Automatic Calibration of a Urban Video Surveillance System through the Observation of Zebra Crossings

A. Broggi, A. Fascioli, R. I. Fedriga, and S. Ghidoni

VisLab – Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Parma
{broggi, fascal, fedriga, ghidoni}@ce.unipr.it

Abstract. In this paper, a method for automatic calibration of a camera stereo pair through the observation of zebra crossing signs is described. It is based on the well-known consideration that it is possible to obtain information about lens distortion and camera orientation by observing how a known pattern appears in the image; moreover, a major advantage of this system is that it does not require any ad-hoc calibration pattern, because it exploits the zebra crossing signs, a pattern usually present in images used for monitoring pedestrians while crossing a road. To achieve this goal, well-known techniques for removing lens distortion and perspective effect are combined with new methods for locating calibration points on the available pattern, and, finally, for evaluating the camera position.

1 Introduction

Calibration is an important issue for stereo systems, because it often happens that algorithms results depend on calibration data. Moreover, the cameras can slightly change their position over time, especially if the system runs for long periods of time, i.e. weeks or months. Therefore, an automatic method capable of understanding the cameras positions, and updating information about them, can be extremely helpful to keep a vision algorithm fully functional.

In the case discussed here, camera calibration is needed by a system whose aim is to monitor pedestrians walking on a zebra crossing. The whole system setup is depicted in Fig. 1: two near infrared (NIR) cameras are mounted on a metal rod, at about 6 m from the ground, looking down; two NIR illuminators help keeping the scene visible also at night. Images size is 768×288 (only one field is used).

The system locates obstacles on the road, and then classifies them in order to detect pedestrians that are crossing the road, measures the time they need to cross, and modifies the traffic light cycle time accordingly, which is the ultimate goal of the whole system. The obstacle detection phase is obtained using a stereo matching algorithm, a well-known technique that compares how the same object appears in the two images, and obtains from that its distance from the cameras.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8

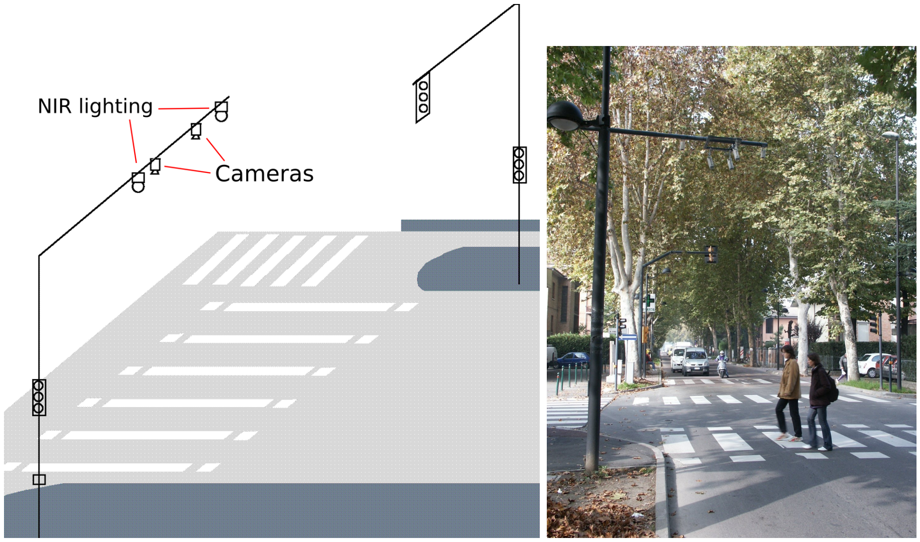
This processing aims to locate objects emerging from the ground plane; therefore, it does not need to exactly measure the distance between the obstacle and the system. When only one point of view is available, distinguishing between an obstacle and its shadow is not a trivial task, but, when dealing with two cameras, it becomes substantially easier, and the results are considerably more reliable. System calibration becomes even more complex when the scene is observed from a camera whose optical axis is almost perpendicular to the ground plane, so that the ground occupies the whole image, and not only a portion of it: in this case, shadows can appear everywhere in the image, and may even extend over a large area depending on where the light source is placed. In real cases, finally, a large number of shadows may appear, due not only to the obstacles appearing in the scene, but also to a great variety of environmental characteristics, like close-by trees, as it can be observed in Fig. 2, showing one of the images taken from the experimental setup used for developing and testing the system.

As previously said, stereovision allows to solve in a simple way some of the main issues that affect monocular systems, but presents an additional cost: to make the comparison between the two stereo images feasible, the two cameras should be correctly placed and carefully oriented. These requirements may be easily met in some cases, but are almost impossible to satisfy in others. Furthermore, in the system being presented, a fish-eye lens must be chosen to guarantee the whole zebra crossing area appears in both images, also when the cameras are placed at a small distance to the ground. Unfortunately, this kind of lenses causes a great distortion, as can be seen in Fig. 2.

There are many other problems when placing cameras in real scenarios: for example, it may not be possible to place them right in the center of the road. In this case the whole crossing is not visible if the optical axes are kept perpendicular to the ground. It is therefore necessary to introduce a deviation from the ideal camera orientation, and install the cameras at an angle, thus introducing an additional perspective distortion. Finally, since the cameras are placed outdoors, temperature variations and vibrations may slightly modify the cameras orientations.

The discussion above highlights that it is almost impossible to completely control the setup; however, working with distorted images taken by a non optimally oriented stereo pair is not easy. The system being presented aims to automatically solve all these problems by observing how the crossing signs (white rectangles) appear in the images, assuming they are all parallel, and of the same size.

The whole algorithm can be divided into three main steps: in the first one, discussed in Sect. 2, the images are analyzed to locate the zebras, and some points on their edges. In the second stage, described in Sect. 3, the coordinates of these points are analyzed in order to compensate for lens distortion. Finally, corresponding points in the undistorted image are further analyzed to evaluate camera orientations, as described in Sect. 4. After all the above processings have been completed, a look-up table is created, which enables subsequent frames to



(a) (b)

Fig. 1. System setup in via Adua, Reggio Emilia, Italy: scheme (a) and picture (b).

be quickly processed. Experimental results are presented in Sect. 6. Some final considerations on the whole system are derived in Sect. 5.

2 Zebra Detector

The first module of the algorithm, called *zebra detector*, looks for the signs painted on the road. The problem of detecting zebra crossings has already been discussed in [1] and [2], but these studies were developed for finding the crossings in the way they appear to a walking person. In the present case, the appearance of the crossing is different, and the signs need to be carefully detected, because the calibration result strongly depends on how they appear in the image. Then, for each sign, some calibration points are selected on the edges. There is no particular requirement on the number of points, nor on their spacing, which can be variable, but some tests on real situations have shown that a reasonable number is around 10 for the long side of the sign, and 6 for the short one. The number of points on each edge should be enough to describe both the distortion and the perspective effect that must be removed: so, the minimum is 3 points for each side, but a larger number is highly desirable to obtain good results.

At this point, misdetection must be carefully controlled: although some small errors on the points positions may be tolerated, and some points can be missed, the erroneous detection of points that do not lie on a sign edge can jeopardize the whole calibration, and therefore must be avoided. On the other hand, the

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

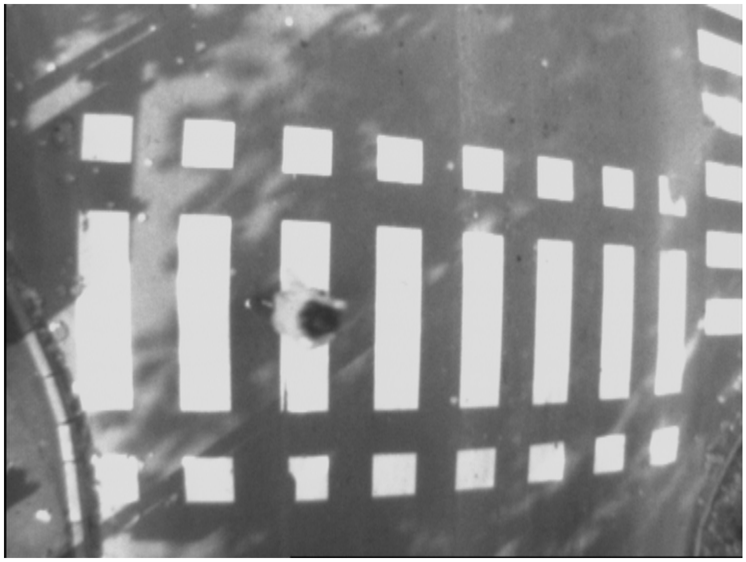


Fig. 2. Image taken from the experimental setup. Nearby trees cast shadows on the ground; a crossing pedestrian is also visible.

recognition criteria should not become too strict, otherwise too few points will be found.

To obtain a high accuracy, the zebra detector first locates the white rectangles painted on the road, then computes the edges in the image, and finally selects only those enclosing very bright and almost uniform areas. Every step has its own requirements. For instance, signs should have a rectangular shape with a precise aspect ratio, and a uniform and light grey level; then, a Sobel operator is applied to the image, and the detector selects the edges that are straight, long and thin, and close to a sign.

In this way, for each sign some calibration points are found, on its four sides. They are organized in groups, so that each one contains points that are aligned in the real world, but not in the image, because of lens distortion. There is a group for each long side of a sign, while points on short sides of all signs are divided into two groups. Figure 3 shows the selected calibration points; it can also be seen that in this case, the shape of the white areas corresponding to the leftmost and rightmost signs is not compliant with the requirements, due to the specific sun-shadow pattern. Therefore, these signs are not detected.

3 Distortion Removal

Once the zebra detector has found the calibration points and grouped them in lines, it provides this result to the subsequent stage of the algorithm, whose

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

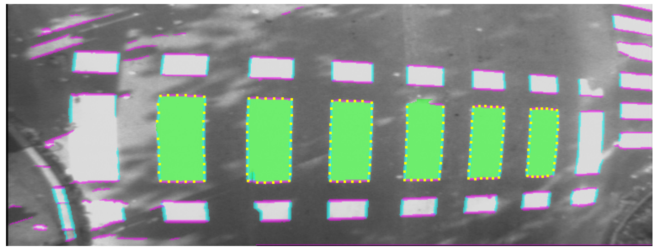


Fig. 3. Result of zebra detection. In green appear areas recognized as signs; cyan and magenta are their sides. In this case 174 calibration points are selected and here depicted in yellow.

task is to remove the distortion introduced by the lens. In [3] this problem is extensively approached, and several lens models are discussed. It turned out that the most suitable one for the fish-eye lenses used in this application is the FOV (field of view) model, that describes the distortion as a radial movement of each point, whose entity depends on its distance to the center of the image. The expressions of the distortion function and its inverse are:

$$r_{\mathrm{d}} = \frac{1}{\omega} \arctan\left(2 r_{\mathrm{u}} \tan\frac{\omega}{2}\right) \quad , \tag{1}$$

$$r_{\mathrm{u}} = \frac{tan\,(r_{\mathrm{d}}\omega)}{2\tan\frac{\omega}{2}} \quad . \tag{2}$$

These equations describe where each pixel should be moved in order to generate the distorted or undistorted image; r_{d} is the distance of a pixel to the image center in the distorted image, while r_{u} has the same meaning in the undistorted image. Therefore, using the above equations it is possible to evaluate r_{d} by knowing r_{u}, and vice-versa.

The FOV lens model depends on one parameter only, ω, that describes the entity of the distortion. It is therefore necessary to find the value of such parameter so that the undistorted image can be obtained by applying (2) to all points. The image is undistorted when all calibration points aligned in the real world are also aligned in the image; it is thus necessary to measure the alignment of points. This is done by the linear least square fitting algorithm, which finds both the best-fit line, and R^2, the sum of the squares of the deviations, a parameter measuring how good is the approximation.

The FOV lens model and the least square fitting are combined together to remove the distortion. A rough estimation of the value of ω is empirically performed, then a minimization algorithm is used to find the value of ω which minimizes a weighted sum of R^2 of each line; the weights are used to give more relevance to those lines that have a higher number of points. The value of ω obtained by the minimizer is put in (1) and (2) to obtain a completely specified lens model; the minimization ensures this is the one offering the best line rectification. The lens model is then used to generate a Look-Up-Table (LUT), that

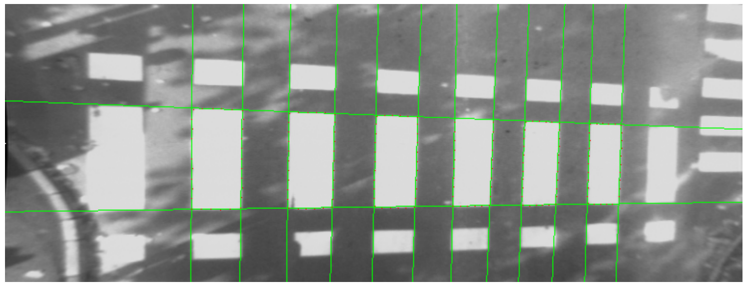


Fig. 4. Undistorted image. The green lines are the best-fitting lines.

is used to quickly remove the distortion from the images. The result of distortion removal is shown in Fig. 4: the green best-fit lines appear, showing that a perspective effect is still present in the image.

4 Perspective Removal

Although after the distortion has been removed the image offers an easier access to the information it contains, the removal of the perspective effect can further simplify the following processing steps. To do that, *Inverse Perspective Mapping* (IPM) [4] has been chosen: it is a method capable of removing perspective, once the camera intrinsic and extrinsic parameters are known. It is in fact an image processing technique that mimics a camera movement, so the resulting image is similar to one that is taken by a camera in a different position. Indeed, it is possible to successfully remove the perspective effect only when the geometry of the world is known; is this case the road is assumed to be flat. Any deviation from this assumption leads to an incorrect IPM. The analysis of how real 3-D objects appear in an image is a key topic of computer vision: in particular, in [5] is discussed the problem of inferring data about the real world by observing how objects appear in an image; while [6] represents a well-known complete survey on this topic.

To obtain an automatic calibration system that exploits the IPM technique, a method has been developed to estimate the camera orientation, again looking at the zebra signs. It is based on the observation of two vanishing points, that can be found in the image as the intersection points of all signs sides; in particular, all the short sides can be connected by two lines only, so just one intersection point is present. The long sides, on the other hand, are many more, so every pair of lines provides an intersection: their center of mass is the point used for the calibration. Because the camera optical axis is almost perpendicular to the ground, the two vanishing points will not be visible in the image.

Every camera has three degrees of freedom, described by the three angles yaw, pitch, and roll. However, the perspective effect is present when the optical axis of the camera is not perpendicular to the ground, so it suffices to control two

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

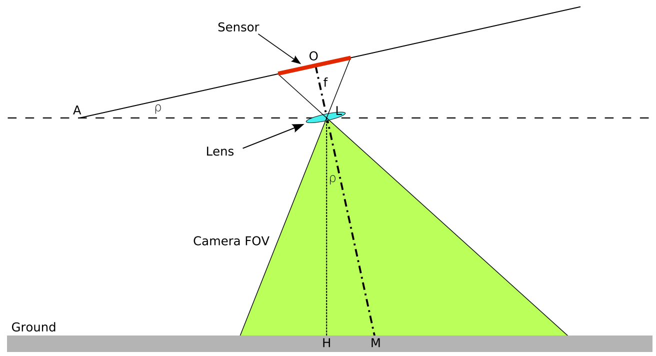


Fig. 5. Scheme explaining how the orientation angles can be derived by the analysis of vanishing points.

angles to remove it; the third angle controls the image rotation. Some geometrical considerations were made to extract information about the camera orientation from the vanishing points found in the image.

To simplify the explanation, a two-dimensional scheme is used to analyze the case of only one vanishing point, so just one angle has to be evaluated; in real cases, both angles can be computed independently and in the same way. The scheme is shown in Fig. 5: the red line segment represents the camera sensor, OM is the optical axis, and ρ is the rotation angle that must be estimated. The triangles LHM and AOL are similar, because they are both rectangled, and $O\hat{L}A = L\hat{M}H$, since they are corresponding angles formed by two parallel lines cut by a transversal. Therefore, the rotation angle ρ can be found by looking for the vanishing point A as described before, calculating the distance to the image center $\overline{AO}$, and finally applying the equation:

$$\rho = \arctan\left(\frac{\overline{OL}}{\overline{AO}}\right) \quad . \tag{3}$$

The segment $\overline{OL}$ is the focal length, so it is supposed to be known.

The accuracy of the camera orientation analysis depends on the precision of the vanishing points detection. In the present case, the perspective effect is very weak; for this reason, the convergence of the sign sides is mild, and even a small error in evaluating a calibration point can sensibly change the location of the intersection of the two lines. However, using the center of mass of all intersections when more than two lines are available reduces such variability.

The described technique gives good results, but, to obtain an even better performance, a further processing has been applied. It is, again, a minimization, capable of changing the camera angles in the IPM. As previously said, two groups of lines that are known to be parallel in the real world are available; this can be exploited by the minimization, that can try to minimize the variance of the angular coefficients for both groups. Finally, another minimizer was used, to

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

find the third camera orientation angle so that all signs in both images appear vertically aligned. Again, a LUT is generated to apply the same transformation to all frames taken from the same camera.

The overall result can be seen in Fig. 6: it represents the frame of Fig. 3 after the application of the two LUTs (undistortion and perspective removal). It looks like the image was taken by a camera whose optical axis is perpendicular to the ground, and without any distortion.

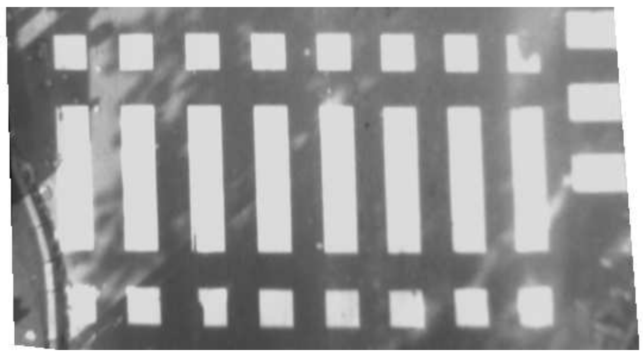


Fig. 6. Result of the perspective removal algorithm applied on the undistorted image of Fig. 3.

5 Experimental Results

The system described so far is a set of automatic procedures that are capable of providing ready-to-use stereo images starting from poorly aligned and non-specifically oriented cameras. It is intended to work in systems that monitor pedestrian crossings, because the main idea is to exploit the fact that both cameras observe a regular and known pattern, from which information about camera orientation and lens distortion is obtained. The ultimate purpose of this system is to help stereo algorithms to work always with well calibrated images, without caring about lens distortion or camera orientation. The latter, in particular, is an important issue, because a zebra crossing surveillance system is supposed to be mounted outdoor, and subjected to vibrations, high temperature variations, and atmospheric events, that can slightly change the cameras position in a way that is not predictable. A daily recalibration may solve these problems.

Dedistortion and perspective removal have been evaluated looking at the parameters on which the minimization is performed. In the case of the dedistortion, the accuracy can be evaluated by looking at the sum of the squares of the deviations of all points, divided by the number of points: this value can be as low as 3.3. The camera orientation is evaluated by minimizing the variance

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

of the angular coefficient of the vertical and horizontal lines approximating the calibration points; this variance turns out to be negligible. The precision of the overall system can be evaluated by looking at how two stereo images are aligned. In Fig. 7 (a) and (b) the stereo input images are shown; to qualitatively evaluate the precision of the system, its output images are centered on the same point, and their difference is computed; the result is shown in Fig. 7 (c): the misalignment of the signs is less than 3 pixels. In Fig. 7 (d), finally, the difference image is computed while a pedestrian is crossing the road: the corresponding pattern is clearly visible.

It is interesting to analyze how the calibration quality depends on the number of points found on the signs. For example, in the case of Fig. 7 (a) and (b), all the signs, except the one at the very left, are recognized by the zebra detector; the calibration points are 232 and 229 for the left and right images, respectively: this working condition can therefore be considered ideal. However, if some obstacle, like a car, hides some of the signs, the number of calibration points dramatically reduces: in (e), they are 129, because only four signs are found. In spite of this fact, the calibration still gives reasonable results, as can be seen by Fig. 7 (f), again showing the difference image when the calibration is based on a reduced number of points. It should be noticed that, even if the points are sensibly fewer that in the optimal case, many of them are very distant from the image center, where the lens distortion is higher, so they can still give an accurate description of the lens properties. In other cases, when a similar number of calibration points are concentrated towards the image center, the calibration error considerably grows.

6 Conclusions

A method for automatic cameras calibration has been proposed. Its aim is to evaluate lens distortion and perspective effect, and to remove them without requiring dedicated procedures or the use of calibration grids, but, rather, exploiting the pattern given by zebra crossings. This method was developed for a pedestrian crossing monitoring system based on stereovision. The distortion removal algorithm is capable of measuring the distortion of the lens, and to compensate it; it needs to be run only once for each camera pair, and was integrated in the system calibration setup procedure, so that the same pedestrian detection system can work with different cameras and lenses, whose choice depends on the crossing characteristics. The second main part of the system evaluates the perspective effect. This is useful since it is difficult to precisely align cameras mounted over a road crossing, and, moreover, it is used to compensate for slight drifts during the operational time of the system.

To obtain high performance in the largest number of different conditions, this method does not require any specific calibration point topology; rather, it is only based on the analysis of points alignment.

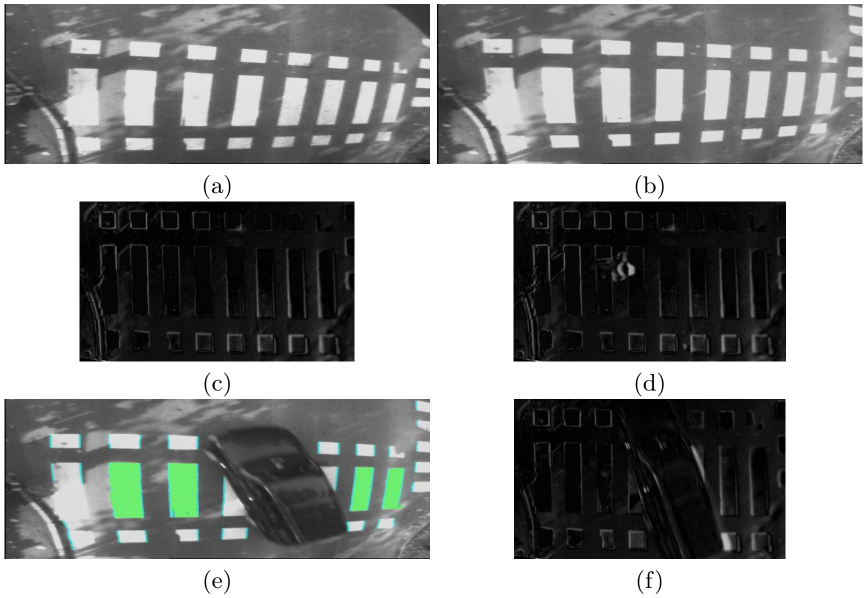


Fig. 7. The left (a) and right (b) input images, and the difference image of the two corresponding outputs (c). (d) shows the difference image when a pedestrian is crossing the road, that is the case of Fig. 2. In (e), system calibration in a difficult case, since a car is hiding some of the signs, reducing the number of calibration points. In (f) the difference between the left and right images, obtained with the calibration based on such low number of points, is still acceptable.

References

1. Se, S.: Zebra-crossing Detection for the Partially Sighted. In: Procs. IEEE Intl. Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, SC, USA (June 2000) 211–217
2. Uddin, M.S., Shioyama, T.: Robust Zebra-Crossing Detection using Bipolarity and Projective Invariant. In: Procs. 8^{th} Intl. Symp. on Signal Processing and Its Applications, Sidney, Australia (August 2005) 571–574
3. Devernay, F., Faugeras, O.D.: Straight Lines have to be Straight. Machine Vision Application **13**(1) (2001) 14–24
4. Bertozzi, M., Broggi, A., Fascioli, A.: Stereo Inverse Perspective Mapping: Theory and Applications. Image and Vision Computing Journal **8**(16) (1998) 585–590
5. Wolfe, W.J., Mathis, D., Sklair, C.W., Magee, M.: The Perspective View of Three Points. IEEE Trans. on Pattern Analysis and Machine Intelligence **13**(1) (January 1991) 66–73
6. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. The MIT Press, cambridge (1993)

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8