Knowledge-Based Systems 38 (2013) 37–47

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Linear fuzzy space based road lane model and detection

Đ. Obradović [a,*], Z. Konjović [a], E. Pap [b,c], I.J. Rudas [c]

[a] Faculty of Technical Sciences, Department of Computing and Automation, 21000 Novi Sad, Serbia
[b] Faculty of Sciences, Department of Mathematics and Informatics, 21000 Novi Sad, Serbia
[c] Obuda University, Institute of Intelligent Engineering Systems, H-1034 Budapest, Hungary

ARTICLE INFO

Article history:
Available online 12 January 2012

Keywords:
Lane model
Fuzzy point
Fuzzy line
Fuzzy collinear
Image processing
Line detection
Linear fuzzy space

ABSTRACT

In this paper, we propose a new road lane model based on linear fuzzy space mathematics, coupled with a robust road lane detection method using fuzzy c-means clustering. The fuzzy line based road lane model presented here describes a lane as a set of fuzzy collinear fuzzy points. The proposed algorithm for road line detection is able to deal with imprecise data and enables reduced computational complexity (proportional to the number of fuzzy points multiplied by the number of fuzzy lines) versus a standard Hough transformation. Experimental results show that the proposed method is fast, and robust enough for use in real-time applications. The proposed method has been implemented as an Android-based mobile phone application.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Demand for traffic safety systems designed to minimize the risk of accidents is increasing. Lane detection plays a significant role in driver assistance systems, and can help to estimate the geometry of the road ahead. According to the International Data Corporation, nearly 300 million smart phones were shipped in 2010; and the smart phone market is expected to grow to 450 million units in 2011. Smart phones are often equipped with a digital camera, GPS and WiFi capability, and represent promising devices for driver assistance systems. Thus, the development and implementation of efficient imprecise space representation models capable of feature extraction and detection in smart phones is increasingly relevant.

A typical complete model-based lane detection system consists of four parts: (1) lane modeling, (2) feature extraction, (3) detection, and (4) tracking. Lane modeling involves the development of mathematical descriptions to represent road lanes; while feature extraction involves identification of particular lane features, such as color, texture or edge, etc. During the detection stage, the lane model is then fit with the set of extracted features. Finally, lane tracking is applied to follow lane changes using reduced system complexity, which is achieved by reducing the search region.

Although the Hough transform [1] remains one of the most commonly applied lane detection techniques [2–4]; other techniques have also been applied to this problem. For example, Pomerleau [5] used neural networks in their ALVINN system; while Kang and Jung [6] used connected-component analysis and dynamic programming. In addition, probabilistic methods, such as Maximum a posteriori estimation evaluated using the Metropolis algorithm, have been reported for use in lane detection systems [7,8]. More recently, Wang et al. [9] proposed the use of fuzzy methods during the feature extraction stage for automatic brightness compensation. With respect to the line model, several different approaches have been reported in the literature, including representing road lane models as: a straight line [10]; B-spline [11]; parabola [4]; or hyperbola [12]. A more detailed survey of lane detection strategies has been published [4]. A common component of nearly all lane detection systems is the use of specialized hardware or a PC, coupled with a vehicle mounted video camera.

Common problems in lane detection arise from the fact that discrete space (digital raster image) is used for real-world element representation, while the spatial relations used apply the rules of continual space. For example, lines are represented as a set of discrete points that typically do not have to be collinear, in contradiction with the definition of a line. In addition, because real-world objects are mapped to a digital raster image via a variety of sensors, the resulting image is only an approximation of the real-world object. Thus, imperfections in either the image data or the edge detector may result in missing points or pixels on lines, as well as spatial deviations between an ideal line and the set of imprecise points obtained from the edge detector. The overall effect is an image containing varying levels of geometric distortion.

This present study focuses on modeling basic planar imprecise geometry objects and the relationships between them. The application of these models to spatial data management systems is then described, and results are briefly presented.

* Corresponding author.
E-mail addresses: obrad@uns.ac.rs, obradovic.djordje@gmail.com (Đ. Obradović), ftn_zora@uns.ac.rs (Z. Konjović), pap@dmi.uns.ac.rs (E. Pap), rudas@uni-obuda.hu (I.J. Rudas).

0950-7051/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2012.01.002

An overview of studies in the literature dealing with imprecise point object modeling has been published [13].

In general, three basic approaches to spatial data uncertainty/imprecision are recognized: (1) Exact Models [14–18,11], (2) Probabilistic Models [7,8,19–23], and (3) Fuzzy Models [13,24–31].

One of the earliest and still most commonly used techniques for feature extraction is known as the Generalized Hough transformation, proposed by Duda and Hart [1] in 1972. The Generalized Hough transformation is based on a voting procedure carried out in parameter space. Line candidates are obtained as local maxima of the Hough transformation, which highly depend on spatial relation co-linearity.

Based on our previous results [13], we introduce here a novel mathematical model of fuzzy lines, as well as models of basic spatial relations, including: coincidence, between and collinear. Practical applications of the results obtained in this paper are based on simple, yet effective, modeling of imprecise data using fuzzy sets, which enables the gradual estimation of object spatial relations. Instead of using a set of discrete 2D points, we propose the use of a set of fuzzy points, which makes road lane detection more robust then the crisp approach, due to the incorporation of a gradual estimation of feature spatial relations.

This work consists of six sections (Sections 1–6). Following this introduction (Section 1), a novel road lane mathematical model is presented, along with definitions of basic spatial fuzzy relations (Section 2). After that, a new algorithm for road feature extraction is described (Section 3), followed by a novel road lane detection algorithm based on fuzzy relations defined in linear fuzzy space (Section 4), and experimental results (Section 5). Finally, concluding remarks and future research directions are discussed (Section 6).

2. The road lane model

The conceptual scheme proposed in this paper, consisting of a driver assistance system integrated into a smart phone, is illustrated in Fig. 2.1. The software components are divided in four functional groups: (1) image capture, (2) road feature extraction, (3) road lane detection, and (4) a decision module. The focus of the present work is on road feature extraction and road lane detection.

Capturing images on Android based smart phones is a part of the Android operating system. The common format of real time image flow is 320×240 pixels with a YUV based color model, which encodes color information by taking human perception into account.

Road lane detection modules depend on good feature estimation, which is conducted during the road feature extraction stage. Usually, road features are extracted as a set of discrete points, using edge extraction or texture extraction methods. In this paper, we propose a modified edge extraction method. The region of interest for road feature extraction is contained in the lower half of the image (the 320×120 gray scale luminance part of the original image).

In this paper, we constructed a 2D lane model using a set of imprecise points extracted during the edge detection stage. Most road lane detection systems are sensitive to the precision of the points extracted by the edge detection method. Fig. 2.2 illustrates how the edge points extraction process works. Specifically, an edge point is located at a position where the plot profile first derivate has a maximum value. In Fig. 2.2, the edge point is located "somewhere" around a distance of 235 pixels.

Instead of exact discrete values, we incorporated fuzzy points, as previously defined [13] for edge point location. Fuzzy points defined in 2D linear fuzzy space are represented as discrete 2D points extended with single non-negative real values. In the following figures, fuzzy points are represented as circles, with a center in core and a radius that corresponds to a support.

The road lane model proposed in this work is actually a set of fuzzy lines and fuzzy points based on the theory of fuzzy sets presented in [13], which is related to fuzzy point, fuzzy line and spatial relations in R^2 linear fuzzy space.

2.1. Fuzzy line in R^2 linear fuzzy space

The imprecise line model given in this paper relies on the concept of fuzzy points and linear fuzzy space introduced in our previous work [13]. Following two preliminary definitions, we will introduce basic operations over linear fuzzy space $\mathcal{H}^2$ defined in $\mathbb{R}^2$, as well as their properties, which are later used in our road lane model.

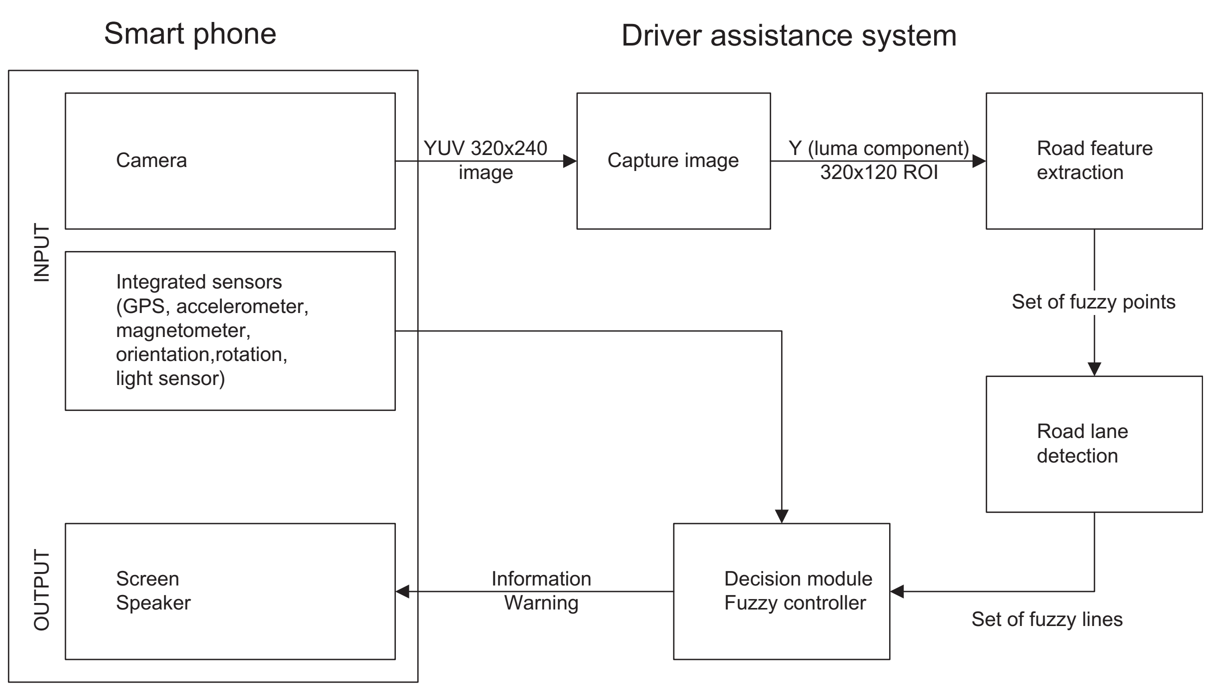
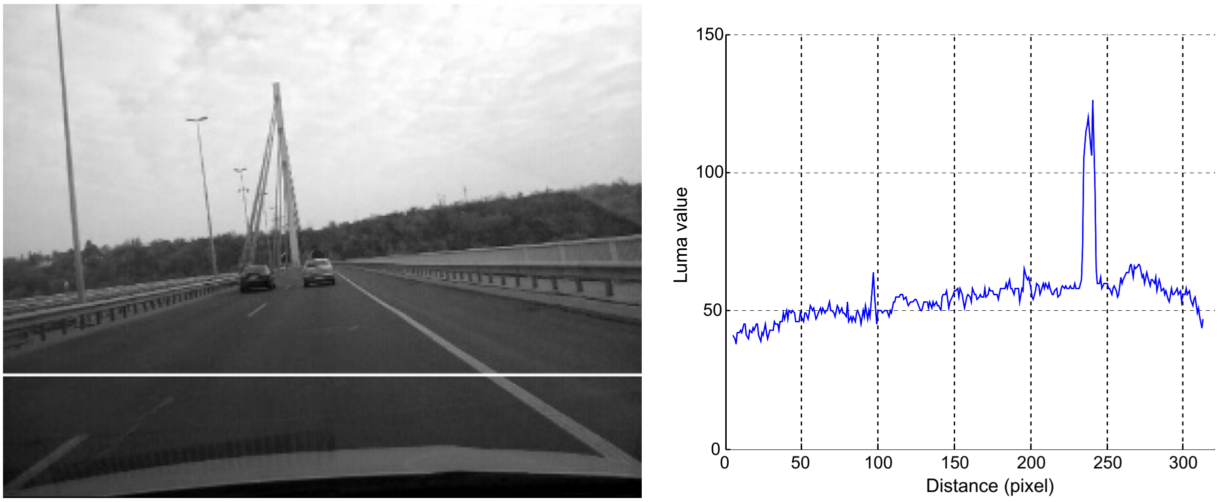


Fig. 2.1. Conceptual schematic of a driver assistance system integrated into a smart phone.



(a) 320x240 luma component of a road picture taken using a smart phone camera

(b) Plot profile of the white horizontal strip line shown in panel a

Fig. 2.2. Edge extraction stage.

Definition 2.1. *Fuzzy point* $P \in R^2$, denoted by $\widetilde{P}$ is defined by its membership function $\mu_{\widetilde{P}} \in \mathcal{F}^2$, where the set $\mathcal{F}^2$ contains all membership functions u: $R^2 \to [0,1]$ satisfying the following conditions:

(i) $(\forall\ u \in \mathcal{F}^2)(\exists_1 P \in \mathrm{R}^2)u(P) = 1$,
(ii) $(\forall\ X_1, X_2 \in R^2)(\lambda \in [0,1])u(\lambda X_1 + (1-\lambda)X_2) \geqslant \min(u(X_1), u(X_2))$,
(iii) function u is upper semi-continuous,
(iv) $[u]^\alpha = \{X | X \in R^2, u(X) \geqslant \alpha\}$ α-cut of function u is convex.

The point from R^2, with membership function $\mu_{\widetilde{P}}(P) = 1$, will be denoted by P (where P is the core of the fuzzy point $\widetilde{P}$), and the membership function of the point $\widetilde{P}$ will be denoted by $\mu_{\widetilde{P}}$. $[P]^\alpha$ denotes the α-cut of the fuzzy point (a set from R^2).

Definition 2.2. R^2 *Linear fuzzy space* is the set $\mathcal{H}^2 \subset \mathcal{F}^2$ of all functions which, in addition to the properties given in Definition 2.1, are:

(i) Symmetric against the core $S \in R^2$:

$$(\mu(S) = 1),$$
$$\mu(V) = \mu(M) \wedge \mu(M) \neq 0 \Rightarrow d(S,V) = d(S,M),$$

where $d(S,M)$ is the distance in R^2.

(ii) Inverse-linear decreasing w.r.t. points' distance from the core according to:
if $r \neq 0$

$$\mu_{\widetilde{S}}(V) = \max\left(0, 1 - \frac{d(S,V)}{|r_S|}\right),$$

if $r = 0$

$$\mu_{\widetilde{S}}(V) = \begin{cases} 1 & \text{if } S = V, \\ 0 & \text{if } S \neq V, \end{cases}$$

where $d(S,V)$ is the distance between the point V and the core S $(V,S \in R^n)$ and $r \in R$ is constant.

Elements of this space are represented as ordered pairs $\widetilde{S} = (S, r_S)$ where $S \in R^2$ is the core of $\widetilde{S}$, and $r_S \in R$ is the distance from the core for which the function value becomes 0; in the sequel parameter r_S will be denoted as the *fuzzy support radius*.

Definition 2.3. Let $\mathcal{H}^2$ be a linear fuzzy space. Then a function $f : \mathcal{H}^2 \times \mathcal{H}^2 \times [0,1] \to \mathcal{H}^2$ is a *linear combination* of fuzzy points $\widetilde{A}, \widetilde{B} \in \mathcal{H}^2$, given by

$$f\left(\widetilde{A}, \widetilde{B}, u\right) = \widetilde{A} + u \cdot \left(\widetilde{B} - \widetilde{A}\right),$$

where $u \in [0,1]$ and operator $\cdot$ corresponds to scalar multiplication of a fuzzy point, and +corresponds to fuzzy point addition.

If the points that represent the path are imprecise, then the whole line should be described similarly to the imprecise point's description. In this section we will present a mathematical model for such a fuzzy line.

Definition 2.4. Let $\mathcal{H}^2$ be a linear fuzzy space and function f a linear combination of fuzzy points $\widetilde{A}$ and $\widetilde{B}$. Then the fuzzy set $\widetilde{AB}$ is a *fuzzy line* if the following holds:

$$\widetilde{AB} = \bigcup_{u \in [0,1]} f\left(\widetilde{A}, \widetilde{B}, u\right).$$

Similar to a fuzzy point, fuzzy lines can be represented as a pair of two fuzzy points. A fuzzy line is a minimal extension of a precise line defined by two discrete points.

2.2. Spatial relations in R^2 linear fuzzy space

Spatial relations (predicates) are functions that are used to establish mutual relations between fuzzy geometric objects. The basic spatial relations are *coincide*, *between* and *collinear*. In this section, we present their definitions and basic properties.

The fuzzy relation *coincidence* expresses the degree of truth that two fuzzy points are located at the same place.

Definition 2.5. Let λ be the Lebesgue measure on the set $[0,1]$ and $\mathcal{H}^2$ a linear fuzzy space. The fuzzy relation $coin : \mathcal{H}^2 \times \mathcal{H}^2 \to [0,1]$ is the *fuzzy coincidence* represented by the following membership function:

$$\mu_{coin}\left(\widetilde{A}, \widetilde{B}\right) = \lambda\left(\left\{\alpha | \left[\widetilde{A}\right]^\alpha \cap \left[\widetilde{B}\right]^\alpha \neq \emptyset\right\}\right).$$

Proposition

"Fuzzy point $\widetilde{A}$ is coincident to fuzzy point $\widetilde{B}$"

is partially true with the truth degree $\mu_{coin}(\widetilde{A}, \widetilde{B})$.

The fuzzy relation *contains* or *between* is a measure that a fuzzy point belongs to fuzzy line or that a fuzzy line contains a fuzzy point.

Collinearity is another fundamental relation between three points in plane geometry. In the following, we present our definition of *fuzzy collinearity* in fuzzy linear space, as well as a method for its practical computation.

Definition 2.6. Let $\widetilde{A}, \widetilde{B}, \widetilde{C} \in \mathcal{H}^2$ be a fuzzy point defined in $\mathcal{H}^2$ linear fuzzy space and λ be a Lebesgue measure of the set [0,1]. The fuzzy relation *coli* : $\mathcal{H}^2 \times \mathcal{H}^2 \times \mathcal{H}^2 \to [0, 1]$ is then *fuzzy collinearity* between three fuzzy points, and is represented by the following membership function:

$$\mu_{coli}\left(\widetilde{A}, \widetilde{B}, \widetilde{C}\right) = \lambda\left\{\alpha | \exists u \in R \wedge \exists X \in \left[\widetilde{A}\right]^{\alpha} \wedge \exists Y \in \left[\widetilde{B}\right]^{\alpha} \wedge \exists Z \in \left[\widetilde{C}\right]^{\alpha} \wedge A = B + u(C - B)\right\}.$$

Proposition

"Fuzzy points $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{C}$ are collinear"

is partially true with the truth degree $\mu_{coli}(\widetilde{A}, \widetilde{B}, \widetilde{C})$.

Remark. The fuzzy relation *contains*for fuzzy line $\widetilde{AB}$ and a fuzzy point $\widetilde{C}$ means that fuzzy points $\widetilde{A}, \widetilde{B}$ and $\widetilde{C}$ are fuzzy collinear.

3. Road feature extraction

In this work, we propose a modified edge extraction method for road feature extraction. The best way to describe a road is to identify lane marks, which define lanes in almost any well-painted road. Usually, this is done by applying some well-known edge extraction method, such as the Canny and Sobel edge detector. Results from these methods usually contain relatively large sets of discrete points. Instead of using precise discrete points, we propose the use of fuzzy points. Also, we apply edge detection for a single line, rather than the more common 2D convolution of an image and operator mask. As can be seen in Fig. 2.3, lane markings are well separated from other features. Fig. 3.1 illustrates the selection process for a single line (row) from an image, and sequentially shifting accumulators A and B along the line.

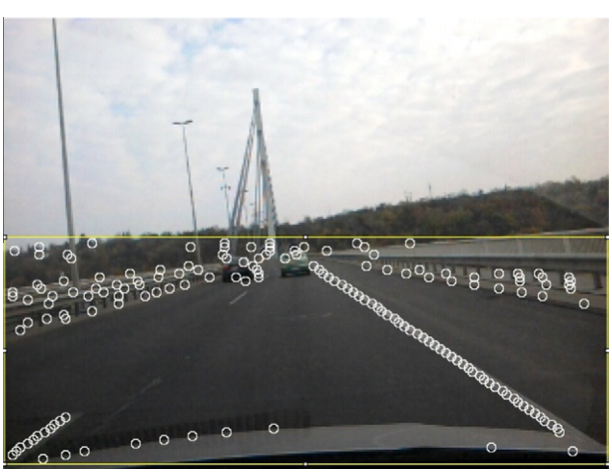


Fig. 2.3. Extracted set of fuzzy points. Each fuzzy point is represented as a white circle.

Two types of points are distinguished during the extraction process:

- Condition 0: $A > B$ and $|A - B| > Threshold$ and
- Condition 1: $A < B$ and $|A - B| > Threshold$

If Condition 0 or Condition 1 is satisfied then the point is a candidate feature point.

Parameters of the presented algorithm are given bellow:

Delta: distance between two analyzed rows.
R: number of pixels that are used in accumulators A and B.
Threshold: sensitivity.

The pseudo code of the road feature extraction algorithm is given in Listing 1:

The center of the extracted fuzzy point *xS* is a midpoint of interval satisfied by Condition 0 or Condition 1, as illustrated in Fig. 3.2.

Lane detection algorithms depend on the precision (position) of extracted road feature points. The approach presented in this paper enables modeling this imprecision using fuzzy set. A consequence is that lane detection algorithms based on fuzzy points are not as sensitive to the precise estimation of edge points. Example extracted fuzzy points are shown in Fig. 2.3.

4. Road lane detection

In this section, we describe a novel algorithm for fuzzy line detection from digital raster images (*FLDetector*). The algorithm is implemented as a modified fuzzy c-means algorithm for clustering sets of fuzzy points. Each fuzzy point belongs to a fuzzy line (cluster centroid) according to a fuzzy relation fuzzy collinear. This method allows one fuzzy point to belong to two or more fuzzy lines.

4.1. Initialization of the FLDetector algorithm

The first step of the algorithm is aimed at identifying initial groups of "*line like*" fuzzy connected sets of fuzzy points. Firstly, all fuzzy points are spatially indexed by a GRID or MESH spatial index structure, where the cell size is not lower than the size of maximal support for all fuzzy points. Then, for every fuzzy point, a set of fuzzy coincident points is determined. At the end of this step, initial clusters are created using information from fuzzy coincident points. In the "*line like*" fuzzy connected set of fuzzy points, "*most of them*" are simply connected with not more than 2 fuzzy points, and the sum of the distances between neighboring points is sufficiently close to the maximal distance between overall points in the set. The "*most of them*" statement is modeled as a fuzzy set defined by the percentage of the number of points that satisfies certain conditions. Fig. 4.1 illustrates two initial clusters of "*line like*" fuzzy connected sets of fuzzy sets.

The aim of this algorithm is to find for each fuzzy point a set of fuzzy coincident points. Fig. 4.2 gives an example of how fuzzy points are assigned to corresponding cells in GRID.

Pseudo code of this initialization algorithm is given in Listing 2. The first step is to fill the GRID with fuzzy points. After the GRID is filled with fuzzy points, for each nonempty cell in GRID we can find fuzzy coincident points only in neighbor cells, which is more efficient than to check each fuzzy point with all others. The result of this step is a graph of fuzzy points mutually connected by the fuzzy relation *coincidence*. Now, it is relatively easy to find all "*connected*" fuzzy points, using for example a breadth-search algorithm. Finally, the fuzzy filter "*line lake*" is used to create an initial set of candidate fuzzy lines.

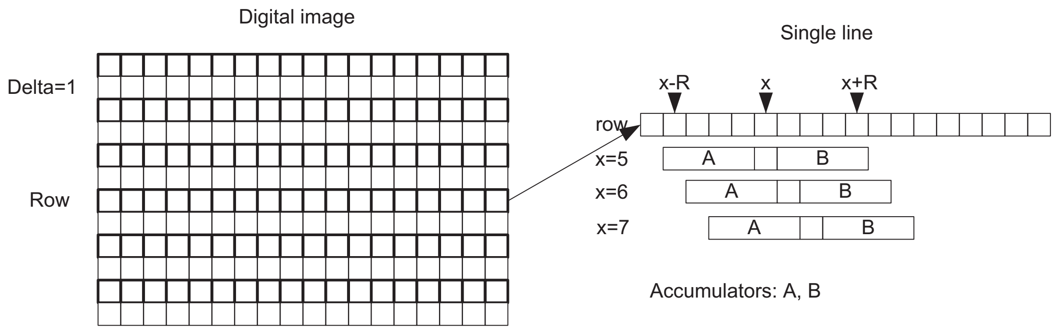


Fig. 3.1. Feature extraction process illustration.

```
for(y=ymin; y<yMax; y+=delta):
  for(x=R; x<width-R; x++):
    A -= row[x-R-1];
    A += rox[x-1];
    B -= row[x];
    B += row[x+R];
    if(A>B && abs(A-B)>THRESHOLD)
      support of fuzzy point(xS, y, delta+1); // type 0
    else if(A<B && abs(A-B)>THRESHOLD)
      support of fuzzy point(xS, y, delta+1); // type 1
```

Listing 1. Pseudo code for the road feature extracting algorithm.

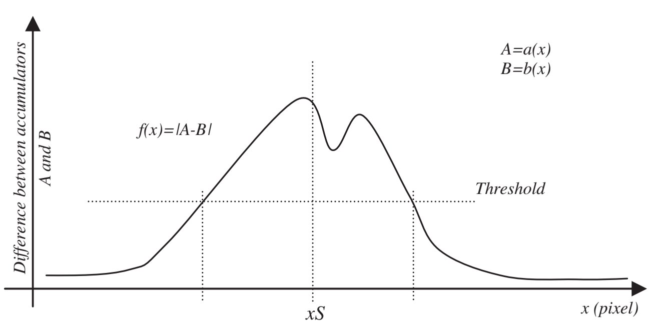


Fig. 3.2. Imprecision in edge detection.

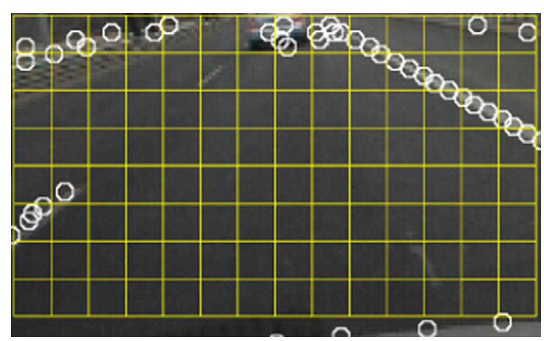


Fig. 4.2. Fuzzy point cell assignments.

The fuzzy filter "*line like*" extracts the set of points containing exactly two single connected fuzzy points (extreme points) and a finite number of *mostly double connected* fuzzy points (fuzzy points that are fuzzy coincident with not more than two other fuzzy points from the set, where the sum of the distances of all fuzzy coincident points from the set is sufficiently close to the distance between extreme points).

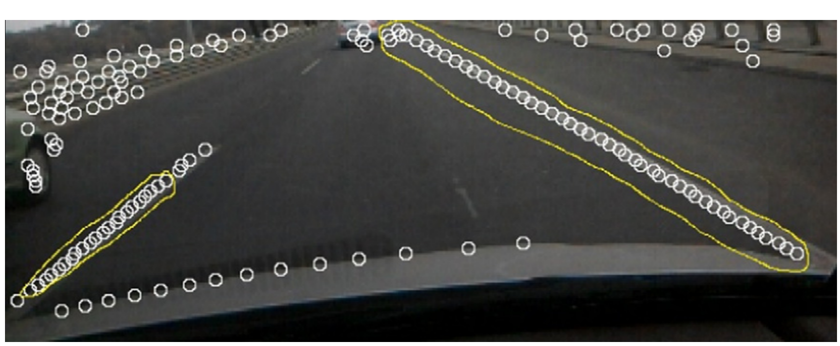


Fig. 4.1. Example of two initial "*line like*" fuzzy connected sets of fuzzy points.

Function: ***Initialization***$(\mathcal{T},\mathcal{L})$
Input: The set $\mathcal{T}=\{\tilde{S}_1,\dots\tilde{S}_n\}$ containing fuzzy points extracted from a raster image.
Output: The set $\mathcal{L}=\{(\tilde{A},\tilde{B})|\forall\tilde{A},\tilde{B}\in\mathcal{H}\}$ containing initial fuzzy lines.
Begin
Fill grid with input fuzzy points $\tilde{T}\in\mathcal{T}$.
Create a graph using the fuzzy relation *coincidence*.
Find *connected* sets of fuzzy points using a breadth search.
Keep only "*line like*" connected sets as $\mathfrak{L}$.
End

Listing 2. Pseudo code for the initialization algorithm.

Function: ***FuzzyLineDetector***$(\mathcal{T},\mathcal{L})$
Input: The set $\mathcal{T}=\{\tilde{S}_1,\dots\tilde{S}_n\}$ containing fuzzy points extracted from a raster image.
Output: The set $\mathcal{L}=\{(\tilde{A},\tilde{B})|\forall\tilde{A},\tilde{B}\in\mathcal{H}\}$ containing detected fuzzy lines.
Begin
1. *Initialization*$(\mathcal{T},\mathcal{L})$ - Create initial clusters of fuzzy collinear fuzzy points. Each clusters' centroid is a single fuzzy line.
2. Based on initial clusters, create an initial set of fuzzy lines $\mathfrak{L}$ such that each fuzzy line "*contains*" (Definition 2.6) all fuzzy points belonging to the cluster.
3. For each fuzzy point from $\mathcal{T}$ determine *fuzzy collinearity* (Definition 2.6) to each fuzzy line from $\mathfrak{L}$. After this step, new clusters are created, and a new centroid is determined by the least squares method.
4. For each cluster determine a new fuzzy line, such that each fuzzy line "*contains*" all fuzzy points belonging to the cluster. The set of newly created fuzzy lines is denoted as $\mathcal{L}_{new}$.
   a) If $\mathfrak{L}$ is sufficiently close to $\mathcal{L}_{new}$ then the result is $\mathfrak{L}$.
   b) Otherwise $\mathcal{L}=\mathcal{L}_{new}$ and go to step 3.

End

Listing 3. Pseudo code for the *FuzzyLineDetector* algorithm.

4.2. Fuzzy c-means based algorithm – FuzzyLineDetector

Now we can rearrange all fuzzy points according to a fuzzy relation fuzzy collinear and create new clusters of fuzzy collinear points. For each cluster, we compute a new centroid which is a fuzzy line that is fuzzy collinear with all fuzzy points from the cluster. The rearranging process is then repeated until two consecutive sets of fuzzy lines are sufficiently close (see Listing 3).

The computational complexity of the algorithms is proportional to the number of fuzzy points multiplied by the number of fuzzy lines.

5. Experiments

The proposed approach has been implemented as a Java class library and tested in two environments: (1) a Java application running on a PC using a data set; and (2) a Java application running on an Android 2.1 platform. The same set of 50 images taken by a smart phone camera was applied. In addition, our new approach was compared with the classical Hough transform using the same feature extraction algorithm. The aim of these experiments was to check the correctness of our proposed approach against a set of images taken in a real environment, and to check its suitability (i.e. is it fast enough) as a real time smart phone application.

5.1. Data set

The data set used in this experiment consists of 50 images taken by a smart phone camera. The overall properties of the data set are shown in Table 5.1.

Table 5.1
Properties of the data set.

Number of images	50
Resolution	320×240
Color model	YUV
Time of day	~15:00 (3 p.m.)

Table 5.2
Feature extraction parameters.

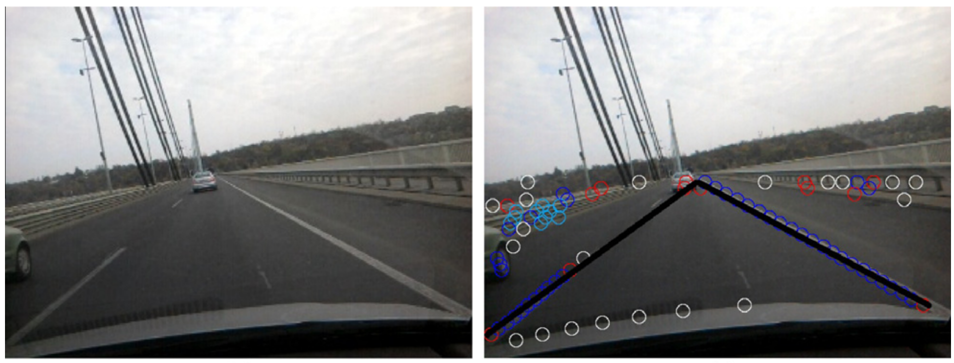
Gray scale image	Y luma component (0 – black 255 – white)
Region of interest	320×120 (*xMin* = 0; *yMin* = 120; *xMax* = 320; *yMax* = 320)
R	5
Threshold	$R*20$
Condition used in extraction	0 transition from dark to light
Delta	Distance between two possible consecutive line values (0,1,2,3)

Table 5.3
Hough line detection parameters.

maxTheta	Resolution of the angle calculation (180 corresponds to 1 degree and 360 corresponds to 0.5 degree between each line) How many discrete values of theta will be checked!
neighbourhoodSize	The size of the neighborhood in which to search for other local maxima. The default value is 4.
Threshold	The threshold percentage above which lines are determined from the Hough array (between 10 and 30)

```
initialize cosCache and sinCache
for each edge point fp
       for theta in (0, maxTheta)
              r = (fp.x − centerX) ∗ cos Cache[theta] + (fp.y − centerY) ∗ sin Cache[theta]
              houghArray[r][theta]++;
filter houghArray using Treshold
find local maxima in filtered houghArray using neighbourhoodSize
```

Listing 4. Pseudo code of the Hough line detection algorithm tested.



(a) (b)

Fig. 5.1. (a) Image #1 from data set, (b) results of experiment 1.

Table 5.4
Experiment parameters and obtained results.

Data set image	1
Number of extracted points	83
Delta	4
Threshold	15
Maxtheta	360
Mean execution time FUZZY	0.047848 ms
Mean execution time HOUGH	0.636239 ms
Quality FUZZY	0.461
Quality HOUGH	0

5.2. Feature extraction parameters

The feature extraction parameters used for the fuzzy and Hough approaches are shown in Table 5.2 (*Note:* the same parameters were used for both approaches).

5.3. Result quality evaluation method

For evaluation of the obtained results, we used formula (5.1) to measure the level of quality. Level of quality is a real number of interval [0, 1], where 0 corresponds to very poor quality and 1 corresponds to the best quality (all points are located on some line).

Let the extracted fuzzy points set contain n points and m lines (the jth line is denoted as $line_j$), and let *maxDistance* be the maximal distance between a single point and line; then the quality, Q, is calculated as:

$$Q = 1 - \frac{\sum_{i=1}^{n} \min_j(\mathrm{distance}(line_j, point_i))}{n \cdot maxDistance}. \tag{5.1}$$

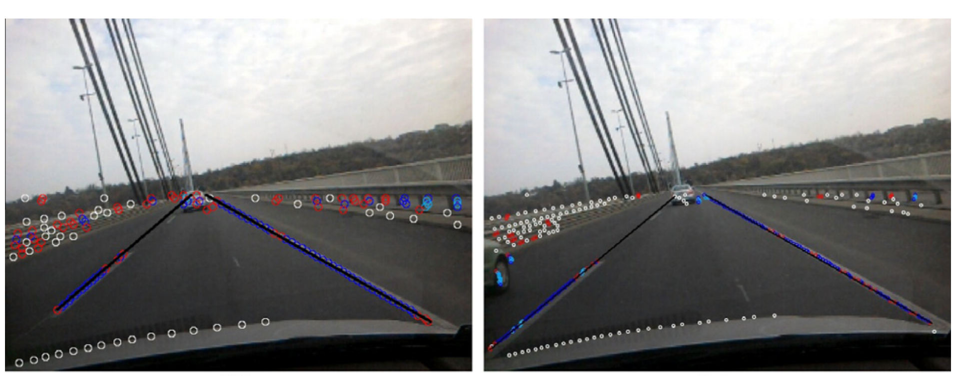
5.4. Hough transform parameters

The Hough line detection algorithm tested here is sensitive to the parameter values chosen. Hough line detection parameters are shown in Table 5.3.

Pseudo code for the Hough line detection algorithm [1] used is given in Listing 4.

5.5. PC Experiments

The following experiments were performed on a PC. Java applications were run on a Windows 7 64-bit Operating system, with



(a) (b)

Fig. 5.2. (a) Extracted points and lines for delta = 2, (b) extracted points and lines for delta = 1.

Table 5.5
Experimental parameters and results for delta = 2 and delta = 1.

Data set image	1	1
Number of extracted points	159	307
Delta	2	1
Threshold	15	30
Maxtheta	360	360
Mean execution time FUZZY	0.108905 ms	0.3311 ms
Mean execution time HOUGH	1.125577 ms	1.409117 ms
Quality FUZZY	0.49363884	0.51133020
Quality HOUGH	0.49594248	0.52376501

4 GB RAM and an Intel Core2Duo 3.0 GHz processor. Execution times were measured 1000 times because of the non-deterministic behavior of the host computer (multithread operating system).

Pseudo code for the experiment performed is given below.

```
for each image in data set and
      for delta in set {1,2,3,4}
            execute: Feature extraction
            for N = 1 to 1000
                    execute: Hough algorithm
                    execute: Fuzzy algorithm
            calculate: mean of execution times for
  Hough algorithm
            calculate: mean of execution times for
  Fuzzy algorithm
```

The experimental results are given in two parts: (1) several representative images with fixed parameters, and (2) a graph of overall execution times.

5.5.1. Experiment 1

In this representative experiment, we analyzed every 4th (delta = 4) line of the image shown in Fig. 5.1(a). Feature extraction time is small, and is related to the small amount of extracted points. Fig. 5.1 shows extracted points represented as circles.

Our proposed fuzzy algorithm performed 1000 times better than the Hough algorithm. Because of the small amount of extracted points, the Hough algorithm returned empty result, whereas the Fuzzy algorithm extracted two lines shown as black bold lines in Fig. 5.1(b). Our fuzzy algorithm processed this image (with 83 extracted points) almost 10 times faster than the Hough algorithm (see Table 5.4).

5.5.2. Experiment 2

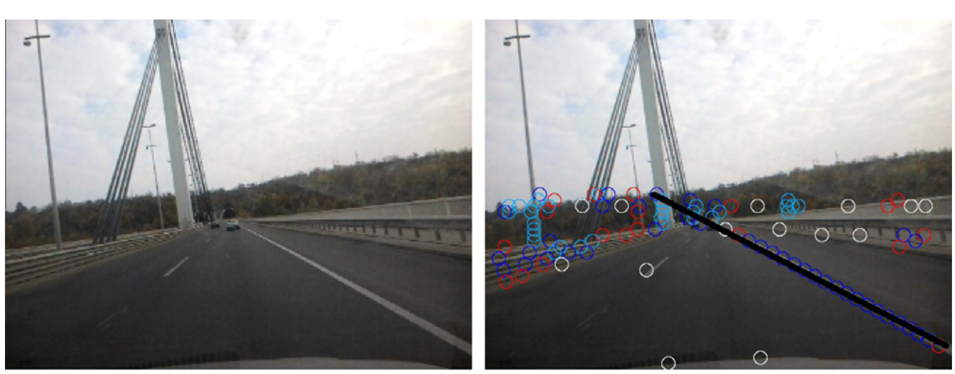
The second experiment was performed on the same image as in Experiment 1 (see Fig. 5.1(a)). However, in the feature extraction stage we analyzed every 2nd line (delta = 2) and every line (delta = 1). In Fig. 5.2, extracted points are represented as circles with half the radius of the fuzzy points extracted in Experiment 1. The qualities of both the Hough and Fuzzy algorithms in this example are almost the same. However, our Fuzzy algorithm was still faster than the Hough approach (see Table 5.5).

5.5.3. Experiment 3

Based on the previous two experiments, results are almost the same in cases where delta = 4, 2 or 1. However, for the case

(a) (b)

Fig. 5.3. (a) Image #10 from data set, (b) extracted lines for image #10 and delta = 4.



(a) (b)

Fig. 5.4. (a) Image #20 from data set, (b) extracted lines for image #20 and delta = 4.

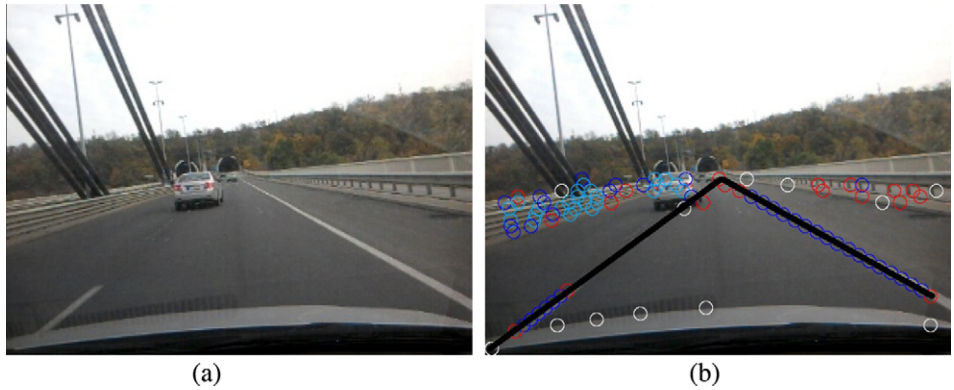


Fig. 5.5. (a) Image #30 from data set, (b) extracted lines for image #30 and delta = 4.

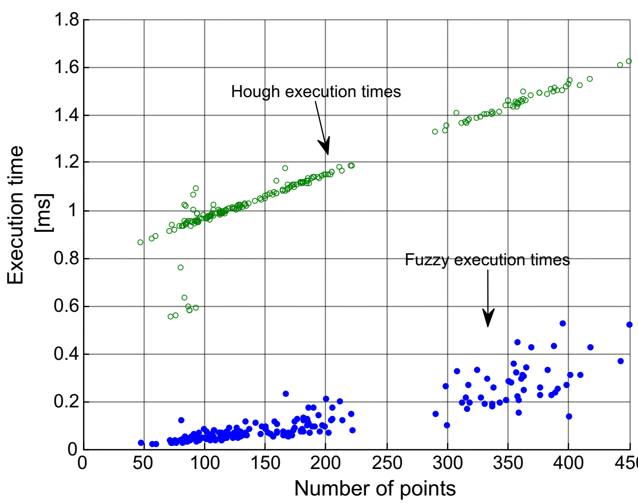


Fig. 5.6. PC execution times.

delta = 4, where we used a small number of points, execution times were extremely fast. In the following experiment, we attempted to extract road lanes using delta = 4 on several images (see Figs. 5.3–5.5).

5.5.4. Experiment 4

In this experiment, we used all images from the Data set and delta values between 1 and 4, to check the behavior of the both the Hough and Fuzzy algorithms with respect to the number of points.

As can be seen in Fig. 5.6, execution times for the Hough algorithm linearly increase with increasing data points, albeit with a nice dispersion. In contrast, our fuzzy algorithm is faster for a small number of points (<200). However, a wider dispersion was observed for the fuzzy algorithm for larger numbers of points (>200), indicating that fuzzy algorithm execution times could also depend on the position of points on an image, and the number of detected lines.

5.6. Smartphone experiments

The following experiments were performed on a Smartphone running Java on an Android 2.1 operating system, with 256 MB RAM and a 667 MHz processor. Due to the relatively slow processor (vs. a PC), execution times were measured 20 times (experiments run on a Smartphone for approximately 133 min $50 * 4 * 20 * 2$ s).

Pseudo code for the Smartphone experiment is shown below.

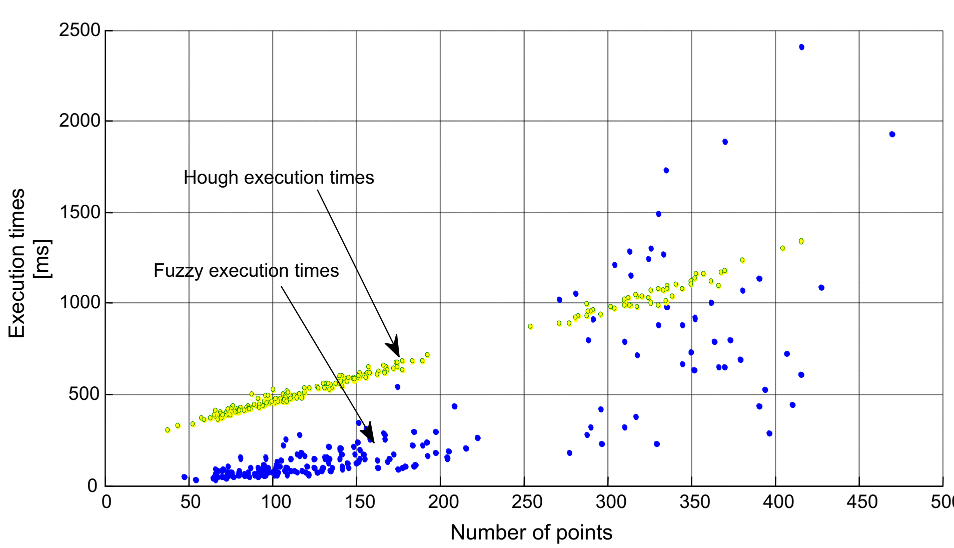


Fig. 5.7. Smartphone execution times.

```
for each image in data set and
    for delta in set {1,2,3,4}
        execute: Feature extraction
        for N = 1 to 20
            execute: Hough algorithm
            execute: Fuzzy algorithm
        calculate: mean of execution times for
 Hough algorithm
        calculate: mean of execution times for
 Fuzzy algorithm
```

The quality of the obtained results was the same as for the PC experiments. However, execution times were several times slower than on PC.

In the following example, we attempted to compare Smartphone results from both the Hough and Fuzzy algorithms, to determine suitable parameters (delta) for real-time applications of our proposed Fuzzy algorithm.

As can be seen in Fig. 5.7, for the case where we have 100 points the Hough algorithm runs in $\sim$500 ms, which is not suitable for real time applications. In addition, in cases with <100 points, the Hough algorithm is practically useless, and of poor quality. However, our fuzzy algorithm for acceptable quality works in almost 200 ms, which is approximately 4 frames per second. In cases with >300 points, both algorithms require longer than 1 s.

6. Conclusion

In this work, we present a model of imprecise road lanes, based on our previously published model of fuzzy imprecise points [13], as the union of a linear combination of two fuzzy points. Using this model, a fuzzy line can be represented with only two fuzzy points, providing a simple, yet descriptive extension of the precise ideal line. Imprecise spatial relations applied in this paper are based on fuzzy relations between fuzzy points and fuzzy lines, while the proposed algorithm for line detection is based on a modified fuzzy c-means clustering algorithm, along with the proposed data models and imprecise spatial relations. In addition to the ability to deal with imprecise data, the proposed algorithm is characterized by reduced computational complexity versus the standard Hough transformation. The algorithm's computational complexity is proportional to the number of fuzzy points multiplied by the number of fuzzy lines.

The proposed algorithm was tested on a data set consisting of 50 road pictures taken from the cockpit of an automobile using a smart phone camera. The last experiment presented in this paper shows that our proposed algorithm is applicable to real time lane detection using a 667 MHz CPU 256 MB RAM smartphone.

However, our experimental results indicate that the proposed algorithm is not well-suited for cases requiring larger numbers of feature points. To overcome this disadvantage for imprecise points and lines, new specialized fuzzy indexing structures analogous to *R* tree, Quad tree and GRID should be developed. In fact, this is one of the main research directions related to the development of fuzzy linear space-based algorithms. Furthermore, in order to improve the speed of lane detection, the introduced fuzzy relations could form a basis for lane tracking algorithms. Finally, another useful future research direction could be the development of an algorithm for vanishing point extraction and prediction.

Acknowledgments

Results presented in this paper are a part of research conducted within the Project "Technology Enhanced Learning Infrastructure in Serbia", Grant No. II 47003 of the Ministry of Education and Science of the Republic of Serbia and Provincial Secretariat for Science and Technological Development of Vojvodina. The third author was supported by Project Number: MPNRS 174009.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.knosys.2012.01.002.

References

[1] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1) (1972) 11–15.
[2] Qing Li, Nanning Zheng, Hong Cheng, Springrobot: a prototype autonomous vehicle and its algorithms for lane detection, IEEE Transactions on Intelligent Transportation Systems 5 (4) (2004) 300–308.
[3] B. Yu, A.K. Jain, Lane boundary detection using a multiresolution Hough transform, in: Proceedings of the International Conference on Image Processing, vol. 2, 1997, pp. 748–751.
[4] J.C. McCall, M.M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, IEEE Transactions on Intelligent Transportation Systems 7 (2006) 20–37.
[5] D.A. Pomerleau, Neural network vision for robot driving, The Handbook of Brain Theory and Neural Networks (1996) 161–181.
[6] D.-J. Kang, M.-H. Jung, Road lane segmentation using dynamic programming for active safety vehicles, Pattern Recognition Letters 24 (16) (2003) 3177–3185.
[7] C. Kreucher, S. Lakshmanan, K. Kluge, A driver warning system based on the LOIS LANE detection algorithm, in: Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles, 1998.
[8] C. Kreucher, S. Lakshmanan, LANA: a lane extraction algorithm that uses frequency domainfeatures, IEEE Transactions on Robotics and Automation 15 (2) (1999) 343–350.
[9] J.-G. Wang, C.-J. Lin, S.-M. Chen, Applying fuzzy method to vision-based lane detection and departure warning system, Expert Systems with Applications 37 (1) (2010) 113–126.
[10] K. Chen, W. Tsai, Vision-based autonomous land vehicle guidance in outdoor road environments using combined line and road following techniques, Journal of Robotic Systems 14 (10) (1997) 711–728.
[11] Yue Wang, Eam Khwang Teoh, Dinggang Shen, Lane detection using B-snake, in: Proceedings of the International Conference on Information Intelligence and Systems, 1999, pp. 438–443.
[12] L. Bai, Y. Wang, M. Fairhurst, An extended hyperbola model for road tracking for video-based personal navigation, Knowledge-Based Systems 21 (3) (2008) 265–272.
[13] Đ. Obradović, Z. Konjovic, E. Pap, N.M. Ralević, The maximal distance between imprecise point objects, Fuzzy Sets and Systems 170 (1) (2011) 76–94.
[14] L. Guibas, D. Salesin, J. Stolfi, Epsilon geometry: building robust algorithms from imprecise computations, in: Proceedings of the Fifth Annual Symposium on Computational Geometry, New York, NY, USA, 1989, pp. 208–217.
[15] M. Löffler, M. van Kreveld, Geometry with imprecise lines, in: 24th European Workshop on Computational Geometry, 2008.
[16] M. Löffler, Data Imprecision in Computational Geometry, Ph.D. Thesis, Utrecht University, Utrecht, 2009.
[17] M. Löffler, M. Kreveld, Largest and smallest convex hulls for imprecise points, Algorithmica 56 (2) (2008) 235–269.
[18] K. Yang, S. Sam Ge, H. He, Robust line detection using two-orthogonal direction image scanning, Computer Vision and Image Understanding 115 (8) (2011) 1207–1222.
[19] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognition 24 (4) (1991) 303–316.
[20] R.S. Stephens, Probabilistic approach to the Hough transform, Image and Vision Computing 9 (1) (1991) 66–71.
[21] C. Galambos, J. Matas, J. Kittler, Progressive probabilistic Hough transform for line detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 1999.
[22] J. Matas, C. Galambos, J. Kittler, Robust detection of lines using the progressive probabilistic Hough transform, Computer Vision and Image Understanding 78 (1) (2000) 119–137.
[23] C. Galambos, J. Kittler, J. Matas, Using gradient information to enhance the progressive probabilistic Hough transform, in: Proceedings 15th International Conference on Pattern Recognition, ICPR'2000, Barcelona, Spain, pp. 560–563.
[24] M. Schneider, Fuzzy topological predicates, their properties, and their integration into query languages, in: Proceedings of the 9th ACM international Symposium on Advances in Geographic Information Systems, 2001, pp. 9–14.
[25] A. Rosenfeld, The fuzzy geometry of image subsets, Pattern Recognition Letters 2 (5) (1984) 311–317.
[26] M. Schneider, Design and implementation of finite resolution crisp and fuzzy spatial objects, Data & Knowledge Engineering 44 (2003) 81–108.
[27] J.J. Buckley, E. Eslami, Fuzzy plane geometry I. Points and lines, Fuzzy Sets and Systems 86 (2) (1997) 179–187.

[28] M. Schneider, Spatial Data Types for Database Systems: Finite Resolution Geometry for Geographic Information Systems, first ed., Springer, 1997.
[29] Đ. Obradović, Z. Konjović, E. Pap, Extending PostGIS by imprecise point objects, in: IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 2010, pp. 23–28.
[30] J. Verstraete, G. Tré, R. Caluwe, A. Hallez, Field based methods for the modeling of fuzzy spatial data, Fuzzy Modeling with Spatial Information for Geographic Problems (2005) 41–69.
[31] S. Haibin, Computational models for computing fuzzy cardinal directional relations between regions, Knowledge-Based Systems 21 (7) (2008) 599–603.