

A Disappearing A-Pillar

DESIGN DOCUMENT

Team Number: sdmay 18-20

Advisers: Jaeyoun Kim

Team Members:

Wenrui Wu - Team Communicator

Yixuan Wang - Image Processing Programmer

Liao Han - User Interface Programmer

Shengliang Liu - Program Tester

Yaowei Lee - Team Website Designer/Image Processing Programmer

Guantong Zhou - User Interface Programmer

Team Email: sdmay18-20@iastate.edu

Team Website: <https://sdmay18-20.sd.ece.iastate.edu/>

Revised: 2017/12/3

Table of Contents

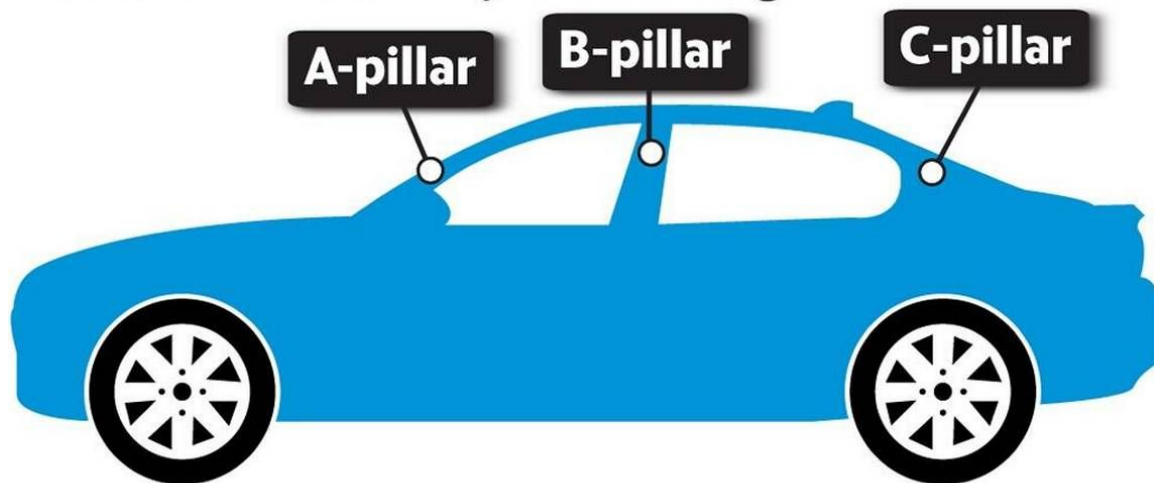
List of figures/tables/symbols/definitions	2
1 Introduction (Same as project plan)	3
1.1 Acknowledgement	3
1.2 Problem and Project Statement	3
1.3 operational Environment	3
1.4 Intended Users and uses	4
1.5 Assumptions and Limitations	4
1.6 Expected End Product and Deliverables	4
2. Specifications and Analysis	5
2.1 Proposed Design	5
2.2 Design Analysis	
2.3 TECHNICAL APPROACH	5
3. Testing and Implementation	6
3.1 Interface Specifications	6
3.2 Hardware and software	6
3.3 Process	6
3.4 Non - Functional test	
3.5 Functional test	
3.6 Design testing	
3.7 Result	
3.8 Implementation Issues	
4 Closing Material	7
4.1 Conclusion	7
4.2 References	7
4.3 Appendices	7

List of figures/tables/symbols/definitions

A pillar: A-pillars are the two pillars at the front of the vehicle holding up the windscreen. Pillars are the vertical or diagonal supports located at the window areas of a car. A car usually has four pillars, namely A, B, C, and D from the front to the rear. In this project, we are going to focus on the A pillar.

Blind spots

While safer, today's cars present a number of blind spots drivers must be aware of when driving.



The Bellingham Herald

1 Introduction

1.1 ACKNOWLEDGEMENT

The purpose of our project is to find a way to improve a driver's vision field while making turns or going in reverse. To be more specific, we are going to achieve this by minimizing the impact of A pillar on what the driver can see. A pillar is the vertical or near vertical support of a car's window area, and a car usually have four pillars, named A, B, C, D moving from the front to rear. In this project, the most

difficult part is how to transfer image onto a tablet and make drivers feel like they are actually looking through the A-pillar.

1.2 PROBLEM AND PROJECT STATEMENT

Based on accident statistics for 2014, there were 3,401 seriously injured cyclists and 113 fatalities. For the same year, there were 5,289 severely injured motorcyclists and 339 fatalities. Despite being only 1% of all road traffic, motorcyclists accounted for 19% of road traffic fatalities. According to Motoring News, The number of crashes caused by blind spots has even increased by 50% over the last two years. Road users such as, motorcyclists, cyclists and pedestrians are most at risk to this blind spot because they lack the protection that a car provides. A specific example would be when drivers are waiting to make a turning at an intersection, they are unable to see what is hidden behind the A pillars, be it pedestrians or cyclists, due to this blindspot. Our project serves to provide a remedy to this problem by increasing the peripheral vision range of the driver.

In order to achieve this goal, we are going to use the camera of a tablet to capture what is going on around the car, especially the region that is obstructed by the A pillar. The images captured are then displayed on the screen of the tablet which is attached to the A pillar, so that the driver can “see through” the A pillar and avoid accidents. The most fundamental part of our project would be to transfer the image from camera to the tablet, because this is the main part and we need to write code using Android Studio and upload it onto our device. Due to the fact that we were previously unexposed to using Android Studio and image processing, these are going to be the most challenging portions of the project. As for the final outputs, we hope that our clients, motorists for example, they can use our tablet to render the A pillar “invisible” and see what is going on behind the A-pillar and avoid oncoming accidents.

1.3 OPERATIONAL ENVIRONMENT

Our product is a software that is designed to be used in a tablet operating with an Android OS, the usage of this product is to assist drivers to get a wider peripheral view. Since this tablet will be affixed to the A-pillar inside the car, we don't need it to be waterproof. However, we need to ensure the tablet is securely fixed onto the pillar so it wouldn't rotate or fall off while driving, which would be potentially hazardous for the motorist and other road users alike.

Since the battery on the tablet is limited, we need to consider charging the tablet's battery while users are using the product. Therefore, the tablet must be continuously charged via a cable connected to the USB port in the vehicle.

1.4 INTENDED USERS AND USES

Our target user base for this project are motorists whose vehicles have wide A-pillars, due to the fact that vehicles with wider A-pillars will obstruct more of the driver's' field of vision. This blind spot will be further exacerbated and ameliorated depending on the size of the A-pillar which differs from vehicle to vehicle. For our project however, it is assumed that the user base mainly drive Sports Utility Vehicles(SUVs) since these cars generally have wider A-pillars.

The motorists can attach the tablet onto the A pillars of their vehicles. The region outside the vehicle obstructed by the A pillar will be displayed on the screens of the tablets, giving the illusion of it being transparent. The section of the camera screen displayed will be parallelogram shaped, mirroring the shape of the A pillar. The angle, width, and height of the parallelogram display can be adjusted by the user depending on the width of the A pillar and the height of the driver.

1.5 ASSUMPTIONS AND LIMITATIONS

Assumptions:

1. The A-pillar should be as wide as possible but no wider than the tablet width.
2. Our product can be easily attached and detached on the A-pillars of any vehicle, so that different users with different vehicles will be able to use this App.
3. The product can be used for all kinds of cars in the world, and our app will store the data of the A-pillars of different cars after people input these information in our database.
4. The images acquired by camera will be cut into a parallelogram shape. They will also be magnified but the image quality and clarity will be as good as pre-magnification, and the cropped out parts will be opaque/blacked out.

Limitations:

1. The cost to produce the end product shall be as low cost as possible, no other external devices are required excluding the tablet.

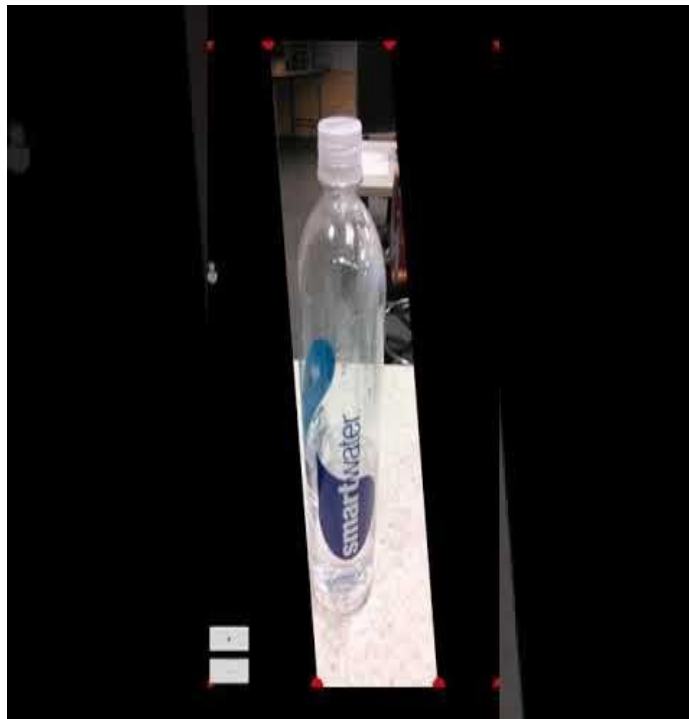
2. The camera used in this project is the built-in camera on the Nexus 7 2013, with outdated camera specifications in comparison to the state-of-the-art UHD cameras available today.
3. The system used is the Android OS, therefore the application will be unusable on IOS devices.

1.6 EXPECTED END PRODUCT AND DELIVERABLES

The end product will be an application installed on a Nexus 7 2013 device. So the item delivered to client is just the tablet. The tablet is rechargeable when it runs out of power.

Functions:

- 1) Customizable width and angle of block size in accordance to that of the A-pillar in different vehicles.
- 2) Zoom in/out functions to provide magnification of images if needed.
- 3) Touch adjustable block shape by dragging the red dots at the 4 edges of the parallelogram as shown in **Fig. 3** below.



2. Specifications and Analysis

2.1 PROPOSED DESIGN

Our mission is to reduce the impact of pillar on the sight range of the drivers as much as possible. We need to make an app that is user-friendly, easy to operate and fully functional. The app can be used in any Android device.

With this app, driver's eyesight can penetrate the A pillar and see the real live scene from the back of A pillar. In order to achieve this, we acquired a tablet, tablet can transfer the image from back camera to the screen. In the way, people can see the blind point scene on the screen from the back camera. We cannot simply use the camera function in tablet. Because there's a camera button on the bottom of the screen which can impede the driver's sight when they are driving. In addition, the UI on the original camera function of the tablet is smaller than in real life. Our goal is to make the A-pillar look like it is "transparent" as much as possible, so that the driver can see what is hidden behind the A pillar. In order to transfer the real live scene from the back camera, we need to use the Android studio to develop an new App to transfer our image.



To do this, we first need to design a user interface. On the main interface, users can easily and quickly find the camera interface. This interface is able to show the real time sight of the outside pillar, and enhance the drives' judgment to the outside. So the camera interface should have a real-time image transmission function.

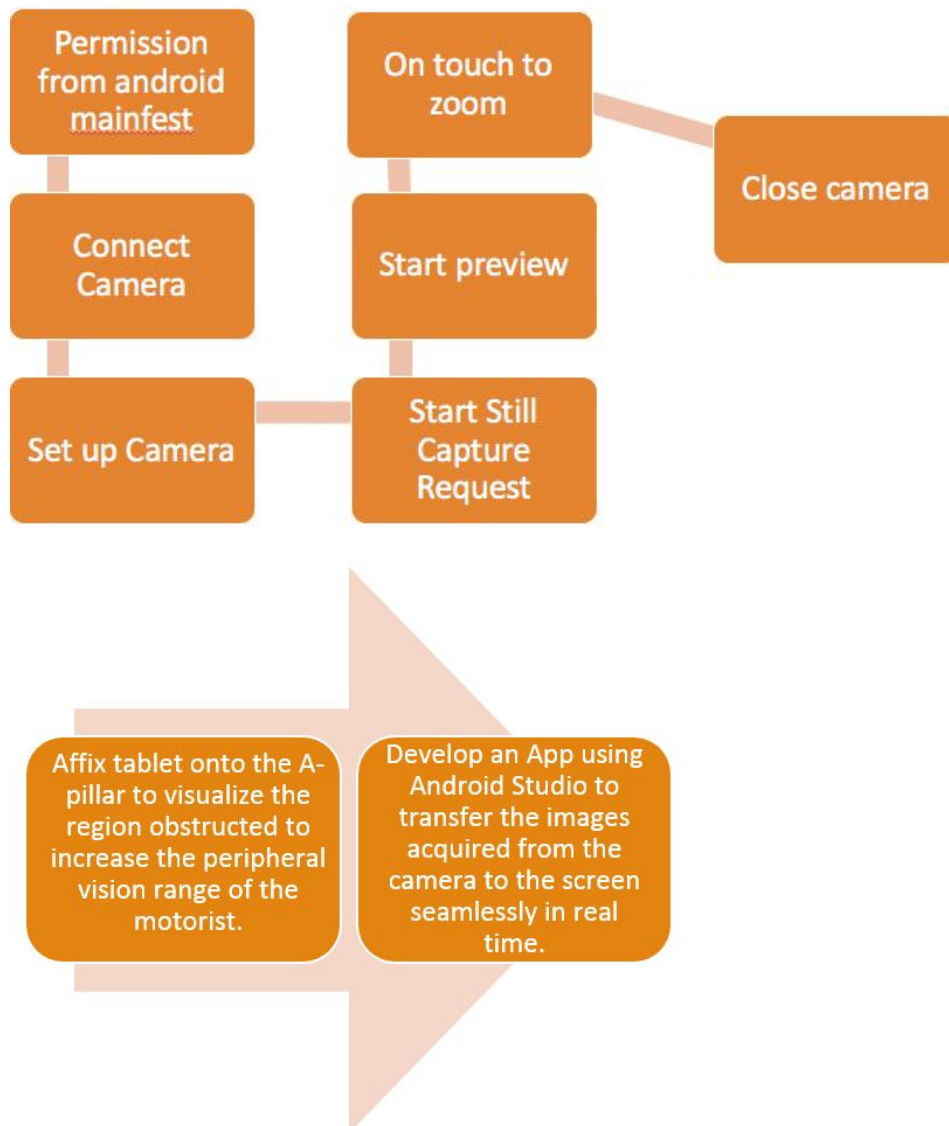
In addition, the app will provide some other auxiliary functions, such as user can customize the angle and width of the image according to their own vehicles. this function is convenient to use. it will ask two input data form users, angle and width. And it will automatically produce the shaped screen corresponding to the number entered by users.

Since we do not expect every customers will be able to measure their vehicles. So we will collect some most common brand vehicles, provide 30 or more

measurement information for users to choose directly. We are going to implement this in “Pillar category”.

Finally, we will complete a subdirectory called “Privacy setting” so that user can save his or her vehicle pillar information inside of the app database. then it can be used more quickly every time they need it. Of course, in order to make it more convenient and efficient for customized to use our products, we will write users’ guidelines after we finish most functions of our app. We will also attach the contact information of our team members so that the customers can contact us in time and give us more feedback in time. This will allow us to update our production depends on the requirements of customers.

Below is the detailed design processes of the main camera function for our project, which is also the most important part of our project.



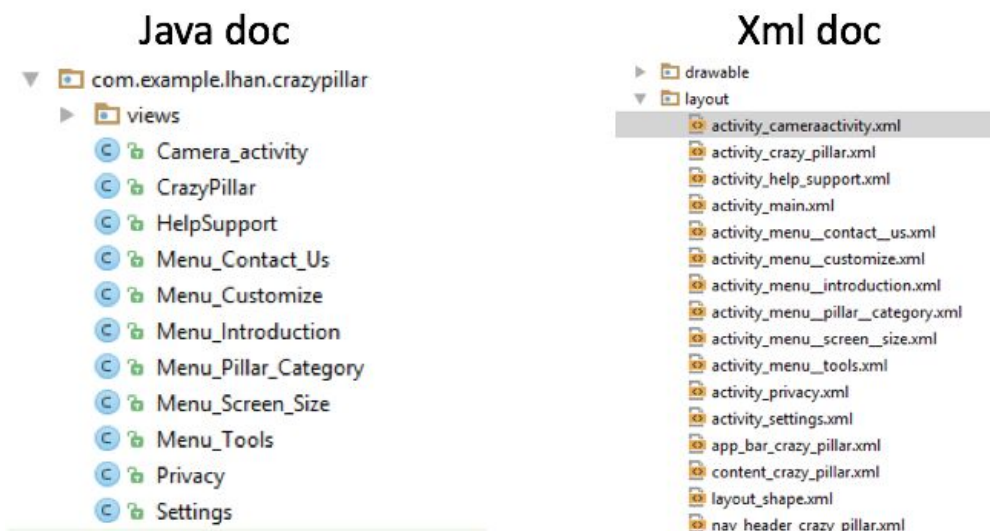
2.2 DESIGN ANALYSIS

In the past six weeks, our team primarily designed and did this project part by part. Starting with the learning of the creating process of android application, we read and collected some articles and videos which are related to Android Studio online. It took us two weeks to design the user interface. In general, our UI design is brief and straightforward. We designed a camera button in the user interface that allows users to access the camera interface most quickly. The “menu” and “settings” tag are added at the upper left and right corner. The catalog contains “Screen size”. “Customized”, “Pillar category” and “General product description”. I will explain the details and usage of these subdirectories following.

For the subdirectory, we were unable to finish all of the sections. But we basically completed our preliminary design according to plan, for example, we added a main camera page which provides a parallelogram-shaped camera UI. In the default cameras of general android devices, the camera UI stretches across the entire screen. In our camera interface, the screen displays images which would have been normally obstructed behind the A-pillar under normal circumstances. And by completing the “Customized” function, we let users have the opportunity to customize the shape and width of the camera block as they please. At present, we have failed to pass the user typing data from other App Activity type class to Textureview type class. we will focus on this problem and figure it out these next few weeks. This could be a significant step to completing our other subdirectories like “pillar category” and “user privacy settings”.

In addition, we finished working on “Contact us” and “About product” by editing the corresponding layouts. This will give users a clear description about our product and team members. We have the plan to add an extra category in menu which will guide the user to use our product effective through some text content or video recording by us.

Below are the functions that we have in our project.



Another thing we need to explain clearly here is the method that we used to achieve the shaped image view in the camera interface. By looking on references online, we found that there are two methods to it. One is to create a shaped image by some complicated image processing work and image rotation. We tried it at the beginning but we gave up. The reason is this method requires a lot on the knowledge of the camera building and image processing. Even if we were to realize the shaped image finally, it would still be hard to make it user-friendly. So at the end, we used the second method which will initialize a full screen image and cover it by a shaped layout as shown on the right. This method is much simpler and the shape is easily customizable. And also, we tried to use the fisheye to make our image bigger, but we found that fisheye cannot make the image on tablet accurate cover the A pillar. We decided to develop the zoom function in the end.

The most important difference between our product and the other car's camera system is that: our product only need a tablet with an android. Compared with the car camera system, the cost of our product is very low. Nowadays, if some cars are equipped with the camera system, it might cost additional \$1000 cost, and most of them are back up camera. The cars with front camera are luxury cars and they are usually sold at exorbitant prices. Our product can be easily added on any kinds of car.

There are some problems that we need to consider. Since we have some hardware limitations, there are some delay on our real time camera. When driving fast, the tablet is not able to show a clear image because of the low quality of our camera.

2.3 technical approach

Our main functions are included in the camera activity. They have 3 main functions, which are customized width and angle of the pillar, touchable and modified block and supported zoom in/out function.

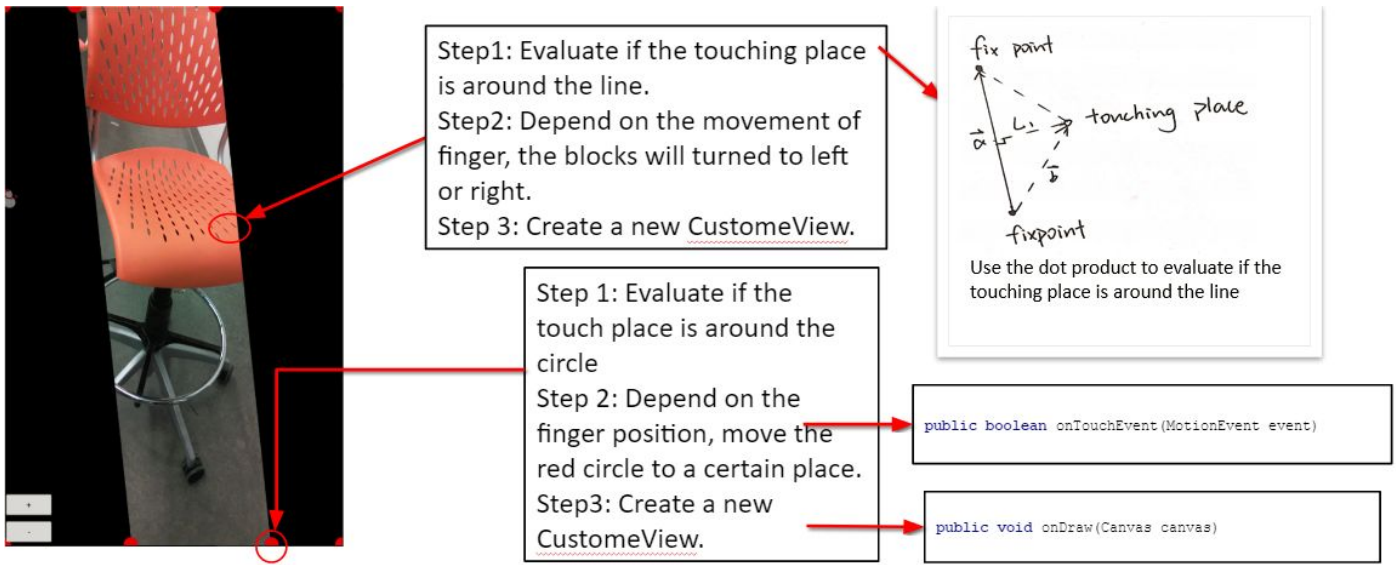
Customized width and angle of pillar: First, we need a specific layout to allow users to type in data as they need. As shown, there are several TextView to guide user to type in the corresponding data depend on their need. Below each text guideline, there is a EditText, which left for user to type in the data. When the user type in their desired data (Length and angle) and hit the button, then it should automatically pass the data to our camera activity. This is the challenge that we faced in this function. We used a method called `button.setOnClickListener(new View.OnClickListener())`. This function will be ran every time when the user hit button. In



the function, it will create a type of variable called Intent. In order to connect this customized function with our camera activity, we use the variable like this: `Intent intent=new Intent(Menu_Customize.this, Camera_activity.class)`, and we use the `putExtra("need changed value",updated value)` to pass those two variables into camera activity, and when the camera get those two variables, it will start to build a new block using the class we wrote as `CustomeView`. There is a function called `public void changeSize(int width,int angle)`. It will ask two parameters which are passed from customized class to camera activity and finally pass it into `CustomeView` to create a new block. After all, it will start the class camera and show a new layout.

Touchable and modified block:

There are two ways to modify the block by finger gestures moving. Those two methods can change the shape of the block by moving the corner and dragging on the line to change the width of pillar as needed. Since all of these functions are based on the finger gestures. so we use a function called `public boolean onTouchEvent(MotionEvent event)`, it will detect if the user put their figure on the screen and make some action corresponding to their finger moving. For example, if we put our fingers on the corner of block, it will evaluate if the point is close enough to the corner, if the answer is yes, it will start following the moving finger to build a new shaped block. The function we called to build block is `public void onDraw(Canvas canvas)`, as long as we get related coordinate on our screen, it will connect those point as line and paint the inside as black. The same as corner moving, the function `onTouchEvent` will evaluate if the point on the line which we want to move. There is a challenge different as the corner moving. Since the line is created by thousands of point. It's impossible to evaluate all of sigal point on the line like what we did before in corner moving. The key is to find the distance from touching point to the line. the method we used is vector cross product. Shown as graph, each line have two fixed point. and we can connect those two fixpoint and touching point to create two vectors. Do the mathematical calculation. we can get the distance from point to the line. If the distance we calculated in the range we setted up, the line will move (actually its two ending fixed point will move) follow by finger moving.



3. Testing and Implementation

3.1 INTERFACE SPECIFICATIONS

Our project is to disappear the A pillar. In order to reduce the cost, we use the tablet to show our test result. At first, we use the Android studio to develop our app, which can show the image after the A pillar directly to make A pillar disappear. Then we upload our app to the tablet to see the results.

In order to make sure that this app is working well, we need to test the app's camera. For the first test, we need to run the APP on our tablet, the app can show the category and the camera icon on the home page. Make sure that all category works well. Then, we need to test our camera activity, the image on the screen captured by the camera will have an angle where it will be directly parallel to the A pillar in the car. In addition, we need to do the practical test on the SUV. In this test, we will see whether this app can accurately transfer the image from the back of A pillar. By changing the block size and the angle between our tablet with the car windshield, we can test whether our tablet could be able to show the appropriate view of driver.

3.2 Hardware and software

Hardware: NEXUS 7 tablet with a camera

Nexus 7 tablet is a long rectangular size tablet. In order to make the A pillar disappearing as possible as we can, the shape of the tablet should be similar to the shape of A pillar. When we set this tablet on car, it can be coincided with the A pillar to disappear the blind point on the car. So that people can see penetrate the A pillar by using this tablet

Software: Android studio

Android studio is a kind of software that help people developing their own app. We learned the function and the method about how to program. And we use it to create an app to disappear the A pillar on the car.

In our project, we use the Android studio to develop our App, we use the Android studio to build an interface of our App for Nexus 7 model. And then we call the camera function in the home page of our app. In order to make this App functional remove the A pillar, we did not call the back camera function directly, we write the code by ourselves, so that we can remove the camera button on the bottom of the tablet. The tablet works more efficient when it transfer the image from the back of camera to the front. We use the emulator on Android Studio to test our App, then download the APK file to our tablet and see if it works.

We measured an angle and the width of the A pillar on a SUV. Then we write the code to read the input from user to set an angle and make the screen size the same width as the A pillar. So the image shown on the tablet can be paralleled with the A pillar.

3.3 PROCESS

In order to see whether our project could accomplish our goal or not, we need to find a way to test it out. First of all, we need to test the camera preview, more specific, we need to know whether our camera is able to show the real time sight of the outside pillar or not, and can it enhance the drives' judgment to the outside. In order to test it out, our plan is to take pictures from driver side through A pillar, then take pictures through our tablet and compare these two pictures, if the picture taken with tablet shows what is missed or blocked by A pillar, then it works. In the next test, we need to see whether users can customize the angle and width of the image according to their own vehicles. In this test, we will first fix our

tablet onto the A-pillar of the model car and then by adjusting the block size of the tablet (because each team car has different height and view angle). If the tablet screen accurately displays what is being obstructed by the A-pillar, then our project has passed this test.

3.4 NON - FUNCTIONAL TESTING

We are mainly responsible for the main function of the disappearing A pillar for the first semester. After one semester of effort, our main function camera activity has been improved and still needs to be optimized. The rest of the non-function include **Menu_Contact_Us**, **Menu_Introduction**, **Menu_tools**, **privacy** and **setting**.

We have completed two of them, introduction and contact us. In the introduction, we roughly introduces the design ideals and purpose of our product, this can give the potential users the most intuitive understanding on our product. We found several potential users to read our product introduction. They gave us very positive feedback and think it is very concise and clear. As shown, We added our app icon include the version number as the header and used the colorful background. The majority of customers feel good about our page design, young dynamic background and concise product introduction.



In our second non-functional activity, we added the picture for each team members, contact information, roles and basic introduction. This page is designed to be convenient for users to know more about our design team. If customers have any opinion and questions about our product, they will found our contact information based on our particular roles. And this will be very important to improve our app in the future. We also find a few potential users to evaluate our introduction page, they think that the role distribution is very clear, but they are looking forward to see more introduction details on our team members just in one or

two sentence. We have adopted this idea , and it will be updated in the next version of the app.

3.5 FUNCTIONAL TESTING

In this part, what we try to test is our program that we wrote in our tablet. In our program, we created an interface where the user can adjust the screen size of the tablet and find the best screen size that fits the size of the A pillar in their car. More importantly, we need to see if our tablet could work transparently that it makes the A pillar looks like it disappeared. How we test it is that we let one of our team member stand directly behind the A pillar, with his whole body completely blocked by the A-pillar. We then attached our tablet onto the A-pillar of the model SUV and adjusted it until we can completely see our team member. If our team member is visible on the tablet screen, then our app works perfectly. If not, then there is still room for improvement.

Because it is dangerous to test the function on the road, we do the test as the photo shows first, it show that the hand which blocked by the A pillar before, we can see now. But, it is not enough because often on the road, cars will not be so close like this. So we need to fix the tablet in front of A pillar and drive the car at very low speed in parking lot. So that we can check whether the car not very close to us can be seen in the tablet by the driver angle.

3.6 Design Testing

As shown in the image below, our test car is Ford Edge. We have used mount to hold the tablet on windshield. Then change the angle until the image in the tablet match the driver's view. We use the phone to record the video from the driver's view when the vehicle is moving. And we let one of our team member standing behind the A pillar, let the pillar completely block him, then we attach our tablet on the model SUV and adjust it until we can see the whole body of our team member from driver view. If we can see it, then it is a success, if not, then we need to adjust angle of our tablet and test it again.



Step 1:
Test car(Ford Edge)



Step 2:
Mount used to hold tablet in place is affixed onto the windscreen



Step 3:
Tablet is attached onto The A-pillar.



Step 4:
A video is recorded while the vehicle is moving from the driver's view.

3.7 RESULTS

So far, the result is pretty good, the tablet could provide an image that shows what's going on outside of A pillar. But there is a slight delay of image, which is what we need to do for our next plan. And we are going to have discussion with team member to deal with those problems.

3.8 IMPLEMENTATION ISSUES

After the testing, we have find there are some issues that we need to handle. First, we have found that the image we record is not very clear. Because we have zoom in after call the camera function, so the image will be obscured. Also, there is a problem when the light reflect on the screen, the driver will not see the tablet clearly. The third issue is that the image we get from camera is delayed, so we need to make sure it changes simultaneously in real time.

4 Closing Material

4.1 CONCLUSION

We have completed the primary goal of our project in accordance with the wishes of our client/adviser, which is to make the camera capture real time images and then crop the images obtained into a size that would fit the size of the A-pillar. This is because we need to show the real time sight of the area outside obstructed by the A-pillar in order to enhance the driver's' visual judgement outside the car. So the camera interface should have a real-time image transmission function.

We have managed to implement a function which can magnify the images on the tablet if needed. We have also managed to implement an auxiliary function which allows users to customize the block size according to the width and angle of the A-pillar in their vehicles. Lastly, we have also added some touch functions on the camera interface such as the ability to increase or decrease the height of the display just by pinching your fingers outwards or inwards and also the ability to cut the size of image in the screen.

As for future plans for semester 2 of Senior Design, we are uncertain for now as our adviser has yet to give us any relevant information. After successfully completing our primary goal of making the A-pillar “invisible”, we would like to continue working on this project. We wish to implement more functions to our App and also to further refine already available functions in the App to higher standards.

4.2 REFERENCES

- <http://www.bellinghamherald.com/news/traffic/rules-of-the-road/article160216674.html>
- <https://www.drivingtesttips.biz/driving-test-tutorials/a-pillar-blind-spots.html>
- <https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>