Universität Karlsruhe (TH)
Forschungsuniversität · gegründet 1825

# Automatic Distortion Correction for a Full Windshield Head-up Display System

Study Thesis

Fakultät für Informatik
Universität Karlsruhe (TH)

School of Computer Science
Carnegie Mellon University

by

**Fabian Blaicher**

Supervisors:

Prof. Dr. Alexander Waibel
Dr. Jie Yang
Dr. Rainer Stiefelhagen

# Contents

# List of Figures

# 1. Introduction

Driver Assistance Applications are supposed to help drivers in difficult driving situations. These situations can be caused by many different reasons. One class of difficult situations is caused by extreme weather, which can limit the drivers ability to recognize the environment or to control the vehicle. Another class consists of situations where the traffic itself is challenging for the driver, for example in unknown areas. The third class of dangerous driving situations can be accounted to things distracting the driver from the street. The distractions can be caused by a range of reasons from disturbing car passengers to the use of electronic devices like cell phones or car electronics.

Driver Assistance Applications have three major functions [6]. They provide real-time information of the surroundings, warn the driver in dangerous situations or warn the driver of possible upcoming dangerous situations.

Several Driver Assistance Applications have already made it into mass production. Navigation Systems, Adaptive Cruise Control and Night Vision Systems are already available from several car manufacturers. But there are still many problems related with those systems. The systems often cannot provide 100% guaranteed reliability. Another problem is the usability of those systems. They often lack human-computer interfaces which appear intuitive to the user. This problem results in a higher mental workload of the driver, which can lead to dangerous traffic situations due to the distraction from the street.

A lot of information is still presented on Head-Down Displays (HDD). This is highly problematic, because the driver has to move its focus away from the street. Head-Up Displays (HUD) can be used to show information to the driver without requiring to move the head or the focus. It has been shown that Head-Up Displays can lower reaction times, improve speed control and cause less mental stress that HDDs[5]. Head-Up Displays have been used in fighter jets for many years and car manufacturers are slowly moving to integrate them in mass produced cars. Current Head-Up Displays in cars display information only on a small part of the windshield, which strongly limits their use.

Head-Up Displays can be used to help the driver in several ways. Their big advantage is that they can enhance the vision of the driver and lower his mental workload, because the driver does not have to relate abstract images on a display to the surroundings. The Display can be used to display arrows for navigation systems. It can be used to highlight interesting parts of the environment, for example road boundaries in different terrain, street signs or it can highlight threats like people or deer walking on the road.

## 1.1 Goal of this Work

This work describes an approach to use a Full-Windshield Head-Up Display from SuperImaging Inc. to enhance the vision of a driver. GM and other car manufacturers have already started research with this device [2]. We develop software to control the laser projector for the HUD. The main problem is to solve for the distortion of the projected image on the windshield. The distortion is due to the curved surface of the windshield.

The full-windshield display (FWD) will be used to highlight street signs. The street sign highlighting software was already presented by Wu [9]. This is useful to direct the attention of the driver to street signs, like speed limit signs or highway exits. The distortion correction work can be further used to display any arbitrary information on the windshield, for example it could be used in combination with a common navigation system to display turning directions.

Finally, we tested the control software in a laboratory. Therefore did we project a recorded care drive to a wall with a video projector and highlight the street signs on the full-windshield display. The control software of the laser could easily adjusted to work in real-time.

## 1.2 Outline

After this introduction chapter we introduce some basics for our work. We introduce the interpolation algorithm we used and give a brief overview about the research field and related work. The third chapter contains the idea how to correct for the non-planar surface distortion. We explained how to correct for the problem and how to get the relevant information needed as input. The fourth chapter goes through the implementation of the process stated in chapter three and explains how the steps were realized. The fifth chapter gives an overview over the final demo and a qualitative evaluation. In chapter six we give a summary about our work and show further areas of interest for research.

# 2. Basics

## 2.1 Run Time Interpolation

We use an interpolation function to prewarp the points which were not precomputed and stored. Our choice was bilinear interpolation as it is fast to compute and can be used for real-time interpolation of image coordinates. We use one interpolation function for each of the coordinates x and y. First comes the description of how to compute the interpolated x value based on surrounding prewarped and precomputed points. The steps for the y coordinate are the analog.
Assume we are looking for the value $x' = f_x(x, y)$, where (x,y) are coordinate points of an input image and x' is the prewarped x coordinate of interest. The idea is to perform interpolation first in x direction and then in y direction. We have four function values $f_x(x_i, y_i), i \in [0, ..., 3]$ of the (x,y) surrounding points $(x_i, y_i)$. The four points surrounding (x,y) are numbered counterclockwise from bottom left to to left 0 to 3. We interpolate first in the x direction:

$$
\begin{aligned}
f_x(x, y_0) &\approx \tfrac{x_1 - x}{x_1 - x_0} \cdot f_x(x_0, y_0) + \tfrac{x - x_0}{x_1 - x_0} \cdot f_x(x_1, y_1) \\
f_x(x, y_1) &\approx \tfrac{x_1 - x}{x_1 - x_0} \cdot f_x(x_3, y_3) + \tfrac{x - x_0}{x_1 - x_0} \cdot f_x(x_2, y_2)
\end{aligned}
$$

The next step of interpolating in y direction gives us the result:

$$
\begin{aligned}
f_x(x, y) &\approx \tfrac{y_1 - y}{y_1 - y_0} \cdot f_x(x, y_0) + \tfrac{y - y_0}{y_1 - y_0} \cdot f_x(x, y_1) \\
&= \tfrac{y_1 - y}{y_1 - y_0} \cdot \tfrac{x_1 - x}{x_1 - x_0} \cdot f_x(x_0, y_0) \\
&+ \tfrac{y_1 - y}{y_1 - y_0} \cdot \tfrac{x - x_0}{x_1 - x_0} \cdot f_x(x_1, y_1) \\
&+ \tfrac{y - y_0}{y_1 - y_0} \cdot \tfrac{x_1 - x}{x_1 - x_0} \cdot f_x(x_3, y_3) \\
&+ \tfrac{y - y_0}{y_1 - y_0} \cdot \tfrac{x - x_0}{x_1 - x_0} \cdot f_x(x_2, y_2)
\end{aligned}
$$

For further refinement of the accuracy of the interpolation one could also use an algorithm implementing bicubic interpolation. As the derivation of the parameters is not trivial one could use the algorithm proposed in the book Numerical Recipes[10].

## 2.2 Related Work

There has been a lot of research about Driver Assistance Applications in general and about distortion correction on arbitrary surfaces. Tardif et al. [3] proposed an approach to project arbitrary images with a video projector to an arbitrary surface. He used special patterns for calibrating the projector(s) to project undistorted images. His work could not applied here, as our laser cannot project the proposed patterns. Another work for automatic calibration of a video projection was done by Takayuki et al. [7], where he proposed a mostly theoretical work to project undistorted images from projectors with unknown poses to a planar surface.

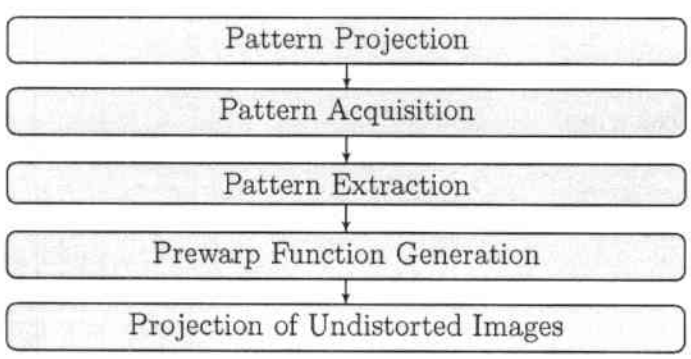
Different research had a more detailed focus to navigation systems, for example Wu et al. [8] presented a multimedia system for route sharing between difference users and video-based navigation by overlaying turning arrows and highlighting relevant landmarks with voice instructions.
Other research groups also worked with head-up displays or proposed ideas for future systems. Sato et al. [1] conducted a similar setup to ours with a full-windshield HUD but used a different projection technology, based on a video projector and a mirror construction. Our system in contrast consists of an advanced FWD and a robust landmark detection and recognition component. Chu et al. [4] recently proposed an idea to use a full-windshield HUD to show various information to the driver. They also proposed the idea to combine a navigation system with the HUD. In their paper was the idea proposed to highlight street signs and display additional information to next to them concerning the attributes of possible routes, for example the distance to the destination or information about traffic jams.

# 3. Design

## 3.1 Overview

To achieve the goal of highlighting street signs, one has to correct the laser projection distortion due to the non-planar surface of the windshield. Therefore one has to measure the distortion on the windshield surface and use the information to prewarp the images before projecting them to the windshield.

Our process to generate a prewarp function is divided in different steps. The first step is to project a pattern with the laser on the windshield. In the following steps is the pattern recorded to video and than is the pattern extracted and the prewarping information is computed. We defined two coordinate systems which will be distinguished in our setup. One coordinate system belongs to the plane of the 2D camera image, its 2D points are denoted by (u,v) and called camera coordinates. The second coordinate system belongs to the input image of the laser projector, its 2D points are denoted by (x,y) and called laser coordinates.




After the prewarp function is generated, the software uses an interface to the street sign detector to display the street sign boundaries. This interface is very simple, as the street sign detector data will not be accessed in real-time. The interface could be easily adjusted to take any format of image coordinates as input even for real-time applications.

The following sections of this chapter describe the steps to generate the prewarp function and its use in detail. The implementation and evaluation of the concept will follow in the next chapters.

## 3.2 Image Distortion on the Windshield

The SuperImaging Inc. full-windshield display is a state of the art head-up display with the capability to project images onto the full windshield with a bright laser. The FWD consists of a laser projector, a controller device and a MediaGlass™ screen. There are two major issues when using the FWD. The first issue is perspective distortion. Perspective distortion means that the projected image appears differently according to the viewing angle. Images projected to the windshield seem to be on different positions on the windshield in relation to the scenery. Figure 3.1 shows a photo of a projected image taken from the left side of the laser projector. Figure 3.2 shows a photo of the same image projected to the same position, but taken from the right side of the windshield where. One can see that the projection seems to be on different position in relation to the background. This problem is not addressed in this work. But to circumvent this issue we used a camera with a tripod and a fixed position in relation to the windshield to have a fixed point of reference. The second

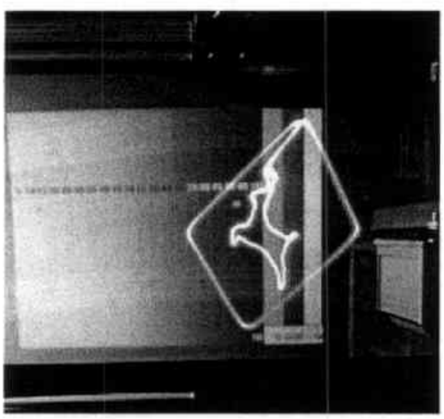


Figure 3.1: A Deer Image viewed from the left side of the Windshield

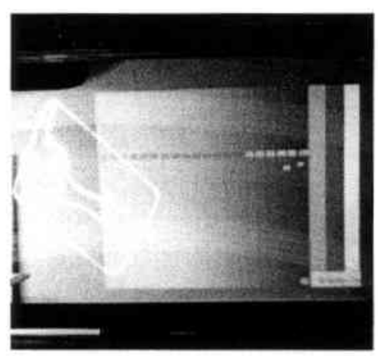


Figure 3.2: A Deer Image viewed from a different Perspective

issue is the laser distortion on the non-planar windshield. This is because of the curved surface of the windshield and the laser is calibrated to project on a planar surface. The correction of this distortion is solved in this work. To compensate for the curved and tilted surface is the distortion on the windshield measured and compensated by prewarping the projected image before the projection.

## 3.3 Experimental Setup

The Setup was built in a laboratory at Carnegie Mellon University in Pittsburgh. The devices of the FWD, manufactured by SuperImaging Inc., were provided by General Motors. Our FWD was a prototype from SuperImaging Inc. which is shipped with a development kit of hardware and software. The main component,

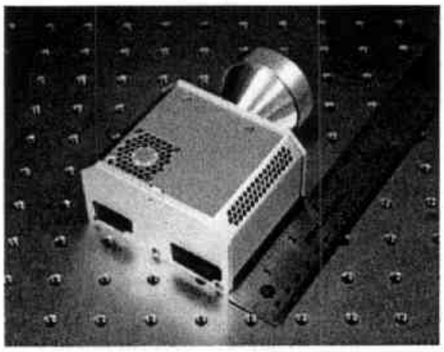


Figure 3.3: Picture of the SuperImaging Inc. Laser Scanner

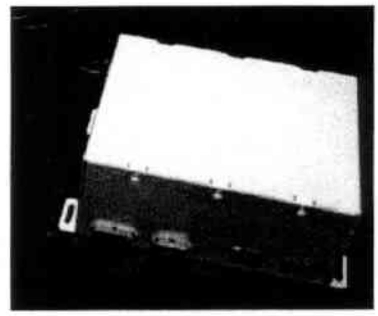


Figure 3.4: Picture of the SuperImaging Inc. Laser Controller

a high speed laser scanner pictured in 3.3 will be referred to in this paper as laser projector. The laser projector is controlled by a USB controller pictured in 3.4. The controller is powered by a 12 V DC power supply. The laser projector is supposed to project to windshield glass, called MediaGlass™ screen, which is covered by transparent phosphors that fluoresce upon absorption of visible light[2]. The special surface offers a strong reflection of the laser ray. This windshield is mounted on a metal frame. The metal frame of the windshield has an extension above the center of the windshield which is oriented to the back side of the windshield. On this extension is another rotatable mount to which the laser projector is attached. This mount is fixed during the whole surface measurement through pattern projection and remains fixed during all experiments. The laser projector can be positioned in two ways. Either it projects from the top of the windshield or from the bottom. In a real car it would therefore project either from the region around the back mirror or from the dashboard.

## 3.4 Pattern Projection and Video Acquisition

Before creating a function to correct the distortion we first have to measure the distortion on the FWD. A common approach to capture distortions of projected images, by different projector types, is to project one or more special patterns and record the patterns either by photo or video cameras. The captured patterns are used to correct for the problematic surfaces. For the correction are functions applied to the images, before they are projected. This is a common way to cope with arbitrary surfaces without having the exact model of a surface.

Our approach is to project a pattern to the windshield, record the pattern with a video camera and use the information to compute a prewarp function. Different patterns were proposed to correct for distortions, for example checkboards. For our setup we chose to project a rectangular grid of unassociated points. The laser coordinates of this pattern form a correct rectangle. The points represent the intersections of grid lines.

The reason for not using patterns, which have already been successfully used, is that our prototype projector cannot project arbitrary patterns. It can not project filled areas. The laser projector can also not project many unassociated grid points

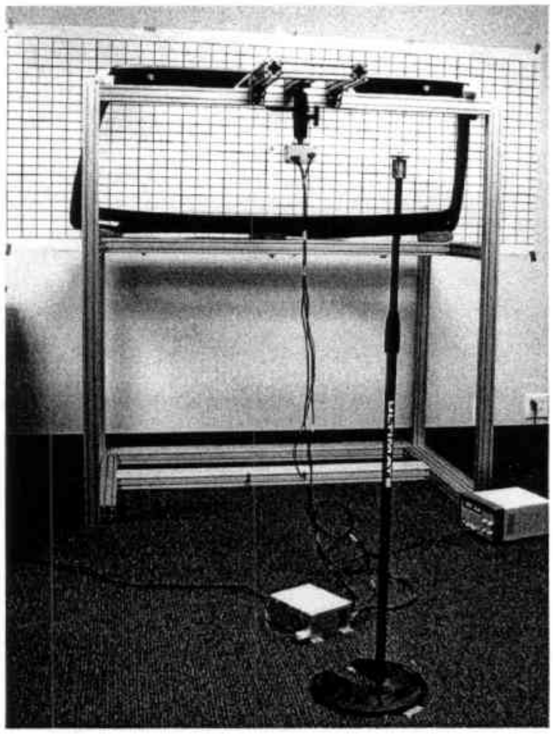


Figure 3.5: The Experimental Setup of the FWD

covering the whole windshield surface. According to information from SuperImaging Inc. are new projectors capable of really projecting arbitrary images, even with filled areas, to the whole windshield.

The projected pattern is finally recorded by a camera. We use a standard consumer video camera with a video recording feature. The camera records the video to a standard video format.

## 3.5 Pattern Extraction from the Video

The next step is to extract the points from the recorded video. We need the coordinates of each position of each blue laser point from the pattern. Therefore we try to extract one coordinate from one frame. After extracting all coordinates we save them to a text file.
The extracted images from the video contain still a lot of disturbing information. To remove the most unnecessary information we crop the images such that the images exactly contain the whole windshield. A common problem when not cropping the images is that there are very bright spots, for example caused by reflections on the ceiling, which are mistaken as the bright laser These spots can result in completely outlying coordinates. The cropped images are the basis for the further extraction of the coordinates.
Now we continue to remove information with the goal to only get the coordinates of the laser points. As we are only interested in finding blue points, we limit the information to the blue color channel of the images.
The next step is to extract the coordinates of the points. Therefore we use a binarization function to set the color of all pixels to black which have a low brightness.

This step returns an mostly black image with a couple of connected white pixels. The centroid of the set of white pixels is used as the coordinate for the grid pattern point.

The list of extracted coordinates must finally be corrected for different problems. Some laser points projected to the windshield are occluded from the projector, this results in missing grid points in the center. As the windshield distorts the pattern some points are projected to the badly reflecting frame and can therefore be not extracted. The last problem is that some points in frames have bright tails which lead to centroids shifted away from the real center.
To solve these problems we propose to use interpolation. The idea is to create one 2nd order polynomial for each row of points. This function is then used to compute coordinates of the correct number of points which are supposed to be in one row.

## 3.6 Prewarp Function Generation

The extracted coordinates of the laser points are the basis for the prewarp function. The function takes points of an image as input and gives the prewarped points as output. If a there is a direct mapping from the input point to one prewarped point, than will be the prewarped point used as the output. In the case that the input point does not fit to one point in the list of precomputed points and therefore has no corresponding prewarped point is interpolation used.
We compute a new grid of points in the coordinate system of the camera (u,v). This grid must be fully covered by the extracted points. We have for all distorted points the source (x,y) coordinates. The idea is to use the relationship between the coordinates of the distorted points and the source (x,y) laser points to compute the (x',y') coordinates of each new grid point. The (x',y') coordinates of the new grid points are called prewarped points.
We use two 2D interpolation functions to compute the prewarped points. One function computes the x' coordinate of each new grid point in the (x,y) laser coordinate system and the second function for the y' coordinate. For each new grid point (u,v) we interpolate the (x',y') values as f(u,v) = x' and f(u,v) = y'. The inputs for the interpolation are the list of extracted and distorted (u,v) coordinates and the corresponding x or y coordinates from the (x,y) laser grid points. The used method is based on Delaunay triangulation.
The prewarped points can now be used to correctly project arbitrary images onto the windshield without distortion.

## 3.7 Projection of Undistorted Images

The program to control the laser is written in C++. It controls the laser via a Riya USA driver. The laser projector is capable of projecting images of line drawings. Images with filled areas cannot be projected with the available prototype. The control program takes the list of prewarped points together with image data as input. Each image will be corrected by the prewarp function based on the prewarped points. The prewarp function uses either table lookup or bilinear interpolation to correct the points. The output of the prewarp function is a distorted iamge. When this image is projected to the full-windshield display it will be distorted because of the non-planar surface and will finally look as the original input image.

# 4. Implementation

This chapter shows how the steps introduced in the design part are implemented. Each step for the creation of the prewarp function is explained in detail and images are provided for an easier understanding. The order is again: pattern projection and video acquisition, pattern extraction, prewarp function generation and the projection of undistorted images on the full-windshield head-up display.

## 4.1 Pattern Projection and Video Acquisition

The first step is projecting a grid of points to the windshield. Figure 4.1 shows a frame extracted from the video. As mentioned in the design part we only project one point of the grid projected at a time. The grid on the wall is not relevant to our setup. The grid of all points is larger than the windshield and therefore are some points projected on the black border. These points cannot be extracted later, as they are not reflected properly.
The camera cannot observe the whole projection as the projector occludes part of the windshield. This problem can be avoided by using a different setup, one solution would be to fix the projector higher up or below the windshield.

The number of grid points was 600. There were 30 points per row and 20 points per column. The minimum number of points which are the required for the setup was required during several empirical tests. All points were projected for 250 ms. This time interval was again the result from different empirical tests. Time intervals below 250 ms are not recommended, as the points will have increasing tails, which shift the centroid of the laser points, and the intensity of the points is lower, too. Higher intervals are possible, but do not provide better results, just lead to a longer time for the pattern acquisition.
The brightness of the room or environment plays a role as lights in the view of the camera or reflections from lights can be so bright that they can be mistaken as the projected grid points. But lighting is only a problem for the pattern acquisition and not for the later operation of the laser projector.
A camera on a tripod takes the role of the driver. This camera has always the exact same position in relation to the windshield, so we don't have to cope with perspective distortion. The position of the camera or driver does not have an effect on laser

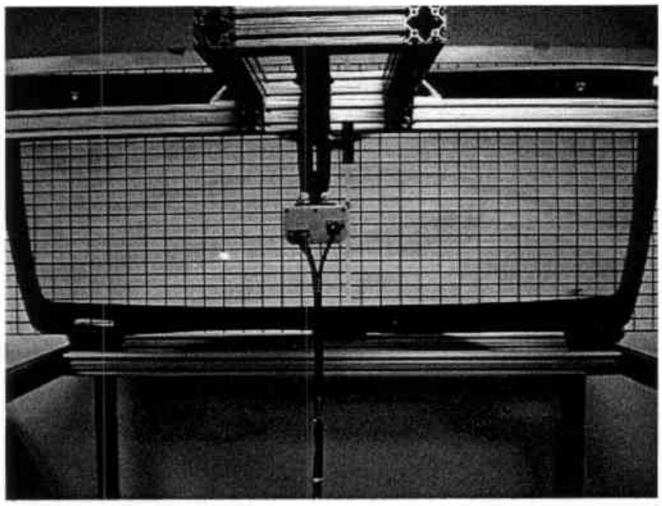


Figure 4.1: Projection of one Calibration Pattern Grid Point

distortion on the windshield. This means the perspective distortion is independent on the curved surface distortion. Therefore one could create a curved-surface distortion correction function and use it for all perspectives of the viewer.
We used a camera Sony Exilim EX-Z77 with a video resolution of 640x480 pixel. Using this camera gave satisfying results, but the earlier use of a camera with a resolution of 320x240 pixel was not sufficient. The video format was Apple QuickTime.

## 4.2 Pattern Extraction from the Video

After capturing a video with the projected grid points we detect the coordinates from the points. Therefore we extract frames from the video file at a frame rate of four frames per second, because we projected each point for 250 ms. We extract each frame in the Bitmap format to limit quality loss. As we extract the images with the projected points from a video there are some frames which contain two blue laser points, because the frame was captured during the movement of the laser point.
Before the coordinates can be extracted one has to correct for disturbances. The first method to limit errors is to crop the image and cut off the borders where the laser projector does not project at all. Reflections in the room with the setup, for example on the metal frame of the full-windshield display will therefore be removed. Figure 4.2 shows a captured frame after cropping.

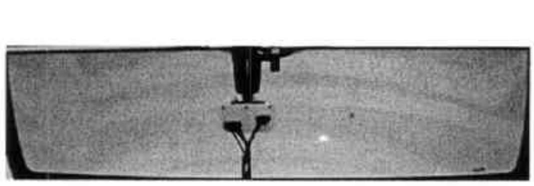


Figure 4.2: Extracted Image after cutting off unnecessary Parts

The next step to increase the accuracy of the laser point detection is to limit the information in the image to the intensity value of the color blue. The grid points are projected with a blue laser, although any color of the laser projector could have been chosen, and therefore can the center of each laser point be assumed as the pixel in the image with the highest intensity of the color blue. Figure 4.3 shows one captured image after limiting the color information to the blue color channel. The image is now a black and white picture, where the brightness of each pixel is corresponding to the blue intensity of the original color image.

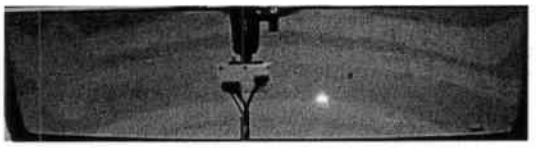


Figure 4.3: Extracted Image only containing the Color Information of the Blue Color Channel

To detect the coordinate of each point one cannot simply look for the single most bright pixel in the image, because in most extracted frames does the laser point cover more than one pixel. Therefore we binarize the image. The points which will be white are determined by a dynamic threshold. The threshold function has the initial threshold of 100% of the intensity and is lowered until at least one point with the corresponding intensity is found. We compute than the centroid of all found and connected pixels, which belong to one point. In case that more than one group of conjoint pixels is found the first centroid is used. If this is the centroid of a bright spot not equal the projected blue laser point, this is not considered problematic as we compensate later for these outliers. Figure 4.4 shows an example of a binarized image.

Figure 4.4: Extracted Image after the Binarization

After extracting all coordinates from the images we have a incomplete list of projected points. Figure 4.5 shows the coordinates extracted from a video. One can see that some points are missing. The points in the middle for example are missing, because the projector occludes them and points on the border are missing, as they are projected outside the reflecting parts of the windshield. As the correspondence between the coordinates from the laser (x,y) and the camera (u,v) is based on the order of the points, we need to have the exact same number of points extracted from the video as the number which were originally projected. Therefore we correct the list of gathered points.

To correct for the missing points and correct for outliers we use spline interpolation. We create a second degree spline for each row of points, based on the gathered points, and recompute the coordinates for all points. Spline interpolation functions

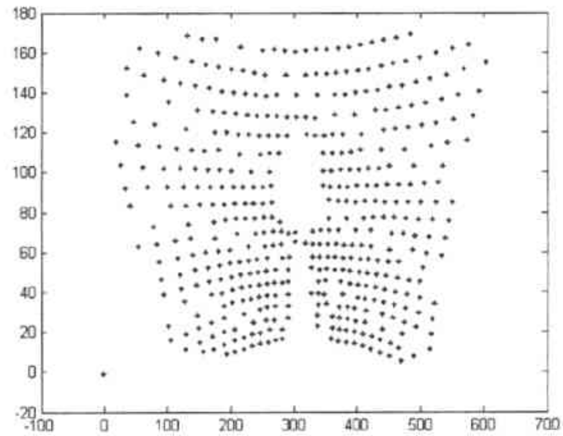



Figure 4.5: Extracted Points from the captured Video

of higher degree do not model the curvature of the windshield very well, therefore we use functions of degree two. Figure 4.6 shows all points after the spline interpolation was applied. The number of points is now equal to the number of projected points, in our case 20x30 = 600 points.

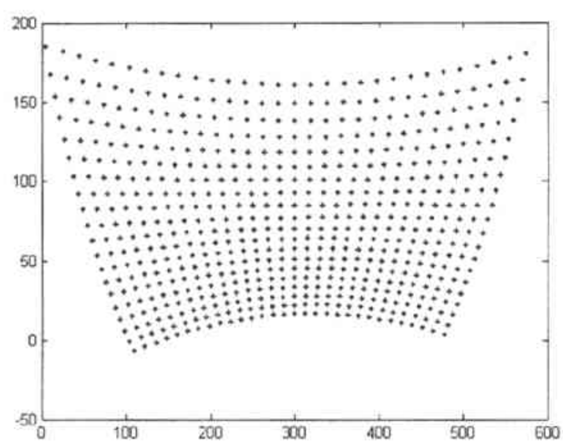



Figure 4.6: Extracted Points after balancing with Splines

This was the last step of extracting a list of useable points as the basis for creating the prewarp function. The next step is now to finally compute the prewarped points.

## 4.3 Prewarp Function Generation

The final step to the distortion correction is to generate the prewarped points. This is the crucial part. The idea is to use the correspondence between the grid points in the laser coordinate system and the captured points in the camera coordinate system to compute prewarped points as basis for the prewarp function.
We create a new rectangular grid of points $(u_i, v_i)$, i $\in$ I = [0,...,n], where we chose n = 30 rows x 50 columns = 1500 points. The coordinates are camera coordinate

system coordinates (u,v). The border points of this shape will be the border of the area where the laser projector will be able to project. The new grid points must be completely in the area of the extracted points. This necessity is due to the input format of the interpolation function, which we use. Figure 4.7 shows the new grid points covered by the extracted grid points.
To get the points in the projector coordinate system (x',y') corresponding to the new grid points we use 2D interpolation functions:

$$f_x(u_i, v_i) = x_i' \forall i \in I$$
$$f_y(u_i, v_i) = y_i' \forall i \in I$$

The coordinates $(x_i', y_i') \forall i \in I$ form the set of prewarped points. We use a 2D linear hyper-surface fitting function for the interpolation to compute the prewarped points.

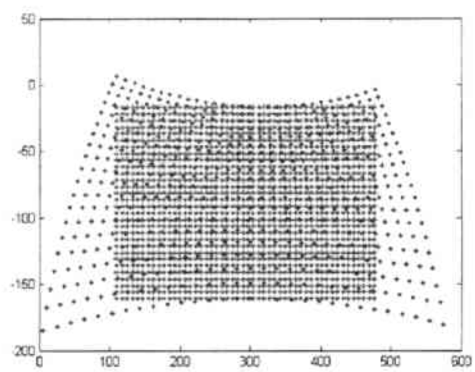


Figure 4.7: New rectangular shaped Grid Points fully covered by the extracted Grid Points

Figure 4.8 shows an example of 1500 prewarped points in a cartesian coordinate system. The viewer in front of the windshield sees an undistorted rectangle when these points are projected by the laser projector.

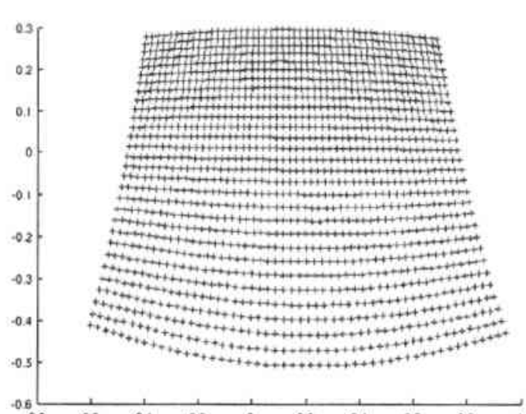


Figure 4.8: Prewarped Points.

## 4.4 Projection of Undistorted Images

The computed prewarped points are the basis for the correction function. This function takes a list of input points and outputs a list of prewarped points, which form the final input for the laser projector. The distortion correction function uses bilinear interpolation. Bi-cubic interpolation would give only minor improvements. The function uses again two 2D interpolation functions, one for the x and one for the y coordinate.

$$\begin{aligned} f_{pw}(x,y) &= (x_{pw}, y_{pw}) \\ \text{where } x_{pw} &= f_{pw,x}(x,y) \\ \text{and } y_{pw} &= f_{pw,y}(x,y) \end{aligned}$$

Figure 4.9 shows one of our reference images. They were used to determine the distortion of the projector. The images consist of a set of image points. The laser projector moves between these points extremely fast that humans cannot see that actually only one laser point is projected. This image contains only blue points, constituted by the blue color, although red points would have been theoretically possible, too, with our prototype.
Figure 4.10 shows the same reference image, but after all points were adjusted by the prewarp function. It is apparent that this image is not just distorted or prewarped, but its size is significantly smaller than the source image. This is due to the fact that the images will be stretched on the windshield. The stretching effect is strongest in the top left and top right corners of the windshield.

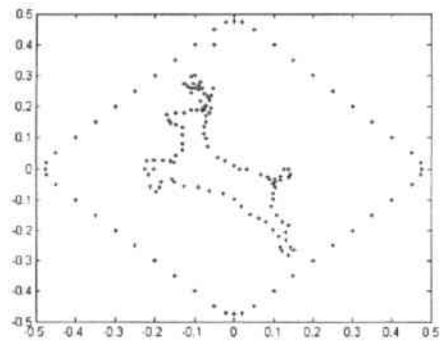


Figure 4.9: Uncorrected Input Image

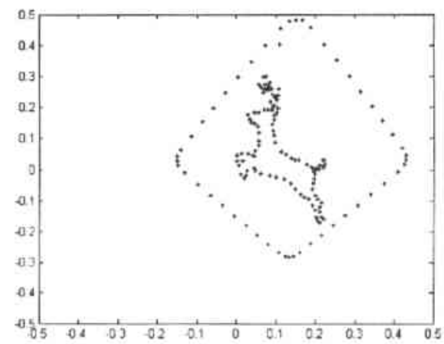


Figure 4.10: Input Image after Prewarping

# 5. Evaluation and Results

After generating the prewarp function we did a qualitative evaluation of our results and tested the setup together with street sign detection. For the qualitative evaluation we did several tests. The basic test was projecting simple images to different positions of the windshield. It was interesting to see if the distortion due to the curved surface was corrected.
For the basic test we projected three different icons to the windshield: a rectangle, an arrow and the reference image of a deer. Figure 5.1 shows two pictures of the projection of the rectangle. The brightness of the blue rectangle is varying, because the input image consisted of only four points. To get a continuously illuminated rectangle one would have to add more points and adjust the illumination for each point. The next tests were performed with two different images of arrows, zoomed

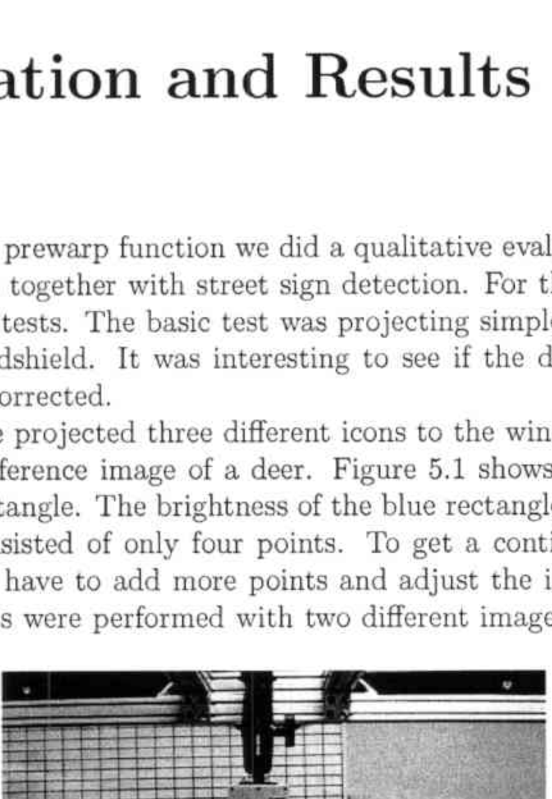


Figure 5.1: Projection of two Squares using the prewarp Function

to different sizes and positions. These tests showed that the prewarp function was successfully correcting for the distortion. The last basic test was projecting the deer reference icon to the windshield. This test showed again that the distortion correction function was working. Though, the result of the test was not satisfying, because on one part of the projected image, where the rectangle is close to the deer icon in the middle, was a connecting line, which was not specified in the input image. This

error was not due to a software error, but due to a problem of the laser projector prototype.
The final test was the highlighting of street signs on the full-windshield display. We fixed a video projector for this setup on top of the windshield frame. This video projector projected videos which where recorded while driving. The videos were recorded for previous work of Wu for the street sign detection algorithm. We used two different videos recorded in the Pittsburgh region. The demo was recorded in a darkened room. This was necessary not because of the weak laser light, but due to the video projector. It only projected a weak image with low contrast of the recorded video to the wall.

This final demo was successful. We showed that highlighting street signs on the

Figure 5.2: Picture of the Final Demonstration where the prewarp Function is combined with Street Sign Highlighting

full-windshield display was feasible. Figure 5.2 shows a picture of the final demo. The room is darkened and a street sign in the video is highlighted. The rectangle which highlights the street sign is not distorted. The demo used precomputed coordinates of the street sign boundaries, but the correction distortion was used in real time for this demo and can also be used for any real time projections of arbitrary images.
We have to note that the street signs were not perfectly marked by the rectangular frames from the laser projector. The reason is that the street sign border detection algorithm we used was still experimental. When the algorithm was applied to the prerecorded videos were the boundaries also not exactly marked.

To conclude this chapter we can say that the process to correct for the curved windshield surface works. All steps of the process to generate the information for the prewarp function can be executed in less than half an hour. The gained prewarp function can be applied to project real-time images to the windshield. The highlighting of street signs is feasible, useful and could be integrated in future navigation systems.

# 6. Summary and Outlook

This work showed how to correct for distortion correction on a full-windshield head-up display. The steps to create the prewarp mapping functions are quick and easy to perform. The first step is to project points of a rectangular grid pattern to the windhshield and record the process with a video camera. The second step was to extract the coordinates of the projected points and use different methods to compensate for problems of the projection and the extraction process. The most important step was to create prewarped points out of the gathered data which form the input for the distortion correction function through interpolation. After executing these steps we gained a prewarp function which is capable of correcting for distortion on the windshield due to the curved surface. The distortion correction algorithm can be used to project arbitrary images to arbitrary positions in real-time to the windshield. We used a camera to avoid the problem of perspective distortion.

After correcting for the distortion we combined the software with a possible future part of navigation systems, namely a street sign detection algorithm. This algorithm was developed by Wen Wu. We created a test setup were we projected different videos of car drives through the area of Pittsburgh and used our software to successfully mark the street signs in the video, while looking through the windshield. The final demo showed that the our software compensated for the distortion and our results gave an idea how highlighting landmarks, like street signs, on a full-windshield display can help the driver in complex driving situations or provide information to the driver without increasing his mental workload. The system can provide information to the driver in a way that does not require dangerous head movements to a display on a dashboard. Thus it lowers the risk for accidents. Additionally allows the full-windshield display the driver to navigate with arrows on the windshield without having to abstract from difficult, abstract images on dashboards displays or irritating voice commands.

Future work could go in different directions. One of the closest problems would be to correct for the perspective distortion. Further research could use head trackers to compensate for the head-movements and to changing angles of view of the driver . A possible optimization would be to track the eyes of the driver.

Further work in this field could analyze the integration of the full-windshield display in cars and the integration of street sign highlighting in navigation systems. The psychological impact of images on the windshield should be analyzed before introducing it to mass production. It would be interesting to see if images on the full-windshield really help the driver or if the images result in irritated drivers.
Different work could be in the integration of real navigation system to display navigation information, like arrows perfectly adjusted to the road, to the driver. As suggested by GMs Research and Development departement could the driver also be assisted in difficult driving situations, for example bad weather. The street boundaries could be highlighted, based on radar or enhanced computer vision techniques. Dangerous situations with other road users could be lowered, too. Pedestrians or bycyclists which might walk or run into the car could be tracked and highlighted on the windshield, such that the driver will notice them and avoid an upcoming accident. Additionally could deer be detected and highlighted before it runs onto the street.

## 6.1 Acknowledgements

This work was funded by General Motors. The visit at CMU was funded by the International Center for Advanced Communication Technologies (interACT). I want to thank my supervisors Jie Yang and Rainer Stiefelhagen for their support of my work at Carnegie Mellon University. Additionally I want to specially thank my advisor Wen Wu who helped and advised me during my stay at CMU. Additionally I want to thank Martin Bäuml for his helpful suggestions.

# Bibliography


[1] Akihiko Sato, Itaru Kitahara, Yoshinari Kameda and Yuichi Ohta: *Visual Navigation System on Windshield Head-Up Display.* 13th World Congress on Intelligent Transportation Systems, 2006.

[2] Buchholz, Kami: *GM Researches Smart Windshield.* Automotive Engineering International Online, 2008.

[3] Jean-Philippe Tardif, Sebastian Roy and Martin Trudeau: *Multi-Projectors for Arbitrary Surfaces without explicit Calibration nor Reconstruction.* Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modelling, 2003.

[4] Kar-Hai Chu, Robert Brewer and Sam Joseph: *Traffic and Navigation Support through an Automobile Heads Up Display (A-HUD).* ICS2008-05-02, Tech Report, University of Hawaii at Manoa, 2008.

[5] Liu, Yung-Ching and Ming-Hui Wen: *Comparison of Head-Up Display (HUD) vs. Head-Down Display (HDD): Driving Performance of Commercial Vehicle Operators in Taiwan.* 2004.

[6] Nan-Ning Zheng, Shuming Tang, Hong Cheng Qing Li Guanpi Lai and Fei-Yue Wang: *Toward Intelligent Driver-Assistance and Safety Warning Systems.* 2004.

[7] Okatani, Takayuki and Koichiro Deguchi: *Autocalibration of a Projector-Screen-Camera System: Theory and Algorithm for Screen-to-Camera Homography Estimation.* Proceedings of the Ninth IEEE International Conference of Comouter Vision, 2003.

[8] Wen Wu, Jie Yang and Jing Zhang: *A Multimedia System for Route Sharing and Video-Based Navigation.* Intl. Conference on Multimedia & Expo (ICME), 2006.

[9] Wen Wu, Xilin Chen and Jie Yang: *Detection of Text on Road Signs from Video.* IEEE Transactions on Intelligent Transportation Systems, 2005.

[10] William H. Press, Saul A. Teukolsky, William T. Vetterling and Brian P. Flannery: *Numerical Recipes.* Cambridge University Press, 2007.