We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK
CITATION
INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Objects Detection and Tracking Using Points Cloud Reconstructed from Linear Stereo Vision

Safaa Moqqaddem, Y. Ruichek, R. Touahni and A. Sbihi

Additional information is available at the end of the chapter

1. Introduction

Object detection and tracking is a key function for many applications like video surveillance, robotic, intelligent transportation systems, etc. This problem is widely treated in the literature in terms of sensors (video cameras, laser range finder, Radar) and methodologies. It is an important task within the field of computer vision, due to its promising applications in many areas. Computer vision is a discipline that tries to reproduce human vision by building models that have similar properties to visual perception. Among the domain of computer vision, stereo vision aims to find relief of a scene. More precisely it allows reconstructing, partially or fully, a 3D scene from two or more images taken under slightly different angles. The key step in a stereo process is matching primitives (pixels, segments, regions, etc.) extracted from the images. There are two broad classes of matching methods [1]. The first one includes the methods using pixel neighborhood correlation that produces a dense disparity map. The second class refers to the methods based on characteristics matching. In this case, the matching process yields to a sparse disparity map. In this work, we are particularly interested in edge points based stereo matching using linear images.

Since the 90s, automatic classification is becoming increasingly important in different areas of engineering sciences such as surveillance and diagnosis, treatment and analysis of signals and images. In the context of our clustering problem, the objective is to segment a cloud of 3D points to obtain classes of points where each class corresponds to an object. The difficulty is that no a priori knowledge on the distribution of 3D points is available and the number of classes is unknown. Hence, classical supervised clustering methods are not useful to achieve this task [2, 3]. To overcome this problem, many approaches have been proposed in the literature. In [4, 5], the authors proposed a method that proceeds with agglomeration partitioning, which considers as much points as isolated groups before eliminating iteratively irrelevant groups by minimizing an objective function until obtaining the correct

number of groups. Other authors proposed division based partitioning, which consists in creating a new group within the current partition, and then readjusts it until reaching an optimality criterion. The PDDP method (Principal Direction Divisive Partitioning), proposed by Boley [6], uses iteratively geometric properties of principal component analysis to divide the points cloud. We can also cite a clustering approach that combines K-means and SVM algorithms to discriminate burnt from unburnt areas [7, 8]. In this technique, the training set is defined automatically by K-means algorithm, which takes into account an entropic term to determine the optimal number of classes.

This chapter is concerned with obstacle detection and tracking in front of moving vehicles using linear cameras based stereo vision. Once the matching process is achieved, the geometric triangulation yields to a list of points represented in a 2D coordinate system of the 3D dimensional world, since linear stereo vision allows to reconstruct only horizontal and depth information[1, 9]. The objective is to segment these points to form clusters that represent objects in the scene. As indicated before, the problem is that there is no knowledge about the number of objects present in the scene. To overcome this problem, we propose a clustering method based on a spectral analysis of the points distribution. The principle is to construct a matrix representing the distance between the points. The spectral analysis consists in selecting significant eigenvalues of a transformed matrix. Different selection techniques are used and tested. The number of the significant eigenvalues corresponds to the number of clusters to be extracted from the reconstructed points. A K-means based clustering algorithm is then applied to extract the clusters that represent the objects present in the scene. The paper proposes also an objects tracking algorithm based on the geometric center of the obtained clusters. A simple Kalman filter is used to estimate the position of the objects. To associate the observations with the tracks a Nearest Neighbour based algorithm is used. The proposed approach is tested and evaluated using real stereo sequences, in the context of obstacle detection and tracking in front of a vehicle.

2. Methodology

Our proposed approach is composed of three principal phases: linear stereo vision, clustering, and tracking. The flowchart of figure 1 illustrates the whole steps of the proposed object detection and tracking approach.

3. Stereo vision with linear camera

Stereo vision is a popular technique for inferring the 3D position of objects seen simultaneously by two or more cameras from different viewpoints. Linear stereovision refers to the use of linear cameras providing line-images of the scene [10-12]. Indeed, the field of view of this type of cameras is reduced to a plane (see Figure 2). Therefore, the information to be processed is drastically reduced when compared to the use of classic video cameras. Furthermore, linear cameras have a better horizontal resolution than video cameras. This characteristic is very important for an accurate perception of the scene in front of a vehicle.
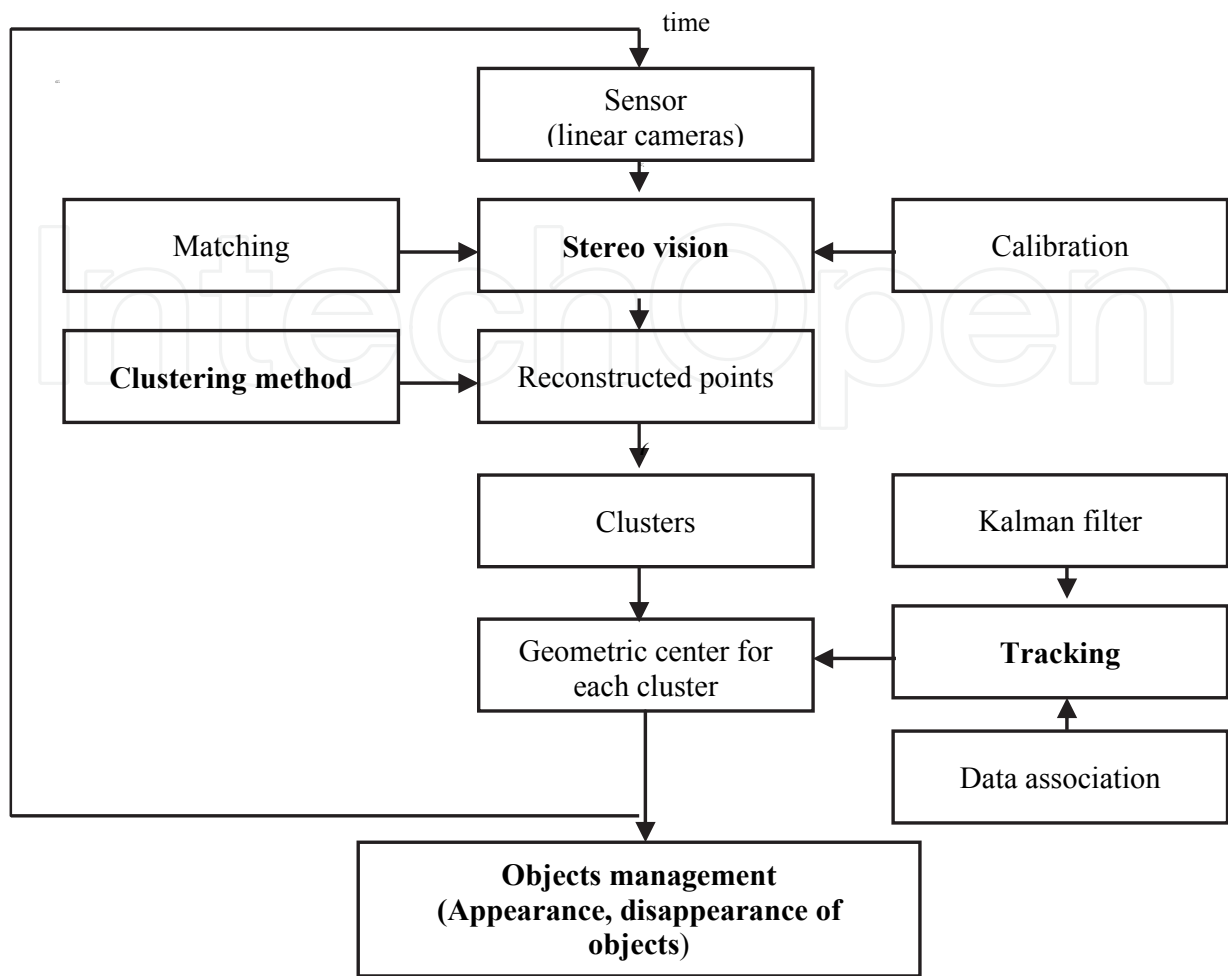
Figure 1. Overview of the proposed object detection and tracking approach.
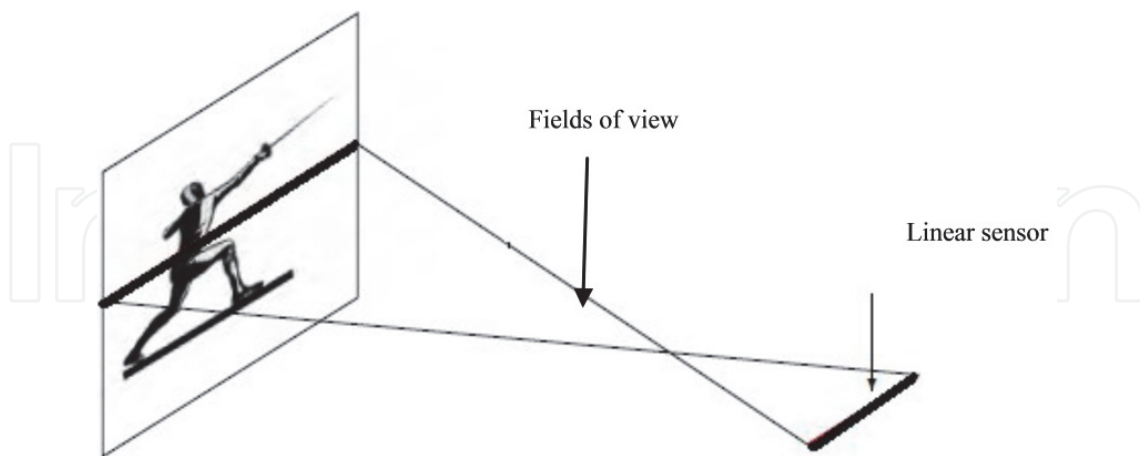

Figure 2. Linear camera

A linear stereo system is built with two line-scan cameras, so that their optical axes are parallel and separated by a distance E (see Figure 3). Their lenses have a same focal length f. The fields of view of the two cameras are merged in the same plane, called optical plane, so that the cameras shoot the same scene. A specific calibration procedure that takes into

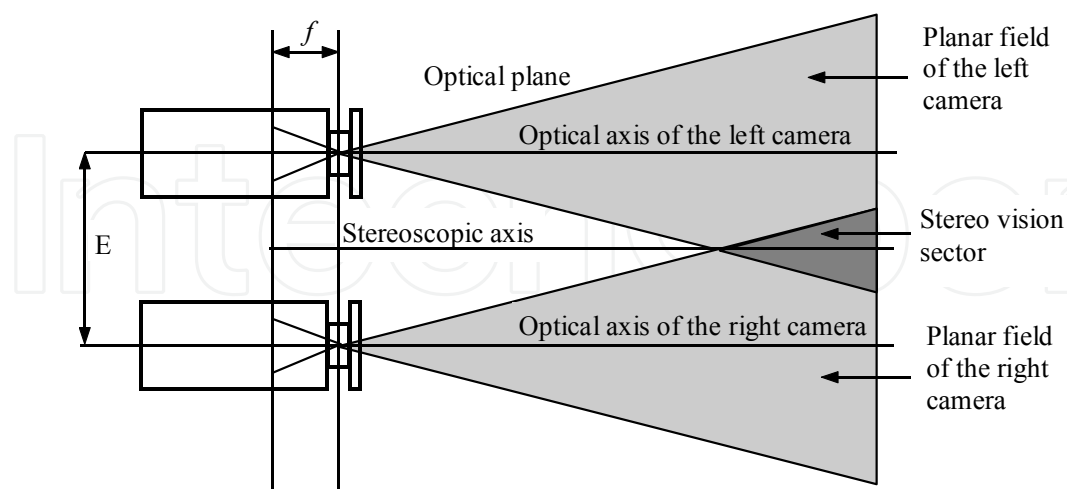account the fact that the line-scan cameras cannot provide the vertical information is developed in [11].

Figure 3. Geometry of the linear stereoscope

3.1. Feature extraction

The first step in stereo vision is to extract from each image the primitives to be matched. In classic video images, one can extract different types of primitives. In the case of linear images, the choice is restricted as a result of the onedimensional nature of the profile of a linear image. The only possibility in this case is to search for edge points corresponding to the frontiers of different objects present in the image (see Figure 4).
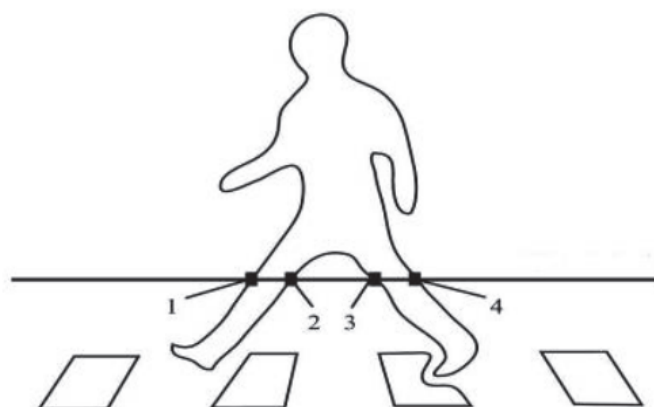

Figure 4. Type of primitives with linear images

The low-level processing of a couple of two stereo linear images yields the features required in the correspondence phase. Edges appearing in these simple images, which are one-dimensional signals, are valuable candidates for matching because large local variations in the gray-level function correspond to the boundaries of objects being observed in a scene.

Edge extraction is performed by means of the Deriche's operator and a technique that selects pertinent local extrema by splitting the gradient magnitude signal into adjacent intervals where the sign of the operator response remains constant [10]. In each interval of constant sign, the maximum amplitude indicates the position of a unique edge associated to this interval when, and only when, this amplitude is greater than a low threshold value (see Figure 5).
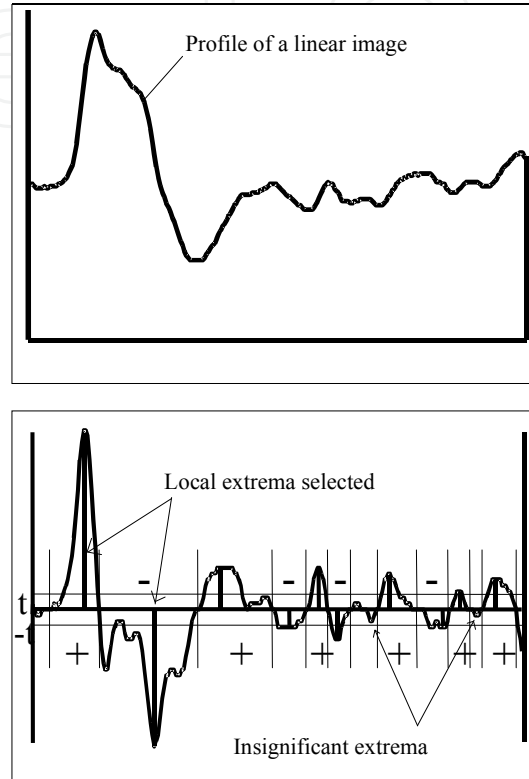

Figure 5. Extraction of edge points

Applied to the left and right linear images, this edge extraction procedure yields to two lists of edges, where each edge is characterized by its position in the image, the amplitude and the sign of the response of Deriche's operator.

3.2. Stereo matching

The edge stereo matching task can be viewed as a constraint satisfaction problem where the objective is to highlight a solution for which the matches are as compatible as possible with respect to specific constraints. Our approach for solving the stereo correspondence problem is based on two types of constraints: local constraints (position and slope constraints) and global ones (uniqueness, smoothness and ordering constraints). The local constraints are used to discard impossible matches so as to consider only potentially acceptable pairs of edges as candidates. Applied to the possible matches in order to highlight the best ones, the global constraints are formulated in terms of an objective function, which is defined so that the best matches correspond to its minimum value. A Hopfield neural network is then used to map the optimization process [10].

Once the matching process is achieved, a simple geometric triangulation allows obtaining for each matched edge pair a 2D point characterized by its horizontal position and depth. Line-scan cameras cannot provide the vertical information.

Let us define the base-line joining the perspective centers O_l and O_r as the X-axis, and let Z-axis lie in the optical plane, parallel to the optical axes of the cameras, so that the origin of the $\{X,Z\}$ coordinate system stands midway between the lens centers (see Figure 6). Let us consider a point $P(x_p, z_p)$ of coordinate x_p and z_p in the optical plane. The image coordinates x_l and x_r represent the projections of the point P in the left and right imaging sensors, respectively. This pair of points is referred to as a corresponding pair. Using the pinhole lens model, the coordinates of the point P in the optical plane can be found as:

$$Z_p = \frac{E.f}{d} \tag{1}$$

$$X_p = \frac{x_l.Z_p}{f} - \frac{E}{2} = \frac{x_r.Z_p}{f} + \frac{E}{2} \tag{2}$$

where f is the focal length of the lenses, E is the base-line width and $d = |x_l - x_r|$ is the disparity between the left and right projections of the point P on the two sensors.
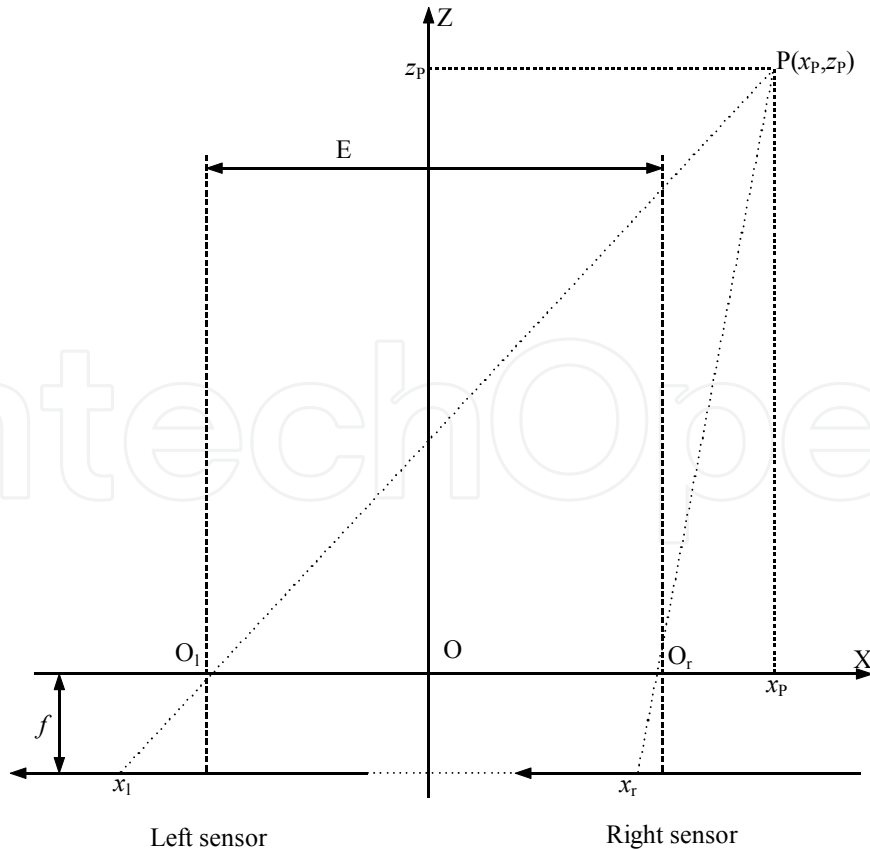

Figure 6. Pinhole model

4. Objects detection

Objects detection is an important and yet challenging vision task. It is a critical part in many applications such as image search and scene understanding. It is still an open problem due to the complexity of object classes and images. In this chapter, we are interested in detecting objects using a cloud of points reconstructed from linear stereovision. The proposed method is based on an unsupervised classification approach using spectral clustering.

4.1. Spectral clustering

Let us consider a list of points reconstructed from a pair of linear images. The objective is to cluster the points so that each cluster corresponds to an object of the scene. The difficulty is that no a priori knowledge on the distribution of the reconstructed points is available. Furthermore, the number of the clusters is unknown. Since classical deterministic classification techniques are not adapted, we propose to use a spectral learning based clustering approach [13, 14]. This approach allows also avoiding the problem of local minima inherent to the most part of classification methods. The principle of this approach is to perform spectral decomposition of a similarity matrix, constructed form the data to be clustered. The decomposition consists in extracting the eigenvectors of a transition matrix, calculated from the similarity matrix. The analysis of these eigenvectors can detect the different structures in the data to classify [15-17].

4.2. Spectral clustering algorithm

Consider a set of n points $L = \{P_1, \ldots \ldots P_n\}$ to be segmented in order to extract the clusters that correspond to the objects observed in the scene. A point P_i is characterized by its horizontal position and depth that are extracted from the linear stereovision process. The spectral clustering algorithm can be summarized as follows:

1. First, one must form a matrix A in R^{n*n}. Called the affinity matrix, this matrix represents the similarity between the point pairs. In our case, more the distance between two points is small more is high their similarity. Hence, the objective is to affect to the same cluster the points that are close each other in their representation space. The similarity can be represented by different forms: Cosine, Gaussian, or Fuzzy function [14]. In this paper, the Gaussian representation which generally the more used in the literature is adopted. The Gaussian similarity matrix is defined by equation (3)

$$A_{ij} = \exp\left(\frac{-d^2\left(P_i, P_j\right)}{\sigma^2}\right) \tag{3}$$

for i # j and $A_{ii} = 0$, where $d\left(P_i, P_j\right)$ is a distance function, which is often taken as the Euclidean distance between the points P_i and P_j, and σ is a scaling parameter which is further discussed in the next section.

2. Define a diagonal matrix D as $D_{ii} = \sum_j A_{ij}$.

3. Normalize the affinity matrix A to obtain a transition matrix N . We use the following normalization form (see Table 1):

$$N = D^{\frac{-1}{2}} A D^{\frac{-1}{2}} \qquad (4)$$

4. Form the matrix $X = [X_1, ..., X_k]$ in R^{n*k}, where $X_1, ..., X_k$ are the k eigenvectors of the matrix N, corresponding to the k significant eigenvalues $\lambda_1, ..., \lambda_k$.

5. Normalize the lines of the matrix X to have a unit module.

6. Consider each line of the matrix X as a point in R^k, and perform a classification using K-means algorithm with k classes.

7. Run M times the K-means algorithm and conserve the optimal partition for which the intra-class inertia is minimal, where M is the number of possible partitions.

8. Assign the point P_i to the class C_j if and only if the line X_i of the matrix X has been assigned to the class C_j.

Table 1 gathers different types of normalization forms applied to the affinity matrix.

Normalization	$f(A, D)$
Division	$N = D^{-1} A$
Symmetric division	$N = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$
Nothing	$N = A$
normalized additive	$N = \dfrac{(A + d_{max} I - D)}{d_{max}}$ $d_{max} = \max_i (D_{ii}) = \max_i (\sum_j A_{ij})$

Table 1. Different forms of the normalization function

The spectral clustering requires the adjustment of two parameters. The first one is the scaling parameter σ, which is used in the expression of the affinity matrix A. The second one is the number of classes k that corresponds to the k significant eigenvalues of the transition matrix N. The goal is to estimate automatically these two parameters, in order to make the clustering process as a nonparametric and unsupervised classification method.

4.3. Estimation of the scaling parameter σ

As expressed in equation (3), the performance of spectral clustering depends on the scaling parameter σ. Thus, choosing optimally the value of this parameter is an important issue. In [17], the authors suggested choosing σ automatically by running their clustering algorithm repeatedly for a number of values of σ and selecting the one providing less distorted

clusters of the rows of the matrix X constructed in step 4 of the clustering algorithm. In [19], the authors propose two selection strategies, manual and automatic. The first one relies on the distance histogram and helps finding a good global value for the parameter σ. The second strategy sets σ automatically to an individually different value for each point, thus resulting in an asymmetric affinity matrix. This selection strategy was originally motivated by no homogeneously dispersed clusters, but it provides also a very robust way for selecting σ in homogeneous cases.

In our case, we adopted the selection strategy proposed in [17] for its simplicity. For that, different values for σ^2 are taken to select the value that provides less distorted clusters of the row of the matrix X.

4.4. Estimation of the number of clusters k

The evaluation of the parameter k can be performed by analyzing the eigenvalues $\{\lambda_i\}$ or the eigenvectors $\{X_i\}$ of the matrix N [19]. In this work, we adopted an eigenvalues analysis. Theoretically, this analysis consists in considering the eigenvalues with a value equal to 1. In practice, significant eigenvalues have to be chosen by applying a thresholding procedure, i.e., eigenvalues that exceed a threshold are retained. We have chosen several forms of thresholding. One can consider also the difference between successive eigenvalues. The disadvantage of this strategy is that the jump between two successive eigenvalues can be big or small [20]. We tested this strategy in order to determine an empirical relationship. After various tests, we found that thresholding analysis gives the best results with a threshold λ_m, which is set to the average of the eigenvalues.

5. Objects tracking

Objects tracking in a sequence of images is a basic problem, but important in many computer vision applications. It consists in reconstructing the trajectory of objects along the sequence. This problem is inherently difficult, especially when unstructured forms are considered for tracking. It is also very difficult to build a dynamic model in advance, without a priori knowledge of objects motion.

5.1. Modeling

In this work, we are interested in tracking objects, where each object is represented by a cluster of points. We recall that the clusters are obtained by the spectral clustering algorithm described in section 4.2. To model moving objects, we consider the hypothesis that the displacement of an object, represented by a cluster of points, is modeled by the displacement of the geometric center of the points. We can therefore apply the fundamental principle of point dynamic to express the following equations:

$$x(t) = x(t - dt) + \dot{x}.dt + \frac{1}{2}.\ddot{x}.dt^2 \tag{5}$$

$$z(t) = z(t - dt) + \dot{z}.dt + \frac{1}{2}.\ddot{z}.dt^2 \tag{5}$$

Where x is the horizontal position and z is the depth of the geometric center of a cluster representing an object.

The most popular approach used for tracking mobile objects is based on Bayesian filters, especially Kalman Filters (KF) under a Gaussian noise assumption. KF is a tool for estimating object's state and smoothing its changes. In our case, KF is used with the Discrete White Noise Acceleration Model (DWNA) to describe object kinematics and process noise [21].

5.2. Kalman filter

Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that it minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown. Kalman filter addresses the general problem of estimating the state $S \in R^n$ of a discrete-time controlled process governed by a linear stochastic difference equation [22]. The discrete-time state equation with sampling period T is expressed as follows:

$$S(k + 1) = F \cdot S(k) + W(k + 1) \tag{7}$$

In this work, the state $S(k)$ is composed with the position and velocity of the geometric center of a cluster representing an object:

$$S(k) = [x \quad v_x \quad z \quad v_z]^t$$

The State Transition Matrix F is given by:

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The target acceleration is modeled as a white noise $W(k)$. The measurement model $Y \in R^m$ is given by:

$$Y(k) = H \cdot S(k) + V(k) \tag{8}$$

where H is the observation model:

$$H = \begin{bmatrix} 1000 \\ 0010 \end{bmatrix}$$

The random variables $W(k)$ and $V(k)$ represent the process and measurement noises, respectively. They are assumed to be independent, white, and with normal probability distributions:

$$P(W) \sim N(0,Q)$$
$$P(V) \sim N(0,R)$$

(9)

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement. In this paper, we assume that they are constant.

Kalman filter can be written as a single equation. However, it is most often conceptualized as two distinct phases: Prediction phase and updating phase (see Figure 7). The prediction phase uses the state estimated from the previous time step to produce an estimate of the state at the current time step. The predicted state estimate is known as the a priori state estimate, because although it is an estimate of the state at the current time step, it does not include observation information from the current time step. In the updating phase, the current a priori prediction is combined with the current observation information to refine the state estimate. This improved estimate is known as the a posteriori state estimate.
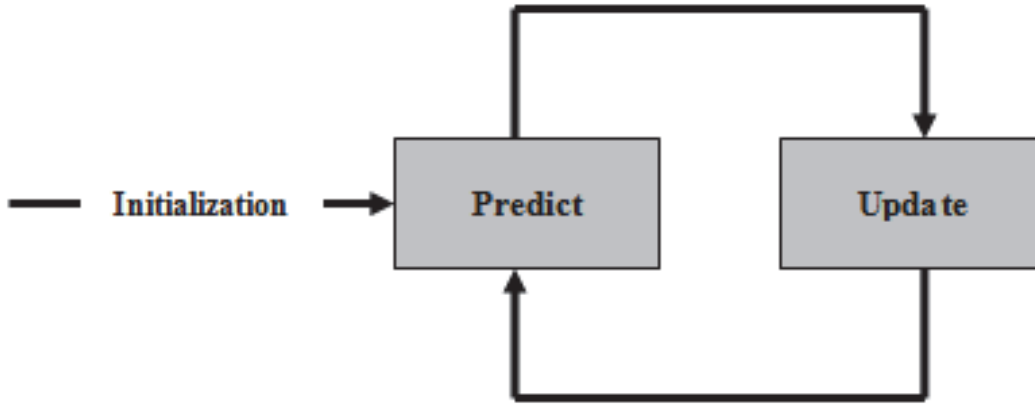

Figure 7. Stages of Kalman Filter

For multiple tracking, the problem of data association must be handled. The proposed data association algorithm is presented in the section 5.4.

5.3. Kalman filter algorithm

• **Initialisation**

$$S_{apos}^i(k-1), P_{apos}^i(k-1), R^i = Q^i = P_{apos}^i(k-1)$$

• **Prediction**

$$S_{apr}^i(k) = F \cdot S_{apos}^i(k-1)$$

(10)

$$P_{apr}^i(k) = F \cdot P_{apos}^i(k-1) \cdot F^T + Q^i \tag{11}$$

- Updating

$$Y_{apr}^i = H \cdot S_{apr}^i \tag{12}$$

$$Res(k) = Y^i(k) - H \cdot S_{apr}^i \tag{13}$$

$$C(k) = H \cdot P_{apr}^i(k) \cdot H^T + R^i \tag{14}$$

$$K^i(k) = P_{apr}^i(k) \cdot H^T \cdot (C(k))^{-1} \tag{15}$$

$$S_{apos}^i(k) = S_{apr}^i(k) + K^i(k) \cdot Res^i(k) \tag{16}$$

$$P_{apos}^i(k) = (1 - K^i(k) \cdot H) \cdot P_{apr}^i \tag{17}$$

where:

S_{apr} is the a priori state estimate; P_{apos} is the a priori estimate error covariance

S_{apos} is the a posteriori state estimate; P_{apos} is the a posteriori estimate error covariance

Y_{apr} is the predicted measurement ; Res is the measurement innovation, or the residual.

C is the innovation covariance; K is the filter gain

Y is the sensor measurement; i corresponds to the ith geometric center to track.

5.4. Data association

Once the prediction step is achieved, one must perform data association between predicted objects and observed ones from measurements provided by the sensor. Data association is a problem of great importance part for multiple target tracking applications. In this section, we describe a method of data association for tracking multiple objects where the number of objects is unknown and varies during tracking.

In the literature, there are many data association algorithms such as Nearest-Neighbour (NN), Probabilistic Data Association (PDA), Joint PDA (JPDA) and multiple hypotheses tracking (MHT) [23, 24]. In this paper, we used the Nearest Neighbour (NN) method, which is simple to implement: for each new set of observations, the goal is to find the most Mahalanobis distance based likely association between an observation and an existing track, otherwise between a new observation and the new track assumption. In our case, we are interesting to track the geometric centers of the obtained clusters representing the objects in the scene.

Mahalanobis distance is defined by:

$$d_m^2(Y,Y_{apr}) = \frac{1}{2}(Y - Y_{apr})^T * C^{-1} * (Y - Y_{apr}) \qquad (18)$$

where:

C is the covariance matrix of Res, which is the measurement innovation (see Equation 14).

Y_{apr} is the predicted measurement (see Equation 12).

Y is the measurement provided by the sensor.

The Mahalanobis distance is a statistical distance that takes into account the covariance and correlation of the elements of the state vector, and is appropriate to solve the data association problem. In our case, the covariance and correlation are determined between the measurements provided by the sensor and the predicted measurement given by Kalman filter.

The first step for data association is to define a search area for candidate points to the association. The size of searching area, which must be defined for each geometric center representing an object, depends on the movement of the object. Uncertainty about the movement defines the search area taken as a circle. Let G_k^i be the searching circle of the predicted object i at time k. The ray of this searching circle is defined by equation (19).

$$ray(G_k^i) = \Delta v(x,z) \qquad (19)$$

where $\Delta v(x,z)$ is the difference between the velocities at times k and $k - 1$.

The data association process is first applied considering the horizontal position x. The results are then validated by the data association process with the depth z.

5.5. Temporal constraint

Tracking requires information about the past of the objects. Indeed, when an object appears for the first time, one cannot decide reliably if the object is real or corresponds to a wrong detection considering that the sensor can generate false detection (i.e the observation does not match any known object). To make objects tracking more robust, an object must be detected and tracked during a sufficient long period in order to assess objects appearance and disappearance. This temporal constraint will allow ignoring objects generated erroneously from the stereo matching process. The temporal constraint consists in associating a minimum lifetime to each object [12]. In our case, we set the minimum lifetime to 5 successive detections: when an object is not detected during 5 successive frames, we estimate that it disappears.

5.6. Fusion of objects

The spectral clustering may sometimes produce two or more distinct objects that represent in reality a single object. Indeed, points representing the same object may be segmented onto two or more clusters of points. To resolve this problem, we propose a clusters fusion technique based on a clusters overlapping strategy. The fusion technique consists in determining an overlapping coefficient, defined as follows:

$$T_C = \frac{dist(o_i, o_j)}{r_i + r_j} \tag{20}$$

with:

o_i and o_j are respectively the geometric centers of the clusters i and j, candidates for a possible fusion.

$dist(o_i, o_j)$ is the Euclidean distance between the geometric centers o_i and o_j.

r_i and r_j are respectively the rays of the search areas of the two tracks i and j. The rays r_i and r_j are determined in the data association step. The ray r_i is calculated as the difference between the estimated (KF-based) and real (observation-based) positions. When the overlapping coefficient T_C is greater than a threshold, the considered clusters are merged. In this work, the overlapping threshold is set experimentally to 0.5.

6. Results and discussion

Our approach is tested and evaluated for obstacle detection and tracking in front a vehicle. The line-scan cameras based stereo set-up (see Figure 8) is installed on top of a car for periodically acquiring stereo pairs of linear images as the car travels (see Figures 9 and 10) [11, 12]. The tilt angle is adjusted so that the optical plane intersects the pavement at a given distance D_{max} in front of the car. The cameras have a sensor width of 22.1 mm, a focal length of 100 mm and deliver images with resolution of 1728 pixels. Within the stereo setup, the cameras are separated by a distance $E = 1m$.

Figure 11 represents a stereo sequence, in which the linear images are represented as horizontal lines, time running from top to bottom. The pedestrian travels in front of the car according to the trajectory shown in (Figure 12). On the images of the stereo sequence, we can clearly see the white lines of the pavement. The shadow of a car, located out of the vision field of the stereoscope, is visible on the right of the images as a black area.

The disparities of all matched edges are used to compute the positions and distances of the edges of the objects seen in the stereo vision sector. The results are shown in (Figure 13), in which the distances are represented in grey levels, the darker the closer, whereas positions are represented along the horizontal axis. As in (Figure 11), time runs from top to bottom.
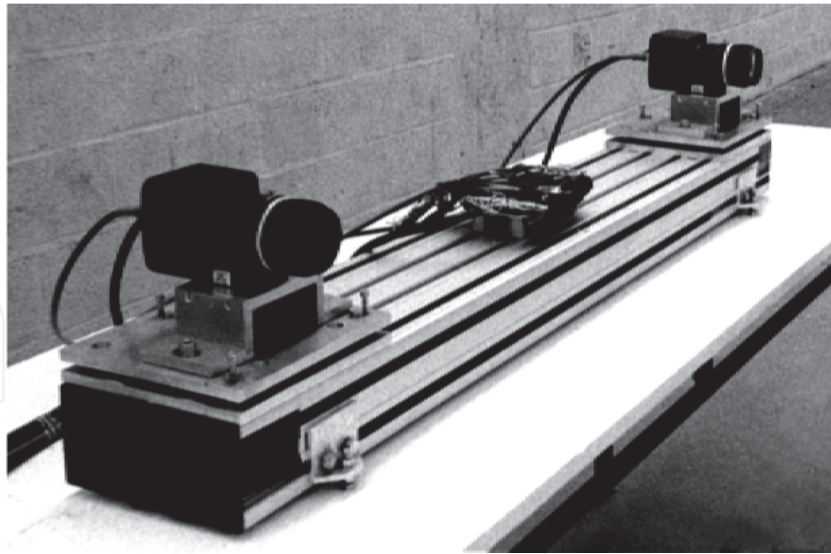
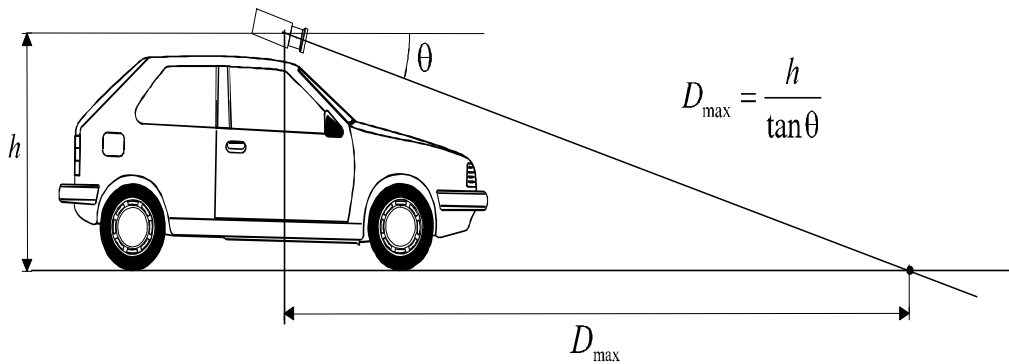**Figure 8.** Linear cameras composing the stereoscope.

$$D_{max} = \frac{h}{\tan\theta}$$
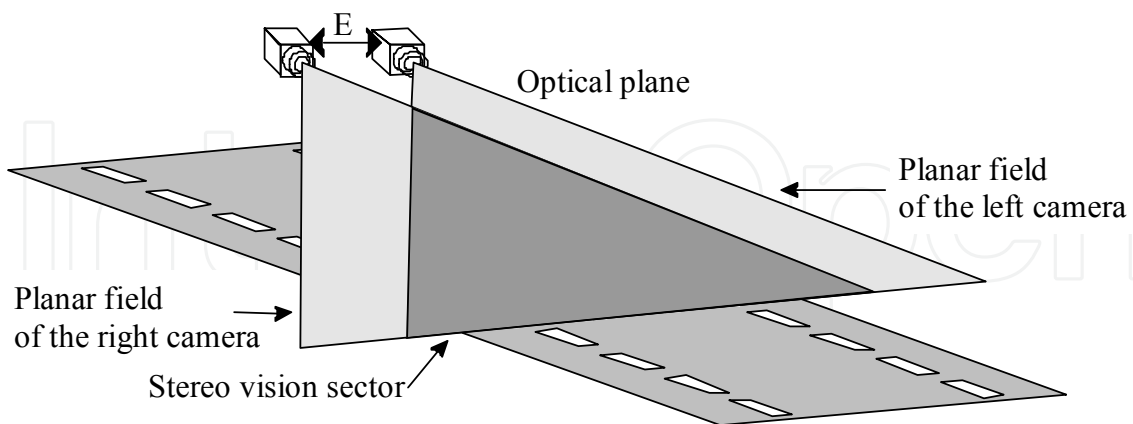
Figure 9. Stereo set-up, side view

Figure 10. Stereo set-up, top view

The clustering stage is performed on the reconstructed points for each pair of stereo linear images. The tracking process is applied to the geometric center of the obtained clusters representing the objects in the scene. The results are illustrated in (Figures 14, 15 and 16), time runs from top to bottom.
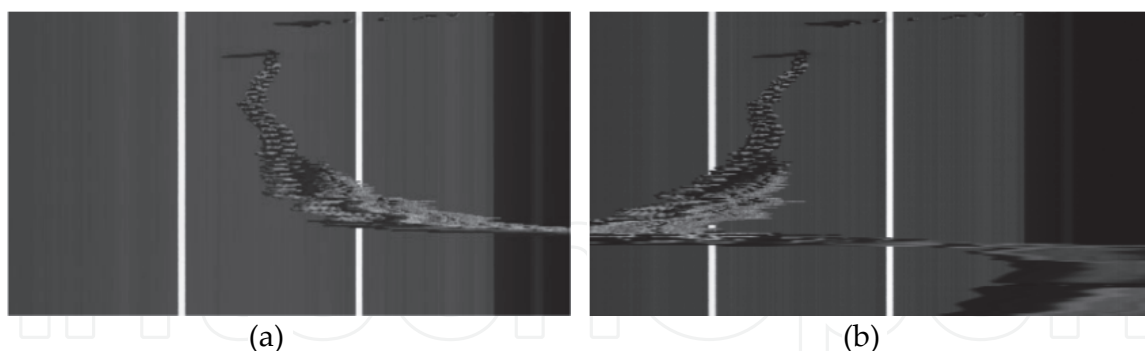
Figure 11. Stereo sequence (pedestrian) a- Left sequence b-Right sequence

The detected and tracked objects are labelled as follows: white lines in blue (with crucifix), shadow transition in black (with crucifix), and the pedestrian in purple (with star), red (with square) and black (with square). One can see that all the objects are detected and tracked correctly. Some errors are identified, especially when occlusions occur at the end of the sequence, i.e., when the pedestrian hides one of the white lines to the left or right camera. These errors are caused by matching the edges of the white line, seen by one of the cameras, with those representing the pedestrian. These errors effect the clustering task and hence the tracking process. Some of these errors could be removed by exploiting the tracking results in the matching procedure. As mentioned before, the clustering process may provide two or more clusters for the same object. This situation occurs when the number of clusters is over estimated by the spectral analysis. In (Figure 14), one can see that this situation occurs for the detection of the pedestrian. To solve this problem, the proposed clusters fusion strategy is applied. The results are illustrated in (Figure 15) in which all of the clusters representing the pedestrian are merged.
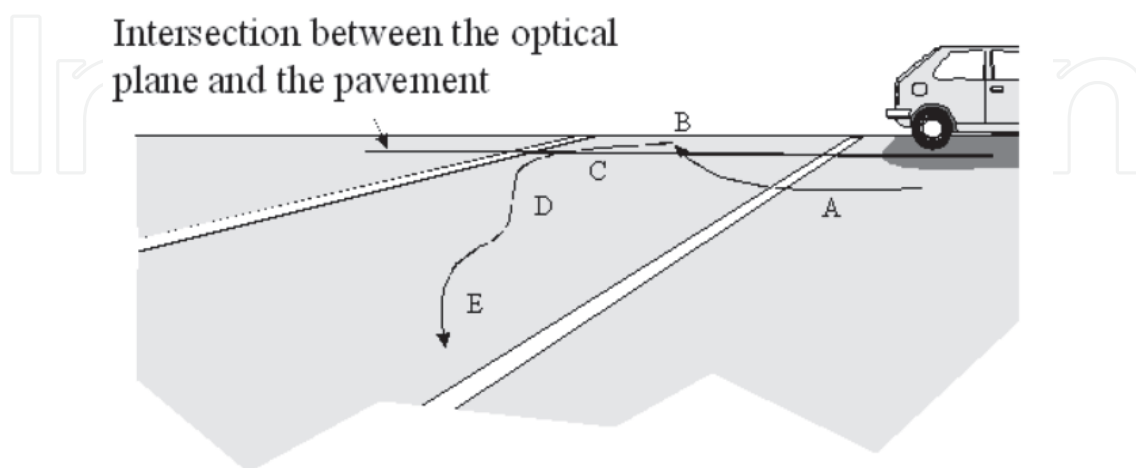

Figure 12. Trajectory of the pedestrian during the sequence

Figure 16 shows the evolution of the detected and tracked objects according the horizontal position x and depth z. In this figure, one can see that the position and depth of the white lines (crucifix in blue) and shadow transition (crucifix in black) is stable. The figure illustrates also the reconstructed trajectory of the pedestrian (stars in purple, and squares in red and black).

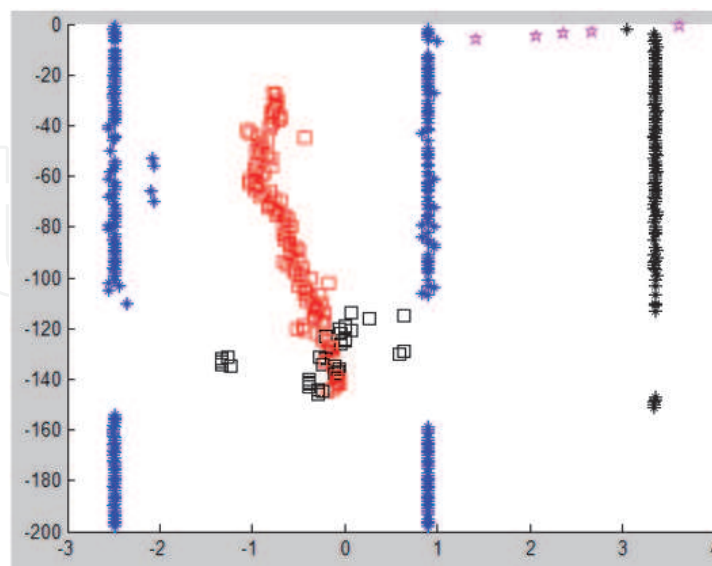**Figure 13.** Image reconstruction of the stereo sequence pedestrian

Figure 14. Objects detection and tracking (plot of the horizontal position x when time runs from top to the bottom)
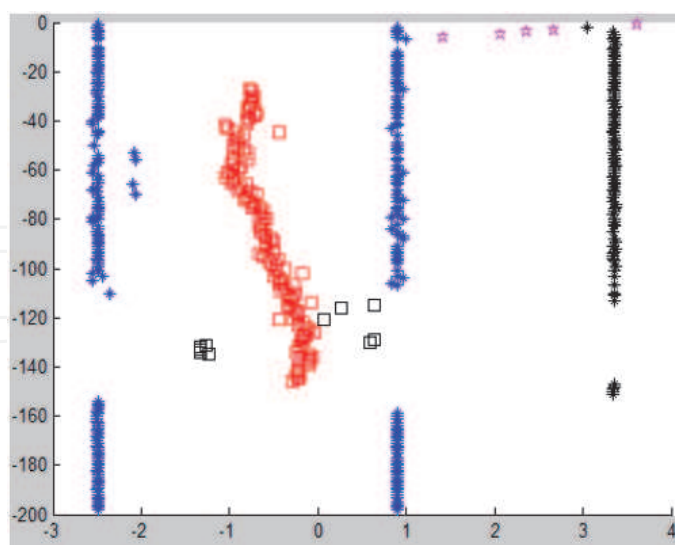
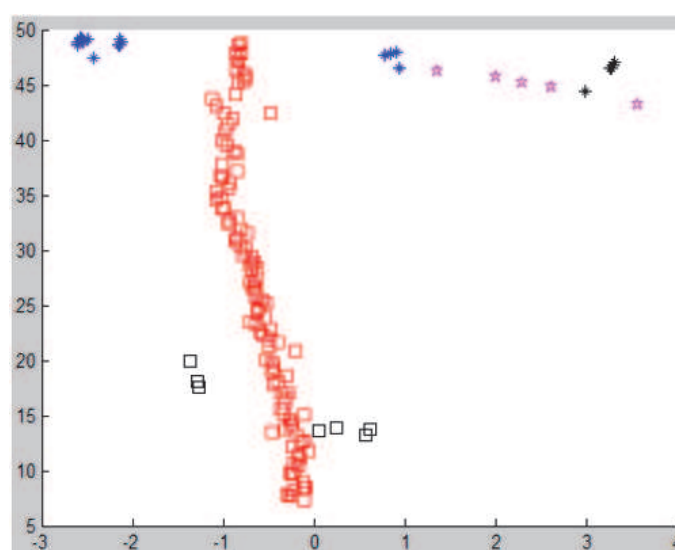**Figure 15.** Objects detection and tracking with the data fusion strategy

Figure 16. Objects detection and tracking with the data fusion strategy (plot of the horizontal position x and depth z)

7. Conclusion

A method for detecting and tracking objects using linear stereo vision is presented. After reconstructing 3D points from the matching edge points extracted from stereo linear images, a clustering algorithm based on a spectral analysis is proposed to extract clusters of points where each cluster represents an object of the observed scene. The tracking process is achieved using Kalman filter algorithm and nearest neighbour data association. A fusion strategy is also proposed to resolve the problem of multiple clusters that represent a same object. The proposed method is tested with real data in the context of objects detection and tracking in front of a vehicle.

Author details

Safaa Moqqaddem
Systems and Transportation Laboratory, University of Technology of Belfort-Montbéliard,
Belfort, France
LASTID Laboratory, Ibn Tofail University of Kénitra, Morocco

Y. Ruichek
Systems and Transportation Laboratory, University of Technology of Belfort-Montbéliard,
Belfort, France

R. Touahni and A. Sbihi
LASTID Laboratory, Ibn Tofail University of Kénitra, Morocco

Acknowledgement

8. References

[1] Banks, Jasmine Elizabeth, Bennamoun, Mohammed, Kubik, Kurt, & Corke, Peter . "A taxonomy of image matching techniques for stereo vision". Queensland University of Technology, Brisbane. 1997

[2] F.Mrabti, H.Seridi. "Comparaison de méthodes de classification réseau RBF, MLP et RVFLNN". Damascus University Journal Vol. (25) - No. (2) 2009.

[3] Teuvo Kohonen. "Self-organizing maps". Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.

[4] H. Frigui et R. Krishnapuram. "Clustering by competitive agglomeration". Pattern Recognition Journal, 30(7) :1109–1119, 1997.

[5] Bertrand Le Saux et Nozha Boujemaa. " Image database clustering with svm-based class personalization". Conference on Storage and Retrieval Methods and Applications for Multimedia / Electronic Imaging symposium (SPIE '04), San Jose, CA, USA, Janvier 2004.

[6] Daniel Boley. " Principal direction divisive partitioning". Data Min. Knowl. Discov., 2(4) :325–344, 1998.

[7] O.Zammit, X.Descombes et J.Zerubia. "Apprentissage non supervisé des SVM par un algorithme des K-moyennes entropique pour la détection de zones brûlées". Colloque GRETSI Groupe d'Etudes du Traitement du Signal et des Images, 11-14 septembre 2007, Troyes.

[8] G. Palubinkas, X. Descombes et F. Kruggel. "An un- supervised clustering method using the entropy minimization". IEEE International Conference on Pattern Recognition, Brisbane, Australie, 1998.

[9] Sergio Nogueira, Yassine Ruichek and François Charpillet " A Self Navigation Technique using Stereovision Analysis ", 295-306p, stereo vision. Edited by Dr. Asim Bhatti.

[10] Ruichek, Y., Hariti, M., and Issa, H., "Global techniques for edge based stereo matching", Scene Reconstruction Pose Estimation and Tracking, Rustam Stolkin (Ed.), I-Tech Education and Publishing, Austria, pp 383–410, 2007.

[11] Bruyelle, J. L, "Conception et réalisation d'un dispositif de prise de vue stéréoscopique linéaire– Application à la détection d'obstacles à l'avant des véhicules routiers", Phd thesis, Université des Sciences et Technologies de Lille, France 1994.

[12] Burie, J. C., Bruyelle, J. L., and Postaire, J. G., "Detecting and localising obstacles in front of a moving vehicle using linear stereo vision", Mathematical and Computer Modeling 22(4–7), 235–246, 1995.

[13] Kamvar, S. D., Klein, D., and Manning, C. D., "Spectral learning" Proc. International Joint Conference on Artificial Intelligence, 2003.

[14] Francis R. Bach and Michael I.Jordan." Learning Spectral Clustering". Report No.UCB/CSD-03-1249. June 2003.

[15] Lihi Zelnik-Manor and Pietro Perona, "Self-Tuning Spectral Clustering", Advances in Neural Information Processing Systems 17, pp 1601-1608, 2004.

[16] Weiss, Y., "Segmentation using eigenvectors: a unifying view", Proc. IEEE International Conference on Computer Vision, pp 975-982, 1999.

[17] Ng A. Y., Jordan, M. I., and Weiss, Y. "On spectral clustering: Analysis and an algorithm". Advances in Neural Information Processing Systems 14, Cambridge, MA. MIT Press, 2002.

[18] Verma D. and Meila M. "A comparison of spectral clustering algorithms", Technical report uw-cse-03-05-01, university of washington, 2003.

[19] Sanguinetti G., Laidler J. and Neil L. "Automatic determination of the number of clusters using spectral algorithms", In IEEE Machine Learning for Signal Processing 2005, 28-30 Sept 2005, Mystic, Connecticut, USA.

[20] Inderjit Dhillon, Yuqiang Guan, Brian Kulis, "Kernel k-means, Spectral Clustering and Normalized Cuts", KDD'04, August 22–25, Seattle, Washinton, USA, 2004.

[21] Bar-Shalom, Y., Li, X., and Kirubarajan, T., "Estimation with applications to tracking and navigation", Wiley, New York, chapter 6, 2001.

[22] Arnaud, E., "Méthodes de filtrage pour du suivi dans des séquences d'images - Application au suivi de points caractéristiques", Phd thesis, Université de Rennes I, France ,2004.

[23] Jaco Vermaak, Simon J. Godsill and Patrick Pérez. "Monte Carlo Filtering for Multi-Target Tracking and Data Association". DRAFT, September 22, 2004

[24] Coué, C., "Modèle bayésien pour l'analyse multimodale d'environnements dynamiques et encombrés : Application à l'assistance à la conduite en milieu urbain, " Phd thesis, Institutational Polytechnique de Grenoble, France, 2003.