Tracking of Multiple objects Using 3D Scatter Plot Reconstructed by Linear Stereo Vision

Safaa Moqqaddem
LASTID Laboratory
Ibn Tofail University K
énitra, Morocco

Yassine Ruichek
IRTES-SET
University of Technology
of Belfort-Montbéliard
90010 Belfort Cedex, France

Raja Touahni LASTID
Laboratory
Ibn Tofail University Kénitra,
Morocco

Abderrahmane Sbihi
LABTIC Laboratory, ENSA
Abdelmalek Essadi University
Route Ziaten, km 10, BP 1818
Tanger, Morocco.

***Abstract*—This paper presents a new method for tracking objects using stereo vision with linear cameras. Edge points extracted from the stereo linear images are first matched to reconstruct points that represent the objects in the scene. To detect the objects, a clustering process based on a spectral analysis is then applied to the reconstructed points. The obtained clusters are finally tracked throughout their center of gravity using Kalman filter and a Nearest Neighbour based data association algorithm. Experimental results using real stereo linear images are shown to demonstrate the effectiveness of the proposed method for obstacle tracking in front of a vehicle.**

***Keywords*—Linear stereo vision; Spectral clustering; Objects detection and tracking; Kalman filter; Data association.**

I. Introduction

Two inseparable aspects coexist in the field of intelligent transportation applications like video surveillance, robotic, etc: detection and tracking. This question that is a challenging 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. Among the domains 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 one 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. Once the matching process is achieved, the geometric triangulation leads to a list of points represented in a 2D coordinate system of the 3D dimensional world, since linear stereo vision permit to reconstruct only horizontal and depth information [1], [2], [3], [4], [5]. The objective is then to regroup these points in order to form clusters, where each cluster of points corresponds to an object of the scene. To perform this task, the difficulty is that there is no knowledge about the number of objects and the distribution of the reconstructed points in the scene. Hence, the classical supervised clustering methods are not suitable to achieve this task [6], [7].

Considering the object detection problem, there are many object detection methods in the literature, which can be classified as point detectors based, segmentation based, background subtraction based, or clustering based [8]. In [9], [10], the authors proposed a method that proceeds with agglomeration partitioning. They consider 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 readjusting it until reaching a criterion optimality. The PDDP method (Principal Direction Divisive Partitioning), proposed by Boley [11], 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 [12], [13]. 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. Considering the second aspect that is devoted to object tracking, there are two categories of tracking approaches in the literature: by matching or by update. Matching track is used to build trajectory characteristics of objects. The principle of this approach is to detect objects and agglomerate them temporally in order to obtain coherent paths over time. Tracking by update consists in detecting and locating objects depending on their state at the previous time. More precisely, tracking consists in estimating the parameters characterizing the objects during the sequence acquisition, such as geometry invariance of the scene or objects, object appearance (photometry or color) or kinematic (space-time constraints). Among the parameters widely used in the literature, one can cite position of center of the objects, to which may be added, depending on the considered application [14], scaling [15] and/or orientation [16] that are used generally for rigid or articulated objects [17]. For deformable objects, the parameters to be estimated are based on modeling contours [18] or modeling appearance using deformable surface models such as active appearance models [19], [20]. All these characteristics define the state of the objects in the scene. Unfortunately, most existing tracking methods are based on a single target model and they are limited to certain specific controlled environments [21]. In the context of our work, we propose a complete solution for localization and tracking objects in static and dynamic

scenes. For the object detection purpose, we propose to use a clustering method based on a spectral analysis of the points distribution whereas the tracking stage is based on a filtering technique and a data association method. The principle of the used object detection method is to perform a spectral decomposition of a transition matrix, constructed from the data to be clustered. The spectral decomposition consists in extracting the eigenvalues of the transition matrix. The analysis of these eigenvalues allows detecting the different structures in the data to be clustered. The spectral analysis leads to a selection of a number of significant eigenvalues that 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 in the scene. The clustering process may provide two or more clusters for the same object. This occurs when the number of clusters is over estimated by the spectral analysis. To deal with this problem, an objects merging strategy is developed to merge the clusters representing the same objects. Finally, the detected objects are tracked throughout the geometric centers of the extracted clusters using Kalman filter and a nearest neighbor based data association technique.

This work is structured into the following sections: Section A presents briefly the principle of linear cameras based stereo vision. Section B details the proposed spectral clustering method. In section C, the tracking procedure is described. Before concluding, experimental results are presented and discussed in section D.

A. Stereo vision with linear cameras

Stereo vision is a popular technique for inferring 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 [5], [6]. 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. In our work, a linear stereo system is built with two line-scan cameras, so that their optical axes are parallel and separated by a distance E. 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 [5]. The first step in stereo vision is to extract from each image the primitives to be matched. In classical video images, one can extract different types of primitives. In the case of linear images, the choice is restricted as a result of the one dimensional nature of the profile of a linear image. The only possibility in this case is to search for contour points corresponding to the frontiers of different objects present in the image. Edge extraction is performed by means of the Deriche's operator and a technique that selects pertinent local extrema [4]. Applied to the left and right linear images, this edge extraction procedure leads 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. To match the edges we used the method presented by the authors in [4]. In this method, stereo matching task is viewed as a constraint satisfaction problem where the objective is to highlight a solution for which the matches are as compatible as possible with specific 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 [22]. 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 [4]. Line-scan cameras cannot provide the vertical information.
Consider that the image coordinates x_l and x_r represent the projections of the point P in the left and right imaging sensors, respectively. 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.

B. Objects detection

Objects detection is an important and yet challenging task in the computer vision field. 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 paper, we are interested in detecting objects using a 3D scatter plot reconstructed from linear stereo vision. The proposed method is based on an unsupervised classification approach using spectral clustering [23], [24]. This approach allows also avoiding the problem of local minima inherent to the most part of classification methods [25]. The principle of this approach is to perform spectral decomposition of a similarity matrix, constructed form 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 data to classify [25], [26].

1) Spectral clustering algorithm:

Consider a set of n points $L = \{P_1, ..., P_n\}$ 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 procedure can be summarized as in the Algorithm 1.

As indicated above, spectral clustering requires first to adjust the scaling parameter σ, which is used in the expression of the affinity matrix A (Equation 3). The second requirement

Algorithm 1: Spectral clustering algorithm

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 [24]. 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:

$$A_{ij} = \begin{cases} exp(\frac{-d^2(P_i,P_j)}{\sigma^2}) & if \quad i \neq j \\ 0 & if \quad i = j \end{cases} \quad (3)$$

Where $d(P_i, P_j)$ 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. Table I gathers different types of normalization forms that could be applied to the affinity matrix. After some preliminary tests, we retained symmetric division normalization (Equation 4), which is more suitable for our application convenient

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

4) Form the matrix X=[X_1,. .....,X_k] in R^{n*k}, where X_1,. ...,X_k are the k igenvectors of the matrix N, corresponding to the k significant eigenvalues λ_1,. ...,λ_k. The determination of value of k is discussed in section B.4.
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 $M = \frac{k^n}{k!}$ is the number of possible partitions.
8) Assign the point P_i to the class C_j if and only if

concerns the determination of the number of classes k that corresponds to the number of significant eigenvalues of the transition matrix N. We propose in this paper an experimental methodology to estimate conjointly σ and k, in order to make the clustering process as a nonparametric and unsupervised classification method.

2) 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 [25], 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 [26], 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, resulting in an asymmetric affinity matrix. Originally, this selection strategy was motivated by supposing that the clusters are non-homogeneously dispersed, but it provides also a very robust way for selecting σ in homogeneous cases. In our case, we adopted the selection strategy proposed in [26] for its simplicity. For that, different values for σ are taken to select the value that provides less distorted clusters of the row of the matrix X [27], [28]. Our common approach is to try different values of σ and retain the best one. Section D describes our experimental methodology to set the value of the parameter σ.

TABLE I: Different forms of the normalization function

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 = $\frac{(A+d_{max}I-D)}{d_{max}}$; d_{max}=$\max_i(D_{ii})$

3) Estimation of the number of clusters k:

The determination of the number of clusters k can be performed by analyzing the eigenvalues $\{\lambda_i\}$ or the eigenvectors $\{X_i\}$ of the matrix N [26]. Theoretically, this analysis consists in selecting 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. One can consider also the analysis of the difference between successive eigenvalues. The disadvantage of this strategy is that the jump between two successive eigenvalues, which can be big or small, is difficult to control [27]. We tested this strategy in order to determine an empirical relationship between the difference of successive eigenvalues and the significant ones. After various tests, we found that thresholding analysis is more adapted for our application. In section D, we will present our experimental methodology to set the threshold value for extracting significant eigenvalues, and then the number of clusters.

It is worthy to note that the clustering process can provide two or more clusters for the same object. This situation occurs when the spectral analysis produces an overestimation of the number of clusters, during significant eigenvalues selection step. To resolve this problem, an object fusion strategy is developed for merging clusters representing the same object. This fusion procedure is described in Section C.6.

C. Objects Tracking

Objects tracking in space is a basic problem, but important in many computer vision applications. It consists in reconstructing the trajectory of objects along time. 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.

1) Modeling:

In this work, we are interested in tracking objects, where each object is represented by a cluster of points. The clusters are obtained by the spectral clustering algorithm described in section B.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:

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

$$z(t) = z(t - dt) + \dot{z}.dt + \frac{1}{2}\ddot{z}.dt^2 \tag{6}$$

where x is the horizontal position and z is the depth of the geometric center of a cluster representing an object. Recall that the reconstruction space is represented by two axes as described in section A. They represent respectively the horizontal position and depth of reconstructed points from linear stereo vision [4].

The most popular approach used for tracking mobile objects is based a kalman filter which represents a particular case of filter bayesian under the 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 [29].

2) Kalman filter:

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 modelled system is unknown. KF 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 [30]. The discrete-time state equation with sampling period T is expressed as follows:

$$S(l + 1) = F \times S(l) + W(l + 1) \tag{7}$$

In this work, the state $S(l)$ is composed with the position and velocity of the geometric center of a cluster of points representing an object: $S(l) = [x\ v_x\ z\ v_z]^t$, where l is time step. 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 $W(l)$. The measurement model $Y \in R^m$ (m=2 in our case) is given by:

$$Y(1) = H \times S(1) + V(1) \tag{8}$$

where H is the observation model: $H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

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

$$\begin{aligned} P(W) &\sim N(0, Q) \\ P(V) &\sim N(0, R) \end{aligned} \tag{9}$$

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.
KF can be written as a single equation. However, it is most often conceptualized as two distinct phases: prediction phase and updating phase. 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.
For multiple objects tracking, the problem of data association must be handled. The proposed data association algorithm is presented in the section C.4.

3) Kalman filter algorithm :

In this algorithm (Algorithm 2), i correspond to the i^{th} geometric center to track. S_{apr} is the a priori state estimate; P_{apr} 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 and Y is the sensor measurement.

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 important 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) [31], [32]. 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 smallest Mahalanobis distance based on the association between an observation and an existing track, or between an observation and a new track assumption. In our case, we

Algorithm 2: Kalman filter algorithm

Initialization :

$$Q = \begin{bmatrix} 0 & 0.0001 & 0 & 0 \\ 0.0001 & 0.0025 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 \\ 0 & 0 & 0.0001 & 0.0025 \end{bmatrix}$$

$$P^i_{apos}(0) = Q$$

$$R = \begin{bmatrix} (0.5)^2 & 00 \\ 0 & (0.5)^2 \end{bmatrix}$$

$$S^i_{apos}(0) = S^i(0)$$

Prediction :

$$S^i_{apr}(l) = F \times S^i_{apos}(l-1) \quad (10)$$

$$P^i_{apr}(l) = F \times P^i_{apos}(l-1) \times F^t + Q \quad (11)$$

Updating :

$$Y^i_{apr}(l) = H \times S^i_{apr}(l) \quad (12)$$

$$Res^i(l) = Y^i(l) - Y^i_{apr}(l) \quad (13)$$

$$C^i(l) = H \times P^i_{apr}(l) \times H^t + R \quad (14)$$

$$K^i(l) = P^i_{apr}(l) \times H^t \times (C^i(l))^{-1} \quad (15)$$

$$S^i_{apos}(l) = S^i_{apr}(l) + K^i(l) \times Res^i(l) \quad (16)$$

$$P^i_{apos}(l) = (I_4 - K^i(l) \times H) \times P^i_{apr}(l) \quad (17)$$

are interesting to track the geometric centers of the obtained clusters representing the objects in the scene. Mahalanobis distance is a statistical distance that takes into account the covariance and correlation of the elements of the state vector, and it is appropriate to solve data association problem. In our case, the covariance and correlation are determined between the measurement (observation) provided by the sensor and the predicted measurement given by Kalman filter. Mahalanobis distance is defined by:

$$d^2_m(Y, Y_{apr}) = \frac{1}{2}(Y - Y_{apr})^t \times C^{-1} \times (Y - Y_{apr}) \quad (18)$$

where C is the covariance matrix of the residual Res, which is the measurement innovation (see Equation 14); Y_{apr} is the predicted measurement (see Equation 12); Y is the measurement (observation) provided by the sensor.
Before applying the Mahalanobis distance based NN data association, one needs to define a search area for identifying potential candidate points (geometric centers) 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. The search area for each object is considered as a circle.

Let G^i_l be the searching circle of the predicted object i at time step l. The ray of this searching circle is defined by equation 19.

$$ray(G^i_l) = \triangle v(x, z) \quad (19)$$

where $\triangle v(x, z)$ is the difference between the velocities at time steps l and $l + 1$.

The data association process is first applied considering the horizontal position x, the ray of the corresponding searching circle is determined by $ray(G^i_l) = \triangle v(x)$. The results are then validated by the data association process according to the depth z, the ray of the corresponding searching circle is determined by $ray(G^i_l) = \triangle v(z)$

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 [6]. 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 must disappear.

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 due to an overestimation of the number of clusters. To resolve this problem, we propose a cluster fusion technique based on a cluster overlapping strategy. The fusion technique consists in determining an overlapping coefficient, defined as follows:

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

Where 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 their Euclidean distance; r_i and , r_j which are determined in the data association step, represent respectively the rays of the searching areas of the two tracks i and j. The ray r_i is calculated as the difference between the estimated (KF-based) and measured (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.

D. Results and discussion

In this section, we present the performance of the proposed object detection and tracking approach, to deal with obstacle detection and tracking in front of a vehicle. As shown in Figures 1 and 2, the line-scan cameras based stereo set-up is

installed on the top of a car for periodically acquiring stereo pairs of linear images as the car travels [4], [6]. The tilt angle is adjusted so that the optical plane intersects the pavement at a given distance D_{max}= 50m 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 3 illustrates a scenario in which a pedestrian is traveling, according a predefined trajectory, in front of the prototype vehicle, which is static. The pedestrian, starting from the right side of the stereoscope (A), is first seen moving to an area located just beyond the intersection of the plane of view and the road (B). When arriving to this area, he leaves the field of view of the cameras and hence disappears in the stereo images (see Figure 5). Then, the pedestrian reappears in the field of view and begins to move towards the left camera (C), before turning slightly to the right camera (D). After that, he moves towards the left camera and then towards the right one before leaving their field of view (E).

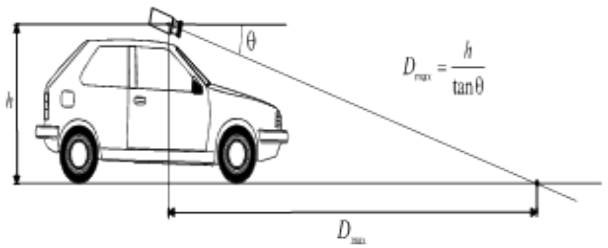

Fig. 1: Stereo set-up, side view.

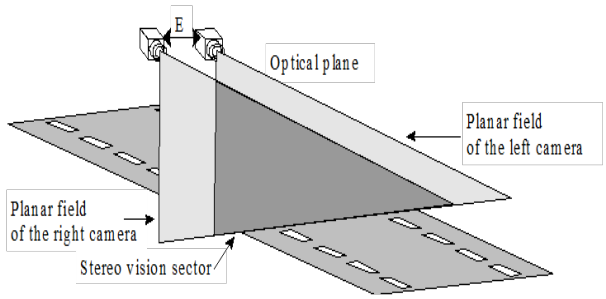

Fig. 2: Stereo set-up, top view.

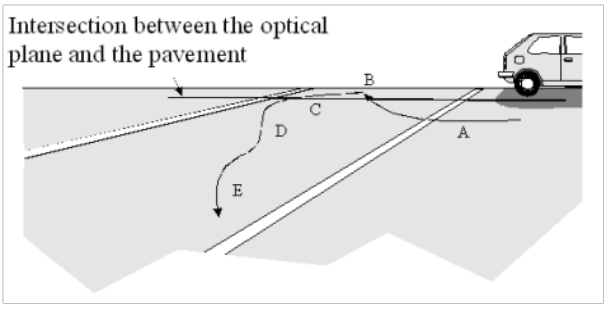

Fig. 3: Stereo set-up, top view.

Figure 4 shows the stereo image sequence representing the scenario of Figure 3. The linear images are represented as horizontal lines, time running from top to down each one the left and right sequences are composed of 200 linear images each. On the images, one can see clearly the white lines of the pavement and the pedestrian who appears with a growing form. 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.

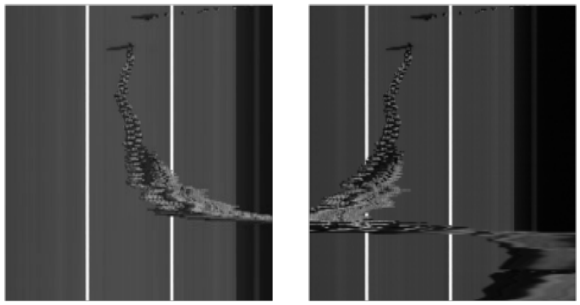

Fig. 4: Stereo sequence (pedestrian).

The stereo sequence is processed with the stereo matching procedure (see section A). The disparities of all matched edges are used in order to compute the positions and distances of the edges of the objects seen in the stereo vision sector. Figure 5 illustrates the obtained reconstruction image where distances are represented as gray levels, the darker is the closer, whereas positions are represented along the horizontal axis. As in Figure 4, time runs from top to down. The edges of the two white lines as well as those corresponding to the transition between the pavement and the area of shadow are correctly matched. Their detection is stable along the sequence as positions and distances remain constant during time. The edges representing the pedestrian are also well reconstructed as their positions and distances are coherent with the trajectory of the pedestrian. One can notice few bad matches when occlusions occur 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 visible white line, seen by one of the cameras, with those representing the pedestrian.

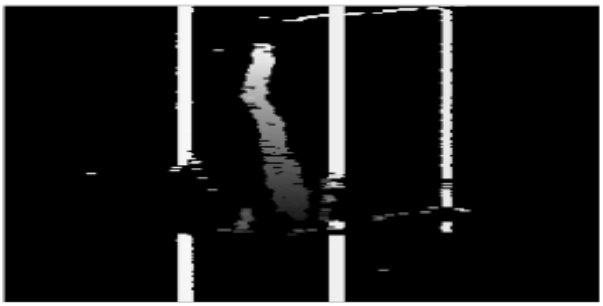

Fig. 5: Image reconstruction of Pedestrian stereo sequence.

The proposed spectral clustering is then applied to the reconstructed points for each stereo couple to detect the objects present in the scene. As discussed in sections B.3 and B.4, we have to set optimally the scaling parameter σ (Equation 3) and

the threshold to apply to the eigenvalues of matrix N (Equation 4) in order to determine the significant ones. The number of significant eigenvalues provides the number of clusters. For that, we apply the clustering process considering several values for the parameter σ^2 and three predefined thresholds. For each couple (σ^2, threshold), we compute the percentage of cases where the detection result is identical to the reality, considering all the stereo couples of the sequence. Table 2 shows the obtained percentages, and Figure 6 gives the real number of objects present in the scene for each stereo couple. One can see that the best couple (σ^2, threshold), providing the high percentage of 73.23%, is obtained with σ^2=1.2 and threshold = 0.5. Consequently, for the tests presented in the sequel of this paper, we opted for these values as optimal spectral clustering parameters.

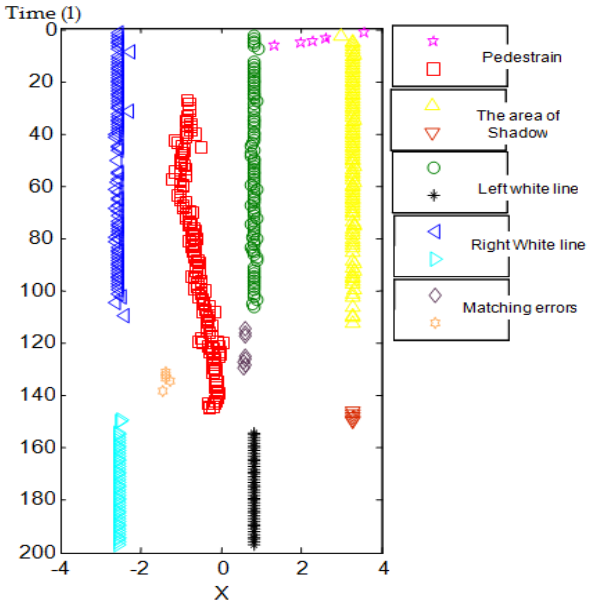

Fig. 7: Objects detection and tracking with threshold = Mean and σ^2=2.

The clustering stage is performed on the reconstructed points for each pair of the stereo sequence. The tracking process is applied to the geometric centers of the obtained clusters characterizing the detected objects in the scene. As stated before (see figure 5), some matching errors occur, especially in presence of occlusions at the end of the sequence, i.e., when the pedestrian hides one of the white lines characterizing the scene. To reduce the effect of these errors on the clustering task, and hence on the tracking process, we apply the temporal constraint that allows ignoring objects generated erroneously from the stereo matching process. Furthermore, and as mentioned previously, 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. To discard this shortcoming, we apply our proposed clusters fusion strategy presented above. Figures 7 and 8 illustrate the obtained detection and tracking results with different values of the spectral clustering parameters (threshold and σ^2). In these figures, each detected and tracked object is represented by a colored symbol. One can see clearly in Figure 9 that all objects presents in the scene are correctly detected and tracked with

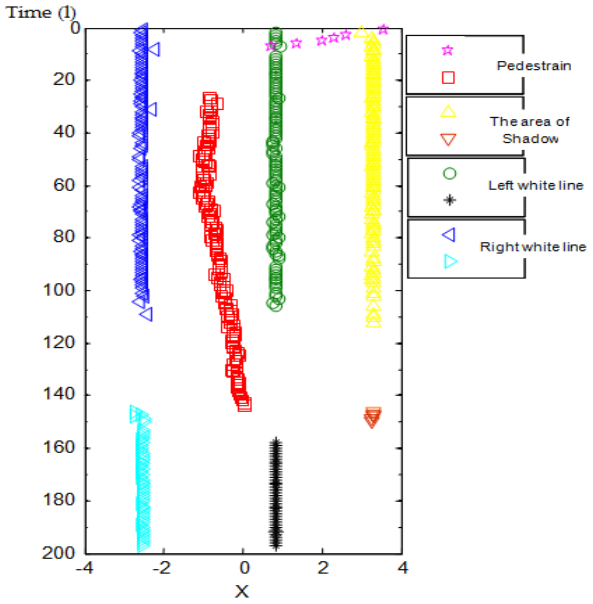

Fig. 8: Objects detection and tracking with threshold = 0.5 and σ^2=1.2.

the optimal parameters (threshold = 0.5 and σ^2=1.2) obtained by the analysis given by Table II . Indeed, clusters representing same object (pedestrian in our case) are fused correctly thanks to the proposed fusion strategy, and, false detections, due to stereo matching errors, are removed thanks to the temporal constraint.

Figure 9 shows the number of objects obtained by detection only and detection/tracking, compared with the real number of objects present in the scene. As we can see, the tracking process allows improving the detection results. In terms of percentage of cases where detection results are identical to ground truth, the rate reaches 85% with tracking instead of 73.23% (see Table II) obtained without tracking. In order to validate the performance of our proposed objects detection and tracking approach, we applied it on a more complex stereo sequence, acquired with the prototype car traveling in highway. Figure 10 illustrates the scenario representing the sequence in which the objects to detect and track are vehicles moving in front of the prototype car equipped with the stereoscopic system. Arrows indicate the relative movements of vehicles relative to the prototype vehicle marked with a cross.

The prototype car travels in the central lane behind another car (A). As the distance is decreasing, the optical plane of the stereo set-up intersects gradually the shadow of the preceding car and then the whole car from the bottom to the top as shown in Figure 11. A third car (B) pulls back into the central lane after overtaking the preceding car (A). Car B is out of the field of view of the stereo set-up. However, and as it can be seen in Figure 11, its shadow captured. The prototype car is itself overtaken by another vehicle (C), which is traveling in the third lane of the road. The partial presence of car C is shown in Figure 11. Figure 11 represents the linear images of the acquired stereo sequence. As in Figure 4, the linear images are represented as horizontal lines, time running from top to bottom. The left and right sequences are composed of

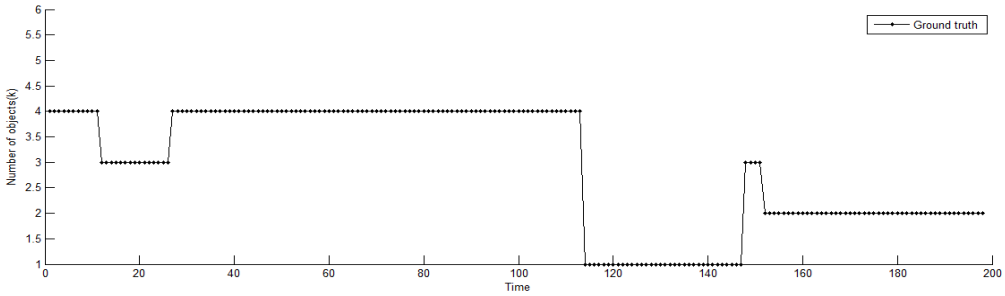

Fig. 6: Real number of objects present in each stereo couple during the Pedestrian sequence.

TABLE II: Percentage of cases where detection result based on spectral clustering is identical to the reality, for different couples (σ^2, threshold). Mean is equal to the mean of all eigenvalues of the matrix N.

σ^2 / threshold	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0.5	72,22	72,22	**73,23**	71,72	71,72	72,22	72,22	72,73	71,72	72,22	72,73
Mean	67,68	67,68	67,68	68,18	68,18	69,19	69,70	69,70	68,18	66,67	67,68
0,9	69,70	70,20	69,70	69,70	69,70	70,20	69,70	70,20	70,20	69,70	68,69

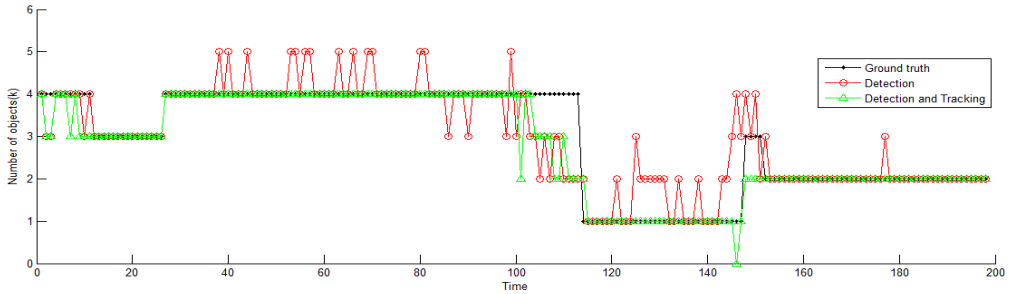

Fig. 9: Number of objects number by detection and detection/tracking, compared to ground truth.

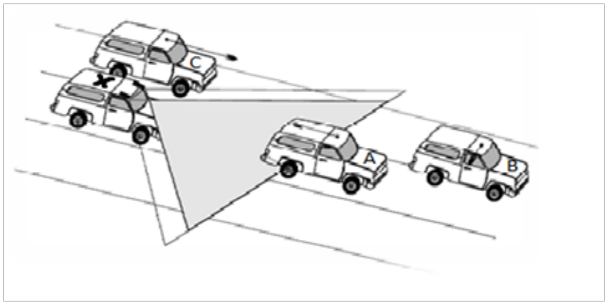

Fig. 10: Displacement of different vehicles during the sequence 2.

200 linear images each. In Figure 11 we can see the white lines, which delimit the pavement of the road, and between these lines, the two dashed white lines and the preceding car in the central lane. The vehicle (C), which is overtaking the prototype car, is seen at the bottom of the left and right sequence on the left-most lane. At the same level, in the middle of the left and right sequences, one can see the shadow of the vehicle, which pulls back in front of the preceding car. The curvilinear aspect of the lines in Figure 11 is caused by the variations of the stereoscope tilt angle, because of the uneven road surface. Depth reconstruction is not affected by these variations, provided that the stereo set-up remains correctly calibrated when the prototype car is running.

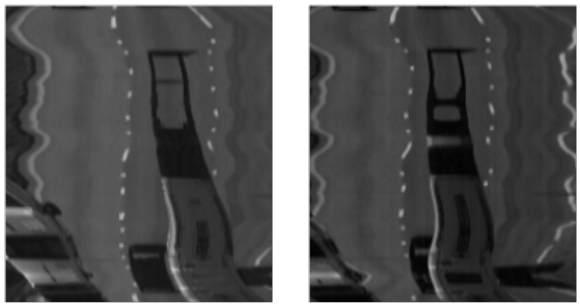

Fig. 11: Stereo sequence 2.

After applying the stereo matching procedure, the obtained reconstruction image is illustrated in Figure 12. The edges of the two dashed lines have been correctly matched. The edges of the lines, which delimit the road, cannot be matched continuously because they do not always appear in the common

part of the fields of the cameras. The preceding vehicle (A) is well detected as it comes closer and closer to the prototype car as time runs. The shadow of the vehicle (B), which pulls back in front of the preceding vehicle, is identified as a white continuous (almost) line at the bottom of the reconstructed image. Finally, at the bottom of the reconstructed image, we can see the dark oblique line, which represents the vehicle (C) overtaking the prototype car.

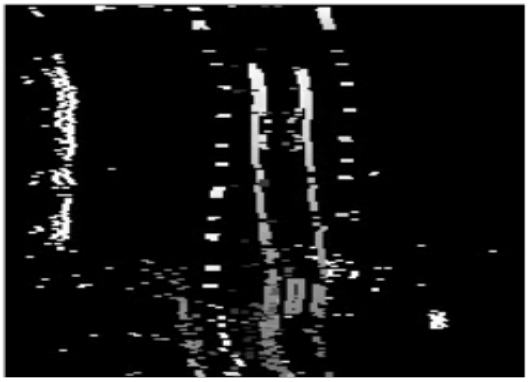

Fig. 12: Reconstruction image of the sequence 2.

Figure 13 shows the objects detection and tracking results obtained by applying the proposed approach on the reconstructed points of Figure 13, using the optimized clustering parameters (threshold = 0.5 and σ^2=1.2). In Figure 13, each detected and tracked object is represented by a colored symbol. All the objects are well detected and tracked. However, the dashed lines and the shadow projected by the vehicle pulling back in front of the preceding car are missed because of the application of the temporal constraint.

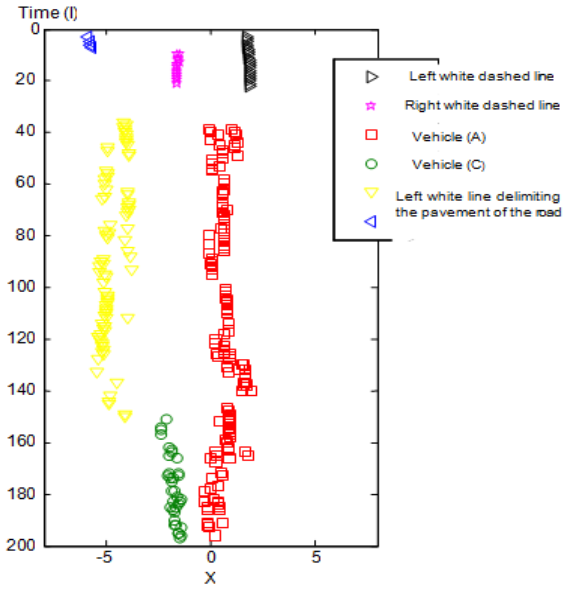

Fig. 13: Object detection and tracking with threshold = 0.5 and σ^2= 1.2.

II. Conclusion

In this paper, we presented a method for detecting and tracking objects using linear stereo vision. The method starts by reconstructing 2D points by matching object edges extracted from linear stereo images. A spectral based clustering algorithm is then applied on the reconstructed points in order to extract where each cluster represents an object of the observed scene. An experimental analysis is conducted to optimize the clustering parameters. Finally, a tracking procedure is performed on the extracted clusters using Kalman filtering and nearest neighbour data association. To improve the detection and tracking results, a fusion strategy is also developed to tackle the problem of the presence of multiple clusters representing a same object. To test and evaluate the proposed method, experiments are performed with real linear stereo sequences for objects detection and tracking in front of a vehicle.

Acknowledgment

The work presented in this paper is a part of Ground Vehicle Intelligence project aiming to develop advanced driving aid systems. The authors would like to thank the CPER 2006-2013, STIC and Volubilis programs for their financial support.

References

[1] Banks, J., Bennamoun, M., P.Corke, Kubik, K.: A taxonomy of image matching techniques for stereo vision. ueensland University Of Technology, Brisbane (1997)

[2] Nogueira, S., Ruichek, Y., F.Charpillet: A self navigation technique using stereovision analysis. Stereo Vision book. Edited By Dr. Asim Bhatti, 295–306 (2008)

[3] Teguri, Y.: Laser sensor for low-speed cruise control. Convergence Transportation Electronics Association (2004)

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

[5] 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)

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

[7] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Journal of Computing Surveys **38**(4) (2006)

[8] Mrabti, F., Seridi, H.: Comparaison de méthodes de classification réseau rbf, mlp et rvflnn. Damascus University Journal **25**(2) (2009)

[9] Kohonen, T.: Self-organizing maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997)

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

[11] Saux, B.L., Boujemaa, N.: 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 (2004)

[12] Boley, D.: Principal direction divisive partitioning. Data Min. Knowl. Discov **2**(4), 325–344 (1998)

[13] Zammit, O., Descombes, X., Zerubia, J.: 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, Troyes, France, 11–14 (2007)

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

[15] D.Comaniciu, V.Ramesh, P.Meer: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5), 564–577 (2003)

[16] Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10), 631–1643 (2005)

[17] Alper, Y.: Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. IEEE Conference On Computer Vision And Pattern Recognition (CVPR) (2007)

[18] Aggarwal, J.K., Cai, Q.: Human motion analysis: A review. Computer Vision And Image Understanding **73**(3), 428–440 (1999)

[19] Revéret, L.: From raw images of the lips to articulatory parameters: A viseme-based prediction. Eurospeech (2011-2014)

[20] Edwards, G., Taylor, C., Cootes, T.: Interpreting face images using active appearance models. International Conference on Face and Gesture Recognition, 300–305 (1998)

[21] Mikram, M.: Suivi d'objets dans une séquence d'images par modèle d'apparence : Conception et evaluation. PhD thesis, Université De Bordeaux I, Spécialité : Automatique, Productique, Signal Et Image Informatique Et Télécommunications, N°3736 (Décembre 2008)

[22] Ruichek, Y.: Perception de l'environnement par stéréovision application à la sécurité dans les systèmes de transports terrestres," hdr. PhD thesis, Université des Sciences et Technologies de Lille, France (2007)

[23] Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. Advances In Neural Information Processing Systems **17**, 1601–1608 (2004)

[24] Y.Weiss: Segmentation using eigenvectors: A unifying view. IEEE International Conference On Computer Vision, 975–982 (1999)

[25] A.Y.Ng, M.I.Jordan, Y.Weiss: On spectral clustering: Analysis and an algorithm. Advances In Neural Information Processing Systems 14, Cambridge, Ma. Mit Press (2002)

[26] D.Verma, Meila, M.: A comparison of spectral clustering algorithms. Technical Report Uw-Cse-03-05-01, University of Washington (2003)

[27] G.Sanguinetti, J.Laidler, L.Neil: Automatic determination of the number of clusters using spectral algorithms. IEEE Machine Learning For Signal Processing Mystic, Connecticut, USA, 28–30 (2005)

[28] I.Dhillon, Guan, Y., Kulis, B.: Kernel k-means, spectral clustering and normalized cuts. KDD'04, August 22–25, Seattle, Washinton, USA (2004)

[29] Y.Bar-Shalom, X.Li, T.Kirubarajan: Estimation with applications to tracking and navigation. Wiley, New York, Chapter 6 (2001)

[30] 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)

[31] Vermaak, J., Godsill, S.J., Pérez, P.: Monte carlo filtering for multi-target tracking and data association. Draft, September 22 (2004)

[32] Coué, C.: Modèle bayésien pour l'analyse multimodale d'environnements dynamiques et encombrés : Application a l'assistance a la conduite en milieu urbain. PhD thesis, Institutational Polytechnique De Grenoble, France (2003)