# Parking Assistance Using Homography

Caleb Myers
University of Maryland
College Park, MD
cmyers17@umd.edu

Dillon Miller
University of Maryland
College Park, MD
dmille19@umd.edu

Hamsavaarthan Ravichandar
University of Maryland
College Park, MD
rhamsaa@umd.edu

## Abstract

*Parking assistance technology has been rapidly improving, starting with simple backup cameras and improving to arrays of cameras surround cars that allow operators to gain a 360 degree view of their surroundings. Often these camera arrays are used to create what appears to be a 360 degree, birds-eye view of their car as they park. This project sought to use 4 wide lens GoPro Cameras to stitch together videos of a car parking in order to create our own parking assistance system. After collecting real data around a car in motion, this project was ultimately able to match together shared features in the corresponding images to create a real-time video of a car's surrounding as it moved through a parking lot.*

## 1. Introduction

According to the National Safety Council, one in five accidents happen in a parking lot. To justify that, have you ever noticed that most of the time you found your car abraded, even after miles and miles of safe driving, is when the car was left alone in a parking lot? Besides, the agony of parallel parking without a rear-view camera or senor is inevitable. What if we could better remove human error in the equation? In return to that question, parking assistance, also known as park assist, is an automotive technology aiming to make parking secure and dependable. Parking assist system (PAS) is one of the most crucial advanced parking-assistance system (ADAS) used in most modern vehicles in order to make driving safe and comfortable [1] by reducing the possible blind spots around the vehicle. PAS are the most extensively used ADAS systems that assist drivers during vehicle parking by providing a top-view (360-degree surround-view) image of the vehicle and its environment. In that way, drivers can see the complete environment of the vehicle on the in-vehicle display and can park easier and safer. With the advanced technology and enhanced computing abilities that we have today, the number of ADAS methods has been significantly increasing [2]. One such conventional method is to use four in-vehicle cameras mounted onto the different sides of the vehicle to generate a single top-view image of the vehicle and its environment [3]. This paper is a profound discussion on implementing the parking assistance system using the concept of homography.

## 2. Background

### 2.1. Field of View (FOV)

Park assist cameras and sensors alert the driver of people and objects in the vehicle proximity while in reverse. These cameras, which have been mandatory in the U.S. since 2019, are a key piece of a vehicle's Advanced Driver Assistance System (ADAS). As a crucial component of both PAS and ADAS, a camera's Field of View (FOV) is one of the clinchers that describes the potential functionality of the PAS system. Field of View (FOV) is the maximum observable area that is seen at any given moment through the camera lens. The coverage of the area can be measured using the horizontal and vertical distances to find the diagonal of the area in degrees. Mathematically, the FOV is calculated using the horizontal dimension of the sensor (h) and the Focal Length (F). FOV is the prime factor that determines the number of cameras that can be mounted on a vehicle to create the desired 360-degree surround view required for PAS. Generally, FOV of cameras in ADAS system varies from 90° to 180°. Cameras with a wider FOV can capture more of what's around the vehicle, but they may sacrifice some coverage for better detail [4]. For example, a 120° to 140° FOV can help you see more of your surroundings, while using a 160° to 180° FOV results is a largely distorted image.

### 2.2. Image Distortion

Lens distortion is a deviation from the ideal projection considered in pinhole camera model. It is a form of optical aberration in which straight lines in the scene do not remain straight in an image [5].

Generally, distortions vary according to the lens models and their configurations such as fisheye, wide-angle, and a super-wide-angle lens. These variations in lens modeling

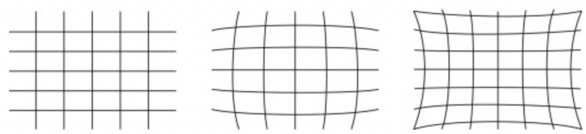

Figure 1. Lens distortion of rectangular grid (left): barrel distortion (center) and pincushion distortion (right) [5].

and specificity make it inflexible for modeling and estimating a common distortion factor for a collective set of lens models [6]. In practice, it is most unlikely even for the two cameras of the same lens model to possess similar distortion parameters, which makes the concept of distortion parameter into a camera-specific distinctive feature [7].

### 2.3. Camera Calibration and Intrinsic Parameters

The camera intrinsic matrix represents the internal parameters of a camera, including the focal length, and it allows to project 3D points in the world onto the 2D image plane, image rectification and undistortion, while the extrinsic matrix is a transformation matrix from the world coordinate system to the camera coordinate system. The parameters of the camera intrinsic matrix are estimated by performing the camera calibration procedure. There is a significant amount of literature on estimating and rectifying the lens distortions in the context of larger FOV lens models[8]. Larger FOV lens models such as fisheye and wide-angle require precise arbitrary calibration target object with known metrics to retrieve the alterations among pixels in lens projections corresponding to the known target object metrics [9]. Usually, this approach can be considered to be manual calibration which is highly prone to procedure-induced errors such as shape irregularities of the calibration targets (imperceptible bending of checkerboard pattern which influences projection of the squared-box edges on to the camera lens). Therefore, precautionary measures should be taken for the calibration scenario to be precise and well-organized to achieve accurate lens calibration [10]. In the context of multiple camera systems used on ADAS platforms, the calibration of an individual sensor unit demands a well-ordered approach and an extra amount of tuning time. With the increase in the demand for exploitation of larger FOV cameras on ADAS platforms [11], industries and companies tend to explore more feasible approaches to calibrate such lens, eluding human intervention and extensive manual labor [12].

## 3. Method

### 3.1. Data Acquisition

The first step when creating a parking assistant system is to collect the data. The parking assistance system created for this project utilizes four cameras, one on each side of the vehicle. The four videos collected will be later modified to be stitched together. In Order to stitch the cameras together, during the data collection phase, the cameras must be calibrated and the frames of the videos must have significant overlap.

#### 3.1.1 Materials Used

The materials used to complete the project are: one 1996 Jeep Cherokee, four GoPro cameras, about 20 ft of PVC pipe, zip ties, random items for feature matching, and 5 identical camera calibration boards. For the best results, the ground features used should be flat, large enough to cover the frame intersection of two cameras, and asymmetrical. This will allow for better feature detection and feature matching.

#### 3.1.2 GoPro Setup

Due to material restraints, the GoPros used were one GoPro Hero7 White, one GoPro Hero7 Black, and Two GoPro Hero5s. The GoPro Hero 7s operated at 1080p, 30 frames per second, and a field of view of around 120 degrees while the GoPro Hero5' operated at 4k and 60 frames per second and a field of view of 141 degrees. The side cameras used


Figure 2. The Data Collection Setup for the Front Camera

were the GoPro Hero 5s as they had a wider field of view. The side cameras were mounted on the side of the car's roof rack using zip ties. All the cameras were oriented straight parallel to the ground and at the same height in order for all the videos to record on the same plane. This will allow for a simpler image stitching pipeline later. As for the front and back cameras, PVC pipe was used to mount the cameras over the hood and back of the vehicle. The front camera used two PVC pipes to mount and stabilize the camera about the hood of the car. All the cameras were placed so that all the ground around the vehicle could be seen and that there was sufficient overlap between the camera's field of view.

#### 3.1.3 Ground Features

Now that the cameras are set up and they have overlapping fields of views there needs to be a way for the cameras to know how to stitch the videos together. The asphalt on the ground does not have enough clear and identifiable features for matching so self made features are needed to be made. The features that were used ranged from camera calibration boards, clothing items, hats, tape, and extra PVC pipe. Around three objects were placed at each of the intersections so the cameras would have sufficient features for image matching.

Figure 3. Examples of Items Placed on the Ground For Features

#### 3.1.4 Camera Calibration Data

The camera's large field of view will result in image distortion in the recordings. In order to remove the distortion in the image the camera's intrinsic matrix must be found. This is done by calibrating the camera using a camera fiducial or camera calibration board. For this project, a 6x4 checkerboard fiducial was recorded by each camera, changing its position and orientation all over the field of view of the camera. Then, 15 images from each camera were taken and used to get the internal parameters of the cameras.

#### 3.1.5 Video Recording

Once everything was set up, all the cameras started recording. The vehicle then drove forward, stopped, then pulled

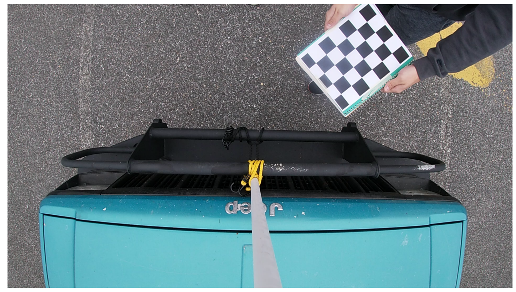

Figure 4. One Calibration frame from the front camera

back into place. After stopping back in place the recordings were stopped. Before the start of the experiment and at the end a sound was played inorder to sync up the videos so they start and stop at the same time. The videos were then downloaded, trimmed to the same length, and the individual frames of the video were extracted for image corrections and image manipulation.

#### 3.1.6 Top Down View of the Car

In the center of the parking assistance video will be a top down view of the vehicle. This was achieved by taking a picture approximately 40ft about the vehicle and then cropping the image to only contain the vehicle. The image was then resized to match the proportions of the vehicle in pixel wise values. The true dimensions of the vehicle are 167"Lx68"W and the image dimensions are 350 x 820 pixels.

Figure 5. The Top Down View of the Test Car

### 3.2. Fisheye Correction

Fisheye distortion of images is a drawback of using wide lenses. Though a wider lens will allow a camera to have a greater field of view and be able to capture more within an environment, it poses problems with calibration and recognition of where the camera frame pixels can actually be related to the physical world.

With our cameras and recorded videos, we see specifically pincushion distortion, meaning that the predicted distance of the center of an image to a pixel is actually much shorter than the actual distance. This is often exaggerated around the edges of images.

Figure 6. The left side of the test vehicle, showing pincushion distortion

As shown in Figure [], there is significant distortion resulting from the wide lens of the GoPro camera, which is most evident from the parking line within the image. In reality, this parking lines is perfectly straight, but in the frame it appears to be curved and grows thinner at the left and rightmost extremities of the image. This distortion can prove to be a problem when manipulating the images, as there is now a variable width of the parking line, and it is not straight.

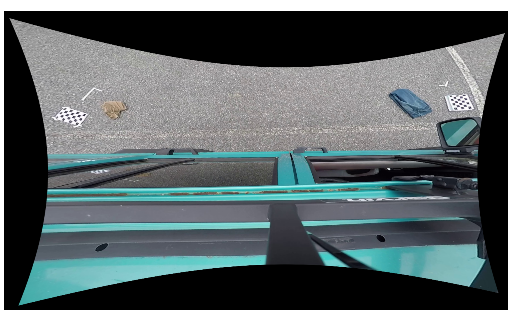

Figure 7. A Rectified Image of the Left Side

## 3.3. Image Manipulation

After the camera's fisheye lens distortion is corrected, the images for each of the cameras will need to be manipulated in order to have easier feature detection, feature matching, and image stitching. Additionally, the current frames still include the black space created when making the undistorted image and the car.

### 3.3.1 Reorient Image

The first step in the image manipulation process is reorienting the image. The cameras that were mounted on the car recorded the video parallel to the side of the vehicle they were attached to. To make feature matching and image stitching easier, the images will be reoriented to match how they would in the end video. The front camera images will remain the same, the left camera images will be rotated 90 degrees counterclockwise, the right camera images will be rotated 90 degrees clockwise, and the back camera images will be rotated 180 degrees.

### 3.3.2 Crop Unwanted Space

After image rotation, the unwanted space will be cropped out of the image. For the left, right, and back images all of the desired ground will be saved while all the remaining space will be removed. For the front image all the black in the image will be cropped out leaving the road and the hood of the car. Then the hood of the car will have a black box placed over it 350 pixels long. The top down image of the vehicle will be placed over the black box.

### 3.3.3 Resize Image

After modifying the images to be understandable, feature matching and feature detection must be used to know where the overlap of the camera frames are. To accomplish this task, a SIFT feature detection and feature matching algorithm will be used to determine how many pixels are needed to cut off the left and right images to have a seamless stitch.

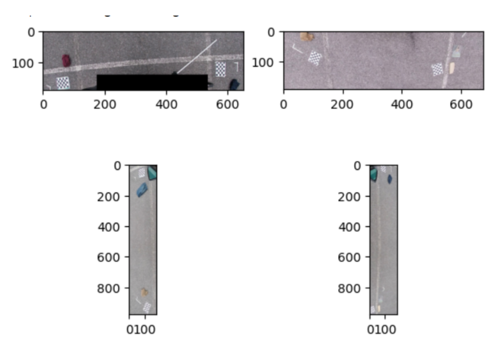


Figure 8. All 4 Camera Perspectives Cropped and Resized

## 3.4. Feature Detection and Image Stitching

After modifying the images to be understandable, feature matching and feature detection must be used to know where the overlap of the camera frames are. To accomplish this task, a SIFT feature detection and feature matcher will be used to determine how many pixels are needed to cut off the left and right images to have a seamless stitch.

#### 3.4.1 SIFT Feature Detection

To detect the features in the images, the feature searching and detection algorithm used was the SIFT algorithm. SIFT or scale-invariant feature transform utilizes various computer vision techniques to detect distinct features. The algorithm uses concepts such as histogram of oriented gradients and gradient thresholds to get good features only.[13] The features detected are not influenced by noise, illumination, image scale, or small viewpoint changes. After all the image manipulation changes the SIFT feature detector is able to effectively detect the features in the images.

#### 3.4.2 Best Feature Matching

All of the features collected from the SIFT algorithm are then saved for each image. Then for each of the four video intersections, the features are compared and matched. For example, all the features on the bottom left of the top image and the features on the top of the left image will be matched up. The feature matching algorithm used was opencv's cv2.BFMatcher(). This is a brute force feature matching algorithm that compares all the features from one image to all the features of another and uses the k-nearest neighbors proximity algorithm to determine what features between the images match. If there is a close enough match between two features the BFMatcher algorithm will pair them up.

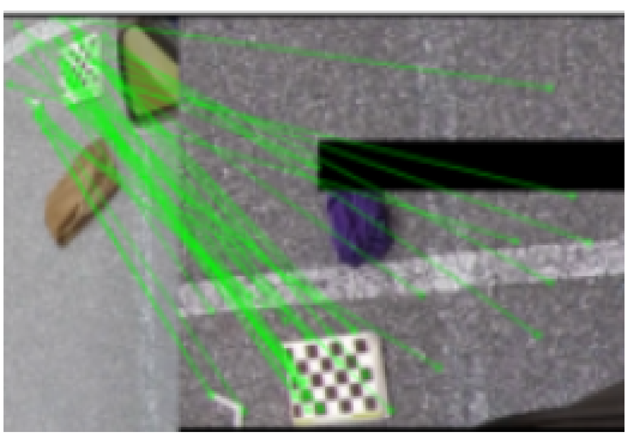

Figure 9. Brute Force Feature Matching for SIFT Features

#### 3.4.3 Image Alignment

The method used to stitch the image together was that the top and bottom images would remain the same after image manipulation while the left and right images would be modified to make a seamless transition between the cameras. How we determined where the side images should be cut off was by finding the matching feature closest to the connection point between images. For example, between the top and left image, the bottom most detected and matched feature from the top image would determine what feature point to look at on the left image. The y-pixel value of the identified feature would be used to crop the image.

#### 3.4.4 Cropped Side Images

After finding the matched feature closest to the intersection of the two cameras, the y-pixel coordinate of the side image would determine where the side image would need to be cut off at. After cropping the image at the desired y location, the side images are resized to meet the pixel size requirements outlined in the image manipulation section for image stitching.

#### 3.4.5 Image Stitching and Video Creation

To stitch the images together, a new blank image with dimensions of 650x1150 pixels will be created. Then the 4 resized car images and the top down view car image will be added to the images in their respective locations. This will create the stitched together image. To make the video, all 1140 frames from the four cameras will undergo the same process to create the final stitched image. The frames will undergo fisheye correction, image reorientation, cropping, resizing, and image stitching. The 1140 final frames will then be saved in a video format at 30 frames per second to show what the cameras would have shown in real time.

### 3.5. Illustrations, graphs, and photographs

## 4. Results

Overall, the project was successful. As seen in the video, the stitching between the right video and the top and bottom frames was complete throughout. The features on the ground can be seen very smoothly transitioning between the three videos. As for the connections on the left frame, they started out very well and overtime got delayed. Towards the end of the video the features almost completely disappeared before reappearing in the following frames. The problem was caused with the different frame rates of the videos and weak feature matching in the image stitching process. The output video showed great success with the project's methodology, approach, and execution.

## 5. Discussion and Conclusion

### 5.1. Challenges

This project faced many challenges throughout its various iterations. One primary challenge that was faced was the issue of feature detection and matching. The captured

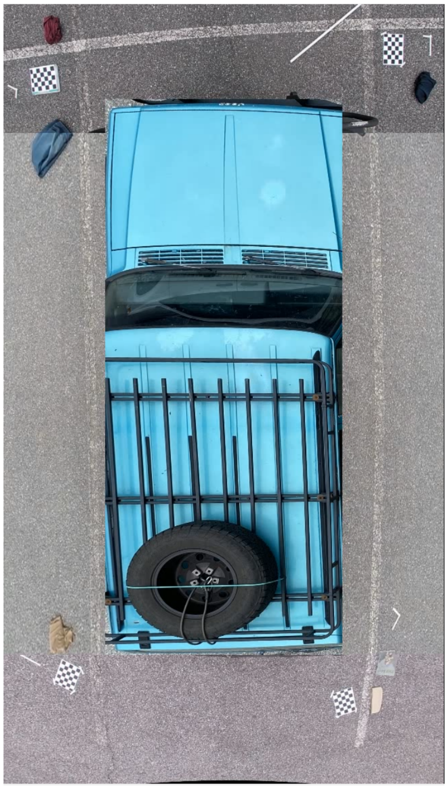

Figure 10. A Frame From the Final Output Video

frames from each image were filled with noise from the pavement, and the items that were placed in the intersecting fields of view did not typically have strong enough features to allow for seamless or accurate image stitching. We were not able to collect more than one data set, and after we began to analyze our data, we found that the angle chosen for thee cameras showed too much of the car, to include the side body of the car and the mirrors. Additionally, this data set lacked solid markers or points with known parallel relationships, making calculating homography incredibly difficult. Lastly, the cameras that were used had smaller FOVs than would be optimal. Ideally a wider lens, like a 180 degree Fisheye lens would give more information about the surroundings, however that resource was not available. Each camera had different frame sizes, frame rates, and settings for exposure and white point, which made it incredibly difficult to have consistent pictures between the four cameras.

## 5.2. Future Work

For continuation of this project, we would like to use Fisheye lenses to expand the field of view, giving the pipeline a greater chance of seeing distinct shared features and creating better combined images of the car's surroundings with more total space visible as well. Additionally, further testing should be applied to the executed pipeline within this project to analyze real-time feasibility with linked camera feeds to quantify the expected frame rate output of the system. Lastly, due to our limited availability of data collection resources, we were only able to collect data once. Upon working, it became apparent that having the cameras tilted at a a higher angle would display more of the surroundings and less of our vehicle.

## 5.3. Conclusion

In this project, we used four cameras, mounted upon a vehicle to create a 360 degree. birds-eye view image of the car and its surrounding. Due to our decision to use image rectification, slicing, cropping, and feature matching, the resulting area around the car that can be seen by the cameras is not large. Regardless, the image processing pipeline that we have created is able to accurately display the nearby spaces in 360 degrees around a car as it moves throughout a parking lot.

## References


[1] Daniel Buljeta, M. Vranješ, Zoran Marceta, J. Kovacevic, "Surround view algorithm for parking assist system", Zooming Innovation in Consumer Technologies Conference (ZINC), May, 2019.

[2] M. Tanneli, R. Toledo-Moreo, L.M. Stanley, "Multi-faceted Driver–Vehicle Systems: Toward More Effective Driving Simulations, Reliable Driver Modeling, and Increased Trust and Safety," IEEE T. Hum-Mach. Syst., Vol. 48, no. 1, Feb. 2018, pp. 1-5.

[3] B. Zhang, V. Appia, I. Pekkucuksen, A. U. Batur, P. Shastry, S. Liu, S. Sivasankaran, K. Chitnis, "A surround view camera solution for embedded systems," IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 2014, pp. 676-681.

[4] JOUR, Kakani Vijay, Kim Hakil, Kumbham Mahendar, Park Donghun, Jin Chengbin, Nguyen Van, "Feasible Self-Calibration of Larger Field-of-View (FOV) Camera Sensors for the Advanced Driver-Assistance System (ADAS) ", July, 2019.

[5] Olgierd Stankiewicz , Gauthier Lafruit, Marek Domański, "Chapter 1 - Multiview video: Acquisition, processing, compression, and virtual view rendering Author links open overlay panel", 2018.

[6] Hughes, C.; Glavin, M.; Jones, E.; Denny, P. Review of Geometric Distortion Compensation in Fish-Eye Cameras. In Proceedings of the IET Irish Signals and Systems Conference (ISSC 2008), Galway, Ireland, 18–19 June 2008; pp. 162–167.

[7] Remondino, F.; Fraser, C. Digital camera calibration methods: considerations and comparisons. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 2006, 36, 266–272.

[8] JOUR, Ricolfe-Viala Carlos, Sánchez-Salmerón Antonio-José, ”Lens Distortion Models Evaluation” , Oct, 2010.

[9] Urban, S.; Leitloff, J.; Hinz, S. Improved wide-angle, fisheye and omnidirectional camera calibration. ISPRS J. Photogramm. Remote Sens. 2015, 108, 72–79.

[10] Luhmann, T.; Fraser, C.; Maas, H.G. Sensor modelling and camera calibration for close-range photogrammetry. ISPRS J. Photogramm. Remote Sens. 2016, 115, 37–46.

[11] Hughes, C.; Glavin, M.; Jones, E.; Denny, P. Wide-angle camera technology for automotive applications: A review. IET Intell. Transp. Syst. 2009, 3, 19–31.

[12] Tai, Y.C.; Hsieh, Y.Y.; Chuang, J.H. A fully automatic approach for fisheye camera calibration. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 November 2019; pp. 1–4.

[13] Singh, A. (2019, October 9). SIFT Algorithm: How to Use SIFT for Image Matching in Python. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/: :text=Frequently