gratings (patterns). We only used MATLAB to go through the design and testing of our project.

5.2 Hardware and software

As stated in the previous section (5.1), we only used MATLAB to do the design and testing. We knew MATLAB could perform the various aspects of our project, such as image processing. For that reason, we decided to use it for our designing and testing purposes.

5.3 Functional Testing

We had two functional requirements : detect the smallest change in strain and show the current level of strain. These requirements are able to be tested using a MATLAB simulation. In our code, we tested by inputting a certain amount of strain and got the results we desired. We wrote the code that would create the two gratings. To test a change in strain, we inputted the level of strain we would like to see and the correct level was shown. In our number grating, we have four levels, so we tested it for those. We were able to confirm that the images that we were getting was what our client desired.

5.4 Non-Functional Testing

We had three non-functional requirements : the moire fringe pattern, which will be numbers, must be large enough to be visible to the naked eye, it must be accessible to a person or a drone and it must be cheap. These requirements cannot be tested because our deliverable is MATLAB simulation, rather than a physical object.

5.5 Process

As stated in section 2, since our deliverable is a MATLAB simulation, we needed to figure out how to create a number grating and line grating on MATLAB. To create a pattern, we plotted vertical lines on an image (white background and four black numbers). The numbers were equally spaced and easy to see. We selected the thickness of each line and the distance between each line based on the size of the image. The length was 467 pixels, so we selected a distance of 3 and thickness of 1.25. We tested out different distances and thicknesses, but the image did not look the way we imagined.

For the line grating, we took a blank white image and plotted vertical black lines on it. We then used a formula to help us figure out what the distance needs to be. The thickness would remain the same. In moire fringes, the thickness of the lines should be the same, but the distance has to be different. Using the formula, we figured the distance to be 3.039033. We had the value, so we plotted the lines.

For the final part of testing, we took the number grating and line grating images and superimposed them. Our goal was to have one number highlighted (darker) than the other three. The other three should be virtually invisible. This part of the project was where we did the most testing, as we did a lot of changing values of the distance between lines and thickness of lines. The process was to change distance and superimpose the images, which we repeated multiple times. In the end, we were able to get the result our client desired.

5.6 Results

Our results are shown in the images below.

Figure 1: Numbers Image

Figure 2: Number Grating

The first image is the original image of the numbers which has height of 160 pixels and length 467 pixels. We took that image and plotted white vertical lines, which is shown in the second image. This image is the number (bottom) grating, meaning it will not move. In this image the distance between the white lines is 3 pixels and the thickness of each is 1.25 pixels. We reached those values through testing different numbers and seeing if the image looked correct.

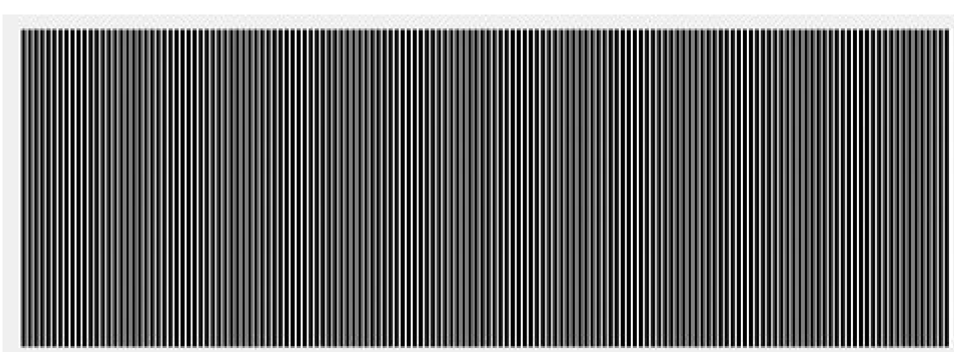


Figure 3: Line Grating

The image above is the line (top) grating, meaning it will slide across the number grating. We took a blank white image that was larger than the number image and plotted vertical black lines on it.

To superimpose, we needed the images to be the same size, so that the gratings would line up properly. We chose the value of 3.015 for the distance between lines, which we got to through testing different distances and picking what worked best. We also created three other line gratings with different distances : 3.027, 3.0545 and 3.089.

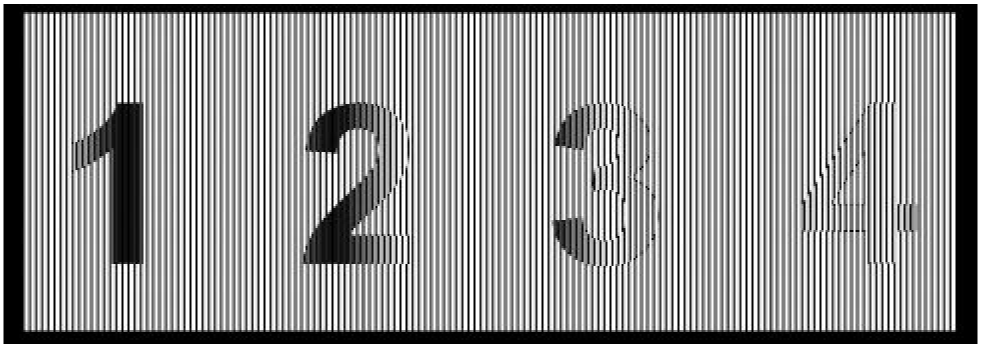


Figure 4: Level 1

This image is the superimposed image of number grating and line grating. The lines in the line grating are separated with a distance of 3.089 pixels. It shows that the level of strain at the moment is 1. This means that the structure is at a resting state i.e no danger at all.

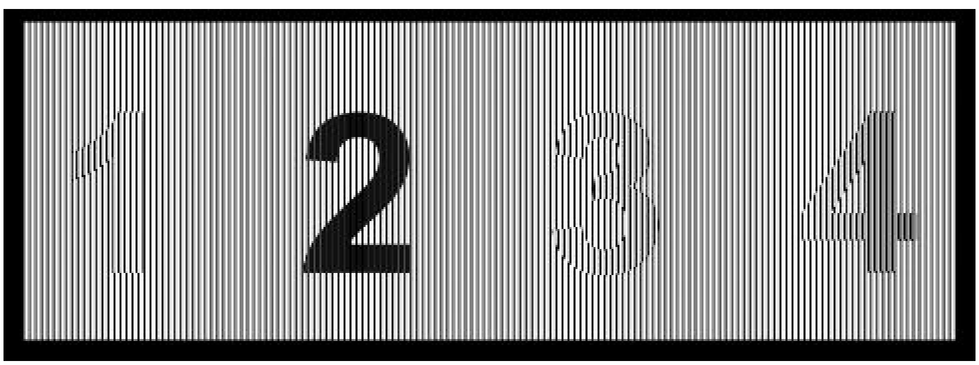


Figure 5: Level 2

In this image, the distance between the lines in the line grating is 3.027 pixels. It shows that the level of strain is at 2. This means that the structure is under minor strain i.e not dangerous but keep an eye on it.

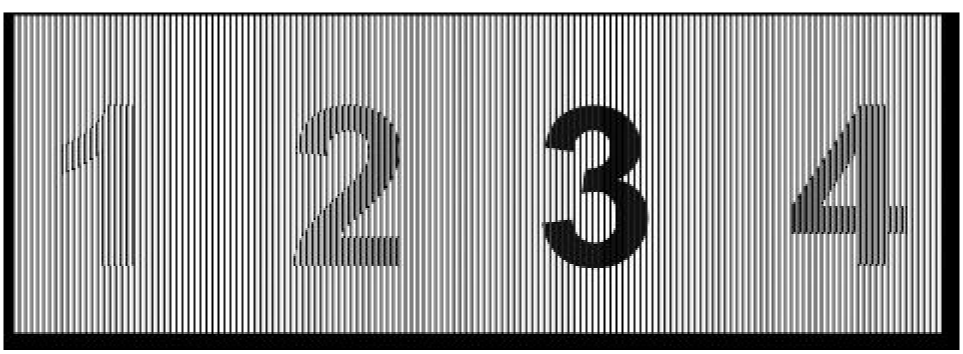


Figure 6: Level 3

In this image, the distance between the lines in the line grating is 3.0545 pixels. It shows that the level of strain is at 3. This means that the structure is under a moderate to high level of strain i.e problem needs to be fixed.

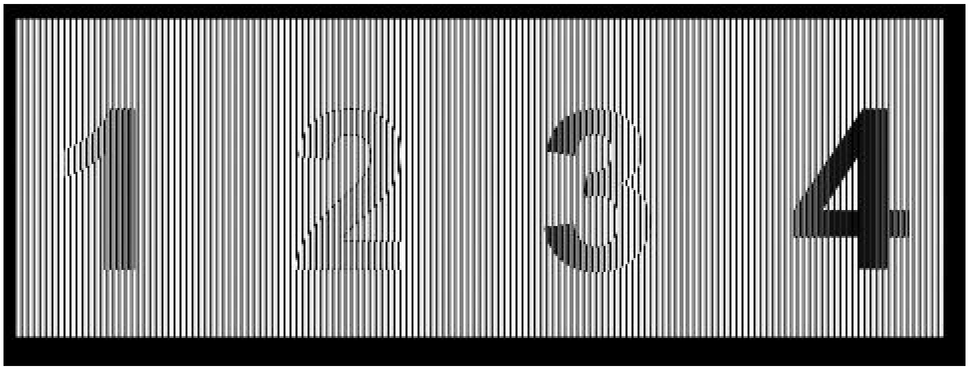


Figure 7: Level 4

In this image, the distance between the lines in the line grating is 3.015 pixels. It shows that the level of strain is at 4, maximum. This means that the structure is under a dangerous level of strain i.e possibility of collapse.

The two images below are the first methods that we tried and were failures. In the first image below, we used horizontal lines to create the pattern, but it showed each number. After that, we decided to try out vertical lines, which looked better but not invisible . We tried to fix that problem by changing the distance between each line in sections of fourths. However, as shown in the second image all the numbers are still visible. Afterwards, we realized that we were blending the number and line gratings, so we changed it to difference. The original line to superimpose was imfuse(TopGratingImage, NumberGratingImage, 'blend');

which we changed to imfuse(TopGratingImage, NumberGratingImage, 'diff');

Figure 8: Horizontal Pattern

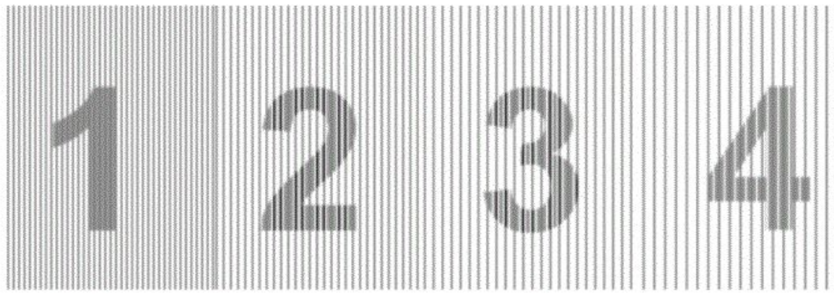


Figure 9: Different Distances in Lines

We faced some challenges due to online learning and the change in the deliverable. We had to learn how to work with MATLAB to create these images. Our entire team was online, so we couldn't meet in person to work together. This caused us to work separately in the beginning. However, we overcame that by meeting once a week to discuss our progress. In the later part of the semester, we started using Webex and Teams to work together. We also used online resources such as Google drive work on a report or presentation together.

6. Closing Material

6.1 Conclusion

We worked on using moire fringes patterns as the transducing signals so that the strain of a structure can be monitored with the naked eye. The goal of this project was to create two patterns, using MATLAB that will produce a moire fringe pattern representing strain level. We have successfully created the code to produce these moire fringe patterns and through testing we have proven our code works as expected. We used MATLAB to build and simulate the moire fringe patterns. This process required us to do image processing to create two gratings. Hopefully, our customers will be able to use our products to prevent the collapse of a bridge or building. Users will also be able to properly maintain any structure under compression by simply looking at the indicated strain level showing from our moire fringe pattern.

6.2 References

Moiré pattern. (2020, April 16). Retrieved April 27, 2020, from https://en.wikipedia.org/wiki/Moir%C3%A9_pattern Takaki, T, et al. "Strain

Visualization Sticker Using Moiré Fringe for Remote Sensing." Bridge Maintenance, Safety, Management, Resilience and Sustainability Bridge Maintenance, Safety and Management, 2012, pp. 2212–2217., doi:10.1201/b12352-330.

6.3 Appendices

Number Grating Code

```
% Number Grating - This code is used to create the number (bottom) grating
% For this code to work the 'number.tif' file needs to be in the same
% folder

NumberGrating = imread('numbers.tif');
% Reads in the number image file
imshow(NumberGrating);
%Shows the image "numbers.tif"
hold on

R = size(NumberGratingImage,1);
% R is the number of rows the image has (height of image) - 160 pixels
C = size(NumberGratingImage,2);
% C is the number of columns the image has (length of image) - 467 pixels

for k = 0:3:C
        x = [k k];
        y = [1 R];
        plot(x,y,'Color','w','Linestyle','-','LineWidth',1.25);
end

% This for loop is used to plot white vertical lines across the
% NumberGrating image with an interval of 3 pixels and each line
% has a thickness of 1.25 pixels
%
% When the figure pops up, save the image as a tif file no compression
% with the name NG1
```

Line Grating Code

```
% Line Grating - This code is used to create the line (top) grating

TopGrating = imread('white.tif');
% Reads in the blank white image
TopGratingResized = imresize(TopGrating, [160 467]);
% Original image needs to be resized so that it matches the NumberGrating
imshow(TopGratingResized);
% Shows resized image

hold on
R = size(TopGratingResized,1);
% R is the number of rows the image has (height of image) - 160 pixels
C = size(TopGratingResized,2);
% C is the number of columns the image has (length of image) - 467 pixels

for k = 0:3.05461:C
        x = [k k];
        y = [1 R];
```

```
        plot(x,y,'Color','k','Linestyle','-','LineWidth',1.25);
end

% This for loop is used to plot black vertical lines across the
% TopGrating image with an interval of 3.015 pixels and each line
% has a thickness of 1.25 pixels
% 3.015 interval - Level 3
% 3.027 interval - Level 2
% 3.0545 interval - Level 4
% 3.089 interval - Level 1
% Save four images with the names TG1 - 3.089, TG2 - 3.027, TG3 - 3.015, TG4 - 3.0545
```

Superimpose Code

```
% This code is used to superimpose the two images from the number grating
% and line grating code
% For this code to work the the two images saved need to be in the same
% folder

clear;
NumberGratingImage = imread('NG1.tif'); % Reads in the image saved from number_grating
code

% TopGrating = imread('TG4.tif');
% TopGrating = imread('TG2.tif');
% TopGrating = imread('TG1.tif');
% TopGrating = imread('TG3.tif');

% Use any of the TopGrating in the for loop below

for i = 0:1:50
        TopGratingImage = imcrop(TopGrating,[i 0 651 273]);
        MoireImage = imfuse(TopGratingImage, NumberGratingImage, 'diff');
        imshow(MoireImage);
        pause(0.25);
end

% This for loop is used to move the top grating across the bottom grating

% The code below is used to represent taking in strain and outputting the
% correct level of strain, when using the bottom code comment out the for
% loop

% prompt = "How much strain is there?";
% strainvalue = input(prompt);
% % Asks for the strain
%
% if (strainvalue > 0 && strainvalue < 2)
%       TopGratingImage = imread('TG4.tif');
% end
%
% if (strainvalue >= 2 && strainvalue < 3)
%       TopGratingImage = imread('TG2.tif');
```

```
% end
%
% if (strainvalue >= 3 && strainvalue < 4)
%         TopGratingImage = imread('TG1.tif');
% end
%
% if (strainvalue == 4)
%         TopGratingImage = imread('TG3.tif');
% end

% MoireImage = imfuse(TopGrating, NumberGratingImage, 'diff');
% imshow(MoireImage);

% The TopGratingImage in the four if statements reads in the images saved
% from the Top_Grating code (will have 4 different tif files). Depending on
% strain level the MoireImage will overlay the two images and then show it
```