OPTIMIZED DRIVABLE PATH DETECTION SYSTEM FOR AUTONOMOUS VEHICLE IN RAIN CONDITION

Olusanya Yinka Agunbiade* Seleman Madison Ngwira and Tranos Zuva
Department of Computer System Engineering, Tshwane University of Technology
Pretoria/South Africa
agunbiadeoulosanya@yahoo.com or agunbiadeOY@tut.ac.za

ABSTRACT
Drivable path detection is a vital researching issue that is increasingly receiving attention from scholars because autonomous vehicle can successfully navigate, if proper drivable paths are detected. Several vision-based techniques have been proposed for drivable path detection and amazing results have been achieved by many researchers. Rain characterized as environmental noise is capable of causing autonomous vehicle accidents because of inaccurate detection of drivable path. However, after investigating the effect of rain, we address the issue of drivable path detection during rain by introducing a filtering algorithm into the drivable path detection system. The filtering algorithm used to optimize the system is capable of eliminating the effect of rain. The optimized system experimental comparison was done qualitatively and quantitatively using the following evaluation scheme: Precision (PRE), False Positive Rate (FPR), Accuracy Rate (ACC), Error Rate (ERR), Total Positive Rate (TPR), False Negative Rate (FNR), and Total Negative Rate (TNR). Results prove the system improvement in drivable path detection during rain scenario and this progress has further enhanced detection and autonomous vehicle navigation.

KEY WORDS

1. Introduction
Drivable path detection is a key problem for researchers and a major necessity for perfect navigation of an intelligent vehicle [1]. Autonomous vehicle system categorized as an intelligent vehicle can navigate and drive on its own without been controlled by any one. For successful navigation, autonomous vehicle system should be able to detect drivable path and non-drivable [2]. Subsequently, previous researchers for many decades have attempted solving this problem with different approach and good result has been achieved towards drivable path detection. Some researcher propose to address the issue of drivable path detection using laser scanner based on low elevation variances of the surface for the extraction of plain terrain which represent drivable part and high elevation area which represent non-drivable path [3, 4]. Certain researchers attempt detecting drivable path by using radar sensors based on five state vector expressed in equation (1), its common means for geometry extraction of drivable part [5, 6].

\[ \hat{x} = [sC \, vO \, s\gamma \, sW \, sS]^T \]  

where \( sC \) represent the curvature, \( vO \) is distance of the vehicle to the road center, \( s\gamma \) is denoted orientation angle of the road, \( sW \) represent road width, \( sS \) signify lane offset, radar distance that exist between left most lane mark and left road boundary.

The drivable path detection using radar sensor is impressive but it is an expensive sensor. In some cases some researchers used sonic sensors for the detection of drivable and non-drivable path using sonic sensor range data of area that are vacant and occupied. Based on this information, areas that are vacant and occupied signify drivable part and non-drivable part respectively, this technique performs better on unstructured roads. However, system using sonic sensors are short-sighted [7]. Recently, cameras are now being provided as sensor to overcome the restriction of autonomous vehicle system using active sensors for navigation. However, environmental noise like rain is a factor that limits the performance of autonomous system using camera for vision. These problem support why some researchers still prefers the use of active sensors [6]. In this research, the autonomous vehicle system is optimized by introducing a filtering algorithm capable of addressing the effects of rain. This algorithm will ensure proper detection of drivable path only in rain scenario with minimal accident which other systems fail to detect when encountering rain [8, 9]. In this paper, the sections are presented as follows: Investigation of various proposed method used for drivable path detection by previous researcher was introduced in section III. Section IV presents the methodology of the drivable part detection system for the autonomous vehicle, section V illustrate pool of pseudo code. Section VI shows the experimental performance of
the optimized system and in section VII, conclusion for the research is drawn.

2. Related Work

Drivable path detection is an essential feature and also an issue to consider when proposing an autonomous vehicle system [1]. Improving the drivable path detection is the purpose of introducing filtering algorithm to the system. However, some scientists using camera sensors for vision have already produced successful result to some extent in drivable path detection. In [8] border detection is used for the detection of road area. The idea proposed to achieved road is based on road area exists between boundaries. The RGB value extraction for road is based on chromatics contrast between pavement and roadsides to find the edge of the border. The chromatic saturation value for each pixel is expressed below in equation (2)

$$S = 1 - \frac{\min(R,G,B)}{\text{Average}(R,G,B)}$$

(2)

where R, G, B signifies color channel red, green and blue.

Based on RGB value of each pixel, scale space analysis is a thresholding algorithm to detect minimum saturation relating to pavement and maximum saturation relating to off road. In [10], methods built on lane detection used by previous scholar was discussed; [11], proposed a system for lane detection in city traffic settings based on color value. [12], proposed a new lane detection system based on color, size, shape and motion for detection of incorrect lane section. [13], developed lane detection approach based on combining three feature cue: lane detection direction, starting position and intensity. This system detects direction of different lane condition even in the presence of shadow or occlusion. [14], also developed a system based on edge distribution function for extraction of lane departure of a vehicle. Hence, they focused on lane extraction because lanes can be used for lateral control, departure warning system and collision prevention. In [15] some certain instructions based on color features are proposed for the detection of road, non-road and uncertain, also canny operation was used to extract the real road boundary because the boundary detection based on color feature might not be consistent due to shadow and road crack. However, experimental presentation of various methods proves remarkable achievement but with limited performance in the presence of environmental noises such as snow, rainfall light intensity, mirage and dust. In this paper, we presented an algorithm (filters) that will enhance drivable path detection for autonomous vehicle system only in rain scenarios.

3. Proposed Optimization Methodology

The optimized system used for drivable path detection includes four stages: features extraction, filtering algorithm stage, segmentation stage, and morphological stage. In feature extraction RGB value based on certain theory’s and law are used for detection. Filtering algorithm stage is used to eliminate the rain effect, whereas, support vector machine (SVM) is used at the segmentation stage for labeling images into drivable path and non-drivable path. Based on the SVM classification, morphological operation used for image enhancement in image processing was introduced to attain better classification result. Figure 1 shows the optimized system connectivity flow of operation for drivable path detection.

3.1 Feature Extraction

Feature extraction Stage of vision based system is based on transforming input data (image) into set of features that can be used for extraction. The most common feature technique used for extraction by previous researcher is color [1, 8]. RGB is the component value
that forms color and is directly obtainable from camera with vital distinctive characteristic based on certain rules or law has been used for object classification in an image. Thus, the hypothesis based on road area having similar color different from non-road has been used for classification and detection with remarkable result achieved [15]. However RGB color values when not corrupted produce better classification result of drivable path than when corrupted by environmental noise [1]. For drivable path estimation we proposed similarity technique at this stage based on RGB color code. We employed the used of histogram euclidean distance for the similarity extraction. Illustration expressed in equation (3)

$$d^2(h,g) = \sum_{R} \sum_{G} \sum_{B} (h(r,g,b) - g(r,g,b))^2$$

(3)

where R, G, B represent red, blue green, r, g, b denoted the corresponding histogram component of red, blue, green while h and g signifies two color histogram.

The full name of authors must be written as the first name, followed by the middle initial and last name. For example, Alex B. Falcon, where “Alex” is the first name, “B.” is the initial for the middle name, and “Falcon” is the last name.

3.2 Filtering Algorithm Stage

The filtering operation plays a crucial role in image processing and it was proposed to optimize the system by selectively suppressing noise caused by environmental factor. Rain has capability to degrade image properties, bringing blurred vision and causing RGB color valve corruption which leads to poor classification result of drivable path at the feature extraction stage of the autonomous vehicle system. The above-mention techniques for drivable path detection in our literature review were all affected by environmental noises. Various techniques can be used to subdue noise but we presented a method based on filtering algorithm to enhance the system performance in drivable path detection in rain scenario only. The detection and removal of rain is based on it temporal property. It was believe by [16] that rain drops that appear far distance from camera has a weak visual effect and appears as fog [17]. We focus our attention on rain drops closer to the camera. They appear in spherical droplets distributed randomly moving at a high speed when closer to the ground [16]. Furthermore, the random distribution of rain drop does not always cover the entire image captured by the camera. Therefore in frames of images for stationary camera, it exhibit two peaks of pixel intensity, background intensity and rain intensity distribution. The application of K-means clustering with K=2 will detect the center cluster for background and rain pixel representing them respectively with smallest and largest intensity on the histogram. The distance between pixel intensity and cluster center is illustrated bellow in equation 4.

$$d(I_p, w) = |I_p - w|$$

(4)

where d represents distance between pixel intensity I_p value of the image and w is denoted as cluster center.

In K-means clustering, image pixel classified as background cluster is based on equation 5 otherwise it rain pixel.

$$d(I_p, w_b) < d(I_p, w_r)$$

(5)

Where W_b and W_r represent cluster center for background and rain respectively.

However, this method might not provide a satisfactory result of rain detection because scene of moving subject due to wind becomes complicated and k-means clustering fails in detecting properly in situations like this[16]. To improve the detection, we proposed also a chromatic property constraint using thresholding algorithm to reduce false detection of rain. Candidate rain pixel classified by K-means clustering whose ΔR, ΔG, and ΔB are approximately the same are also classified as rain pixel otherwise it not a rain pixel. The removal process of rain pixel is accomplished by replacing rain pixel colour with the corresponding background color detected by the K-means clustering. To enhance the removal process, Gaussian blurring and dilation are applied on detected rain pixels and are used as alpha channel (α) to remove streak of rain by (α)-blending. This implies that a new color of a pixel is generated by (α)- blending of rain affected color and background color: Illustration of the new color of the pixel generated is expressed in equation (6)[16].

$$C = αC_b + (1-α)C_r$$

(6)

where C_r and C_b represent respectively rain affected color and background color.

3.3 Segmentation Stage

Image segmentation is a vital operation in image analysis. In computer vision, digital image can be partitioned into several parts using this segmentation algorithm. More precisely, it involves classifying pixels in images for a visual distinctive based on pixel having similar label. Hence, this feature can be used for objects detection and boundaries in an image. Thus, this stage
was proposed to improve the system to achieve a better detection and classification of drivable path and non-drivable path because extracted image pixel at this stage is not corrupted by rain. However, at this stage, support vector machine (SVM) was presented as the segmentation algorithm because of better result in generalization ability and pattern recognition [18]. In image, the SVM tries to address the classification problem using hyperplane surfaces (H) illustrated in equation (7). Hyperplane is the distance that exist between closest non-vector (non-drivable path) [9].

$$H = w \cdot x + b$$  \hspace{1cm} (7)

where $x$ is denoted as point on the hyperplane, $w$ is represented as $n$-dimensional vector perpendicular to $H$, $b$ is indicated as distance of the closest point on hyperplane to the origin.

Applying Lagrange multiplier on image for linearly separable, the optimization problem is shown in equation (8):

$$\text{Maximize } \sum_{i=1}^{1} \lambda_i \left( \sum_{i=1}^{1} \lambda_i y_i x_i + x_j \right)$$  \hspace{1cm} (8)

Subjected to constraint (9)

$$\sum_{i=1}^{1} \lambda_i y_i = 0, \lambda_i \geq 0, i = 1..l$$  \hspace{1cm} (9)

For classification and recognition of drivable path, decision function shown in equation (10) is applied on image.

$$F(x) = \text{sign} \left( \sum_{i=1}^{1} y_i \lambda_i (x, x_i) + b \right)$$  \hspace{1cm} (10)

where $i$ is represented as images pixels, $\lambda_i \ldots \lambda_l$ signifies the vector of non-negative Lagrange multiplier corresponding to constraint in equation (9). The vector $x_i, y_i$ when $\lambda_i \neq 0$ are classified as support vector, others are termed as training vectors when $\lambda_i = 0$ [9].

3.4 Post Processing Stage

Post processing as proposed to the drivable path detection system is a vital stage because of its role in computer vision. It was employed for geometrical analysis of digital image structure using set rules, matrix rules, surface structure and random function. The system post processing is based on morphological operation it was employed because of good performance result from previous researchers [1, 19, 20]. In morphological operation, the assumption is based on road areas are connected and associated together. Based on implementing this hypothesis, connecting component algorithm shown in equation (17) was employed to extract the largest connected area (drivable path) and other part of the image different from the connected part are classified as non-drivable part. Image reconstruction using morphology is based on the following operation

Binary dilation as shown in equation (11) performs the laying of structure element B on the image A

$$A \oplus B = \bigcup_{b \in B} A_b$$  \hspace{1cm} (11)

Binary erosion as shown in equation (12). This operation is similar to dilation but it tries to shrink [21].

$$A \Theta B = \{ p | p + b \in A \forall b \in B \}$$  \hspace{1cm} (12)

These are the main operations (dilation and eroding), but filters as shown in equation (1) and (14) are proposed for smoothing the operations of dilation and erosion.

Erosion follow by dilation $A \Theta B = (A \Theta B) \oplus B$  \hspace{1cm} (13)

Dilation follow by erosion $A \bullet B = (A \oplus B) \Theta B$  \hspace{1cm} (14)

The boundary of the road image represented by $\beta(A)$ is extracted by performing set differences between A and it erosion as shown in equation (15)

$$\beta(A) = A - (A \Theta B)$$  \hspace{1cm} (15)

Component connecting algorithm was proposed for drivable path detection of the image using image dilation, complement and intersection as shown in equation (16).

$$X_k = (X_{k-1} \oplus B) \cap A', k = 1, 2, 3, \ldots$$  \hspace{1cm} (16)

where $X_0 = P$ (initial point), $B$ is represented as symmetric structure element, $\cap$ is denoted as interception operator, $A'$ indicates the complement of set image A, $X_k$ hold all the filled holes. At iteration K the algorithm stop when $X_k = X_{k+1}$ [19].

4. Pools of Pseudo Code

4.1 Pseudo Code for Drivable Path at the Feature Extraction Stage

Input: A of Stream 1.....k images, where A is represented as frame
Output: A of Stream 1…k images with extracted RGB drivable path features.

Step 1: Select A.
Step 2: Color extraction of A using RGB color space.
Step 3: Similarity extraction for region classification using histogram Euclidean distance.
Step 4: Drivable path of A is extracted for every stream 1….k images.

In algorithm 1, A signifies frame size captured by camera. Image captured are made up of RGB component value which are available when color camera are used. The RGB components are a major technique for image analysis. Using similarity hypothesis it was used for drivable path detection.

4.2 Pseudo Code for Eliminating Rain Effects

Input: Extracted RGB component of A form streams 1…..k images.
Output: Uncorrupted RGB component of A for every stream 1……k image with.

Step 1: Read in RGB color feature A for every stream 1…k image.
Step 2: Apply temporal and chromatic property.
Step 3: Apply K-means clustering.
Step 4: Apply, Gaussian blurring and dilation.
Step5: Extracted A for every frame 1….k images with no rain effect.

In algorithm 2, the temporal and chromatic properties are used to address false detection of rain to avoid removing important information to miss-detection as rain. Gaussian blurring and dilation is used to remove the rain by replacing it effect using background color.

4.3 Pseudo Code for SVM Labelling Algorithm at the Segmentation Stage

Input: Uncorrupted RGB color value of A for every stream 1…k images
Output: Drivable path classification of A for every stream 1….k images.

Step 1: Read A (uncorrupted RGB) from streams 1…..k.
Step 2: Apply Kernel function to solve constrained optimization problem of A from 1....K.
Step 3: A from stream i…k images form a new optimized problem. Apply equation 8
Step 4: Drivable and non-drivable path classification of A for stream i….k images. Apply equation.
Step 5: Drivable Path detection of A for every stream 1….k images.

In algorithm 3, segmentation algorithm has the ability to provide an improved classification results than feature extraction stage. Its idea of classification is based on finding corresponding distance between two non-vectors (non-drivable) which will be set as a new border.

4.4 Pseudo Code for Drivable path Labelling at the Post Processing Stage

Input: Drivable path detection of A for every stream 1…..k images.
Output: Optimized drivable path detection of A for every stream 1……k images.

Step 1: Read in A from every stream 1…..k images. Classification result from segmentation stage
Step 2: Image dilation on A for every stream 1…k images.
Step 3: Image erosion on A for every stream i…k images.
Step 4: morphological filters on preceding operation of A for every Stream 1....k images.
Step 5: edge detection of A for every stream 1….K images for boundary detection.
Step 6: region filling for drivable path classification of A for every stream i….k images.

In algorithm 4, region filling algorithm in morphology is used for classification. This phase produce the best result of drivable path detection with improved navigation for the autonomous vehicle system.

5. Experimental Result and Evaluation Scheme

In this paper, images used to test the system are publicly available sourced via internet, based on these images the result evaluation of the optimized drivable path detection system is observed using visual comparison and qualitative approach. Visually, real life images (input image) is compared with the output vision of the image as illustrated below in figure 2a and 2b.

2a                                          2b
Figure 2: The visual comparison result

Furthermore, assessment scheme was used to test the output image based on ground truth classification to measure the success rate at which the filtering algorithm introduced to the system has enhanced the classification result. Evaluation comparison using 100 frames images on an average 10 are used to test the drivable path detection system with and without filtering algorithm. Figure 3 shows the graphical representation of the optimized drivable path detection system achieving better performance than the system without filtering algorithm. The evaluation schemes used to test the system based on ground truth classification are as follows in equation (17)-(23) respectively, where ACC is denoted as accuracy rate: Proportion of total numbers of drivable and non-drivable
paths that are predicted correctly. ERR represent error rate: Proportion of total numbers of drivable and non-drivable paths that are predicted incorrectly. TPR is the total positive rate: Proportion of drivable path case that are correctly classified. FNR represent the false negative rate: Proportion of drivable path case that is incorrectly classified as non-drivable path. TNR denote total negative rate: Proportion of non-drivable paths case that are classified correctly. FPR represent false positive rate: Proportion of non-drivable paths cases that are classified as drivable paths. PRE is denoted as precision: Proportion of the predicted drivable paths that is correct.

\[
\text{ACC} = \frac{TP_i + TN_i}{N_i} \tag{17}
\]

\[
\text{ERR} = \frac{FN_i + FP_i}{N_i} \tag{18}
\]

\[
\text{TPR} = \frac{TP_i}{TP_i + FN_i} \tag{19}
\]

\[
\text{FNR} = \frac{FN_i}{TP_i + FN_i} \tag{20}
\]

where \(TP_i\) is represented as true positive: numbers of drivable pixel properly classified in the \(i^{th}\) image, \(TN_i\) denote true negative: non-drivable pixel that are properly labeled in the \(i^{th}\) image, \(FP_i\) is signified as false positive: numbers of non-drivable pixel labeled as drivable in the \(i^{th}\) image. \(FN_i\) is represented false negative: numbers of drivable pixel that are labeled as non-drivable in the \(i^{th}\) image and \(N_i\) denote the number of drivable pixel and non-drivable pixel in \(i^{th}\) ground truth labeling.

\[
\text{TNR} = \frac{TN_i}{TN_i + FP_i} \tag{21}
\]

\[
\text{FPR} = \frac{FP_i}{TN_i + FP_i} \tag{22}
\]

\[
\text{PRE} = \frac{TP_i}{TP_i + FP_i} \tag{23}
\]

Fig 3 (a) shows the accuracy representation and Fig 3(b) shows error rate representation.

**4. Conclusion**

In this research, the proposed method has verified the vision enhancement of autonomous vehicle capability of detecting drivable path for a smooth navigation in rain weather condition. The filtering algorithm based K means clustering is studied as a means of detecting and removing the effect of rain. However, the output at this stage is an uncorrupted RGB image value which gives the segmentation algorithm (SVM) capability of proper
classification of the image into drivable and non-drivable path. Furthermore, enhancement of the classification result of the segmentation algorithm is done at the morphological stage because of its feature characteristic. The performance result of our method is clarified by testing the system with different road images in rain scenarios and remarkable results are reached. This impact has contributed to improvement in path detection and navigation of autonomous vehicle.

Acknowledgement

The author will like to thank the department of Computer System Engineering, Web and Multimedia computing of Tshwane University of Technology for making resources needed for this research available.

References


