A modular interpretable system for single camera autonomous vehicle navigation localisation

Michael McDonnell*
UNSW Canberra at ADFA.

An autonomous vehicle navigation capability requires an ability to identify road features such as intersections and plan driving routes through said features. While high end autonomous vehicles operate using a fusion of sensor data, a single camera minimal solution has the advantage of lowering the barrier to entry as well as providing a redundancy option in the event that high end autonomous sensors fail.

This paper outlines a system developed to identify, track and provide driving lines through route features. The system is designed to be interpretable at each stage including interfacing using human understandable inputs and outputs. The system developed uses inverse perspective mapping and histogram backprojection to develop a simple model of the road surface. Route features are identified using a feature mask which is compared to the detected road surface. The feature mask is developed from navigation data nodes and includes a bezier curve interpolated driving line. Once detected the feature is tracked using the Gunnar Farnebäck method to determine mean optical flow and estimate an updated feature position which is confirmed using model masking as per the detection stage.

The system was developed using $512px \times 512px$ images from a customised simulation and ran at over 70Hz while in the feature matching state, maintaining over 10Hz while feature tracking. The system was validated with live video which was tested at a lower resolution of $400px \times 174px$. The road surface detection accuracy was in the order of 75% and optical flow estimation errors less than 5% per frame.

The system is modular with each element being purely defined by input and output format which allows incremental system improvement as improved approaches to individual system functionalities are identified. The outputs of this system can be used to implement a controller to autonomously navigate through a desired route using only road node based mapping data. This system provides not only a 'low barrier to entry' option for autonomous navigation but also provides an effective redundancy for more advanced systems in the event of sensor fusion failure.

*CAPT, School of Engineering and Information Technology, ZEIT4902

I. Introduction

This paper outlines a method for navigation localisation and driving line identification for an autonomous vehicle using a single camera system. This system can be considered as a robust entry level navigation localisation system or, more critically, as redundancy for more complex systems in the event of sensor failure. After a brief review of related works, a high level overview of the system is given. Subsequent sections investigate discrete elements of the system in order, road surface detection, route feature matching and route feature tracking. A general discussion including limitation and opportunities follows before concluding remarks.

In order for an autonomous vehicle to navigate effectively there are some key challenges. The vehicle must have a mechanism to sense the local environment as well as the ability to identify and track transient aspects such as other vehicles and on road obstacles. In addition to the local area, the vehicle must also have the ability to reconcile navigation data with the current location which is the focus of the system described in this paper. A supporting concept to this is that of map matching which calculates vehicle location by using the geographical information from sensors such as GPS position, inertial data and map information from a mapping service (Zhao et al., 2018). This paper outlines the developed 'low tech' system to provide a minimum navigation capability that does not rely on more advanced tools such as LIDAR.

The system is not a controller, rather is designed to be an interpretable system for navigation localisation in environments where other vehicles are not encountered. The design simplifying assumptions involved single direction road surfaces without the requirement to consider other vehicles or traffic control. With appropriate control logic and GPS sensor fusion, the underlying route feature detection and tracking may also be incorporated into vehicles operating in more complex environments if sufficient control is in place to permit feature detection. The use of this system as presented is envisaged to be automation of tasks in remote areas, for example logistics movements through large properties such as farms and mining areas. Alternatively this method provides a single camera redundancy which may assist the capability to 'limp home' in the event of main sensor system failures in more complex autonomous vehicles. While there is no overarching benchmark to validate the system, the accuracy and error rate of core aspects of the system was investigated and the system as a whole was developed using simulation images and validated using live dash camera (BlackVue DR750S-2CH) footage.

II. System overview

Preece et al. (2018) highlight a growing desire for interpretability for machine learned models, both regarding the inner workings of a model and explanations of how a conclusion was reached. Kim and Canny (2018) also discuss the importance of explainable and interpretable systems and propose a system to allow interpretability of deep neural perception and control networks to support it. The solution discussed in this paper[a] is developed to be deliberately an inherently interpretable system; at all key stages in the navigation localisation process a human observer can understand intermediate products and the logic of how they were derived. More importantly, this allows the entire system to be considered as a modular series. As improved systems, approaches and algorithms are developed, assuming they adhere to the same input and output data interface, they can be seamlessly integrated in place of less ideal approaches. The interpretability of intermediate products also means these are available for use in other systems that may require that information. A high level overview of the system is outlined in Fig. 1.

The developed system uses inverse perspective mapping to develop a 'top down' projection from the front facing vehicle camera image. An averaged histogram of local road pixels is used to develop a probabilistic road surface map using histogram backprojection. The system uses route 'features' to localise the navigation goal. Route features used in the development of the system were road intersections[b] however any point on a route with a clear visual pattern for the approach and exit(s) can be used. Route features are developed from mapping data (such as Open Street Map or Google Maps data) using road position nodes. A binary image mask of the upcoming route feature is developed using mapping data and is overlaid on the detected road surface. The feature is deemed detected when the proportion of feature mask covered by road surface is

[a]The developed system discussed in this report was implemented in Python3 with OpenCV using camera inputs from a customised simulation built in Unity3D. The system was subsequently validated using live dash camera footage for frame inputs.

[b]Examples in this paper generally consider an approach to a T junction in a custom built simulation.

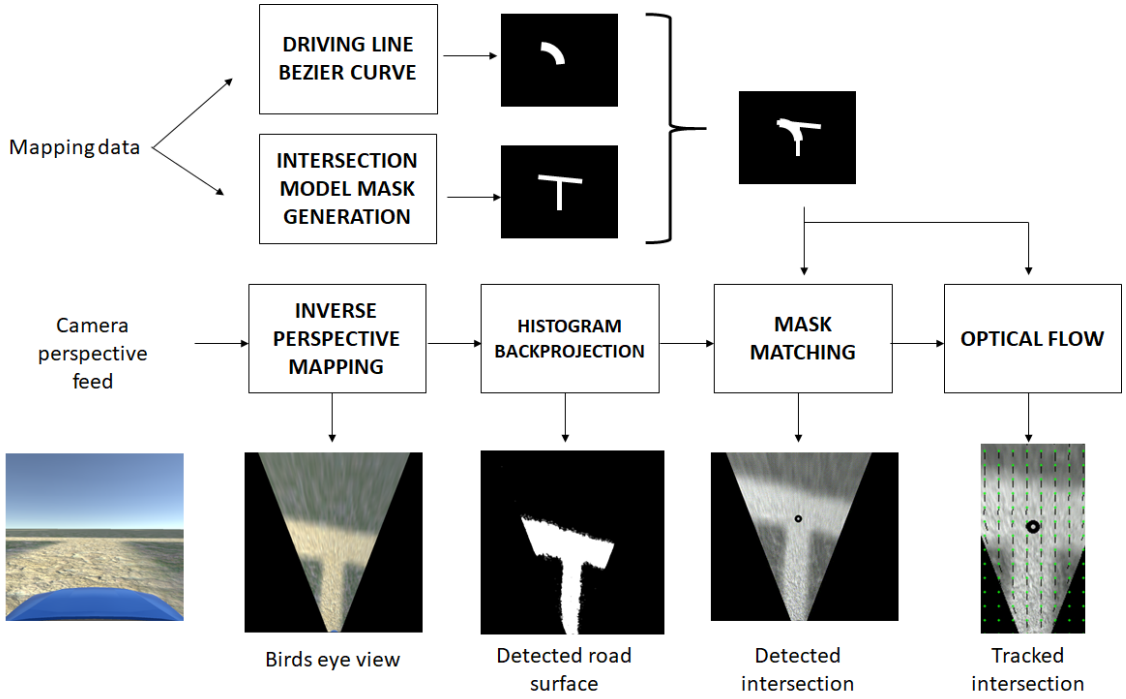

Figure 1. High level overview of system stages, inputs and outputs.

above a threshold. Once initially detected, the feature position is updated in subsequent video frames using a mean optical flow between frames and confirmed by remasking. A driving line through the feature can be developed and tracked by using a Bezier curve of the desired route through the feature.

The discrete elements of the system with input and output as follows:

- **Road surface detection**.
  - **Input:** Camera feed. (24 bit $m \times n$ image, F frames per second)
  - **Output:** 'top down' road surface map. (24 bit $m \times n$ image, F frames per second)
- **Route feature matching**.
  - **Input:** 'top down' road surface map. (8 bit $m \times n$ image, F frames per second)
  - **Input:** Route feature generated from mapping information. (8 bit $m \times n$ image per feature on demand)
  - **Output:** Position in image of target route feature, if detected. (Image coordinate)
- **Route feature tracking**.
  - **Input:** Previous image frame and detected location of target route feature for that frame. (24 bit $m \times n$ image and Image coordinate, F frames per second)
  - **Output:** Updated location of target route feature in subsequent image frame. (Image coordinate)

III. Related works

Current cutting edge self driving vehicles require high fidelity 3D maps to operate effectively which are time consuming to develop and not adaptive to rapid local changes. Despite this, position localisation improvements have been achieved without high fidelity 3D mapping data using a data fusion of GPS and inertial navigation system data (Qingmei Yang and Jianmin Sun, 2007) correlating a detected back lane registry supported with computer vision, GPS and inertial data with map data (Vivacqua et al., 2017) and through use of Kalman filters and LIDAR in more complex environments (Bosse and Zlot, 2008).

Jun Wang et al. (2014) also used inverse perspective mapping with k-means clustering and open uniform B-spline model for lane fitting. Muad et al. (2004) outlined the effectiveness of Inverse Perspective Mapping for lane detection and Assidiq et al. (2008) outlined a robust lane detection approach using Canny edge detection and the Hough transform. Mallot et al. (1991) highlighted the effectiveness of inverse perspective mapping for optical flow computation, a finding that is strongly supported by the optical flow results obtained in this system.

Histogram backprojection has been used for basic (Vergs-llah et al., 2001) and multi model object tracking (Jung-ho Lee et al., 2003), (Tse Min Chen et al., 1999), image indexing (Swain and Ballard, 1992) and region of interest detection (Gossow et al., 2008). It has been used with sensor fusion including thermal mapping for road detection (Bayerl et al., 2015) and to effectively refine spatial fuzzy clustering road detection (Bao et al., 2018).

Mechat et al. (2012) outlined an approach involving the use of a Support Vector Machine to segment image road pixels from non-road pixels and an identification of road edges and control points defining the road curve. This approach is somewhat less interpretable however provides a suitable alternate and possibly more robust option for road detection. Additionally the input and output interfaces are the same as in this system which allows the option to 'sub in' this approach if it is preferred.

Gonzalez and Ozguner (2000) investigated road classification from histogram based segmentation. Other road detection approaches include road surface mapping from near field driving surface models (Dahlkamp et al., 2006) and matching road curvature models to detected lane boundaries (D. Crisman and E. Thorpe, 1991). Supervised convolutional neural network lane detection has had success through fully convolutional (Zang et al., 2018) and instance segmented (Neven et al., 2018) approaches. Spline based representation using random sample consensus for bezier splines based off road edge detection (Aly, 2008) has also shown to be effective.

Todd R. Kushner (1987) outlined a model based recognition approached which matches intersection models to a series of road boundary points and Ort et al. (2018) demonstrated through practical experiments that effective localisation using Open Street Map data and global waypoints is possible using a LIDAR sensor suite for local trajectory generation.

IV. Road Surface Detection

In order to localise navigation elements a reasonable estimate of the current road surface is required. This is achieved via Inverse Perspective Mapping (IPM) and histogram backprojection. These aspects are discussed in detail in the following subsections. The road surface detection implemented is a basic road surface detection algorithm and can be outperformed by other more advanced methods. While it was suitable for developing the proof of concept system, more robust alternatives would be suggested in a live system.

A. Inverse Perspective Mapping

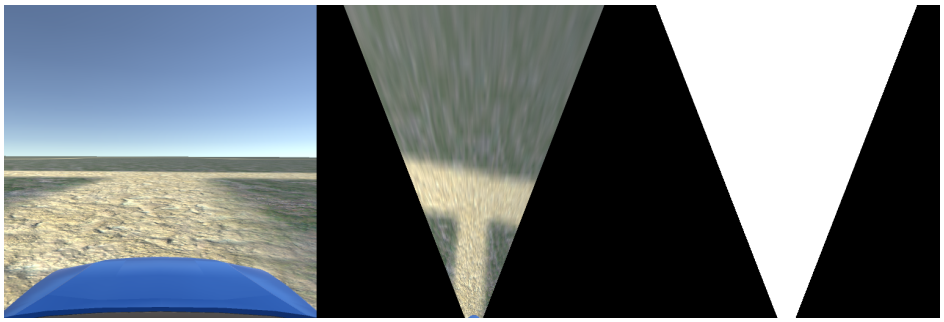

Figure 2. Example image from simulation (left) with IPM applied (centre) and derived IPM mask (right).

IPM is an established technique that involves a matrix transformation operation to remap pixels from a forward facing camera perspective to a 'top down' view (Ballard and Brown, 1982). The specification of original and transformed pixel locations determines the observable distance in the IPM transformed image.

As part of this process the pixel to real world distance ratio of the inverse transformed image is determined based on known distances in the transformed image. An example of the technique applied to an image from the simulation used is included as Fig. 2. IPM assumes the perspective image is from a flat plane (Muad et al., 2004) and while road surfaces are not a flat plane, in most circumstances the local road plane can be considered approximately linear as the scale of large road undulations do not result in significant changes locally to a vehicle. Further the system developed is robust enough to tolerate error introduced by reasonable undulation in the order of ±15%. Bertozzi et al. (1998) have discussed an IPM approach that removes the flat plane assumption; however, it relies on a stereo camera so is out of scope for this system. This approach may prove more resilient and is worth consideration in the event a purely stereo camera based system is required.

The IPM transformed image has a 'null' area in the bottom edges where all pixels are black due to the perspective shift. This represents areas of the original perspective image that are outside the camera lens field of view. An 'IPM mask' is developed which is used in later stages to ensure that any feature matching attempts are not penalised by the zero values in this area. The derived IPM mask is also included in Fig. 2. This is discussed again in Section V. While correction for the camera lens effect should be considered (Lin et al., 2010), the effects were determined to be negligible in testing thus it was determined no correction was required for this system as developed.

B. Probabilistic road surface detection

The road surface detection module uses histogram backprojection to identify probable areas of road surface. Histogram backprojection takes a provided histogram of the target, in this case the road surface, and divides it by an image histogram before convolution with a small mask to gain an estimation for the probability for each pixel in the image belonging to the target histogram (Swain and Ballard, 1992). For this system, initially a road surface area of interest is defined, based off the near portion of road surface from the vehicle front, as indicated in Fig. 3. While this relies on the assumption that the vehicle starts on a road surface, it has the benefit of flexibility in detecting dynamic road surfaces. A histogram of this area is used to inform a rolling average histogram incorporating the preceding five frames. This histogram is backprojected over the image to obtain a road surface probability for each pixel. An example of the detected road surface from an IPM transformed live camera feed is included as Fig. 4.

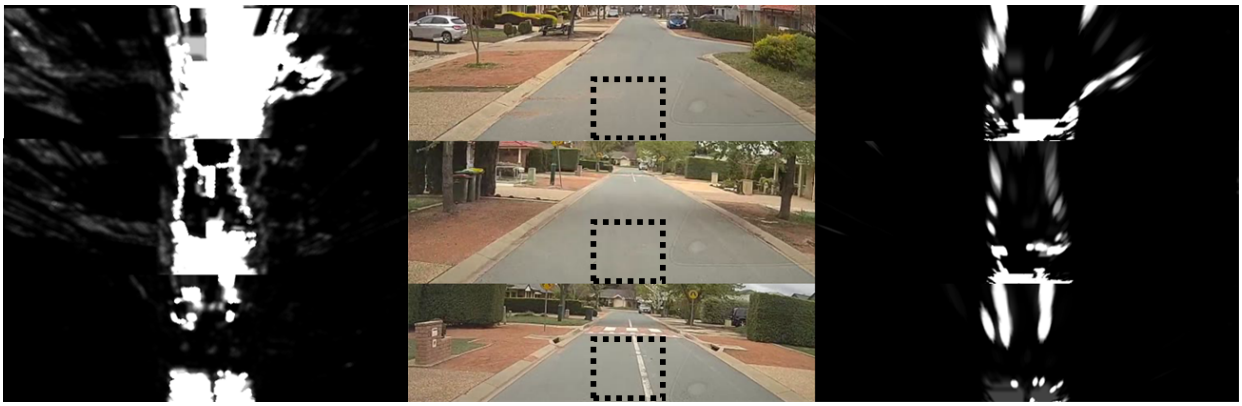

Figure 3. Comparison of histogram backprojection conducted after IPM transformation (left) and prior to IPM transformation (right). Raw image used (centre) includes indicative sampling area of interest used.

While the road detection approach used for this system works with both the IPM transformed image and the raw camera perspective image, in this instance it was applied after IPM transformation. Initial tests indicated that road surface detection accuracy was degraded by as much as 50% when the IPM transformation occurred after road surface detection. Fig. 3 demonstrates the significant degradation in detected surface quality when IPM transformation is applied after the road surface detection. It was also noted that when IPM was applied after road surface detection, probabilities were skewed due to pixel stretching as a result of the IPM transformation. For this reason it was determined that road surface detection should occur after IPM transformation.

Bao et al. (2018) outline a shortfall with this style of road surface detection is that it can be prone to

Figure 4. Detected road surface from inverse perspective mapped live camera.

error when the road surface has little colour contrast to the surrounding environment. The solution proposed was to include thermal sensing, which is outside the scope of the single camera system. If the system is required to operate in a more difficult to segment environment a more robust road detection approach may be needed.

V. Route Feature Matching

The system matches route 'features' based on points defining relevant road segments. While GPS mapping data format can vary, the general common factor is that roads are defined by a series of points (nodes). These points can be used to develop a model of the road and in this case, key features. Features considered involved road intersections such as T junctions and side roads. The 'main feature node' is considered to be the node central to the feature, for example the node in the middle of an intersection. A developed feature mask is used to determine if the detected road pixels match the feature. The following subsections outline the development of the route feature mask, the driving line mask and the subsequent matching process.

A. Feature mask development

In order to develop a feature model, the feature point (node central to the intersection in this case) was placed as a starting point centrally on a blank mask. The feature mask is then developed by linearly connecting adjoining nodes. The pixel distance between nodes is defined based on the real world distance between nodes scaled using a pixel to real world distance ratio that is identified during the IPM process. It is important to note at this stage that the feature mask consists of multiple sub masks; each connection to the feature point is kept as an independent mask [c]. The width of these connections is scaled to the desired projected width for vehicle movement. An overview of the feature mask creation is included as Fig. 5 with the individual sub masks being represented by differing shading in the central sub figures.

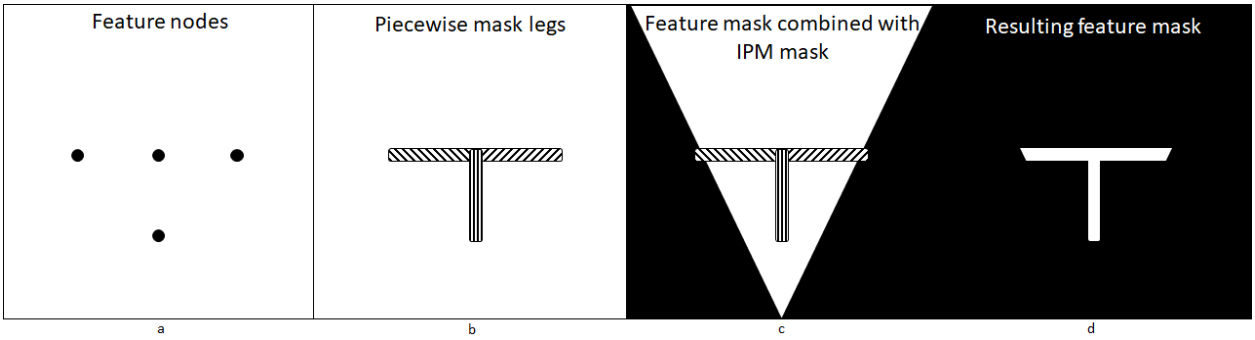

Figure 5. Process of developing feature mask developed feature nodes. A blank mask the same resolution as the detected road surface is created. a) Nodes are placed (feature node centrally with connecting nodes placed at appropriate real world pixel distances) and b) individually connected to main feature node. Shading differentiates between sub masks. c) The initial feature mask is then combined with the IPM mask to d) obtain the final model mask.

The final step in developing the feature mask is to mask it to the non-null areas of the IPM transformed image. This is done using a bitwise AND operation between the developed feature mask and the IPM mask. The purpose of this step is to ensure that portions of mask features such as side roads are not considered

[c]Maintaining individual sub masks results in a more robust feature detection as each sub feature must meet the specified detection threshold.

if they fall outside the inverse projected image space. For this system, the mask development process only considered the feature node and immediate connecting nodes.

B. Driving line curve mask

In order to allow a controller to effectively action system outputs there needs to be consideration given to vehicle turning arcs. This is addressed by incorporating a 'driving line curve' into the mask to ensure the detected feature has room for the vehicle to turn through it. The driving line curve mask is developed using a quadratic bezier curve defined by the approach node, the feature node and the exit node. Once the driving path curve mask is developed it is masked by the IPM mask and added to the route feature masks in order to ensure the developed driving line is also on a detected road.

The quadratic bezier is generated simply by a series of linear interpolations over the range $t = 0...1$. Linearly interpolating t between the approach node to feature node and the feature node to exit node provides two new points, *p0* and *p1*. Linearly interpolating t between *p0* and *p1* provides a point $\mathbf{B}(t)$ along the quadratic bezier curve. The full bezier curve is given by $\mathbf{B}(t)$ for the range $t = 0...1$. A visualisation of this is included as Fig. 6.

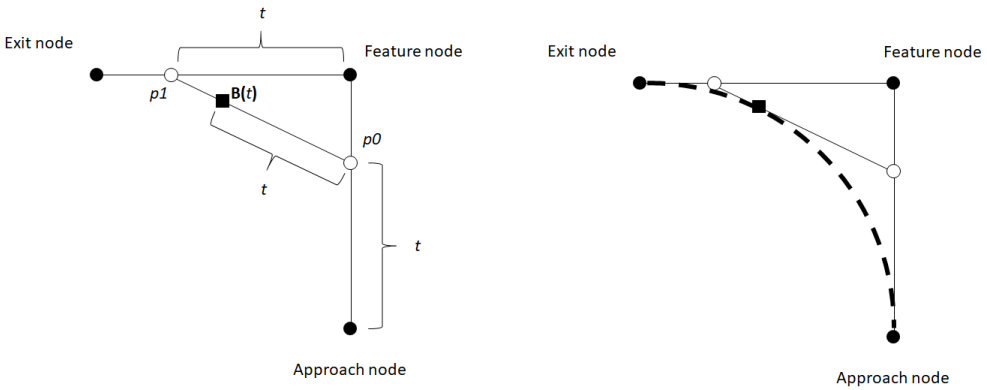

Figure 6. Bezier curve interpolation: Generating $\mathbf{B}(t)$ for $t \approx 0.7$ (left) and the full bezier curve (right)

The development of the driving line must be customised for vehicle model specific parameters such as turning radius. This is implemented by amending the approach and exit node to be a desired distance from the feature node such that a developed bezier curve represents the turning circle capability of the vehicle.

C. Feature mask matching

Features are considered detected based off a probability threshold; that is a feature is considered as a binary 'detected' or 'non-detected' based on a comparison between a defined threshold and the determined probability. Let $P(\mathbf{F})$ be the probability that a feature $\mathbf{F}$ is detected for a given detected road surface $\mathbf{R}$ and $P(\mathbf{F}_{sub,k})$ be the probability that the k'th sub feature $\mathbf{F}_{sub,k}$ is detected. $P(\mathbf{F}_{sub,k})$ is determined by the summation of the Hadamard product (elementwise product) between the feature sub mask and the detected road surface divided by the element sum of the feature sub mask, as outlined in Equation 1. The assessed probability that the route feature is detected is the minimum sub feature probability as per Equation 2. The probability threshold for $P(\mathbf{F})$ is a design decision that can be amended based on factors such as noise and road surface detection output, for example considering a probabilistic or binary thresholded detected road mask.

$$P(\mathbf{F}_{sub,k}) = \frac{\sum_{i,j=1}^{n} (\mathbf{R} \circ \mathbf{F}_{sub,k})_{ij}}{\sum_{i,j=1}^{n} \mathbf{F}_{sub,k,ij}} \tag{1}$$

$$P(\mathbf{F}) = min\{\mathbf{F}_{sub,k} : k = 1, ..., n\} \tag{2}$$

Initial detection approaches involved using a single full feature mask as per Equation 1 in lieu of sub features however this approach is significantly less robust. In the event of a noisy surface detection it is

conceivable that the sum total of all road pixels under the mask may meet a generous threshold even if one element of the feature is not detected at all. Considering all elements of the feature as separate masks and using the minimum detected probability eliminates this and ensures that each feature element has a minimum detection.

Once the feature is detected, the centre point is then the main feature node location from the initial mask. It is possible to further refine this by 'bracketing' about the detected feature point by testing feature points in the vicinity. This may not be required however as assuming the vehicle is central on the road surface the midpoint of the detected feature will be aligned to the midpoint of the route feature and the inclusion of the driving line in the feature mask assures the chosen line is on the detected road surface.

VI. Route Feature Tracking

Mean optical flow was used to estimate an updated feature node location in order to track the detected route features. This was then confirmed using the masking approach discussed in Section V. In general, optical flow options can be considered as sparse where a few key points are tracked, or dense where a large number of points up to individual pixels are tracked. It should be noted this categorisation represents ends of a continuous range rather than a binary category option.

The Gunnar Farnebäck method (Farnebäck, 2003)

The Pyramidal Implementation of the Lucas Kanade Feature Tracker outlined by Bouguet (2000) relies on key tracking points in an image. The Lucas Kanade approach was considered initially as a sparse option due to efficiency however it did not perform well. It is assessed that the random surface textures typical of driving surfaces lack the definition to be tracked effectively via this method. The Two Frame Estimation proposed by Farnebäck (2003) (the 'Gunnar Farnebäck method') is a dense approach which considers a polynomial expansion to approximate pixel neighbourhood and minimises an error function for an approximated local displacement. This approach performed significantly better in the image frames that are typical of this problem. Similar relative performance results between these two methods were identified by de Boer and Kalksma (2015) when considering moving images of near grass surfaces. The Gunnar Farnebäck method was employed to track pixels in a small region of interest on the IPM transformed camera image. The region of interest is comparable to the histogram region of interest for backprojection as discussed in Section IV-B. The smaller region of interest mitigated the use of a dense algorithm while improving mean optical flow calculation accuracy.

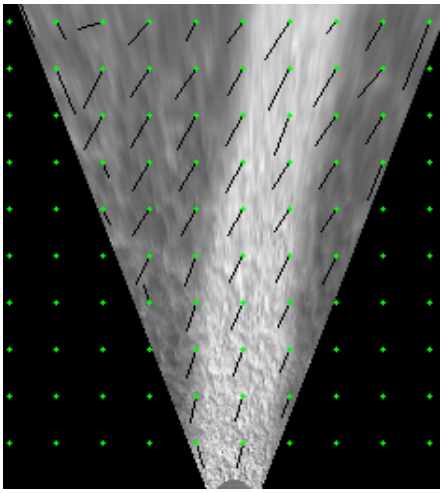

Figure 7. Selected optical flow lines visualised. Black tails indicate flow vector from each point.

The benefits of optical flow tracking in a smaller region of interest of the inverse perspective map are as follows:

- Allows effective processing of dense optical flow due to a small total number of pixels.
- Avoids noisy areas such as extremity pixels which are warped by IPM.
- Optical flow space occurs in the same space as the feature matching allowing a direct application of mean flow to the new estimated position.

A visual example of optical flow output for a small selection of pixels is included as Fig. 7. It can be seen from this image that not all flow vectors are uniform and in particular at the edges of the mask there are large distortions. These distortions are expected due to the discontinuity and do not affect the result as the system only considers the optical flow from a range of pixels to the direct front of the vehicle. The average of these more uniform flow vectors is used to provide an effective estimate of the updated feature location. The mean optical flow approach involving a small region of interest of 100px high by 36px wide performed effectively in cases where the mean flow was in the order of 5px or less, corresponding to an approximate 5% of the window height. Further discussion on optical flow performance is included as Section VII-B-2.

Once the mean optical flow has been developed mean flow vector is used to update the estimated feature position and feature model masks accordingly. Once the model masks have been updated, the IPM mask is

applied and the new approximated feature position is confirmed using route feature matching as discussed in Section V. If the new location of the approaching node places the curve defined by the driving path at the bottom of the image, the vehicle has arrived at the tracked feature and the route feature matching can generate the next route feature. In the event the feature is otherwise no longer detected, a bracketing approach can be used to attempt to relocate it.

VII. Discussion

This system as developed effectively identifies route features and driving lines through features. The system was developed and refined in simulation with verification conducted using live dashcam footage. The system outlined in this paper specifically assumed a single road surface with no other vehicles. Multi lane detection and detection and tracking of other vehicles is outside the scope of the system as developed. While a controller may manage these factors, there is no explicit consideration for non traversable road surfaces (for example a lane with opposite direction of travel) or on road obstacles. Within these specified limitations however, the system as outlined demonstrated an ability to approximate a road surface and very effectively identify and track route features and driving lines where the road surface identification was effective. The system is easy to understand at a high level, as per Fig. 1 and has interpretable inputs and outputs which can be used in other aspects by both human and machine applications. This provides a system suitable for navigation localisation redundancy or alternatively in a low cost entry level setup.

This system as discussed provides an initial capability however there is scope for extension and future work. A discussion on implementation details and opportunities for future work will be discussed following an overview of the system parameters and analysis of system performance.

A. System parameters

The system has a range of parameters to be specified which allow customisation to both vehicle and route specifications. System parameters include:

- **Camera parameters.**
  - **Field of view.** The camera field of view affects the visibility of road surfaces off the direct line of travel at the close edges of the vehicle, especially after IPM is applied. If the field of view is too narrow, route features such as side streets may be lost prematurely. As such, a field of view with reasonable vision to the left and right of the driving surface in the near distance is required.
  - **Frame resolution.** This system remains effective at lower resolutions so the full resolution of the camera frame may not be required. While larger frames contain more detail in general the system can trade some resolution for computational time savings without significant losses. Frame resolutions used in testing varied from 512px to 150px widths.
  - **Frame processing frequency.** While not strictly a camera parameter, the frequency with which the system processes the frames is a key consideration and will generally relate to the processing speed of the implementation. A lower frequency of frame processing results in a higher likelihood of route feature tracking errors and less redundancy in any detection failures. Frame frequency for this test was as low as 10Hz with no noticeable degradation in feature detection or optical flow tracking. This parameter should be considered in context with resolution to ensure that optical flow tracking remains effective.
- **Road surface detection.**
  - **Assumed road surface region.** The histogram backprojection relies on a specified area in front of the vehicle to be sampled for the histogram for backprojection. This region will vary based on the physical setup of the vehicle and camera mount and should be chosen carefully to avoid introduction of additional noise such as road edges. The size of this region is dependant on the camera field of view; a narrow camera field of view will restrict the area that can be reliably sampled as road surface.
  - **Feature detection probability threshold.** This value is considered when determining if a feature has been detected. A value of 1.0 will require each pixel of the feature mask to be over a road surface pixel with a probability of 1.0. The only occasion this is feasible is in the case the

detected road surface is a purely binary thresholded mask. Realistically the detection threshold probability needs to be less than 1.0 to account for noise and uncertainty in the detected road surface.

- **Feature development**
  - **Image pixel to real world distance ratio.** This can be determined from the IPM procedure based on known real world distances as projected after IPM occurs. This is required for the development of route feature masks at an accurate scale.
  - **Route feature mask width.** The pixel width to use when developing the route feature mask. This width does not need to represent a full vehicle width as the driving line mask will ensure the vehicle can pass through the feature. In the development of this system, the route feature mask width was 25% of the vehicle width. If this width is too great there is a risk that the combined driving line and feature mask may be wider than the feature itself, leading to a failure to detect the feature.
  - **Driving line mask width.** Related to above however this needs to have a real world width to comfortably accommodate the vehicle width to ensure driving line through the feature will ensure the vehicle remains on the detected road surface.

These parameters directly impact the effectiveness of the system however are generally easy to tune as the entire system is human interpretable thus the effect of a parameter can be directly observed and understood.

B. Performance

It is somewhat difficult to effectively quantify the performance of the system as a whole as there is no relevant system benchmark and the 'detection' module is deterministic in the sense that if the road surface is identified sufficiently, the feature mask matching process will identify the feature. Despite these factors it is important to quantify the performance of the system. The two elements of the system which rely on quantifiable accuracy are the road surface detection and the feature tracking. These elements are discussed in detail in following subsections.

In general the system has a computational complexity of $\mathbf{O(mn)}$ where the image frames are of dimension $m \times n$. The complexity directly scales based on input image dimensions which can be an initial simple parameter to consider for performance. The system was developed using simulated $512px \times 512px$ images and was validated with live video which was tested at a lower resolution of $400px \times 174px$.

It should be noted that many of the module operations as outlined are trivially parallisable. Specific implementations will require individual optimisation to identify the optimal operating parameters based on hardware available and road surface complexity. The test computer used had a 2013 model i5-3570k CPU. With the base implementation code implemented in Python, OpenCV and Numpy (with no optimisations), the system took on average[d] 13.88ms to process a single frame ($512px \times 512px$) in the Feature Matching state and processed individual frames in an average of 72.06ms when tracking features with a region of interest of ($260px \times 55px$) using a three level Gunnar Farnebäck detection.

1. IPM and Histogram Backprojection

While the IPM and Road Surface Detection modules have been discussed individually, it was identified that the accuracy of the detected road surface was dependant on the IPM implementation. When defining the IPM transformation initially, the 'far' distance of the IPM transformed image is defined by the choice of pixel locations to transform (and their transformed locations). Tests were done using live image data with IPM transformation maximum projection distances consisting of far (100m), mid (30m) and near (15m). Indicative examples of the results of these distances for straight and curved roads is included as Fig. 8.

An analysis of the road surface detection accuracy demonstrates that the near IPM range performs more consistently and, importantly, suffers from less 'false positives' where road surface is detected outside the true surface. Table 1 outlines the accuracy values in these cases where it can be noted that the near maximum distances detects in the order of 75% of the road surface consistently.

Road surface detection was also applied prior to IPM transformation as part of the validation process. Testing over varied road stretches identified that the quality of detected road surface degraded by as much as

[d]Multiple tests were benchmarked for time during final testing. Exact frame count was not obtained but consisted of data from at least 15000 frames of feature detection and 5000 frames of feature tracking.

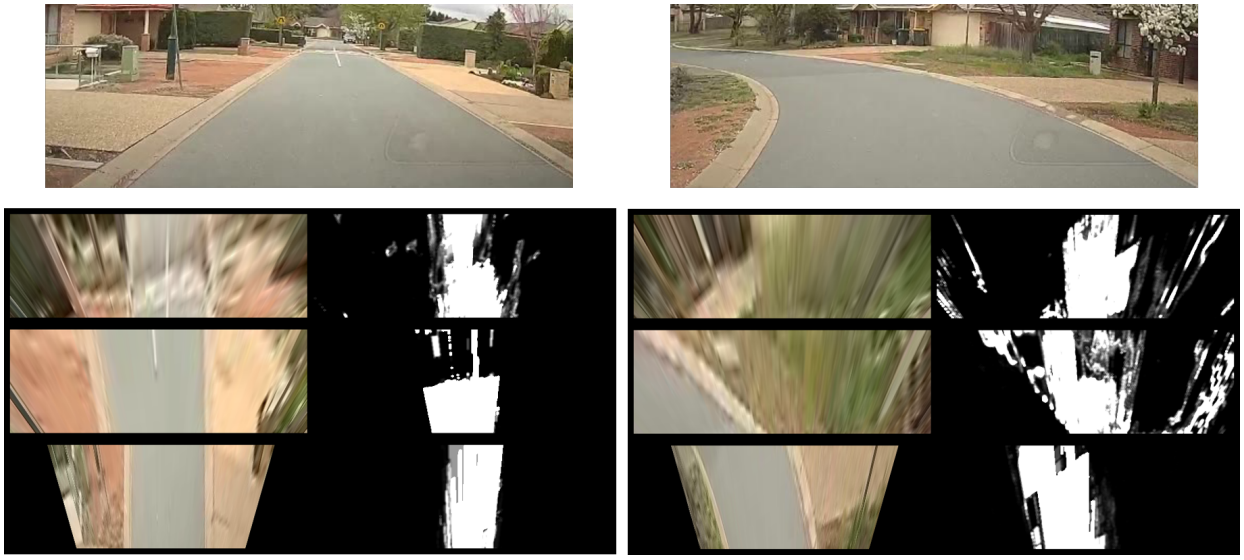

Figure 8. Demonstration of Histogram backprojection at far (top), mid and near (bottom) ranges for straight (left) and curved (right) road surfaces.

Road type	IPM range	Correct road surface ratio	Falsely detected road surface ratio
Straight	Far	67.19%	1.15%
	Mid	43.76%	0.39%
	Near	74.87%	0.49%
Curved	Far	0%	20.56%
	Mid	0%	30.80%
	Near	75.26%	0.82%

Table 1. Accuracy of detected road surface for varying IPM maximum ranges for straight and curved roads

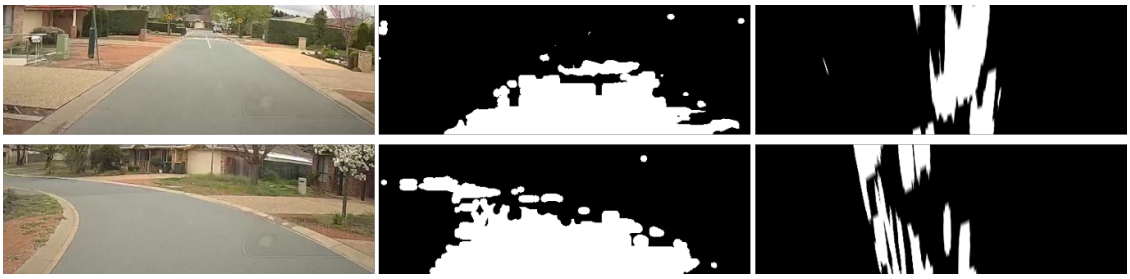

Figure 9. Road surface detection performance when applied prior to IPM transformation for roads in Fig. 8.

50% if detection was applied prior to IPM transformation. Fig. 9 demonstrates the result of applying road surface detection to the perspective image and applying the IPM transformation to the detected road surface. This detection example corresponds to the 'near' range detection in Fig. 8 and clearly demonstrates additional distortion to the IPM transformed road surface and the benefits of applying the IPM transformation prior to road surface detection.

2. Optical Flow reliability

The optical flow reliability is the second core element of the system that can be quantitatively investigated. The performance of this module was determined to be a combined function of image resolution and processing frequency. Optical flow error or 'slippage' was defined as the apparent slipping of the detected feature; this was due to the computed mean optical flow being less than the true optical flow which resulted in the feature location not being moved the full delta each frame. It was determined that the resolution of the 'area of interest' used for optical flow calculation was an effective comparison point and, as highlighted in Fig. 10, the 'slippage' has a generally linear relationship with increasing relative pixel flow. Increasing the resolution of the optical flow area of interest or increasing the processing frequency will result in a smaller relative pixel flow per frame which in turn will reduce the feature tracking error.

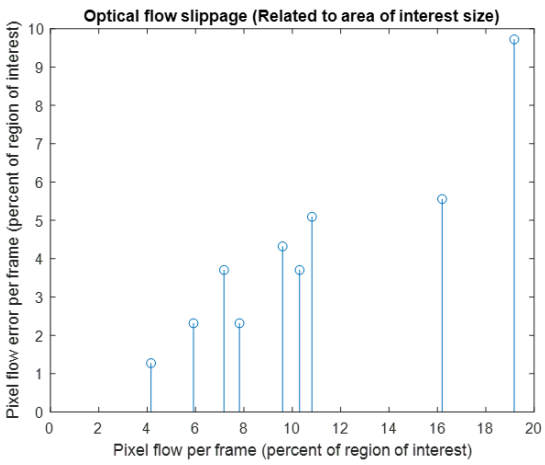

Figure 10. Optical flow estimation error results.

C. Opportunities for future work

1. Estimation of road surface in IPM voids

In the current implementation, the 'void' in the IPM transformed image (where no information is contained) is not considered. There is scope for road surface estimation in this area based off preceding frames. As detected road surface moves outside the IPM transformed image, there is scope for future work to interpolate the estimated position of road surface from prior 'in frame' detection.

2. Feature bracketing in lieu of optical flow estimation

It was determined the optical flow estimation added in the order of 60ms processing time per frame. This allowed effective feature tracking in testing however a heuristic for an intelligent 'bracketing' update procedure may result in computational savings as the 60ms addition from optical flow represents 5-6 feature mask operations.

While this may be an avenue of interest, by itself it does result in a non-smooth positional tracking as the feature location will 'jump around' based on bracketing steps. This may be addressed further through Kalman filtering in a more advanced implementation.

3. Integration with Fuzzy Logic.

Stelzer et al. (2007) present results demonstrating a very effective combination fuzzy control system for a sailing boat including the ability to tack and jibe. Subramanian et al. (2009) outlined significant improvements in using fuzzy logic for sensor fusion and control of autonomous vehicles operating in citrus grove alleys. The system discussed in this paper involves probability based decisions throughout the process and may be a good candidate for integration with Fuzzy logic decision making and/or control.

4. Known road map histograms.

The histogram backprojection uses an average histogram over the previous 5 frames. This is effective for slowly changing road surfaces however can lead to brief road surface loss when moving from one surface to another. An alternate option is to have a bank of representative road surfaces, including areas under specific lighting conditions, to use for the backprojection and consider the maximum probability over the range of surfaces as each pixel's probability of being a road surface. This can also be combined with the rolling average to provide a more robust assessment as well as categorise road surfaces.

5. Optimisation of system parameters

There are a large range of system parameters that need to be manually tuned for implementation including selection of IPM transformation points as outlined in Section 5, regions of interest for histogram backprojection and optical flow and processing frequency and resolution. While some of these parameters are difficult to automate, parameters such as image frame resolution and optical flow window bounds may be suitable for multi objective optimisation automation, especially when considering the tradeoff between resolution accuracy and individual frame computational processing time.

D. Implementation Discussion

1. Road surface centreline detection

While vehicle position and orientation on the road is a critical input to a controller, it was not explicitly covered as part of this system. A general approach slightly modified from that outlined by Assidiq et al. (2008) to suit this system is to identify the central point on a road surface is to apply edge detection to the detected road surface pixels followed by identifying straight line candidates using the Hough transform. The basic approach would be as follows:

- Identify local window of interest in the near ground to vehicle.
- Use edge detection such as the Canny algorithm to identify road edge pixels inside window of interest.
- Identify candidate lines using Hough Transform.
- Select strongest candidate lines for left and right portions of the window, corresponding to the estimated left and right road edge lines.
- Identify road centreline as midline between left and right detected edge lines.
- Determine desired steering direction based on detected centreline pixel offset from vehicle centreline [e]

This approach can be applied to either the camera perspective image or the IPM transformed image. In the former case the perspective effect where the left road edge line will be angled to the right and vice versa for right edges, tending to the vanishing point, allows easy segregation of left and right road surface lines. By contrast the IPM transformed road edges will be parallel thus rely on positional information only to separate the left and right road surface lines. Alternatively more advanced methods such as the Support Vector Machine approach suggested by Mechat et al. (2012) may be more robust and may additionally inform the road surface detection.

[e]The vehicle centreline will correspond to the image centre assuming the camera is position centrally. If the vehicle mounted camera is offset, this 'centre pixel line' will need to be manually identified and stored prior to operation.

2. Extension of feature masks.

As designed, the current system only considers the main feature node and immediate connections. As the feature node is placed centrally, the approach node distance results in only masking a small portion of the full image. An improvement is to extend the sub masks from the feature node to the full extent of the mask image size. This would involve each sub mask drawing additional relevant node connections until the relative position takes the line off the edge of the image. The resulting mask will be a full representation of the approach route to the feature.

3. Rotation of feature masks.

The current implementation has the system generating feature masks of the route feature as it will be approached and assumes a linear approach. This is a reasonable assumption for close in detection however it may fall down if complex approaches to route features are encountered. A more robust solution would involve developing the extended feature mask as outlined above and aligning the mask to the approach direction at the frame the feature will be detected. This would entail developing the mask and once developed, the mask can then be rotated so the approach line is vertical at the base of the image. On arrival at a feature node on the approach, the mask will be regenerated and reoriented. This approach takes advantage of the fact that the route between two nodes can be assumed as linear and will ensure the feature mask orientation will match the detected road more closely.

An example of a complex approach to a feature is included as Fig. 11. In this instance as the vehicle encounters the first bend, the feature model mask will no longer have the correct orientation and result in a loss of feature tracking.

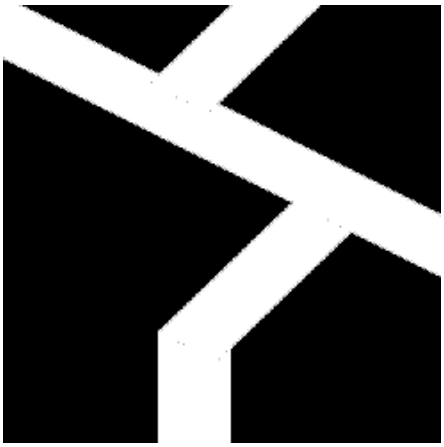

Figure 11. An example of a complex approach to a feature which may undermine the current system feature detection.

4. Navigation extension options.

While it was not explicitly considered as a feature, any road feature with a sharp enough angle between points can be used as a feature. This is particularly helpful if an aim is to identify a 'true' position outside GPS error as bends in a road can be used as additional features without requiring additional intersecting roads. Additionally while this system was designed with a supporting GPS in mind to estimate anticipated feature arrival, as developed it does not rely on positional information specifically. Indeed assuming that relevant route feature data is stored, this system can work 'offline' using the features as directions in a similar way a human navigator may. Integrating the output of this system with other sensors such as GPS offers additional localisation benefits. Thrun et al. (2005) discuss Markov and extended Kalman filter localisation techniques that may be applicable to this system and have the potential to result in improved positional estimation.

5. IPM maximum distance

Initial implementations of the system used IPM transformations which aimed to maximise the visible 'top down' road surface by choosing points close to the horizon in the perspective image to transform which corresponded to distances of approximately 100m. While this approach is effective where there are long stretches of visible road it has significant failure points. If the relative distances are maintained it results in a narrowing of the road surface in the top down view. Additionally it generates significantly more interpolated data ('stretched pixels') and results in a small portion of detected road surface where the road is not straight which is particularly evident when approaching a corner.

Distances in the order of 15-20m were determined to be the most effective for road surface detection in this implementation. This results in less interpolation of data and a reasonable road surface coverage even when approaching corners. The downside to this approach is it is only 'looking ahead' a shorter range so any controller that is implemented will likely need to anticipate approaching features to manage velocity changes.

VIII. Concluding remarks

The system as developed has a road surface detection accuracy of in the order of 75% and identified all intersections in testing. The optical flow approach used in feature tracking provided an effective method of estimating the feature location in subsequent frames. As implemented the optical flow error per frame was less than 5% which was small enough that detected intersections were not 'lost' prior to arrival at the start of the driving line curve. The bezier curve generated driving line demonstrated an effective approach to vehicle route planning through detected features. The resulting system has specified limitations as discussed in Section I and in particular the road surface detection approach represents a basic implementation. Alternate road surface detection implementations may provide a more robust or higher accuracy detection option.

In addition to the societal considerations for interpretable AI, the focus on interpretable steps has the significant benefit of having a clear 'plug and play' interface with other systems. As a result, when an improved road surface detection algorithm is identified it can easily be integrated to the system in place of the existing element. Redundancy in autonomous systems is a critical consideration and should be included as part of any system design. A system such as the one proposed in this paper provides a low barrier to entry for simple vehicular navigation automation tasks and offers a final redundancy option in the event that a 'high end' autonomous system suffers a significant sensor suite failure.

References

M. Aly. Real time detection of lane markers in urban streets. In *2008 IEEE Intelligent Vehicles Symposium*, pages 7–12, June 2008. doi: 10.1109/IVS.2008.4621152.

A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan. Real time lane detection for autonomous vehicles. In *2008 International Conference on Computer and Communication Engineering*, pages 82–88, May 2008. doi: 10.1109/ICCCE.2008.4580573.

D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall, 1982. ISBN 0131653164.

J. Bao, Y. Zhang, X. Su, and R. Zheng. Unpaved road detection based on spatial fuzzy clustering algorithm. *EURASIP Journal on Image and Video Processing*, 2018(1):26, Apr 2018. ISSN 1687-5281. doi: 10.1186/s13640-018-0260-3. URL `https://doi.org/10.1186/s13640-018-0260-3`.

F. Bayerl, T. Luettel, and H.-J. Wuensche. Following dirt roads at night-time: Sensors and features for lane recognition and tracking. 09 2015.

M. Bertozzi, A. Broggi, and A. Fascioli. An extension to the inverse perspective mapping to handle non-flat roads. In *IEEE International Conference on Intelligent Vehicles. Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles*, volume 1, 1998.

M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based slam. *I. J. Robotic Res.*, 27:667–691, 06 2008. doi: 10.1177/0278364908091366.

J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. *OpenCV Document, Intel, Microprocessor Research Labs*, 1, 01 2000.

J. D. Crisman and C. E. Thorpe. Unscarf–a color vision system for the detection of unstructured roads. volume 3, pages 2496 – 2501 vol.3, 05 1991. doi: 10.1109/ROBOT.1991.132000.

H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. R. Bradski. Self-supervised monocular road detection in desert terrain. In *Robotics: Science and Systems*, 2006.

J. de Boer and M. Kalksma. Choosing between optical flow algorithms for uav position change measurement. 2015.

G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In J. Bigun and T. Gustavsson, editors, *Image Analysis*, pages 363–370, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-45103-7.

J. P. Gonzalez and U. Ozguner. Lane detection using histogram-based segmentation and decision trees. In *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493)*, pages 346–351, Oct 2000. doi: 10.1109/ITSC.2000.881084.

D. Gossow, J. Pellenz, and D. Paulus. Danger sign detection using color histograms and surf matching. In *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 13–18, Oct 2008. doi: 10.1109/SSRR.2008.4745870.

Jun Wang, Tao Mei, Bin Kong, and Hu Wei. An approach of lane detection based on inverse perspective mapping. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 35–38, Oct 2014. doi: 10.1109/ITSC.2014.6957662.

Jung-ho Lee, Woong-hee Lee, and Dong-seok Jeong. Object tracking method using back-projection of multiple color histogram models. In *2003 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages II–II, May 2003. doi: 10.1109/ISCAS.2003.1206062.

J. Kim and J. Canny. *Explainable Deep Driving by Visualizing Causal Attention*, pages 173–193. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98131-4. doi: 10.1007/978-3-319-98131-4_8.

C.-T. Lin, T.-K. Shen, and Y.-W. Shou. Construction of fisheye lens inverse perspective mapping model and its applications of obstacle detection. *EURASIP J. Adv. Signal Process*, 2010:8:1–8:23, Feb. 2010. ISSN 1110-8657. doi: 10.1155/2010/296598. URL `http://dx.doi.org/10.1155/2010/296598`.

H. Mallot, H. Blthoff, J. Little, and S. Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64:177–85, 02 1991. doi: 10.1007/BF00201978.

N. Mechat, N. Saadia, N. K. M'Sirdi, and N. Djelal. Lane detection and tracking by monocular vision system in road vehicle. In *2012 5th International Congress on Image and Signal Processing*, pages 1276–1282, Oct 2012. doi: 10.1109/CISP.2012.6469683.

A. M. Muad, A. Hussain, S. A. Samad, M. M. Mustaffa, and B. Y. Majlis. Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system. In *2004 IEEE Region 10 Conference TENCON 2004.*, volume A, pages 207–210 Vol. 1, Nov 2004. doi: 10.1109/TENCON.2004.1414393.

D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. Towards end-to-end lane detection: an instance segmentation approach. pages 286–291, 06 2018. doi: 10.1109/IVS.2018.8500547.

T. Ort, L. Paull, and D. Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. pages 2040–2047, 05 2018. doi: 10.1109/ICRA.2018.8460519.

A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty. Stakeholders in explainable ai. *arXiv.org*, 2018. URL `http://search.proquest.com/docview/2115560388/`.

Qingmei Yang and Jianmin Sun. A location method for autonomous vehicle based on integrated gps/ins. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, pages 1–4, Dec 2007. doi: 10.1109/ICVES.2007.4456376.

R. Stelzer, T. Proll, and R. I. John. Fuzzy logic control system for autonomous sailboats. In *2007 IEEE International Fuzzy Systems Conference*, pages 1–6, July 2007. doi: 10.1109/FUZZY.2007.4295347.

V. Subramanian, T. Burks, and W. Dixon. Sensor fusion using fuzzy logic enhanced kalman filter for autonomous vehicle guidance in citrus groves. *Transactions of the ASABE*, 52(5):1411–1422, 2009. ISSN 21510032.

M. J. Swain and D. H. Ballard. Indexing via color histograms. In A. K. Sood and H. Wechsler, editors, *Active Perception and Robot Vision*, pages 261–273, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. ISBN 978-3-642-77225-2.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005. ISBN 0262201623. URL `https://www.ebook.de/de/product/3701211/sebastian_thrun_wolfram_burgard_dieter_fox_probabilistic_robotics.html`.

S. P. Todd R. Kushner. Progress in road intersection detection for autonomous vehicle navigation, 1987. URL https://doi.org/10.1117/12.968232.

Tse Min Chen, R. C. Luo, and Tsu Hung Hsiao. Visual tracking using adaptive color histogram model. In *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.99CH37029)*, volume 3, pages 1336–1341 vol.3, Nov 1999. doi: 10.1109/IECON.1999.819405.

J. Vergs-llah, J. Ar, and A. Sanfeliu. Object tracking system using colour histograms. In *9th Spanish Sym. Pattern Recog. Image Anal*, pages 225–230, 2001.

R. P. D. Vivacqua, R. F. Vassallo, and F. N. Martins. A low cost sensors approach for accurate vehicle localization and autonomous driving application. In *Sensors*, 2017.

J. Zang, W. Zhou, G. Zhang, and Z. Duan. Traffic lane detection using fully convolutional neural network. pages 305–311, 11 2018. doi: 10.23919/APSIPA.2018.8659684.

J. Zhao, B. Liang, and Q. Chen. The key technology toward the self-driving car. *International Journal of Intelligent Unmanned Systems*, 6(1):2–20, 2018. doi: 10.1108/IJIUS-08-2017-0008. URL https://doi.org/10.1108/IJIUS-08-2017-0008.