Expert Systems with Applications 41 (2014) 1532–1542

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Hierarchical camera auto-calibration for traffic surveillance systems

S. Álvarez *, D.F. Llorca, M.A. Sotelo

Computer Engineering Department, Polytechnic School, University of Alcalá, Madrid 28871, Spain

ARTICLE INFO

Keywords:
Auto-calibration
Pan-tilt-zoom cameras
Vanishing points
Intelligent transportation systems
Urban traffic infrastructures

ABSTRACT

In this paper, a hierarchical monocular camera auto-calibration method is presented for applications in the framework of intelligent transportation systems (ITS). It is based on vanishing point extraction from common static elements present on the scene, and moving objects as pedestrians and vehicles. This process is very useful to recover metrics from images or applying information of 3D models to estimate 2D pose of targets, making a posterior object detection and tracking more robust to noise and occlusions. Moreover, the algorithm is independent of the position of the camera, and it is able to work with variable pan-tilt-zoom (PTZ) cameras in fully self-adaptive mode. The objective is to obtain the camera parameters without any restriction in terms of constraints or the need of prior knowledge, to deal with most traffic scenarios and possible configurations. The results achieved up to date in real traffic conditions are presented and discussed.

1. Introduction

Camera calibration is a fundamental stage in computer vision. The process is the determination of the relationship between a reference plane and the camera coordinate system (extrinsics), and between the camera and the image coordinate system (intrinsics). These parameters are very useful to recover metrics from images or applying prior information of 3D models to estimate 2D pose of targets, giving an idea of the size of the objects and making their detection and tracking more robust to noise and occlusions.

In a previous paper Álvarez et al. (2012), the authors presented a target detection system for transport infrastructures based on manual camera calibration through vanishing points. After that, the approach was improved, as described in Álvarez et al. (2013), with a preliminary automatic calibration method based on camera zooming and zebra-crossings. The current paper extends these works with a hierarchical camera auto-calibration system, which deals with most traffic scenarios and configurations with no restrictions. The work begins with the paper presented in Álvarez et al. (2011).

The standard method to calibrate a camera is based on a set of correspondences between 3D points and their projections on image plane as presented by Hartley and Zisserman (2000) and Tsai (1986). However, this method requires either prior information of the scene or calibrated templates, limiting the feasibility of surveillance algorithms in most possible scenarios. In addition, calibrated templates are not always available, they are not applicable for already-recorded videos and if the camera is placed very high their small projection can derive in poor accurate results. Finally, in case of having PTZ cameras, using a template each time the camera angles or zoom changes is not feasible. One novel method which solves the problem of the template is the orthogonal calibration proposed by Kim (2009). The system extracts the world coordinates from aerial pictures (on-line satellite images) or GPS devices to make the correspondences with the image captured. However this system is dependent on prior information from an external source and it does not work indoor. Therefore auto-calibration seems to be the more suitable way to recover camera parameters for surveillance applications.

One of the distinguished features of the perspective projection is that the image of an object that stretches off to infinity can have finite extent. For example, parallel world lines are imaged as converging lines, which image intersection point is called *vanishing point*. Caprile and Torre (1990), developed a new method for camera calibration using simple properties of vanishing points. In their work, the intrinsic parameters of the camera were recovered from a single image of a cube. In a second step, the extrinsic parameters of a pair of cameras were estimated from an image stereo pair of a suitable planar pattern. The technique was improved by Cipolla et al. (1999), who computed both intrinsic and extrinsic parameters from three vanishing points and two reference points from two views of an architectural scene. However these assumptions were incomplete, because as demonstrated by Hartley, Zisserman and Liebowitz in different publications, and summarized in Hartley and Zisserman (2000), it is possible to obtain all the parameters needed to calibrate a camera from three *orthogonal* vanishing points. From the mentioned works, a lot of research has been done to calibrate cameras in architectural environments (Rother, 2002; Tardif, 2009, etc...). All these methods are based on scenarios

* Corresponding author. Tel.: +34 91 885 6702.
E-mail address: sergio.alvarez@aut.uah.es (S. Álvarez).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.050

where the large number of orthogonal lines provide an easy way to obtain the three orthogonal vanishing points.

Nevertheless, in absence of so strong structures, as usual in the case of traffic scenes, the vanishing point-based calibration is not applicable. In this context, a different possibility is to make use of object motion. The complete camera calibration work using this idea was introduced by Lv et al. (2006). The method uses a tracking algorithm to obtain multiple observations of a person moving around the scene. Three orthogonal vanishing points are then computed by extracting head and feet positions in their leg-crossing phases. The approach requires accurate localization of these positions, which is a challenge in traffic surveillance videos. Furthermore, the localization step uses FFT based synchronization of a person's walk cycle, which requires constant velocity motion along a straight line. Finally it does not handle noise models in the data and assumes constant human height and planar human motion, so the approach is really limited. Based on this knowledge, in Junejo (2009) it is proposed a quite similar calibration approach for pedestrians walking on uneven terrains. There are no restrictions as with Lv's work, but the intrinsic parameters are estimated by obtaining the infinite homography from all the extracted points in multiple cameras.

To manage these inconveniences, the solution lies in computing the three vanishing points by studying three orthogonal components with parallel lines in the moving objects or their motion patterns. Zhang et al. (2013) presented a self-calibration method using the orientation of pedestrians and vehicles. The method seems to extract a vertical vanishing point from the main axis direction of the pedestrian trunk, perpendicular to the ground plane. Additionally, two horizontal vanishing points are extracted by analysing the histogram of oriented gradients of moving cars. The idea is interesting and it was initially implemented for this work. However, the straight vehicles used by Zhang differ from the modern ones, usually with more irregular and rounded shapes. Finally, the pedestrian detection step is not described and results are not depicted in the paper. Hodlmoser et al. (2010) present a different approach. They use zebra-crossings with known metrics to obtain the ground plane information, and pedestrians to obtain the vertical lines. The problem is the maximum distance that the camera can be from the ground and the necessity of knowing real distances from the scene.

In this paper, a self-calibration procedure based on vanishing points is presented. It is done through a hierarchical process which covers most of traffic scenarios and possible configurations. The objective is to obtain both intrinsic and extrinsic camera parameters without restrictions in terms of constraints (restrictions mentioned in previous paragraphs, vehicles driven in only one road direction (Hue et al., 2008), deprecated camera roll (Schoepflin and Dailey, 2003), etc.) or the need of prior information, except for the camera height.

After the present introduction, the remainder of the document is organized as follows. Section 2 describes the developed camera auto-calibration method, based on vanishing points, and the hierarchical system proposed. In Section 3, an application of this technique in the context of traffic surveillance is depicted with the developed segmentation and tracking algorithms. The results obtained are presented and discussed in Section 4 and finally Section 5 contains the conclusions and future work.

2. Camera autocalibration

The camera model used and the equations to obtain the calibration parameters from orthogonal vanishing points are described in the previous paper, Álvarez et al. (2013). In summary, the conclusion is that it is possible to calibrate a camera if the principal point and two orthogonal vanishing points are known; or by computing the principal point as the orthocentre of the triangle formed by three orthogonal vanishing points as vertices. The current work is focused on the way to extract these points from common elements of traffic scenarios. Fig. 1 summarizes the proposed camera calibration process.

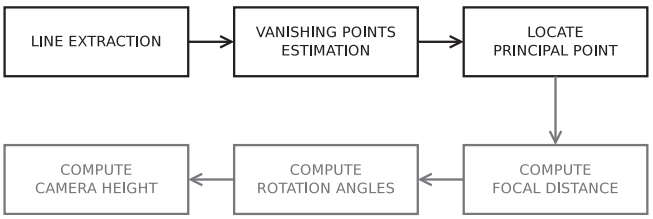

Fig. 1. Camera auto-calibration process.

2.1. Hierarchical auto-calibration

This section presents the proposed method to extract the vanishing points from the image through a hierarchical process. Depending on which elements appear in the scene and the chance of using camera zoom, 5 levels have been established to determine the hierarchy of each developed method and the priority of the solution adopted. Before presenting the hierarchical tree of Fig. 2, and to make its comprehension easier, the different options developed to obtain the vanishing points and optical center are described in the following paragraphs:

- *Zoom:* when zooming, if several features of the image are matched between frames they converge in a common point which corresponds to the optical center.
- *Crosswalk (cross):* the alternate white and gray stripes painted on the road surface provide a perfect environment to obtain two perpendicular sets of parallel lines. It means that two vanishing points of the ground plane can be extracted.
- *Pedestrians (ped):* humans are roughly vertical while they stand or walk. This characteristic makes them very useful to extract perpendicular lines to the ground.
- *Vehicle motion (vmot/vperp):* if one vanishing point from the ground plane is needed, it can be obtained from vehicles moving along the main motion direction (vmot). In case of a perpendicular intersection (in 3D coordinates), vehicles along the two main directions will provide perpendicular sets of parallel lines corresponding to the two ground plane vanishing points (vperp).
- *Structured scene (struct):* in case of scenes with a considerable number of architectural elements, the orthogonal vanishing point extraction can be done by brute force gradient analysis.
- *Optical center assumption (OC):* when it is not possible to obtain one of the three vanishing points, the optical cener can be assumed as the center of the image, although a small error is committed.

The different possible cases are also summarized in Fig. 2 and Table 1.

2.2. Principal point through camera zoom

When zooming, if several features of the image are matched between frames, the lines which join the previous and new feature positions converge in a common point which corresponds with the optical center. This effect is demonstrated in Álvarez et al. (2013) and represented in Fig. 3: an image was taken before and

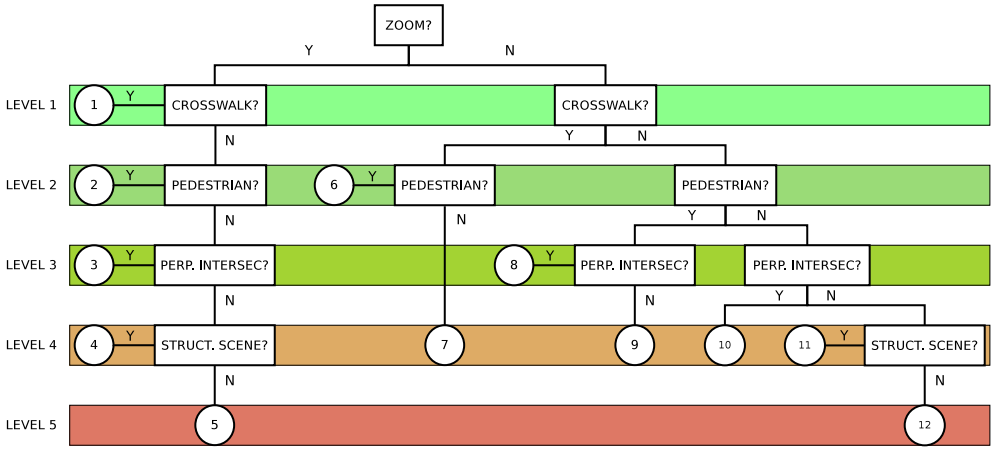

Fig. 2. Hierarchical calibration tree used. Note: *Perp. Intersec* means perpendicular intersection.

Table 1
Cases of the hierarchical tree.

CASE	ZOOM	CROSS	PED	VMOT	PERP	STRUCT	OC	MANUAL
1	x	x						
2	x		x	x				
3	x				x			
4	x					x		
5	x							x
6		x	x					
7		x					x	
8			x		x			
9			x	x			x	
10					x		x	
11						x		
12								x

after zooming and the matched features converge to the same point, the optical center.

2.3. Zebra crossing vanishing point extraction

The alternate stripes painted on the road surface provide a perfect environment to obtain two perpendicular sets of parallel lines. It means that two vanishing points from the ground plane can be computed. The crosswalk detection method is also explained in Álvarez et al. (2013)), and illustrated in the Fig. 4. Firstly, the background model image is binarized, and the lines are extracted by gradient analysis and grouped by angle. After that, a RANSAC-based filter is applied to get the final candidates. The red line is the one which best fits the candidate. Bipolarity and transition analysis is then done in order to obtain a confidence factor. Finally, the vanishing points are computed.

2.4. Pedestrian vanishing point extraction

Humans are roughly vertical while they stand or walk. This property makes them very useful to get perpendicular lines to the ground, to compute the vertical vanishing point. One option is to extract the vertical component of each pedestrian to form the necessary set of parallel lines, as done by Hodlmoser et al. (2010). However, the cameras in common traffic scenarios are usually located quite higher than the situations proposed by the authors in the paper, and small pedestrians can derive into erroneous lines extractions. Traffic scenes provide a lot of structured elements with vertical components (walls, lampposts, traffic lights, etc.), that can be used to increase the performance and quality of the system. The developed algorithm is based on this idea, and it is divided into the following steps: pedestrian detection with vertical component extraction, scene analysis and vanishing point computation.

The aim of this method is to detect pedestrians with no false positives, to avoid lines that are not perpendicular to the ground. Therefore, it is not crucial to detect all the pedestrians in the image but it is important to be sure that the detected objects are humans. In order to obtain useful candidates for vertical lines extraction two kinds of parameters for every moving object are obtained: the motion direction and the main axis direction. The difference of these directions is quite significant for moving pedestrians while it is very small for vehicles. It is evident that this classification is not very accurate, but in practice it is good enough to get valid pedestrians useful to extract vertical lines.

To compute the main axis direction of the blob θ, three different approaches have been used: moment analysis, principal component analysis and RANSAC estimation. The direction estimated by moment analysis is defined as:

(a) (b) (c)

Fig. 3. Principal point computation through camera zoom. (a) Image before zooming and extracted features. (b) Image after zooming and extracted features. (c) Feature matching. The common point corresponds to the optical center.

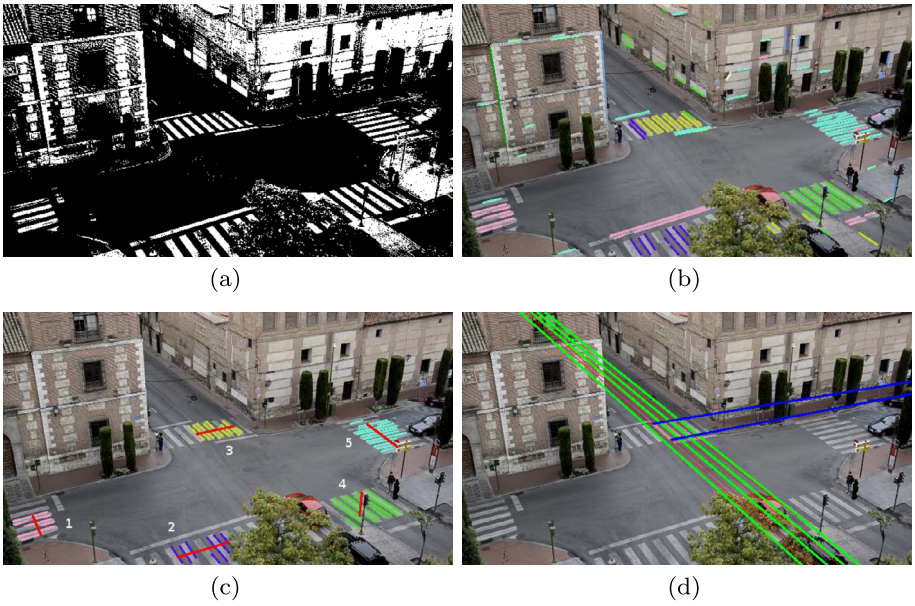

Fig. 4. Crosswalk detection example. (a) Binarized background model. (b) Line extraction. (c) Grouped candidates with testing lines in red. (d) Parallel lines to compute the vanishing points. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

$$\theta_{moment} = \tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (1)$$

where μ_{pq} is the central moment of order (p,q).

Principal Component Analysis (PCA) is equivalent to major axis regressions, so the largest axis can be considered as the vertical component. And finally RANSAC algorithm takes the centroid of each candidate row to estimate the line that corresponds to the main axis of the pedestrian. When these three methods obtain similar results and the blob aspect ratio is valid, the candidate is considered a pedestrian. At the same time, a gradient line extraction of the image is done in order to extract all the possible structured elements. The angle of the vertical components of the pedestrians is compared to the lines extracted and, in case of matching, the lines will be saved to compute afterwards the vanishing point. Due to the perspective of the camera, a perpendicular line to the ground in the image has different angles depending on the position. Moreover, because of the negative *pitch* the vertical vanishing point has to be positive. Therefore the image is divided into five quadrants.

Fig. 5 depicts an example of the developed method. Fig. 5(a) represents the lines extracted from the scene, with different colors depending on the belonging quadrant. Fig. 5(b) shows the detected pedestrian inside a green box with the estimated vertical component in red, and the matched vertical lines in magenta. Finally, Fig. 5(c) depicts the estimation of the vertical vanishing point with all the accumulated vertical lines. Red lines are the outliers and green lines the inliers for a RANSAC-based method to obtain the intersection point.

2.5. Vehicle motion vanishing point extraction

One of the properties of the traffic scenarios is that many vehicles drive in the same or inverse direction of the 3D world. Therefore the main axis of these vehicles are parallel to each other, and also parallel to the ground plane. This supplies important information to extract horizontal vanishing points.

As explained in the hierarchical calibration tree (Fig. 2), there are cases that need only one ground plane vanishing point while others need two. In case of computing the optical center (either by zooming analysis or assuming it as the image center) and detecting pedestrians, only one vanishing point from the ground plane is needed, in any direction. On other hand, in case of needing two ground plane vanishing points and if a perpendicular intersection (in 3D coordinates) is present in the scene, vehicles moving along the two main directions will provide perpendicular sets of parallel lines corresponding to the two ground plane vanishing points. In both cases the followed process is similar, done either for one direction or two respectively.

Firstly, the main motion directions are extracted. For this purpose, a feature optical flow analysis of the foreground blobs is done and their motion direction is saved into an histogram. Once it is

Fig. 5. Vertical vanishing point extraction example. (a) Extracted scene lines divided by 5 quadrants. (b) Detected pedestrians with red vertical component and vertical matches in magenta. (c) RANSAC vanishing point estimation with red outliers and green inliers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

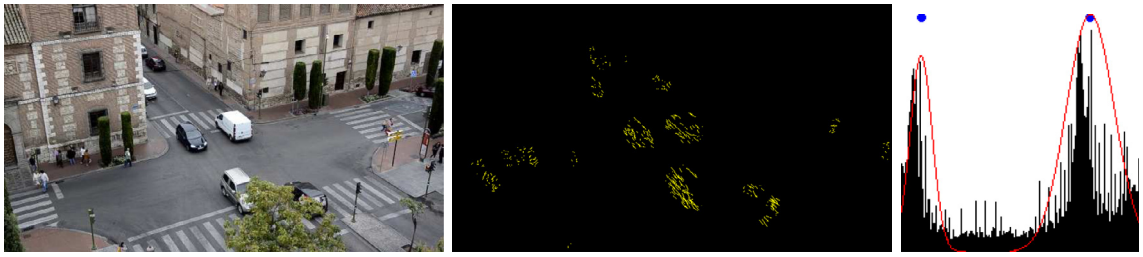

Fig. 6. Example of main motion directions extracted in a perpendicular intersection. (a) Perpendicular intersection. (b) Foreground optical flow analysis. (c) Histogram of directions and fitted gaussians in red. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

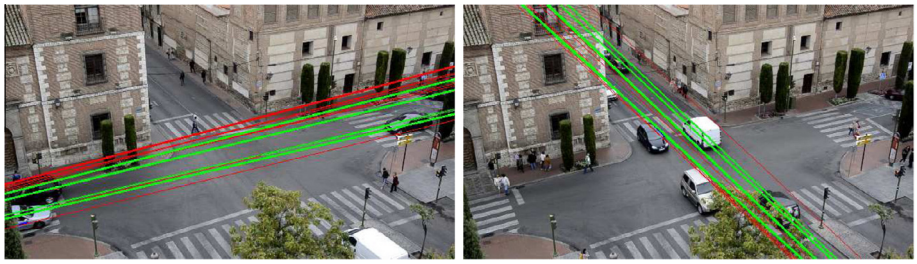

Fig. 7. Example of ground plane vanishing point extraction in a perpendicular intersection.

constructed after a determined number of frames, an EM algorithm is used to fit the histograms into gaussians in order to get the principal components of the movement. Fig. 6 shows an example of a perpendicular intersection, where the features of the foreground objects are tracked by optical flow and the motion direction histogram with the gaussian components in red is computed. The vertical axis corresponds to the frequency of the angle, and the horizontal axis corresponds to the angle value between 0° and 180°. These values are not perpendicular in image coordinates due to the perspective projection.

After getting the main directions of the scene, the motion of each foreground blob is analysed. In case of detecting motion in the computed directions, the gradients of the blob are extracted in order to look for parallel lines with the mentioned angles. Once obtaining a representative number of parallel segments, a RANSAC-based method to obtain the intersection point is used. Fig. 7 shows an example of two ground plane vanishing point extraction using the method explained in this section.

2.6. Structured scenarios vanishing point extraction

In the case of having a considerable number of architectural elements in the scene, a last option for an autonomous calibration is available (although less common and effective). It consist on extracting the vanishing points by brute force gradient analysis, assuming that the three sets of parallel lines with most number of lines are orthogonal. To group the lines, J-Linkage algorithm (Toldo and Fusiello, 2008) is used. This method is based on the work of Tardif (2009), although he does not look for orthogonal vanishing points. Fig. 8 shows the orthogonal lines extracted in a structured scenario to compute the three orthogonal vanishing points.

3. Traffic target detection and tracking

After calibrating the camera, an approximate size of pedestrians and vehicles in the image can be obtained using a standard size for them in world coordinates. This step will give the system a notion

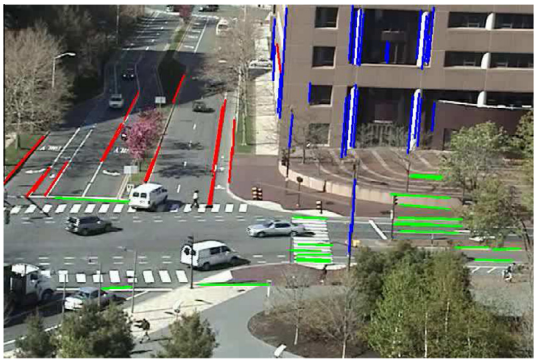

Fig. 8. Extracted lines in a structured scenario to obtain three orthogonal vanishing points automatically.

of how big are the searched elements. In this section, a multilevel framework to detect and track pedestrians and vehicles is presented. Fig. 9 illustrates the flowchart of the proposed framework, which consists of 3 levels: 1) *Image segmentation level*, to create and handle a background model and to obtain the foreground objects without image noise, camera vibrations or illumination effects; 2) *features level*, which extracts and follows features of the foreground objects; 3) *objects level*, which is in charge of managing occlusions and create and track object clusters. The first and second levels are similar than the ones described by the authors in Álvarez et al. (2012), and the *objects level* is improved in the current work as explained next.

3.1. Objects level

Usually, feature grouping works associating features directly into objects using proximity and motion history. However, the distance between two features that belong to the same object can be much larger than two features that belong to two nearby objects, which can confuse the system. To efficiently deal with the problem,

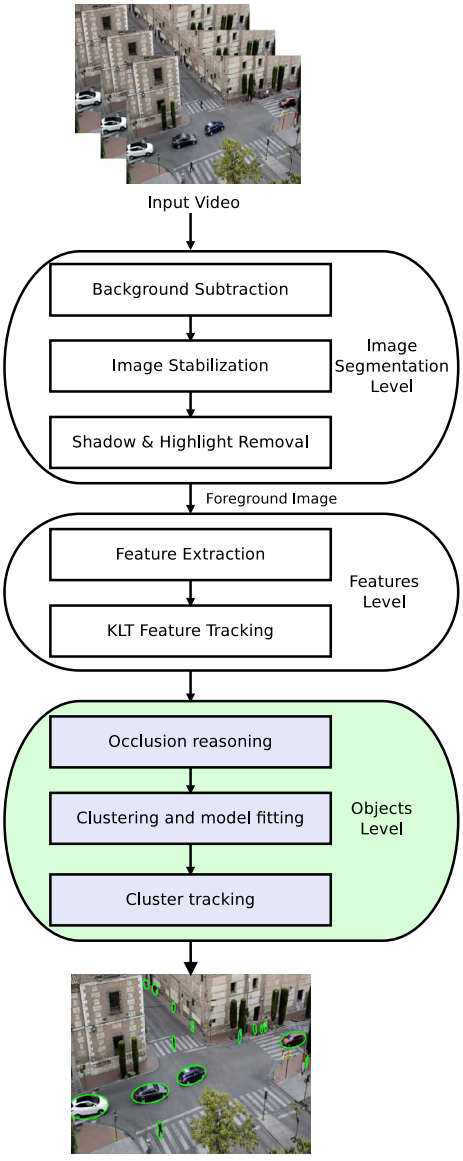

Fig. 9. Flowchart of the proposed framework to detect and track pedestrians and vehicles.

a multilevel grouping algorithm is presented. First, an occlusion reasoning step is done in order to split foreground blobs from different objects. After that, the individual features are associated to a blob and grouped into clusters depending on their motion and 3D sizes. Finally, the objects are tracked.

3.1.1. Partial occlusion reasoning

The first step when considering this problem is to observe the shapes of the objects involved in an occlusion. A common characteristic is that the shapes generated by an occlusion are not uniform: non-occluded objects are generally convex, whereas the shape of partially occluded objects become concave. An example of non-occluded and occluded objects is given by comparing their convex hull in Fig. 10.

Fig. 10. Object occlusions and convex hull. The convex hull is represented in white and the foreground blob in gray color.

It can be seen that non-occluded objects can reach a good fit by their convex hull, which does not hold for occluded objects. Accordingly, if there is an approximate idea of the searched objects sizes (through the camera calibration), an occlusion can be figured out by studying the blob shapes and their convex hulls. In particular one simple shape descriptor has been widely used in this task: the *shape compactness*. It is an intrinsic characteristic of the object shapes defined by:

$$C = \frac{P^2}{A} \tag{2}$$

where A is the shape area and P is the shape perimeter or boundary length. This way to measure shape compactness is taken from the isoperimetric inequality (Montero and Bribiesca, 2009). The next step to evaluate if a blob is the result of an occlusion is to compare the shape compactness of the object (C_o) and the one of its convex hull (C_h). Obviously C_o is always greater than C_h, because the area of an object is smaller than the one of its convex hull, whereas the boundary length of an object is greater. Therefore, for non-occluded objects C_h is close to C_o, and for occluded ones C_h is smaller than C_o. The ratio between both descriptors is used to discriminate both situations. It is called *compactness ratio* and it is defined by:

$$CR = \frac{C_h}{C_o} \tag{3}$$

Another parameter used to detect occlusions is the *convexity*, determined by the ratio between areas as:

$$R_A = \frac{A_o}{A_h} \tag{4}$$

where A_o and A_h represent the area of the object and the area of the object's convex hull respectively. Since the denominator is always greater than the numerator, R_A is always less than one. For a non-occluded object its shape is convex and R_A is close to 1, whereas for occluded objects R_A is far less than 1.

The third estimator to consider a blob as an occlusion is its size. After calibrating the camera, the approximate sizes of the pedestrians and vehicles located in the ground plane are known. In case of occlusions these sizes will be considerably increased. If the three parameters described above indicate an occlusion, the occlusion reasoning method is run as described in the flowchart of Fig. 11.

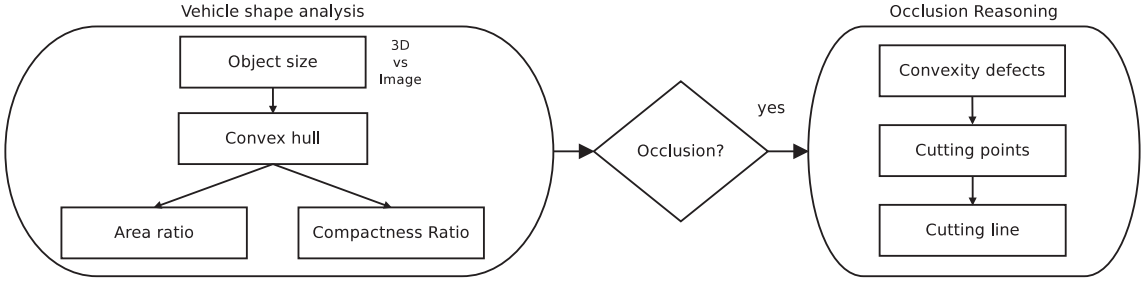

Fig. 11. Flowchart of the occlusion reasoning method.

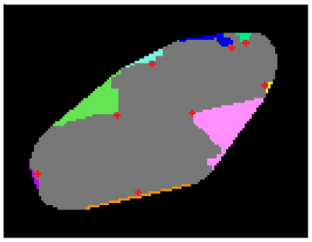

Fig. 12. Blob, convex hull and convexity defects in an occlusion example.

An useful way to understand the shape of an object contour is to compute its convex hull and convexity defects. Fig. 12 illustrates these concepts using the image of a vehicle occlusion. The gray area corresponds to the foreground blob and the coloured areas represent the different defects of the convex hull. Finally, the red marks correspond to the farthest points from the convex hull within each defect, also called *defect points*.

The distance between the farthest defect point and the convex hull is taken, and this point is selected as the first *cutting point*. The next objective is to find an optimum second *cutting point* to create a *cutting line* which separates the blob into two different objects. To extract the second point, the occluded object is sequentially cut by segments that join the cutting point with the rest of defect points. For every line, the area and compactness ratios for each new blob are computed. The chosen *cutting line* is the one that brings the maximum ratio given by the Eq. (5).

$$Ratio = \sum_{i=1}^{2} \frac{R_{A_i} + CR_i}{2} \tag{5}$$

Fig. 13 depicts some examples of occlusion reasoning using the method explained before. This procedure does not require prior knowledge but the known measures from camera calibration. By using this method, most partial occlusions can be effectively handled.

3.1.2. Feature clustering and model fitting

To group all the features from the same object, a 2-stage 3D clustering algorithm is used. First the individual features are assigned to a blob (after the occlusion reasoning) and grouped into clusters depending on their motion. Finally these clusters are grouped into objects depending on the 3D sizes and motion. Therefore, if a blob corresponds to a single object, all its features will have a similar motion and will be grouped together. Otherwise, they will be clustered into multiple objects associated to different motion characteristics. As an unsupervised stage, it is necessary to identify the number of clusters and the correspondence of the samples automatically. Mean Shift (Comaniciu and Meer, 2002) is used as a non-parametric method which does not require prior knowledge of the number of clusters, and does not constrain their shape.

The main idea behind mean shift is to treat the points in the d-dimensional feature space as an empirical probability density function where dense regions in the feature space correspond to the local maxima or modes of the underlying distribution. For each data point in the feature space, one performs a gradient ascent procedure on the local estimated density until convergence. The stationary points of this procedure represent the modes of the distribution. Furthermore, the data points associated with the same stationary point are considered members of the same cluster. The quality of the output is controlled by a kernel *bandwidth*, and it is not critical due to objects moving with different angles or velocities generate features with a strong different component. Fig. 14 depicts an example of the feature clustering step.

As mentioned before, an approximate size of vehicles is known thanks to the information provided by the camera calibration. Therefore, a vehicle which has been split into several blobs due to errors in the foreground or a misclassified occlusion can be merged. If the clusters fits into the 3D size of an standard target in the corresponding 2D coordinates and have similar motion, the clusters are merged into a final object. Fig. 15 shows an example of blob merging after splitting the blob due to an occlusion with a tree.

3.1.3. Cluster tracking

After detecting consecutively a cluster several times, a tracking stage combined with a multi-frame validation process takes place. This final step is used to reinforce the coherence of the detected objects over time, obtaining a more stable position, avoiding occlusions in case the previous methods fail, and minimizing the effect of both false-positive and false-negative detections. The multi-frame validation and tracking algorithm relies on the Kalman filter theory in 2-D space, with a state vector based on the ellipse parameters: centroid, axis and angle. For the data association, Hungarian assignment is used. Fig. 16 depicts a performance example of the detection system after the steps described.

Fig. 13. Examples of occlusion management by the proposed algorithm.

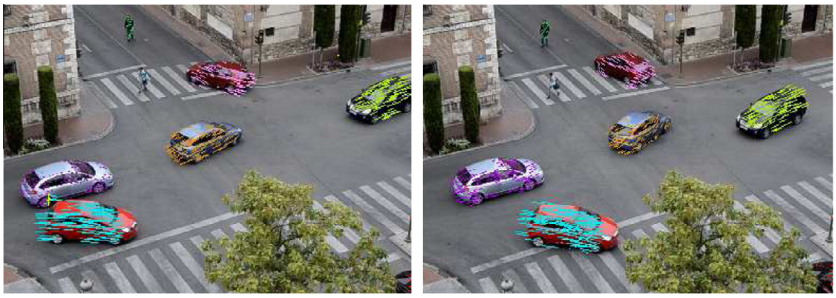

Fig. 14. Examples of feature clustering represented by coloured features.

Fig. 15. Examples of cluster merging.

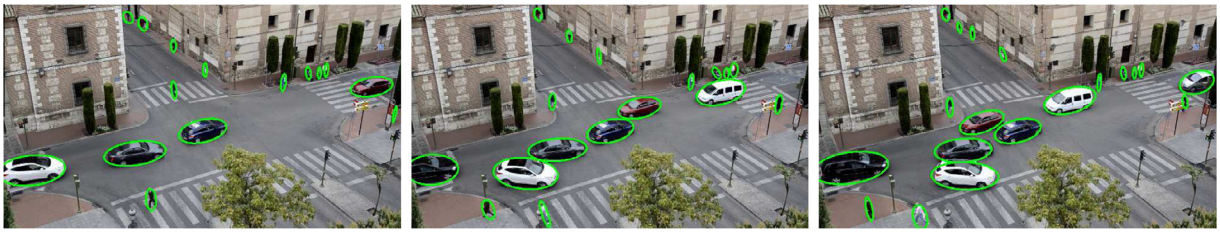

Fig. 16. Example of the target detection system.

4. Experimental results

4.1. Camera auto-calibration

For the auto-calibration method, 30 sequences from different scenarios and conditions have been used, testing the 12 cases of the hierarchical tree. As a result, a comparative table (Table 2) has been constructed with the average errors of the main intrinsic and extrinsic parameters extracted (focal distance, pitch and roll), compared to the cases 5 and 12 which are considered the ground-truth. Yaw is not used because its variation does not modify the ground plane and does not have impact into the 3D projection.

As can be seen, case 1 is the best solution due to the strong parallel component of their orthogonal elements and the zooming chance. Near it, cases 6 and 7 have similar results. It was expected because they are the relative cases to the first one, but without zoom. On the other hand, the worst options are cases 3, 4, 10 and 11, based on perpendicular intersections (not always available or strictly perpendicular), and structured scenes (not always with strictly orthogonal components).

The obtained results are really satisfactory: the low error obtained proves the strength of the system, and the multiple options of the hierarchical tree provide high versatility to cover most of the possible traffic scenarios. Furthermore, the system is able to adapt the calibration parameters in case of PTZ camera displacements without manual supervision. Even if there is no chance to auto-calibrate the camera (due to absence of orthogonal components), the manual input of lines remains as a valid option which allows the user to control the system in a short time.

Table 2
Auto-calibration errors comparative table.

Case	Focal (%)	Pitch (°)	Roll (°)
12/5	0.00	0.00	0.00
1	**2.29**	**1.68**	**0.30**
2	4.69	2.83	0.34
3	5.14	2.55	0.65
4	6.68	3.05	0.67
6	3.52	1.46	0.51
7	3.88	2.05	0.51
8	4.05	2.25	0.69
9	4.40	2.57	0.26
10	7.47	3.11	0.64
11	7.18	3.16	0.65

4.2. Target detection system

Firstly, the performance of the proposed object occlusion reasoning framework has been quantitatively evaluated. The results are summarized in Table 3, separated by occlusion class and depending on the level of the algorithm where they were detected and managed. *Detected* columns stand for the number of occlusions detected by each level, and *handled* is the number of occlusions correctly managed. The *total* column contains all the detected

Table 3
Quantitative evaluation of the occlusion reasoning framework.

	Occlusion level		Clustering level		Together			
	Detected	Handled	Detected	Handled	Detected	Handled	Total	Rate
Ped& Ped	226	213	11	11	237	224	251	0.89
Car& Car	124	115	19	18	143	133	142	0.93
Car& Ped	53	51	29	26	82	77	85	0.90
Result:	403	379	59	55	462	434	478	0.91

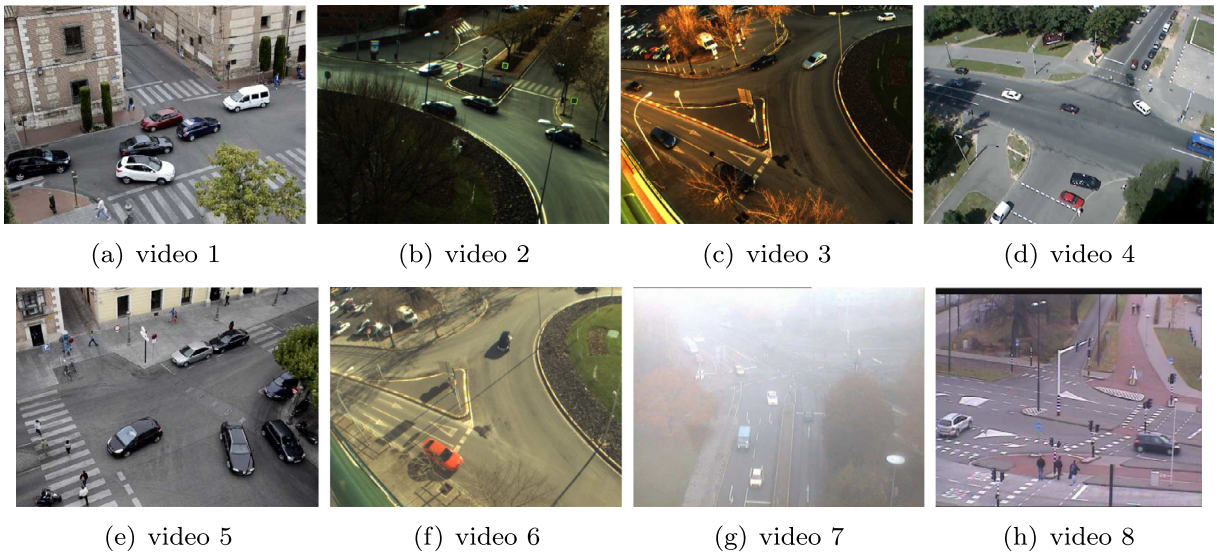

Fig. 17. Samples of testing scenarios.

and non-detected occlusions and it is used to evaluate the rate values as Handled/Total. Occlusion level always takes part in the process and only if it can not detect or handle the occlusion, the algorithm passes through the next level.

The global occlusion management ratio (91%) is very reasonable. It is important to emphasize that this analysis is single frame. Therefore an error due to an occlusion in a particular frame is not important in the whole path of an object. Moreover, after the tracking stage this value is increased to 95% because several occlusions are managed by the multi-frame validation. The advantage of the approach is the use of a multi-level framework that allows to solve an occlusion from different and complementary points of view.

To analyse the performance of the global target detection system, the algorithm has been tested on over 2 h of traffic videos with more than 2000 objects between vehicles and pedestrians. The sequences include different camera views, illumination effects, shadows, etc., in order to evaluate the method in a wide range of situations. Some examples of the testing scenarios used are shown in Fig. 17 and described in Table 4.

The global results of the application are depicted in Table 5 in terms of object detection rate, recall and precision. The *Detection Rate* (DR) is the percentage of correctly detected objects, the *Recall* (R) measures the system's ability to identify positive samples, and the *Precision* (P) is the fraction of retrieved instances that are relevant. where TP stands for the number of true positives (objects correctly detected at least the 80% of their path), FP stands for the number of false positives (unexpected detections or object splits) and FN is the number of false negatives (missing detections).

From a total amount of 2269 objects, the system has obtained a detection rate of 93.3%, which is considered a good result and valid for the proposed application. To analyse the importance of the calibration stage, this value is compared with the one obtained without calibrating the camera as represented in Fig. 18. Firstly, the system works correctly with the parameters obtained by an initial camera auto-calibration. Between frames 44 and 54, the camera changes its angles and zooms out. Then, a comparison between

Table 4
Description of testing videos.

Video	number of frames	Resolution	Conditions	Source
video1	16402	640 × 480	Cloudy	Own sequence
video2	5244	640 × 480	Dusk (dark)	Own sequence
video3	3332	640 × 480	Dusk (bright)	Own sequence
video4	18296	640 × 480	Sunny	Lunds Univ.
video5	15921	640 × 480	Cloudy	Own sequence
video6	3585	640 × 480	Sunny	Own sequence
video7	630	768 × 576	Fog/rain	Karlsruhe Univ.
video8	4290	352 × 288	Cloudy	Candela

Table 5
Results of the target detection system. **N**: number of samples. **TP**: number of true positives. **FP**: number of false positives. **FN**: number of false negatives. **DR**: detection rate. **R**: recall. **P**: precision.

Scenario	N	TP	FP	FN	DR	R	P
Sunny (shadows)	901	832	39	32	0.923	0.963	0.955
Cloudy	885	841	23	43	0.950	0.951	0.973
Dusk	312	291	17	15	0.933	0.951	0.945
Rain/snow	171	152	17	13	0.889	0.921	0.899
Total	2269	2116	96	103	0.933	0.954	0.957

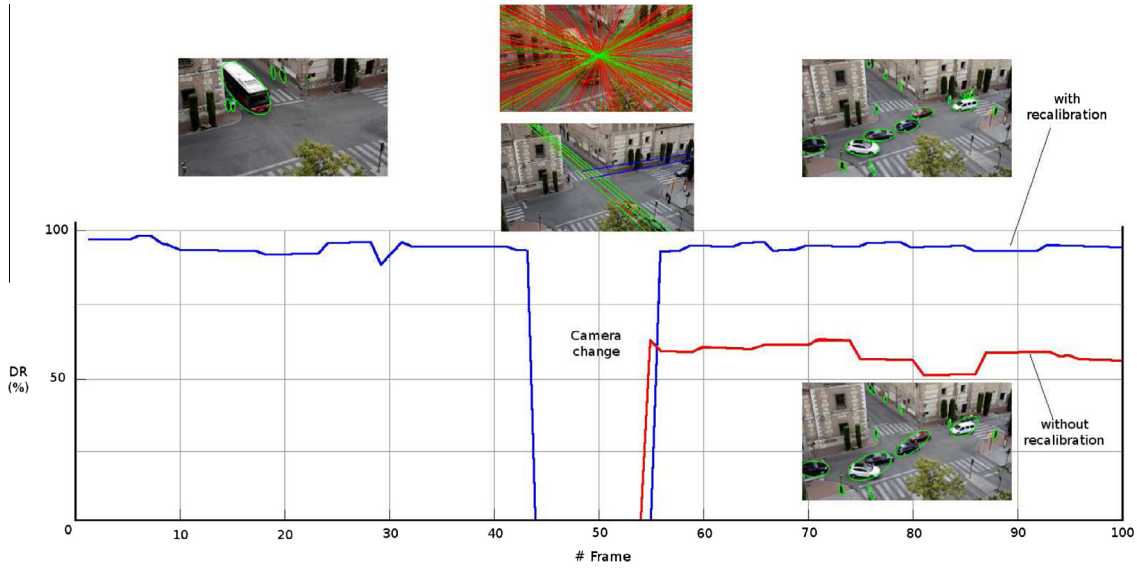

Fig. 18. Comparative example of the system with and without recalibration after a camera change.

the system with and without auto-recalibration is done (through OC computation and vanishing points extracted from a crosswalk). The blue line represents the detection rate of the auto-recalibrated system (near 93%) and the red line represents the DR in case of the calibration parameters remain constant (near 50–60%) with graphical examples. It demonstrate the need to work in fully self-adaptive mode.

5. Conclusions and future work

In this paper, a novel hierarchical camera self-calibration procedure based on vanishing points has been presented. Depending on which elements appear in the scene and the chance of using camera zoom, 5 levels have been established to determine the hierarchy of each developed method and the priority of the solution adopted. It is an important step for many possible applications, because it provides very useful information to compute an approximate size of the searched objects. In this context, a monocular system has been developed to detect and track vehicles and pedestrians for applications in the framework of Intelligent Transport Systems. Trough the auto-calibration step, the algorithm requires no object model or prior knowledge (only an approximate size of the searched objects in world coordinates), it can work indoor and outdoor, in different conditions and scenarios. Moreover, it is completely autonomous (“plug & play”), independent of the position of the camera and able to manage PTZ changes in fully self-adaptive mode.

From the results and conclusions of the present work, several future lines for each treated topic are devised. With respect to the camera auto-calibration, an interesting improvement is related to the recalibration process in case of PTZ displacements. The idea is to develop a segment tracking, to use the same set of orthogonal lines to find the new position of the previously used vanishing points. Besides that, due to the high diversity of camera views, operating conditions and observation objectives in traffic surveillance, there is an important lack of a common framework and most authors use their proprietary sequences. This condition has generated a large diverse body of work, where it is difficult to perform direct comparison between the proposed algorithms. It would be very important to generate a public traffic database, with a wide range of scenarios and conditions, to be able to make these comparatives.

Acknowledgment

This work has been supported by the Spanish Ministry of Science and Innovation by means of Research Grant ONDA-FP TRA2011-27712-C02-02.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.eswa.2013.08.050.

References

Álvarez, S., Sotelo, M. A., Llorca, D. F., & Quintero, R. (2011). Monocular vision-based target detection on dynamic transport infraestructures. In *Lecture notes in computer science* (pp. 576–583).

Álvarez, S., Llorca, D. F., Sotelo, M. A., & Lorente, A. G. (2012). Monocular target detection on transport infrastructures with dynamic and variable environments. In *IEEE intelligent transportation systems conference*.

Álvarez, S., Llorca, D. F., & Sotelo, M. A. (2013). Camera auto-calibration using zooming and zebra-crossing for traffic monitoring applications. In *IEEE intelligent transportation systems conference*.

Caprile, B., & Torre, V. (1990). Using vanishing points for camera calibration. *International Journal of Computer Vision, 4*, 127–140.

Cipolla, R., Drummond, T., & Robertson, D. (1999). Camera calibration from vanishing points in images of architectural scenes.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 603–619.

Hartley, R., & Zisserman, A. (2000). *Multiple view geometry in computer vision*. Cambridge University Press.

Hodlmoser, M., Micusik, B., & Kampel, M. (2010). Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Hue, T., Lu, S., & Zhang, J. (2008). Self-calibration of traffic surveillance camera using motion tracking. In *IEEE conference on intelligent transportation systems*.

Junejo, I. N. (2009). Using pedestrians walking on uneven terrains for camera calibration. *Machine Vision and Applications, 22*, 137–144.

Kim, Z. (2009). Camera calibration from orthogonally projected coordinates with noisy-ransac, In *IEEE workshop on application of computer vision*.

Lv, F., Zhao, T., & Nevatia, R. (2006). Camera calibration from video of a walking human. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(9), 1513–1518.

Montero, R., & Bribiesca, E., (2009). State of the art of compactness and circularity measures. In *International mathematical forum*.

Rother, C. (2002). A new approach to vanishing point detection in architectural environments. *Image and Vision Computing, 20*, 647–655.

Schoepflin, T., & Dailey, D. (2003). Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Transactions on Intelligent Transportation Systems, 4*(2), 90–98.

Tardif, J. P. (2009). Non-iterative approach for fast and accurate vanishing point detection. In *IEEE conference on computer vision*.

Toldo, R., & Fusiello, A. (2008). Robust multiple structures estimation with j-linkage. In *European conference on computer vision* (pp. 537–547).

Tsai, R. (1986). An efficient and accurate camera calibration technique for 3d machine vision. In *IEEE conference on computer vision and pattern recognition*.

Zhang, Z., Tan, T., Huang, K., & Wang, Y. (2013). Practical camera calibration from moving objects for traffic scene surveillance. *IEEE Transactions on Circuits and Systems for Video Technology, 23*, 518–533.