International Journal of Engineering Research and General Science Volume 5, Issue 5, September-October, 2017
ISSN 2091-2730

EFFICIENT LANE DETECTION USING THE HYBRIDISATION OF ARTIFICIAL BEE COLONY & MODIFIED HOUGH TRANSFORM

Er Rajni Multani , Chetan Marwaha

Guru Nanak Dev University(Amritsar), rajnimultani21@gmail.com

Abstract— This paper discusses lane coloration in real time vehicular adhoc system. Lane detection is normally helpful to localize path limits, determine undesired lane variations and to enable approximation of the upcoming geometry of the road. There are different types of methods that are used for detecting lines i.e. Hough transform, clustering and curve fitting. These methods are working efficiently but generally the however they are unsuccessful or otherwise not provide efficient results when there are curved lane road images. The objective of this paper is to improve lane coloration algorithm based edge detector by modifying the Hough transform i.e. hybridization of additive Hough transform with artificial bee colony edge detection to detect curve lanes.

Keywords— Lane Detection, ABC Algorithm, Modified Hough Transform, Curved lanes, Clahe, Segmentation, Region of Interest

INTRODUCTION

Nowadays in this contemporary world, travelling has turned into an essential part of regular life. Consequently, in the last few years, the volume of automobiles around the world population has enhanced to massive degree. A particular adverse component of this kind of development are the traffic incidents that take numerous lives in day- to- day life. The primary factors that cause these types of injuries are fatigue and ineptness. Automated lane detection is a key component of urban driving that plays a significant role in driver assistance systems and finds application in both autonomous as well as manned vehicles. Lane detection is generally helpful to localize road boundaries, to discover undesired lane variations as well as to enable estimation of the forthcoming geometry of the road. [3] Due to the fact that the volume of accident victims has increased annually with the expanding amount of vehicles on the road so as a result lane detection has become a crucial exploration area for smart automobile technologies. A lot of accidents develop from insufficient awareness about driving ailments caused by driver negligence as well as visual interference. The goal of the lane detection is to discover lane marks out of background debris and in addition to track out their actual location by utilizing specific appliances or by using machine vision based techniques. The principle connected with Lane Departure Warning System (LDWS) as well as Forward Collision Warning System (FCWS) is the curve lane detection. Lane detection becomes a difficult task under challenging conditions such as the dashed lane markings and automobile occlusion because of unreliable lane feature points on the structural paths.[5] Lane detection is an important procedure in large number of these applications as lanes provide important information like region-of interest for further processing. Due to huge variety of road types such as (rural, urban, suburban), varying weather conditions like (sunny, cloudy, rainy, snowy, foggy) as well as illumination variations such as (shadow, glare and also darkness) stating that it gets difficult to develop a powerful lane detection algorithm that always works efficiently for many different road environments. So far, several methods have been proposed to detect lane markings painted on the road surface like RADAR[12], DBSCAN[15], Spiking Neural Networks and Hough Transform[16],Edge Detection with Median Filter and Weiner filter[18] etc. All these methods have failed in the case of curved lanes and are mainly used for detection of straight lanes. Further, the condition of lanes in the presence of noise, shadow, illumination conditions are not detected. So in this paper we discuss an Artificial Bee Colony & Clahe based algorithm by using hybrid hough transform for efficient lane detection. Finally we study the performance of traditional hough transform and proposed modified hough transform based approach.

RELATED WORK

The literature survey of existing research in lane detection has been presented in the form of table as follows:

SNo	Year	Authors	Technique used	Limitations
1.	2014	Sukriti Srivastava et al	Lane detection using median filter, weiner filter and integrated hough transform	did not solve the problem of detecting lanes in case of heavy rain as well as different lighting conditions
2.	2015	Upendra Suddamalla et al	Edges are extracted using adaptive thresholding	Poor results in terms of weather conditions
3.	2015	Xue Li et al	Lane detection based on spiking neural network and hough transform	Failed in case of optimization algorithms
4.	2015	Jianwei Niu et al	Two stage feature-extraction along with curve fitting	does not address explicitly lane occlusions and is not applicable to lanes with large curvatures
5.	2014	Jongin Son et al	Real time illumination invariant lane detection for lane departure warning system	Not applicable in complicated contexts for example blur conditions, low sun angle conditions as well as lane cracks
6.	2014	Abdelhamid Mammeri et al	MSER Algorithm, Hough Transform and Kalman Filter	Lanes cannot be detected in case of noisy conditions
7.	2013	Andreas Richtsfeld et al	B-Spline Curves in order to detect RGB-D images	Sharp edges are not considered
8.	2013	Payam S. Rahmdel et al	multilayer fractional Fourier transform (MLFRFT) and the state-of-the-art advance lane detector (ALD)	High computational time required
9.	2013	Dajun Ding et al	Adaptive Road ROI Determination Algorithm	Absence of appropriate road data and the effect of disturbances causes problems
10.	2013	Elif Deniz Yigitasi et al	Edge detection algorithm	Corner pixels and pixels of frame around the image are often either ignored or taken as zero.

PROPOSED WORK

In the lane detection algorithm initially an image is captured with the help of a camera and then preprocessing is done with the help of artificial bee colony algorithm in order to extract region of interest. After that scaling and segmentation are done to extract the segmented portions of an image. Firstly, adaptive thresholding and then fuzzy c-means is used for the segmentation technique. The Vertical histogram is constructed to get the estimation about the mid lane of segmented image. To locate the sharp images of segmented image, prewitt operator is used to found the gradient magnitude of an image. Then the inner margins of the lane marks are recognized and colored. Modified Hough Transform is applied to detect and color lanes and finally its performance is evaluated. Various steps used for lane detection are as shown below:

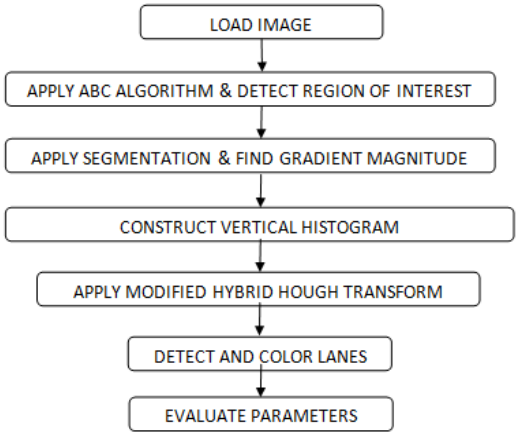

Fig 4.1: Framework of Proposed Algorithm

In this paper we use artificial bee colony algorithm to find region of interest in order to enhance the results. Artificial Bee Colony is a novel optimization technique involving natural behavior of honey bees in search for the best food resources. The Pseudocode of our algorithm is as follows:

Step1: Read input image and resize it

Step 2: Define problem specific variables

$objfun = 'Distancefn';$

$D = 100;$

$ubd = ones(1, D) * 100;$

$lbd = ones(1, D) * (-100);$

$for\ r' = 1{:}\ runtime$

Step3: Initialization of food sources in the range]lbd, ubd[

$Range = repmat((ubd - lbd), [foodnum\ 1]);$

$Low = repmat(lbd, [foodnum\ 1]);$

$Foodsrc = randm(foodnum, D) * Range + Low;$

Step4: Evaluate objective function and update fitness value

$ObjVal = feval(objfun, Foods);$

$Fitness = calculateFitness(ObjVal);$

Step5: memorize the best source of food

$BestI'n'd' = find(ObjVal == \min(ObjVal));$

$BestI'n'd' = BestI'n'd'(end);$

$GlobalMin = ObjVal(BestI'n'd');$

$GlobalParams = Foods(BestI'n'd', :);$

$itern = 1;$

$while\ ((itern \leq maxCycle))$

Step6: Phase of Employed bee

$for\ \mathrm{i} = 1{:}\ (FoodNum)$

$Para2Change = fix(randm * D) + 1;$

$neighbr = fix\big(randm * (FoodNum)\big) + 1;$
$while(neighbr == i)$
$neighbr = fix\big(randm * (FoodNum)\big) + 1;$
$end;$
$soln = Foods(i,:);$
$soln(Para2Change) = Foods(i,Para2Change) + (Foods(i,Para2Change) - Foods(neighbr,Para2Change)) * (randm - 0.5) * 2;$
$i'n'd' = find(soln < lbd);$
$soln(i'n'd') = lbd(i'n'd');$
$i'n'd' = find(soln > ubd);$
$sol(i'n'd') = ub(i'n'd');$
$ObjValSoln = Feval(objfun, soln);$
$FitnessSoln = calculateFitness(ObjValSoln);$

Step7: A Greedy approach is applied between the current solution k and its mutant

$if\ (FitnessSoln > Fitness(i'))$
$Foodsrc(i',:) = soln;$
$Fitness(i') = FitnessSoln;$
$ObjVal(i') = ObjValSoln;$
$trial(i') = 0;$
$else$
$trial(i') = trial(i') + 1$
$end;$
$end;$

Step8: Probability is measured by using fitness values and then normalized by dividing maximum fitness value

$probab = (0.9.* Fitness./\max(Fitness)) + 0.1;$

Step9: ONLOOKER BEE PHASE

$while(t < FoodNum)$
$if(randm < probab(i))$
$t = t + 1;$
$Para2Change = fix(randm * D) + 1;$
$neighbr = fix\big(randm * (FoodNum)\big) + 1;$

Step10: Arbitrarily chosen solution should vary from the solution i

$while(neighbr ==)$
$neighbr = fix\big(randm * (FoodNum)\big) + 1;$
$end;$
$soln = Foodsrc(i,:);\ sol(Para2Change) = Foodsrc(i,Para2Change) + \big(Foodsrc(i,Para2Change) - Foodsrc(neighbr,Para2Change)\big) * (randm - 0.5) * 2;$

Step11: Generated parameter value is shifted onto the boundaries

$i'n'd' = find(soln < lbd);$
$soln(i'n'd') = lbd(i'n'd');$
$i'n'd' = find(soln > ubd);$
$soln(i'n'd') = ubd(i'n'd');$

Step12: Find out a new effective solution

$ObjValSoln = feval(objfun, soln);$
$FitnessSoln = calculateFitness(ObjValSoln);$
$if\ (FitnessSoln > Fitness(i))$
$Foodsrc(\mathrm{i},:) = sol; Fitness(i) = FitnessSoln;\ ObjVal(i) = ObjValSoln;$
$trial(i) = 0;$
$else$
$trial(i) = trial(i) + 1;$
$end;$
$end;$
$i = i + 1;$
$if(i == (FoodNum) + 1)$
$i = 1;$

$end;$
$end;$

Step13: The most effective source of food is memorized

$i'n'd' = find(ObjVal == \min(ObjVal));$
$i'n'd' = i'n'd'(end);$
$if(ObjVal(i'n'd') < GlobalMin)$
$GlobalMin = ObjVal(i'n'd') * p;$
$GlobalParams = Foods(i'n'd', :);$
$end;$

Step14: Determine the sources of food whose trial counter exceeds the "limit" value.

$i'n'd' = find(trial == \max(trial));$
$i'n'd' = i'n'd'(end);$
$if(trial(i'n'd') > limit)$
$Bas(i'n'd') = 0;$
$sol = (ubd - lbd).* randm(1, D) + lbd;$
$ObjValSoln = feval(objfun, soln);$
$FitnessSoln = calculateFitness(ObjValSoln);$
$Foodsrc(i'n'd', :) = soln;$
$Fitness(i'n'd') = FitnessSoln;$
$ObjVal(i'n'd') = ObjValSoln;$
$end;$
$itern = itern + 1;$
end
$GlobalMins(r) = ceil(GlobalMin);$
$end;$
$soln = (p * GlobalMins)/(n^3)$

Step15: Apply ABC based Clahe

$abc_{factor} = stretchlim(Image, sol);$
$abc_{image} = imadjust(Image, abc_{factor}, [\ ]);$

Step16: Evaluate and update region of interest

$imgData = Image(:, ;, 1);$
$[A, B] = find(imgData \cong 255);$
$xmin = \min(A);$
$xmax = \max(A);$
$ymin = \min(B);$
$ymax = \max(B);$
$width = xmax - xmin;$
$height = ymax - ymin;$
$imgSelect = imcrop(imgData, [xmin, ymin, xmax, ymax]);$
$imgSelectrgb = imcrop(Image, [xmin, ymin, xmax, ymax]);$

Step17: Apply fuzzy c-means based segmentation

$data = reshape(Image, [\ ], 1);$
$[center, member] = fcm(double(data), 3);$
$[center, cidx] = sort(center);$
$member = member';$
$member = member(:, cidx);$
$[maxmember, label] = max(member, [\ ], 2);$
$binary1 = Image >$
$level = \frac{\max(data(label == 1)) + \min(data(label == 2))}{2};$ $if\ level > 1$
$level = \frac{level}{255};$
end

Step18: Evaluate Gradient Magnitude Of Image

$yaxis = fspecial('prewitt');$
$xaxis = yaxis';$

$Imageyaxis = imfilter(double(Image), yaxis, 'replicate');$
$Imagexaxis = imfilter(double(Image), xaxis, \quad 'replicate');$
$gradmag = sqrt(Imagexaxis.^2 + Imageyaxis.^2);$

Step19: Apply canny based hough transform

$BW = edge(binary, \quad 'canny');$
$[H, T, R] = hough(BW);$
$P = houghpeaks\left(H, 5, 'threshold', ceil\left(0.3 * \max(H(:))\right)\right);$
$x = T(P(:,2)); y = R(P(:,1));$

Step20: Apply morphological operations

$binary = bwareaopen(binary, 46);$
$s'e' = strel('disk', 2);$
$b'w' = imclose(b'w', se);$
$b'w' = imfill(b'w', 'holes');$
$[B, L] = bwboundaries(b'w', 'noholes');$

Step21: Demonstrate the particular label matrix as well as draw each and every boundary

$for\ k = 1: length(B)$
$boundaryy = B\{k\};$
$plot(boundaryy(:,2), boundaryy(:,1), 'b', 'LineWidth', 2)$
end
$stats = regionprops(L, 'Area', 'Centroid');$
$for\ k = 1: length(B)$

Step22: Estimate the basic approximation of the object's perimeter

$delta_{sq} = diff(boundaryy).^2;$
$peri = sum\left(sqrt\left(sum\left(delta_{sq}, 2\right)\right)\right);$

Step23: Determine the calculated area corresponding to label 'k'

$area = stats(k).Area;$

Step24: Calculate the roundness metric

$$metric = 4 * pi * \frac{area}{peri\text{^}2};$$

$metric_{string} = sprintf('\%2.2f', metric);$
$metricn(k) = metric;$

Step25: Mark objects over the threshold with a black circle

$count = count + 1;$
$centroid = stats(k).Centroid;$
$plot(centroid(1), centroid(2), 'ko');$
end
$lowthreshold = .009;$
$BW_{filled} = imfill(b'w', 'holes');$
$boundaries = bwboundaries(BW_{filled});$

Step26: Find lines and plot them

$liness = houghlines(BW, T, R, P, 'FillGap', 5, 'MinLength', 7);$
$max_{len} = 0;$
$for\ k = 1: length(liness)$
$x'y' = [liness(k).point1; liness(k).point2];$

Step27: Determine the endpoints of the longest line segment

$length = norm(liness(k).point1 - liness(k).point2);$
$if(length > \max_{len})$
$\max_{len} = length;$

$x'y'_{long} = x'y';$
end
end

Step28: Return lane colorized image

```
if count~ = 0
hold on
for k = 1:count
b = boundaries{k};
if metricn(k) < threshold && metricn(k) > lowthreshold
plot(b(:,2),b(:,1),' g',' LineWidth',3);
hold on
end
end
end
```

RESULTS -

In an effort to measure the performance of existing and proposed algorithm criteria, different lane metrics are usually considered. For this we have taken 10 set of various images in .jpg format.

A. Balanced Error Rate (BER)

The Balanced error rate is the average of the errors on each class and is denoted by formula:

temp_BER = 100 * (1 -temp_BCR)

According to the need this error rate should decrease and in addition should be minimum.

Table 2 shows the different values of balanced error rate for set of 10 different images.

Input Image	Existing work	Proposed work
1	49.2835	39.5502
2	32.5438	29.7490
3	27.0138	20.0426
4	32.1851	6.1262
5	24.3058	19.4388
6	26.4659	21.0262
7	37.9812	8.8886
8	45.2974	36.5385
9	24.6136	22.7887
10	12.3718	7.9066

Table 2: Balanced Error Rate

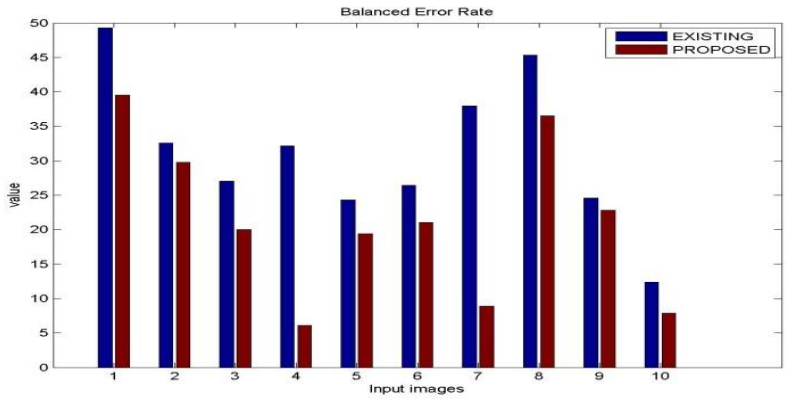

Fig 3: Balanced Error Rate

Fig 3 shows a graph of balanced error rate analysis. It is seen that the proposed technique has lower balanced error rate as compared to existing technique.

B. F-measure

F-measure is used for computing the average values of precision and recall. Table 3 shows the different values of f measure for set of 10 different images.

Input Image	Existing work	Proposed work
1	2.8254	34.5734
2	51.7557	57.6533
3	62.9877	74.9333
4	52.5398	93.4740
5	67.8895	75.8707
6	64.0087	73.3757
7	38.7586	90.2442
8	17.1934	42.4242
9	67.3501	70.4853
10	85.8815	91.4145

Table 3: F-measure

F-measure is expressed with the help of following formula :
F-measure=2(precision.recall) / (precision + recall)
Graphical representation of f measure is shown in fig 4. From the graph it is clearly highlighted that the proposed technique is much better than simple lane detection algorithm.

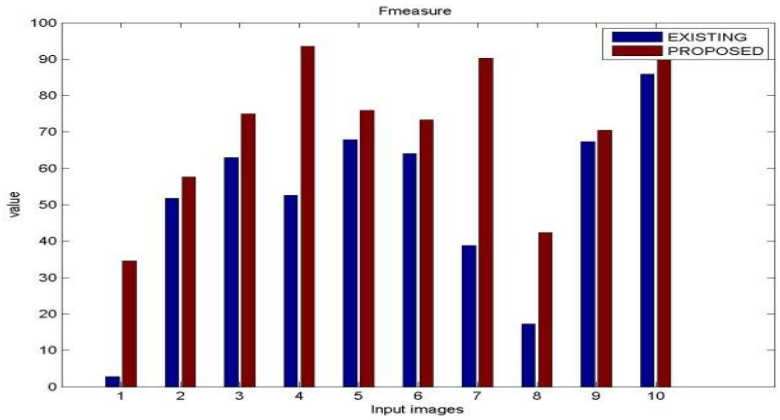

Fig 4: F-measure

C. Recall

Recall is the measure of how many positives does our proposed algorithm gives. It also determines the sensitivity of our network. Recall is expressed with the help of following formula :

Recall = (Number of true positive assessments) / (Number of all positive assessments)

Table 4 shows the different values of f measure for set of 10 different images.

Input Image	Existing work	Proposed work
1	0.0143	0.2090
2	0.3491	0.4050
3	0.4597	0.5991
4	0.3563	0.8775
5	0.5139	0.6112
6	0.4707	0.5795
7	0.2404	0.8222
8	0.0941	0.2692
9	0.5077	0.5442
10	0.7526	0.8419

Table 4: Recall

The value of recall should be higher. As higher the value of recall better is the performance of our algorithm.
Fig 5 shows the graphical representation of recall analysis.

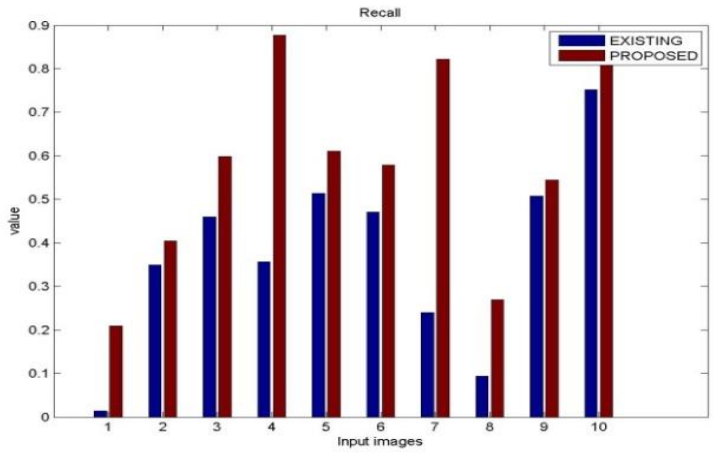

Fig 5: Recall Analysis

CONCLUSION

Lane detection enables you to obtain the position as well as direction of the vehicle along with lane information. There are different types of methods that are used for detecting lines. The methods formulated until now are operating effectively as well as providing beneficial results in scenario when the straight lane images are generally there. However challenge is simply because that they are unsuccessful or otherwise not provide successful outcomes whenever there are curved lane road images. In this modify Hough transform i.e. additive Hough transform with artificial bee colony based edge detector is used to improve straight lane as well as curved lane road images. The proposed technique has been designed and implemented in Matlab simulator 2010 by using image processing toolbox. The comparison has been drawn between Hough transform and Hybridization of additive Hough transforms with ABC edge detection by using various parameters balanced error rate, f-measure and recall. The proposed methods outperforms over existing methods.

REFERENCES:

[1] Srivastava, Sukriti, Manisha Lumb, and Ritika Singal. "Lane detection using median filter wiener filter and integrated hough transform." *Journal of Automation and Control Engineering* 3.3 (2015).

[2] Suddamalla, Upendra, et al. "A novel algorithm of lane detection addressing varied scenarios of curved and dashed lanemarks." *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*. IEEE, 2015.

[3] Li, Xue, et al. "Lane detection based on spiking neural network and hough transform." *Image and Signal Processing (CISP), 2015 8th International Congress on*. IEEE, 2015.

[4] Niu, Jianwei, et al. "Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting." *Pattern Recognition* 59 (2016): 225-233.

[5] Son, Jongin, et al. "Real-time illumination invariant lane detection for lane departure warning system." *Expert Systems with Applications* 42.4 (2015): 1816-1824.

[6] Mammeri, Abdelhamid, Azzedine Boukerche, and Guangqian Lu. "Lane detection and tracking system based on the MSER algorithm, hough transform and kalman filter." *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2014.

[7] Mörwald, Thomas, et al. "Geometric data abstraction using B-splines for range image segmentation." *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013.

[8] Rahmdel, Payam S., Daming Shi, and Richard Comley. "Lane detection using Fourier-based line detector." *Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on*. IEEE, 2013.

[9] Ding, Dajun, Chanho Lee, and Kwang-yeob Lee. "An adaptive road ROI determination algorithm for lane detection." *TENCON 2013-2013 IEEE Region 10 Conference (31194)*. IEEE, 2013.

[10] Yigitbasi, Elif Deniz, and Nurdan Akhan Baykan. "Edge detection using artificial bee colony algorithm (ABC)." *International Journal of Information and Electronics Engineering* 3.6 (2013): 634.

[11] Bottazzi, Vitor S., Paulo VK Borges, and Jun Jo. "A vision-based lane detection system combining appearance segmentation and tracking of salient points." *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013.

[12] Li, Nan, et al. "A spatial clustering method with edge weighting for image segmentation." *IEEE Geoscience and Remote Sensing Letters* 10.5 (2013): 1124-1128.