**Part 1. Feature Extraction:**

To break down the task of lane detection, we used a two-step approach. First, we processed the image to find interesting points i.e. pixels that exhibit strong intensity gradients from their surrounding neighbor pixels. These pixels are known as keypoint pixels and are to be used as feature descriptors for the image.

Instead of using first-order gradients to identify keypoints, as is usually done with edge detection algorithms [1], we found the second-order derivatives *Ixx*, Iyy and Ixy in both horizontal and vertical directions of all pixels in the image [2]. Since, derivative computations are sensitive to image noise, we combined the Hessian matrix *H* computation with a Gaussian smoothing filtering step with a tuning parameter of *sigma (σ)* as shown in equation 1.

(1)

The determinant image of the Hessian matrix was computed for each pixel with a 3x3 window non-maximum suppression to preserve the maximum determinant value in that window. A thresholding step was followed to select the final keypoints. Figures 1 through 7 show the interest points detected with varying Gaussian sigma *(σ)* and threshold *(θ)*. Figures 1 through 6 show camera images taken at night while figures 7 and 8 are different images taken during the day.



Figure 1: σ = 0.5, θ = 0.5

Figure 2: σ = 0.5, θ = 0.3

 Figure 3: σ = 0.5, θ = 0.8



Figure 4: σ = 0.3, θ = 0.3

 Figure 5: σ = 0.8, θ = 0.4

 Figure 6: σ = 0.3, θ = 0.4



Figure 7: σ = 0.3, θ = 0.4



Figure 8: σ = 0.5, θ = 0.5

**Part 2. Model Fitting:**

We would then estimate a mathematical model that accurately represents the lanes using the interest points or keypoints that were detected as described in the previous section. To fit the model, we assume the lanes are represented geometrically as straight lines or polynomial curves. We used RANSAC (RANdom SAmple Consensus) [3] to estimate the parameters of the lane model.

Figures 9 and 10 show the RANSAC fitting results with different stopping criterion values.

We did the RANSAC process twice, once to model each lane, by dividing the image longitudinally into two halves and using keypoints lying in a certain half to model the lane for that half. This results in only detecting the two borders of the lane the car is traveling in. Additional lanes could be easily added if needed. The RANSAC algorithm could be easily summarized as follows:

1. Randomly sample *m* points from the image.
2. Fit a straight line or a polynomial curve and find the parameters of the fit.
3. Find the inlier ratio *r* i.e. *the number of points that minimize the error fit (sum of squared errors) divided by the total number of selected points* and *ε* using

(2)

1. Repeat k times until *ε* is less than a predefined criterion which is the probability that no correct sample is found within k repetitions.

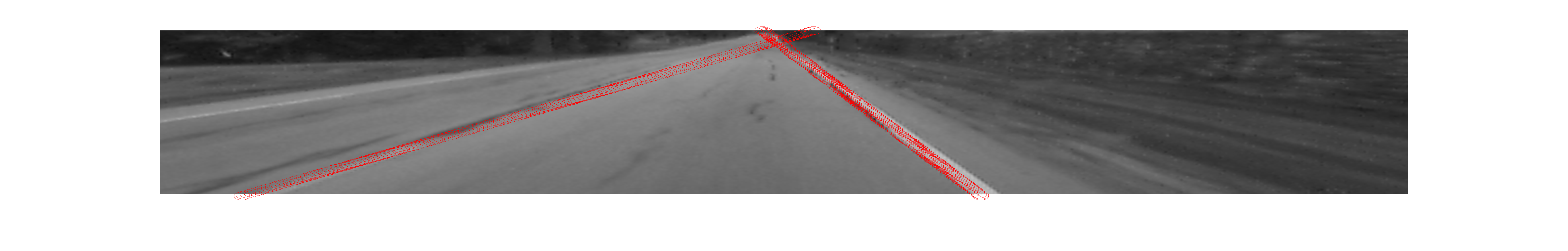


Figure 9: Fitting a straight line using the RANSAC algorithm and keypoint features to find the lane on the road.



Figure 10: Fitting polynmial curves using the RANSAC algorithm and keypoint features to find the lane on the road. The left lane border is represented by a quadratic polynomial curve while the right lane border is represented by a cubic polynomial curve.

**References:**

[1] Gonzalez, R., Woods, R.E., “Digital Image Processing”, 3rd Edition [2007], ISBN-13: 978-0131687288

[2] Beaudet, P., “Rotationally invariant image operators”, Proc.4th International Joint Conference on Pattern Recognition’, pp.579–583 [1978]

[3] Fischler, M., Bolles, R.,”Random sampling consensus: A paradigm for model ﬁtting with application to image analysis and automated cartography”, Communications of the ACM 24,381– 395 [1981]