Simple Single View Scene Calibration

Bronislav Přibyl and Pavel Zemčík

Brno University of Technology, Faculty of Information Technology, Graph@FIT,
Božetěchova 2, 612 66 Brno, Czech Republic
{ipribyl,zemcik}@fit.vutbr.cz
http://www.fit.vutbr.cz/research/groups/graph

Abstract. This paper addresses automatic calibration of images, where the main goal is to extract information about objects and relations in the scene based on the information contained in the image itself. The purpose of such calibration is to enable, for example, determination of object coordinates, measurements of distances or areas between objects in the image, etc. The idea of the presented work here is to detect objects in the image whose size is known (e.g. traffic signs in the presented case) and to exploit their relative sizes and positions in the image in order to perform the calibration under some assumptions about possible spatial distribution of the objects (e.g. their positioning on a plane in the presented case). This paper describes related research and the method itself. It also shows and discusses the results and proposes possible extensions.

Keywords: Scene calibration, scene reconstruction, Euclidean reconstruction, single view, automatic parameters estimation, principal component analysis

1 Introduction

The quality of cameras as well as computational power of devices has been increasing steadily in last few years which has led to a growing interest in computer vision methods and their utilisation in a variety of applications. Many applications exist which do not need to work with calibrated images; however, some applications demand scene calibration in order to perform successfully. Mobile robot navigation, forensic engineering, object parameters estimation, or different kinds of scene reconstruction are typical classes of algorithms which benefit from the calibrated scene.

A dependence exists between precision of the calibration process and the amount and quality of input information required for it. On one side of the scale lies the approaches such as image-based rendering, which do not extract too much geometric information. They are not very precise in terms of the reconstructed 3D structure, but they do not have special demands on the input data. On the other side of the scale lies the more precise methods, such as stereometry, which can be very accurate but can also demand complex and exact input data (i.e. precise camera locations and their internal parameters). A similar dependence often arises between the precision and complexity of the calibration process and

further – between the complexity and type of structure which can be extracted from the scene. Methods generating projective description of the scene tend to be simpler than methods working with an affine description. However, Euclidean reconstruction usually requires far more complex methods [1].

We developed a simple method of scene calibration which does not have comprehensive demands on the input data. It requires only one image of the scene without knowledge of either external or internal camera parameters. However, objects of a known size must be present in the scene and their distance from a planar surface or ground-plane must be known. Also, some basic assumptions about the camera are made. Our goal is to obtain a partial Euclidean description of the scene. In order to be more specific, we want to estimate certain parameters of objects in the scene, such as their size, distance between them, or sizes of specific areas in the scene. This can be achieved by estimating parameters of the ground-plane and real object positions.

1.1 Related Work

As uncalibrated images allow only projective reconstruction, either camera or scene calibration is needed to obtain an affine or Euclidean scene description. Scene calibration is a task of setting-up a relation between real-world coordinates and image coordinates by exploiting the scene constraints. This task has often been confused with (internal or external) camera calibration, which is a different task altogether and is generally not needed for scene calibration.

Methods of scene calibration vary considerably, mainly between single and multiple-view approaches. When using two views of a scene, enough information is provided for depth reconstruction. More than two views help us to reduce uncertainty or allow us to check the consistency of features matched between individual images [2]. When the camera motion between views is known, we speak about stereo vision and principles of epipolar geometry that can be exploited. Such methods are able to produce an Euclidean description of the scene but often require knowledge of internal camera parameters in addition to external ones - i.e. Salman and Yvinec [3] produce a highly accurate scene representation in the form of a triangle mesh. When camera motion between views is unknown, we speak about structure from motion problem where epipolar geometry is also very useful. Szelisky and Kang [4] do not perform camera calibration which makes only projective reconstruction possible, but their achievement was to recover a dense depth map of the scene from multiple views. Koenderink and Van Doorn [5] use just two views for affine structure recovery, but they require the internal camera parameters to be known. Christy and Horaud [6] showed in their work that it is possible to obtain even a Euclidean scene description efficiently. Multiple views of the scene are often not available in practise, so single view techniques are needed.

In general, one view alone does not provide enough information for a complete 3D reconstruction. Ambiguity has to be resolved by using some apriori information (i.e. by utilising geometric relations of objects in real world). Avitzour

assumes that objects rest on a planar ground and that the camera internal parameters are known in his calibration procedure [7]. Based on these preconditions, it is possible to estimate parameters of the ground-plane. Criminisi et al. does not need to know anything about the camera, he only needs some special geometric primitives like parallel lines to be present in the image. It is then possible to determine the vanishing line and vanishing points with use of affine geometry and perform affine measurements in the scene [8]. Similarly Masoud and Papanikolopoulos use regular geometric primitives like directional roadway markings to calibrate traffic scenes [9]. Huynh does not work with either external or internal camera parameters; however he specialises in scenes which contain some symmetrical objects or object configurations [10].

Our method described in the following sections works with only a single view of the scene and it does not require knowledge of any camera parameters. It is able to calibrate scenes without regular geometric primitives, in contrast to the majority of other single view methods.

2 Proposed Calibration Method

Let us suppose that objects in the image, whose approximate size is known, exist and that they rest on a planar surface or their distance from that surface is known. The goal of scene calibration is to find optimal projection parameters in the described case. These parameters affect backprojection of image coordinates of captured objects onto real-world coordinates. Optimal projection parameters are achieved when backprojected points form a plane. Because 3 points in any configuration in the space define a plane, at least 4 points (i.e. objects of the known size) are needed to determine optimal parameters of the ground-plane and to successfully perform the calibration.

2.1 Projection Model

We use a model of perspective projection because images acquired by most of the cameras are produced through the naturally occurring perspective projection. Since dimensions of the sensing element are not known, it is modelled as a planar lattice with an arbitrary aperture which can be arbitrarily positioned on the optical axis. As the freedom of both of these variables is redundant we choose to fix the aperture, thus working with the position of the screen in our calculations. We presume that the rays coming from the scene converge in the centre of projection and draw an image on the screen.

Three types of entities are used – entities related to real-world objects (denoted by symbols without any index, e.g. A, x); entities related to planar images of objects (denoted by symbols with one apostrophe, e.g. A', x'); and entities related to spherical images of objects (denoted by symbols with two apostrophes, e.g. A'', x''). A right-handed Cartesian coordinate system, which is attached to the camera, has been used to determine world coordinates. Its origin is identical with the centre of projection, where x-axis goes horizontally from left to

right, y-axis goes vertically from top to bottom and z-axis goes perpendicularly through the screen into the scene as shown in Fig. 1. We assume that the z-axis is identical with the optical axis of the camera (i.e. the principal point of the camera is in the middle of the image). Practically, the principal point is often displaced by a few pixels from the centre of the image. However, experiments show that inaccuracy caused by this fact is insignificant. The origin of the image coordinate system is in the middle of the image and the x'-axis and y'-axis are parallel to their real-world counterparts, x-axis and y-axis respectively. Transition from a world coordinate system to an image coordinate system is, therefore, simply done by discarding the z-coordinate.

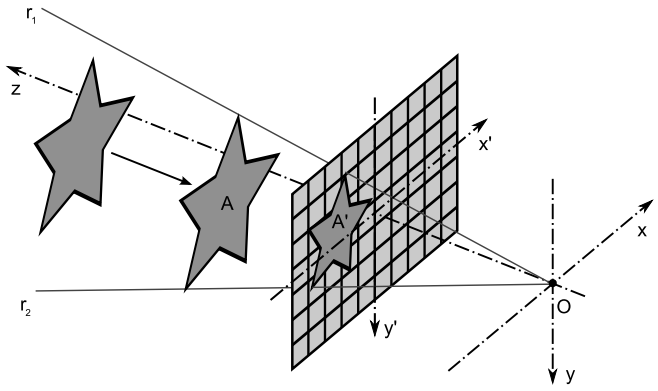

Fig. 1. Object image A' on the planar screen is backprojected using rays r_1 and r_2 which belong to a conic beam. The real distance of star-shaped object A from the centre of projection O is determined by the position in which the object clips precisely into the beam.

Real-world object coordinates are computed from image coordinates using backprojection together with a known size of real objects. The procedure can be visualised by casting rays from the centre of projection through border points of an object image. Such rays form a conic beam. The real object must be situated inside this beam. The distance of the object from the centre of projection can be determined by exploiting the fact that a real object must clip precisely into the beam (see Fig. 1 for illustration). In practice, only one ray per object has to be cast to determine its distance. This ray should go through the same point of each object (e.g. centre of gravity, bottom left corner, etc.).

In order to determine the object coordinates, it is necessary to know the scale of the map between the real object and its spherical image. The scale s is computed as a fraction of object height h and the height of its spherical image h'':

$$s = \frac{h}{h''} \ . \tag{1}$$

Sizes of both planar and spherical images of the object depend on out-of-plane rotation of the object (regarding the image plane here). If the chosen object detector does not provide information about out-of-plane rotation, it is necessary to introduce an abstraction – for example, to assume that bounding spheres of objects are detected instead of the objects themselves. In cases where the chosen object detector does not detect bounding spheres of objects, it is possible to simulate such behaviour simply: based on the knowledge of width and height of the real object, it is possible to decide which of its dimensions is less distorted in the image. The other dimension can then be appropriately enlarged so the distortion will be virtually equal in both dimensions. This abstraction causes exactly the same consequence as if all the detected objects were oriented towards the camera. Sizes of images of object bounding spheres, which are of equal size and of equal distance from the centre of projection, depend on the distance of the object from the optical axis – objects further from the optical axis has larger planar images. It is possible to eliminate this dependency by converting the planar image into the spherical image. That is why we compute with sizes of spherical images instead of planar images (see Fig. 2 for an illustration and [11] for a proof of this phenomenon).

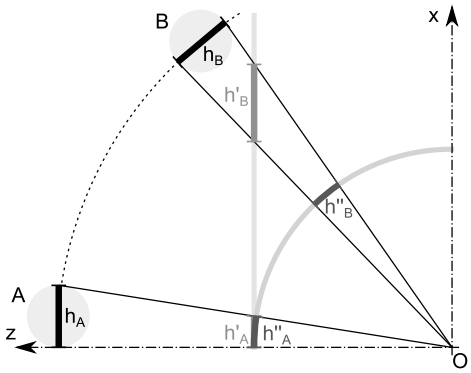

Fig. 2. Perspective projection on a plane and on a sphere. Objects A and B and their bounding spheres of equal size $h_A = h_B$ are in equal distance from the centre of projection O. Size h'_A of planar image and size h''_A of spherical image of the object A are of nearly equal size, because A is situated near the optical axis z. However, this is not true for size h'_B of planar image and size h''_B of spherical image of the object B. Sizes h'_A and h'_B of planar images of both objects differ significantly, whereas spherical images of both objects are of the same size $h''_A = h''_B$.

the height of a spherical image h'' is given by top and bottom y-coordinates y'_{t} and y'_{b} of the planar image and also by distance k of the screen from the centre of projection.

$$h'' = 2\pi k \ \cdot \frac{\left|\arctan\frac{y'_\mathrm{t}}{k} - \arctan\frac{y'_\mathrm{b}}{k}\right|}{2\pi} = k \ \cdot \left|\arctan\frac{y'_\mathrm{t}}{k} - \arctan\frac{y'_\mathrm{b}}{k}\right| \ . \quad (2)$$

Vector $\boldsymbol{v'}$ going from the centre of projection O to the planar image A' is created afterwards:

$$\boldsymbol{v'} = \overrightarrow{OA'} \ . \quad (3)$$

Consequently, this vector is normalised to the magnitude of k (which means a conversion from a planar image to a spherical image) and multiplied by the map scale s:

$$\boldsymbol{v} = \frac{k \ \cdot \boldsymbol{v'}}{\|\boldsymbol{v'}\|} s \quad . \quad (4)$$

The resulting vector $\boldsymbol{v}$ has its initial point in the centre of projection O and its terminal point in some point A of the real-world object. Thus, the object is at coordinates

$$A = O + \boldsymbol{v} \ . \quad (5)$$

The described procedure is only valid if all objects are elevated at the same known height e above the surface. If this height is nonzero, it is necessary to shift the computed plane vertically downwards by e to approximate the real ground-plane. Although the vertical direction in the scene is not known, it can very well be estimated from the average top-down orientation of all detected objects. If the assumption about no out-of-plane rotation of objects is made, as stated above, the vertical direction can then be estimated as an average in-plane rotation angle $\bar{\alpha}$ of all detected objects, where α is provided by the object detector. If the objects are elevated at different heights above the surface, it is first necessary to unify their heights in order to be able to estimate the ground-plane. The unification can then be done (e.g. by a vertical projection of the centre of each object on the surface). Such points will be called a "foot" of the object A and will be referred to as A_f. Image coordinates A'_f of the foot must be estimated by shifting image coordinates A' of the object by vector $\boldsymbol{d'}$ given by the known object elevation e and its in-plane rotation angle α.

$$A'_\mathrm{f} = A' + \boldsymbol{d'} \ , \quad (6)$$

where

$$\boldsymbol{d'} = \left[\sin\alpha \cdot e \cdot \frac{h'}{h} \ ; \ \cos\alpha \cdot e \cdot \frac{h'}{h}\right] \ . \quad (7)$$

Image coordinates A'_f of the foot will replace image coordinates A' of the object itself in further calculations, assuming the foot is approximately at the same distance from the camera as the object itself.

2.2 Finding Optimal Solution

It is possible to search for the optimal value of projection parameter k with the ability to determine real-world 3D coordinates of objects of known size for a certain value of k. The assumption that all objects rest on the ground, or are situated at the known height above it, is exploited here.

This means we search for such a k when it is possible to lay a plane through a cluster of 3D points with minimal effort (minimising an error function). Parameters of the plane can be determined by means of Principal Component Analysis (PCA) [12], because we actually search for the subspace (i.e. the plane), in which the 3D points will be orthogonally projected by minimising the error, expressed as the sum of squared distances of the points from such a plane.

First, the mean $[\bar{x}, \bar{y}, \bar{z}]^T$ of all 3D coordinates is subtracted from each coordinate to eliminate the bias of the coordinate set. This mean expresses the point, through which the resulting plane will pass. Then, $n \times 3$ matrix $\mathbf{M}$ is constructed

$$\mathbf{M} = \begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ y_0 & y_1 & \cdots & y_{n-1} \\ z_0 & z_1 & \cdots & z_{n-1} \end{bmatrix} \quad , \tag{8}$$

where n is the number of objects whose 3D coordinates have been computed. Each column of the matrix contains coordinates of one object. Therefore, it is possible to create a covariance matrix $\mathbf{\Sigma}$ of matrix $\mathbf{M}$

$$\mathbf{\Sigma} = \mathbf{M}\mathbf{M}^T \quad . \tag{9}$$

Afterwards, a triplet of eigenvalues and eigenvectors of 3×3 matrix $\mathbf{\Sigma}$ is computed by means of PCA. Eigenvectors associated with the two largest eigenvalues lie in the searched plane. The third eigenvector associated with the smallest eigenvalue is perpendicular to both of the other eigenvectors and it is a normal vector of the plane. The smallest eigenvalue is equal to the sum of squared distances of all 3D points from the given plane; thereby, it expresses a mean square error of the solution for a given k.

Because we work with a non-linear system, analytical calculation of eigenvalues and eigenvectors is very complex. Therefore, we search for an optimal projection parameter k by searching for the minimum of an error function. It has been experimentally found that the solution error as a function of k has a typical behaviour as depicted in Fig. 3. It usually has several smooth maxima alternating with sharp minima for small values of k. For large values of k, the function asymptotically approaches some value dependent on the input parameters of the calibration problem. Based on the behaviour of the error function, it is possible to find its global minimum and thereby also the optimal projection parameter k.

The function value can be sampled, for example, with an exponentially growing step. If some minimum is found, its location is refined by dividing the sampling step repeatedly. Afterwards, the step size is reinitialised and the search for another minimum takes place. If there has been a global minimum in the course of the function, this approach has led to its localization in every examined case.

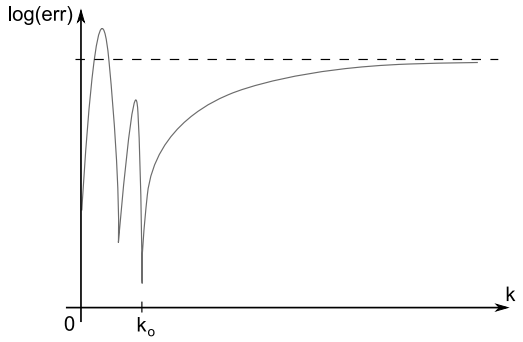

Fig. 3. Typical behaviour of the solution error (logarithmic scale) as a function of k (distance of the screen from the centre of projection). For small values of k several smooth maxima alternating with sharp minima appear. For large values of k the function asymptotically approaches some value dependent on input parameters of the calibration problem. Global minimum occurs at optimal value k_o of the projection parameter k.

3 Experimental Results

The described method has been implemented in C++ language and tested on a set of artificial as well as real scenes. Traffic signs were used as objects of known size because they appear quite frequently in urban scenes, most of them are of a unified size and good traffic sign detectors were also available. The examined scenes contained typically $4 - 7$ traffic signs. Every time the surface in the scene was planar, the minimum of the error function was found. The order of error of the optimal solution was usually about $10^{-5} - 10^{-17}$. If the scene surface was not planar but curved, the error function did not contain any minima, thus the method could not be applied. Examples of both artificial and real calibrated scenes follow in Fig. 4 and Fig. 5.

The whole procedure suffers from small inaccuracies, the main sources of them being a geometric distortion of the image caused by the acquisition process and imperfect detection of objects. The inaccuracies are further amplified when backprojecting object images back onto real-world coordinates, which are later utilised in the surface plane estimation.

If parameters of the camera are not known, geometric distortion of the image cannot be easily dealt with. However, the accuracy of the object detection can be controlled very well. The effect of detection inaccuracy on the calculation of object real-world coordinates is considerable, especially when the images of objects are small (i.e. far or small objects). See Fig. 6 for an illustration of displacement. For example, 1 px detection inaccuracy causes approximately 28 % displacement for an object whose image is only 10 px large. The detection inaccuracy affects much less objects with larger images, because the linear growth of object image size causes quadratic growth of maximal tolerable detection inaccuracy, which will cause constant displacement. When maximal acceptable displacement is,

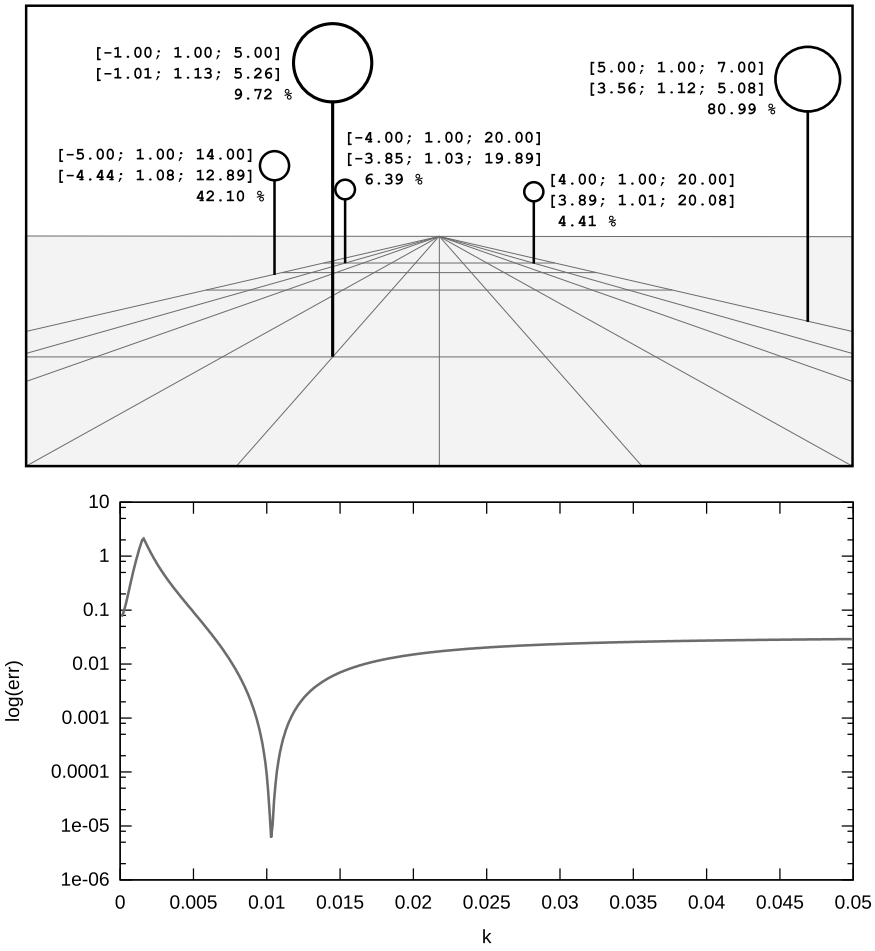

Fig. 4. Artificial scene. Top: scheme of the captured image – circles represent detected traffic signs with a diameter 0.7 m on 2.27 m long rods. The image has been constructed using perspective projection of the scene on a screen with a pixel size of 1.651 μm, which was positioned 1 cm away from the centre of the projection. Each traffic sign is marked (from top to bottom) with actual coordinates of the contact point with the ground-plane (the foot), its computed coordinates and displacement relative to the real size of each object. Bottom: course of the error function for the depicted scene. Our method has found the minimum at value $k = 0.0103192\,\mathrm{m} = 1.03192\,\mathrm{cm}$ with error $5.82 \cdot 10^{-6}$, which is very close to the actual distance of the screen from the centre of the projection.

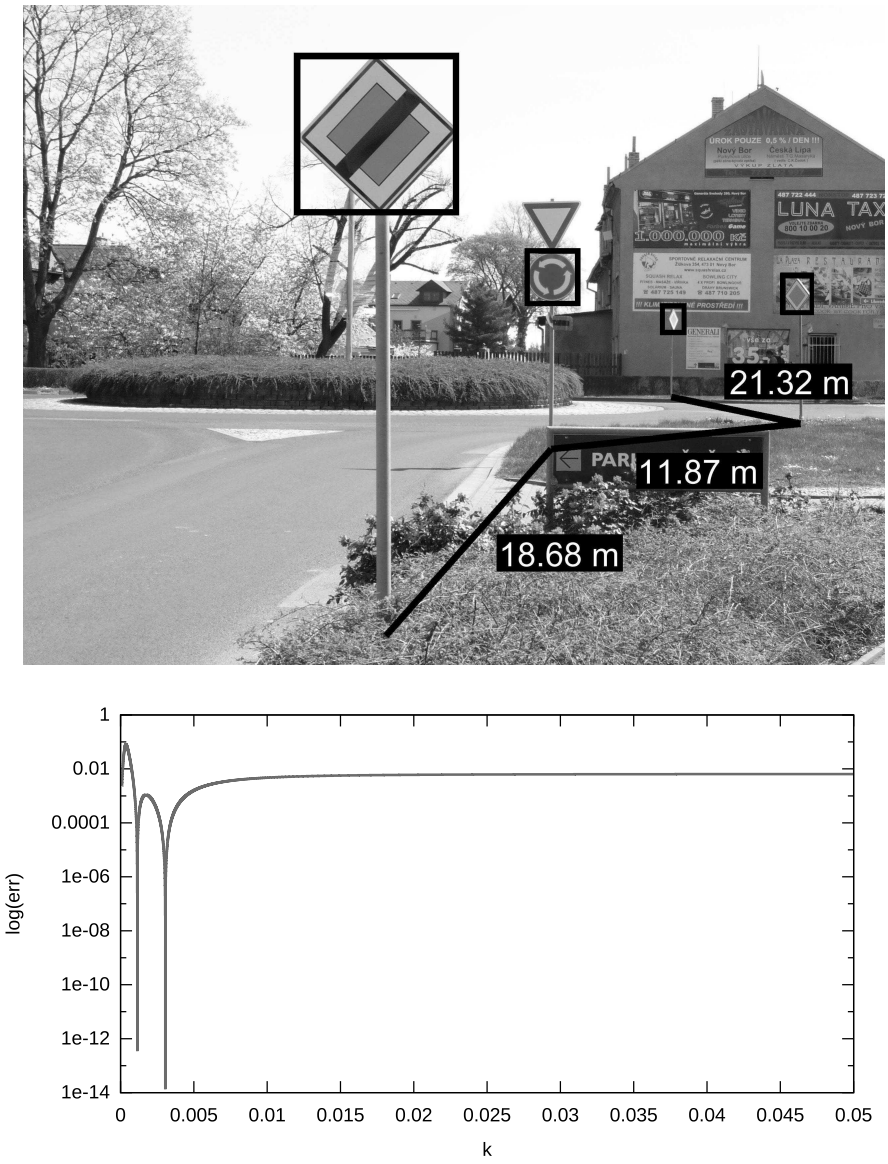

Fig. 5. Real scene with a roundabout. Top: image of the scene with 4 detected traffic signs (*black rectangles*). Computed distances between signs are stated (*black lines*). Bottom: course of the error function of the solution for the depicted scene. Our method has found the global minimum at value $k = 0.00305137\,\mathrm{m}$ with error $2.86114 \cdot 10^{-14}$. (note that the error function has 2 minima in this case). However, the ground truth is unknown in this case.

for example, 5 %, the uttermost detection inaccuracy can only be a fraction of a pixel for object images which are about 10 px large, 3 px for images that are about 40 px large and 10 px for images which are about 70 px large. Thus, it is desirable to exploit the presence of rather bigger and nearer objects when calibrating a scene. A crucial aspect of automatic calibration is also the choice of an accurate object detector or an accurate manual marking of the objects when using semi-automatic calibration.

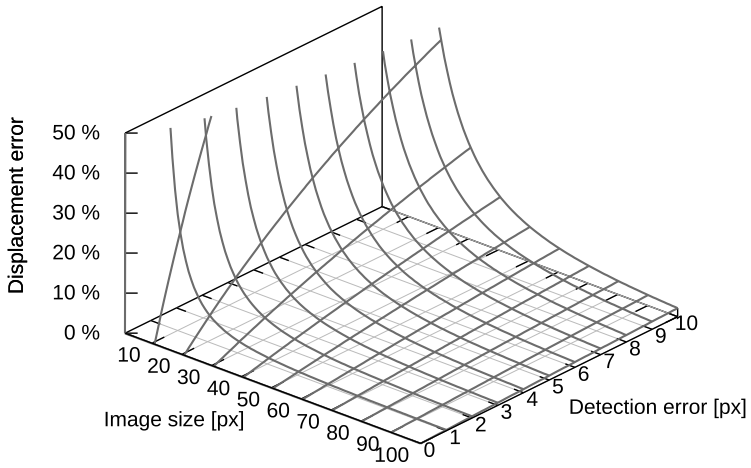

Fig. 6. Object displacement (in %) is the relative error of computed real-world coordinates with respect to object size. The displayed chart shows how displacement depends on the size of the object image and on the error of object detection. Objects with small images suffer from immense displacement while, objects with mid-sized and big images are much more resistant to object detection inaccuracies.

4 Conclusions

We developed a simple scene calibration method, which demands only a single view of the scene. The basic idea of the described method is to exploit relative sizes and positions of known-sized objects in the image under the assumption that the objects lie in a plane. This method is able to work semi-automatically if objects are marked manually, as well as automatically if object detectors are used. The calibrated scene makes possible the determination of coordinates of objects lying on the ground-plane, measurement of distances between the objects or measurement of areas between them.

Our approach is limited by the fact that it works only with planar scenes; despite this limitation, the method is practically usable in various environments (e.g. in urban environment). The prerequisite is the presence of known-sized

objects in the image and the knowledge of their positions with respect to the ground-plane.

The method has been tested on a set of artificial and real scenes. In the scenes with planar surface, a solution has been always found. The precision of the solution is sufficient for many applications, which demand a calibrated scene. Future research includes processing of the scenes with a non-planar surface and deeper sensitivity analysis of the method.

Acknowledgements

This work was supported by the European Commission under the contract FP7-215453 "WeKnowIt" and by the Ministry of Education, Youth and Sports of the Czech Republic by projects MSMT 2B06052 "Biomarker" and MSM 0021630528 "Security-Oriented Research in Information Technology".

References

1. Faugeras, O.: Stratification of 3-D vision: projective, affine, and metric representations. Journal of the Optical Society of America 12, pp. 465–484 (1995).
2. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003).
3. Salman, N., Yvinec, M.: High Resolution Surface Reconstruction From Overlapping Multiple-Views. In: Proceedings of the 25th annual symposium on Computational geometry, pp. 104–105. ACM (2009).
4. Szeliski, R., Kang, S.B.: Direct Methods for Visual Scene Reconstruction. In: Proceedings of the IEEE Workshop on Representation of Visual Scenes, pp. 26–33. IEEE (2002).
5. Koenderink, J.J., Van Doorn, A.J.: Affine Structure from Motion. Journal of the Optical Society of America 8, pp. 377–385 (1991).
6. Christy, S., Horaud, R.: Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, pp. 1098–1104 (2002).
7. Avitzour, D.: Novel Scene Calibration Procedure for Video Surveillance Systems. IEEE Transactions on Aerospace and Electronic Systems 40, 1105–1110 (2004).
8. Criminisi, A., Reid, I., Zisserman, A.: Single View Metrology. International Journal of Computer Vision 40, pp. 1105–1110 (2000).
9. Masoud, O., Papanikolopoulos, N.P.: Using Geometric Primitives to Calibrate Traffic Scenes. Transportation Research Part C: Emerging Technologies 15, pp. 361–379 (2007).
10. Huynh, D.Q.: Affine Reconstruction from Monocular Vision in the Presence of A Symmetry Plane. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, pp. 476–482. IEEE (2002).
11. Sohon, F. W.: The Stereographic Projection. Chemical Publishing Company, Brooklyn (1941).
12. Jolliffe, I. T.: Principal Component Analysis. Springer Verlag (2002).