A self-calibration method for smart video cameras*

Georg Nebehay
Pattern Recognition and Image Processing
Vienna University of Technology
A-1040 Vienna, Austria
georg.nebehay@student.tuwien.ac.at

Roman Pflugfelder
Video and Security Technology
Austrian Institute of Technology
A-1220 Vienna, Austria
roman.pflugfelder@ait.ac.at

Abstract

The automatic calibration of the intrinsic camera parameters such as the focal length and the camera orientation is an important pre-requisite for many computer vision algorithms in video surveillance. Despite its importance only a few number of methods show their applicability in embedded systems. This paper shows new results of previous work done in camera self-calibration on images of the York Urban data-set. These 102 images show typical urban scenes one might expect in practice. The evaluation shows that in 52 of 102 images the proposed method achieves less than 5 % relative error in the focal length at a mean computation time per image of 14.45 s on a standard PC. We believe that these results show a fair balance between accuracy and computational performance and encourage an embedded implementation on a smart camera.

1. Introduction

A rough knowledge about the scene geometry and the camera is a prerequisite for many applications in the area of visual surveillance. For example, length and area ratios are useful to recognize objects. Some object classification methods need image rectification to remove projective and affine distortion. The camera projection matrix allows to project objects in a certain world point onto the image. This is very useful for model-based applications. Some object tracking applications use homographies to operate on a world ground plane. Homographies between cameras or between cameras and a common world ground plane are a powerful cue to track objects in a multi-camera system. Such systems are attractive, because object occlusions can be handled more effectively than in the single camera case.

Unfortunately, to know these homographies or to know the camera projection matrix needs information about the scene geometry and the camera. Usually, this information is encoded as prior knowledge in many vision algorithms. To retrieve all those parameters with procedures utilizing calibration targets is time-consuming, requiring an expert and becomes quickly infeasible for higher number of cameras. Therefore, researchers have been focusing on *self-calibration* methods to make cameras autonomous as much as possible. This paper exactly concentrates on self-calibration of a single, static camera from vanishing points in man-made environments. Vanishing points of orthogonal and rectilinear scene structure typically available in man-made environments [3] capture valuable information about the internal camera parameters, mainly the focal length, and the orientation between the camera and the scene structure. Usually two or three vanishing points of orthogonal world directions are identifiable in practice, thus, cameras with zero skew and square pixel or known aspect ratio are assumed.

Prominent papers that use line segments for self-calibration were written by Caprile and Torre [2], Rother [18] and Kosecka and Zhang [11] who exploit the rich geometric information of vanishing points. Liebowitz and Zisserman [13] showed the flexible combination of scene and camera constraints on the image of the absolute conic (IAC) which is in a one-to-one relationship with the internal camera parameters [7]. All these methods compute the vanishing points in an initial step and calibrate afterwards.

Coughlan and Yuille [3] proposed to use the image gradients in a Bayesian framework instead of an edge detector. Deutscher, Isard and MacCormick [5] showed an interesting Bayesian calibration approach within this framework. Schindler and Dellaert [20] significantly extended this framework to group line segments and to compute vanishing points whereas the optimization takes place in the calibration parameter space and vanishing points are an intermediate result. Unfortunately, it is not clear how to automatically initialize their methods.

Pflugfelder proposed a new framework [17] that incor-

*This research has been funded by the Vienna Wissenschafts-, Forschungs-, und Technologiefonds WWTF, project ICT-08-30.

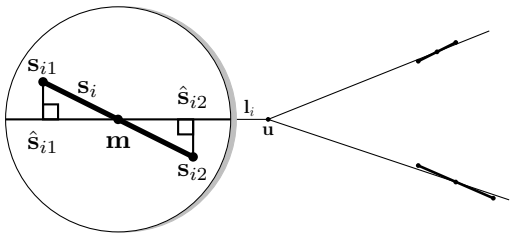


Figure 1: The closest fit of a straight line to a line segment $\mathbf{s}_i$ passing exactly through $\mathbf{u}$ is $\mathbf{l}_i$. The endpoints of $\mathbf{s}_i$ are $\mathbf{s}_{i1}$ and $\mathbf{s}_{i2}$ respectively. $\mathbf{m}$ is the intersection point of $\mathbf{l}_i$ and $\mathbf{s}_i$. It is not necessarily the midpoint of the line segment. $\hat{\mathbf{s}}_{i1}$ and $\hat{\mathbf{s}}_{i2}$ are the normal projections of $\mathbf{s}_{i1}$ and $\mathbf{s}_{i2}$ onto $\mathbf{l}_i$.

porates Liebowitz's error in the endpoints [13], a novel RANSAC based initialization with similarities mentioned in Rother's PhD thesis [19], the direct optimization in the parameters instead optimizing the vanishing points [20], the EM-framework [1], necessary geometric criterions [22] and the consideration of the lens distortion [23]. This paper shows results of this method on the newly available York Urban data-set [4] which is a compilation of images showing urban environments. The evaluation shows that in 51 of 102 images our calibration method achieves less than 5 % relative error in the focal length at a mean computation time per image of 13.36 s on a standard PC. We believe that these results show a fair balance between accuracy and computational performance and encourage an embedded implementation on a smart camera.

Section 2 gives a brief overview of Pflugfelder's self-calibration method. A detailed description can be found in [17]. Section 3 presents a comprehensive evaluation on the new data-set as the contribution of this paper. We finally give conclusions in Section 4.

2. The self-calibration method

The basic information to estimate vanishing points are line segments which are detected by the Canny operator, followed by a straight line fitting. Imagine the existence of N line segments $\mathbf{s}_1 = (\mathbf{s}_{11}\ \mathbf{s}_{12})^\top$, $\mathbf{s}_2 = (\mathbf{s}_{21}\ \mathbf{s}_{22})^\top$, ..., $\mathbf{s}_N = (\mathbf{s}_{N1}\ \mathbf{s}_{N2})^\top$ with endpoint measurements $\mathbf{s}_{i1} = (s_{i11}\ s_{i12})^\top$ and $\mathbf{s}_{i2} = (s_{i21}\ s_{i22})^\top$, $1 \leq i \leq N$, in an image. See Fig. 1 for an illustration.

Following the idea by Liebowitz [12] to compute straight lines passing exactly through an intersection point $\mathbf{u}$ and elongating optimally the line segments $\mathbf{s}_1, \ldots, \mathbf{s}_N$ in a statistical sense one has to minimize the total error

$$\mathrm{f}_d(\mathbf{u}, \mathbf{s}_1, \ldots, \mathbf{s}_N) = \sum_{i=1}^{N} \mathrm{d}^2(\hat{\mathbf{s}}_{i1}, \mathbf{s}_{i1}) + \mathrm{d}^2(\hat{\mathbf{s}}_{i2}, \mathbf{s}_{i2}) \quad (1)$$

between all line segments and concurrent lines given $\mathbf{u}$. $\mathrm{d}^2(\cdot)$ is in general the Mahanalobis distance given a Gaussian error model in the endpoints. $\mathrm{f}_d(\cdot)$ enjoys three major advantages compared to other error functions [16, 10] as it has (i) no singularities, (ii) will be optimal under a Gaussian error mode, (iii) it is independent of the location of vanishing points in the image plane. In practice, the minimization is carried out by an iterative, numerical method such as the popular Levenberg-Marquardt (LM) algorithm [14]. The computation of $\hat{\mathbf{s}}_{i1}$ and $\hat{\mathbf{s}}_{i2}$ is intrinsically solvable in closed form given $\mathbf{u}$ and $\mathbf{s}_i$. This is important, because this computation is done for hundreds of line segments. Details are in [12].

2.1. The robust intersection of line segments

Line segments that do not belong to a particular set of concurrent line segments are gross outliers which cause severe errors in the estimation of intersection points and therefore should be identified.

Following the idea of RANSAC [8], a line segment $\mathbf{s}_i$ will be an inlier and thus part of the consensus set, if both Euclidean distances $\|\mathbf{s}_{i1} - \hat{\mathbf{s}}_{i1}\|$ and $\|\mathbf{s}_{i2} - \hat{\mathbf{s}}_{i2}\|$ are smaller than a threshold t. We set $t = \chi\sigma$ with χ be the α-quartile of a χ^2-distribution. The number of samples n and the proportion w of inliers to the total number of line segments N is determined adaptively [9]. An early stopping criterion as it is described by Hartley and Zisserman is not used, because we simply do not know the number of inliers. However, the adaptive probing of the data via the consensus set includes a statistically correct termination of the algorithm and works well in practice when α is chosen conservatively, for example, $\alpha = .99$.

2.2. The robust grouping of line segments

A simple and self-evident idea to group concurrent line segments is the consecutive estimation of intersection points. At the beginning, all line segments are used to estimate an intersection point. After the estimation, the consensus set is removed from the set of line segments. The set of outlier line segments is used in a consecutive step to estimate a further intersection point. This process can be repeated until all line segments are grouped into concurrent line segments or intersection points of a particular number M are identified.

To avoid equivalent groups of line segments a statistical test evaluates the pairwise coincidence between all intersection points. This is possible, because uncertainties are propagated in each computation. If an intersection point lies with a specific confidence within the uncertainty ellipse the intersection point will be rejected.

Threshold t during the RANSAC execution controls the grouping. If t is too small then the grouping for a particular intersection point will mistakenly exclude true line segments. Otherwise, if t is too large then the grouping for

a particular intersection point will include some wrong line segments. However, by assuming an error model in the endpoints, t has a clear statistical interpretation as shown in the last section.

2.3. Calibration

Intersection points supported by most of the line segments are typically vanishing points in a man-made world, that is, $M \in \{2, 3\}$. Now, vanishing points $v_1, \ldots, v_M$ give $\binom{M}{2}$ constraints on the IAC $\omega_{3\times3}$. ω can be interpreted as a metric of the uncalibrated space. ω is a symmetric matrix and has 5 degrees of freedom. Zero skew s and a constant aspect ratio r yield two more constraints on ω. Assuming that the principal point $p = (u_0\ v_0)^\top$ is close to the image center gives further two constraints on ω. All these constraints can be formulated compactly by

$$\begin{aligned} v_i^\top \omega v_j &= 0 \quad 1 \geq i, j \geq m,\ i \neq j && (2)\\ (1\,0\,0)^\top \omega (0\,1\,0) &= 0 && (3)\\ (1\ r\ 0)^\top \omega (0\ -1\ 0) &= 0 && (4)\\ \omega p &= (0\,0\,1)^\top && (5) \end{aligned}$$

For $m > 1$ a solution of ω can be computed. If more than two vanishing points are detected, the linear equations system formed by the equations in 2-5 is over-determined. An optimal solution of ω in a least-squares sense can then be computed using Singular Value Decomposition. To know ω is equal to know the internal camera parameters

$$\omega = K^{-\top} K^{-1} \text{ with } K = \begin{pmatrix} f & s & u_0 \\ 0 & rf & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (6)$$

f is the focal length. K can be computed using a Cholesky decomposition of ω.

The computation of a camera's rotation to the orthogonal scene structure is straightforward: Consider the pairwise, orthogonal rays

$$\mathbf{d}_{ij} = K_j^{-1} \mathbf{v}_{ij}, 1 \leq i \leq 3. \quad (7)$$

For any two rays $\mathbf{d}_{ij}$ and $\mathbf{d}_{kj}$, the third, orthogonal ray $\mathbf{d}_{lj}$ is determined by the cross product

$$\mathbf{d}_{lj} = \mathbf{d}_{ij} \times \mathbf{d}_{kj} \quad (8)$$

with $1 \leq i, k, l \leq 3, i \neq k \neq l$. Each ray $\mathbf{d}_{ij}$ passes exactly through the vanishing point $\mathbf{v}_{ij}$ and, obviously, each ray represents a coordinate axis of the camera coordinate system. The rotation matrix R_j that transforms the canonical world coordinate system into the camera coordinate system is given by

$$R_j = \left(\pm \frac{\mathbf{d}_{ij}}{\|\mathbf{d}_{ij}\|} \ \pm \frac{\mathbf{d}_{kj}}{\|\mathbf{d}_{kj}\|} \ \pm \frac{\mathbf{d}_{lj}}{\|\mathbf{d}_{lj}\|} \right). \quad (9)$$

Note that each column of R_j is normalized to one and $\det(R_j) = 1$, because rotation matrices are per definition orthonormal.

There is no guarantee that the result is valid as all these constraints are necessary but not sufficient for a correct result. A plausibility test with a known focal length of the lens can overcome the problem. Critical cases such as infinite vanishing points can have severe influence on the accuracy of the result. More details on how to proceed in such cases are in [17].

2.4. Optimized calibration

RANSAC may fail to assign a true inlier line segment to the consensus set of a vanishing point due to the particular choice of t. Expectation Maximization (EM) [15] provides here a solution where the grouping of line segments is done in the E-step and the estimation of the vanishing point is alternately done in the M-step. EM estimation of vanishing points can be combined with successive calibration [11]. Schindler and Dellaert [20] criticized this bottom-up calibration approach and replaced it by a top-down view where the actual unknown parameters are K and R. In fact, the three orthogonal vanishing points of a man-made structure are the columns of the matrix $H = KR$.

Lens distortion is another reason why it is difficult to find an appropriate t, because it violates the error model in the endpoints. It has further a significant influence on the calibration result. This fact is not astonishing, because every kind of nonlinear distortion will impede the localization of concurrent lines. Hence, this method considers a first-order radial lens distortion model (distortion coefficient k, radial center $\mathbf{c}_k = (c_{k1}\ c_{k2})^\top$) [6] with undistortion function

$$\mathrm{L}(x, k, \mathbf{c}_k, \mathbf{c}) = \mathbf{x} + k(\mathbf{x} - \mathbf{c}_k) \frac{\|\mathbf{x} - \mathbf{c}_k\|^2}{\mathbf{c}^\top \mathbf{c}} \quad (10)$$

where the undistorted point $\mathbf{x}'$ is computed from image point $\mathbf{x} = (x_1\ x_2)^\top$ by

$$\mathbf{x}' = \mathrm{L}(x, k, \mathbf{c}_k, \mathbf{c}). \quad (11)$$

Following this approach the EM optimization problem is as follows: We want to maximize the posterior probability density function of the unknown internal camera parameters, the lens distortion and the rotation, given the line segments, while marginalizing over the unknown grouping of line segments J with respect to the three orthogonal vanishing points. Mathematically, we write

$$\underset{K,R,k,\mathbf{c}_k}{\operatorname{argmax}} \sum_{J \in \mathcal{J}} \mathrm{p}(K, R, k, \mathbf{c}_k, J | \mathbf{s}_1, \ldots, \mathbf{s}_N). \quad (12)$$

$\mathcal{J}$ are all combinations of possible labels that can be assigned to the line segments. Labels are either a specific vanishing point or noise. Note that a proper R's parametrization

is important to avoid problems in the objective function like singularity [21]. More details on the E-step and M-step can be found in [17].

3. Evaluation

Denis, Elder and Estrada [4] have established the York Urban data-set to benchmark camera calibration. This data-set contains 102 images (45 indoor, 57 outdoor) of urban environments in downtown Toronto taken with a calibrated Panasonic Lumix DMC-LC80 digital camera. Each image has been hand-labeled to identify the set of line segments that conform to the 3-D orthogonal frame of the scene structure in the urban environment which has been further used to compute the three orthogonal vanishing points and from these the rotation matrix between camera and scene structure. Both the York Urban data-set images and images produced by urban surveillance cameras show scenes aligned to Manhattan frames, which is the main assumption of Pflugfelder's method. Therefore, the data-set images show a good profile one might expect in practical situations of urban video surveillance and are suitable for our evaluation.

The radial distortion was not taken into consideration by the authors of the data-set. It became clear during experimentation and after a closer look to the images that undoing the radial distortion did not have a significant impact on the results, as the distortion of the Panasonic camera lens is obviously weak.

By this evaluation we tried to answer the question if this method is mature enough in terms of accuracy, robustness and run-time to be implemented on embedded systems. Hence, it was of interest how the algorithm performs on the data-set in terms of precise calibration results. The second main topic was to evaluate run-time and potential run-time improvements.

3.1. Accuracy and robustness

Four implementations of the method were run on the whole data-set:

DPUD	C, double precision, undo distortion
SPUD	C, single precision, undo distortion
SP	C, single precision
SPOPT	optimized C, single precision

The principal point was set to the image center. The interested reader is referred to [17] for a description of the exact method's parameter setting. Table 1 shows the error in the focal length and the largest error in the Euler angles. No significant loss could be ascertained if the number precision was reduced from double to single. In the same manner, ignoring radial distortion did not have an impact on the quality of the results, even though this might not be true for every image or camera.

rel. error [%]	DPUD	SPUD	SP
< 5	52	52	52
$5 \leq \ldots < 10$	13	12	11
≥ 10	31	33	34
no result	6	5	5
total	102	102	102

(a) Focal length

rel. error [%]	DPUD	SPUD	SP
< 5	79	80	78
$5 \leq \ldots < 10$	9	9	9
≥ 10	8	8	10
no result	6	5	5
total	102	102	102

(b) Euler angles

Table 1: Accuracy and robustness: The number of images that achieve a certain relative error between ground truth and estimate are classified. The method terminates without any result when it is unable to obtain at least two orthogonal vanishing points or in critical cases such as the IAC is not a positive definite matrix.

The rotation error is below 5 % in nearly 80 % of all cases while only half of the data-set gives errors in focal length below 5 %. This result supports the observation that the focal length is much harder to estimate than the rotation, because rotation depends on the relation between the pairwise distances of vanishing points and not their exact location. The average error in the Euler angles is 1.32 deg with a standard deviation of 0.89 deg. This result is approximately three times better than the average error reported in [4]. Fig. 2 depicts exemplarily some results.

3.2. Run-time

Experiments were run on an Athlon XP 3000+ CPU with 2GB of RAM. Statistics are shown in Table 2. The best mean run-time per image for SPOPT is 14.45 s with a standard deviation of 9.55 s. Denis, Elder and Estrada reported a run-time for their implementation between 25 s and 30 s. The time used for initializing the method and for performing the RANSAC step can be neglected since almost the total run-time is consumed by the EM step. A correlation between runtime and errors in the estimates was not observable. As expected a correlation between the number of line segments and the time of one EM iteration can be observed (Fig. 3a), however, the speed of convergence is independent

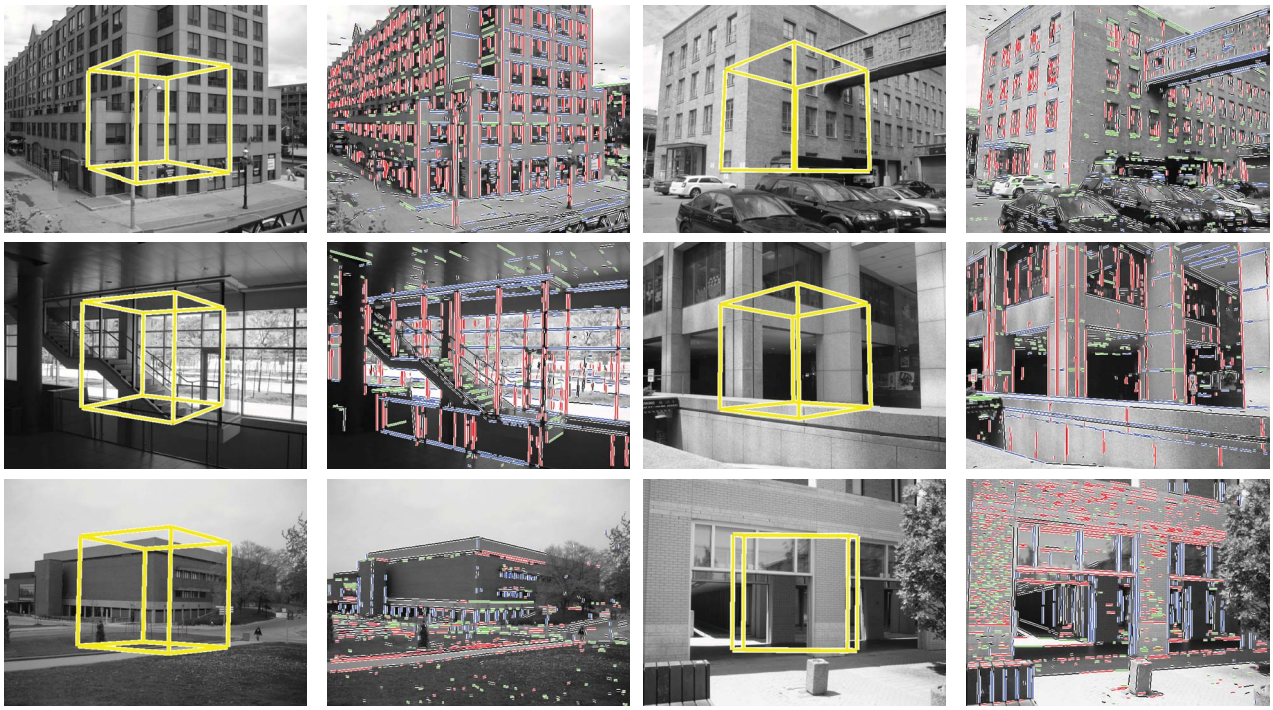


Figure 2: Qualitative exemplary results: The top two rows show images (first row: P1080079, P1080104, second row: P1020829, P1080025) where the error in focal length is below 5 %. The orthogonal Manhattan frame (yellow) was re-projected into the images. Vanishing points are marked by the different colored line segments. The last row (P1040822, P1020856) shows a medium ($5 \leq \ldots < 10$) and poor (≥ 10) result. The algorithm is confused by the presence of line segments (originating from the pathway) not belonging to the Manhattan Frame in the former case, while in the latter RANSAC delivered poor initial values for K and R due to the lack of line segments in the direction of the third vanishing point.

	mean [s]	**std** [s]	**median** [s]	**min** [s]	**max** [s]	**total** [s]
DPUD	66.85	68.65	48.50	9.00	555.00	6351.00
SPUD	33.26	21.63	28.50	4.00	108.00	3170.00
SP	17.53	11.51	14.50	2.00	60.00	1683.00
SPOPT	14.45	9.55	11.50	1.00	51.00	1373.00

Table 2: Run-time results. When vanishing points are well defined by the line segments - no critical case - run-times within a few seconds are reached.

of the number of line segments (Fig. 3b).

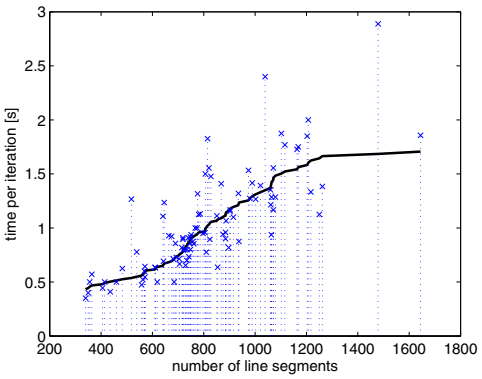
Neglecting the radial distortion reduced the computation time by almost 50 %. Compiling the code using an optimization flag (-O2 for gcc) yielded another 20 % speed-up (SPOPT).

3.3. Algorithmic Enhancements

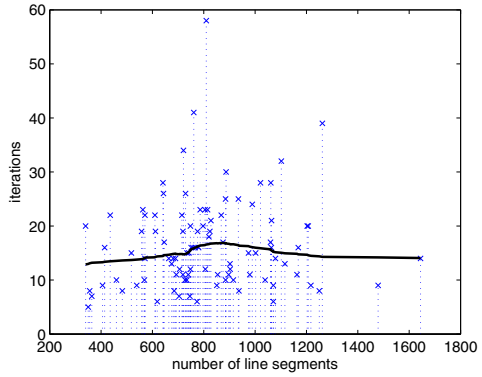
In order to improve the run-time of our code we concentrated on reducing the number of line segments used in the EM step since a large number of line segments is a major obstacle in achieving satisfying run-time results. A simple measure for reducing line segments is to keep only n line segments for each vanishing point after the RANSAC step. We chose to keep those line segments exhibiting the least error for each vanishing point respectively. Table 3 shows the results for $n = 15$. By applying this measure, we achieved a mean run-time of 1.73 seconds. The loss of accuracy is shown in table 4 and is not acceptable for an implementation in a real environment. The key in achieving good run-time results while preserving accuracy might lie in reducing the number of line segments in a manner less harmful to the EM step.

	mean [s]	**std** [s]	**median** [s]	**min** [s]	**max** [s]	**total** [s]
SPUD, $n = 15$	1.73	1.25	1.00	0.00	7.00	166.00

Table 3: Run-time results when keeping only 15 line segments exhibiting least error for each vanishing point after the RANSAC step.



(a) One EM step



(b) Speed of convergence

Figure 3: Correlation between the number of line segments and the run-time.

3.4. Applicability to Embedded Systems

Applying our approach to an embedded device is easy to perform since it is written in portable C code. Since many embedded devices, such as the Sony XCI-V100C, run Windows XP embedded or a customized Linux, our code is directly applicable to them.

It is clear that an Athlon XP 3000+ still outperforms for example a 1GHz VIA Eden built into a SONY XCI-V100C smart camera. However, the results from this evaluation are still applicable to embedded devices, because camera calibration is usually invoked at specific and rare time instants such as the start-up phase or when the camera recognizes itself that the calibration parameters have changed which is usually not real-time critical.

rel. error [%]	SPUD, $n = 15$
< 5	25
$5 \leq \ldots < 10$	19
≥ 10	53
no result	5
total	102

Table 4: The accuracy of the calibration of the focal length decreases when reducing the number of line segments.

4. Conclusion

This paper presented results on a newly available dataset of an existing self-calibration algorithm. The evaluation supports our expectation in terms of accuracy, robustness and run-time that this approach is mature enough to be implemented on an embedded system. Our next step will be to adapt this approach for Sony's XCI-V100C smart camera. The initial detection of line segments and the subsequent detection of vanishing points play a major role in achieving good results for focal length. The reduction of line segments might play a role in achieving very low run-time results, however, we showed that limiting the EM iterations by only using the n "best" (lowest Liebowitz error given a vanishing point) line segments does not give an acceptable accuracy for the focal length. The estimation of the focal length still remains a hard problem which is successful only for 50 % of the images, because focal length is so highly sensitive to vanishing point locations. We observed that vanishing points dramatically change when reducing the line segments even to the n "best" ones.

References

[1] M. E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 282–289, Hilton Head Island, SC, USA, 2000.

[2] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision (IJCV)*, 4:127–140, 1990.

[3] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 941–947, Corfu, Greece, September 1999. IEEE.

[4] P. Denis, J. Elder, and F. Estrada. Efficient edge-based methods for estimating manhattan frames in urban imagery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2, pages 197–210, 2008.

[5] J. Deutscher, M. Isard, and J. MacCormick. Automatic camera calibration from a single manhattan image. In *Proceedings of the 7th European Conference on Computer Vision*, volume 2353, pages 175–188. IEEE, May 2002.

[6] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001.

[7] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. In *Proceedings of the European Conference on Computer Vision*, pages 321–334, 1992.

[8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[10] K. Kanatani. *Statistical optimization for geometric computation*. Elsevier, 1996.

[11] J. Kosecka and W. Zhang. Video compass. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, volume 2353, page 476f. Springer-Verlag, May 2002.

[12] D. Liebowitz. *Camera Calibration and Reconstruction of Geometry from Images*. PhD thesis, University of Oxford, Dept. Engineering Science, June 2001. D.Phil. thesis.

[13] D. Liebowitz and A. Zisserman. Combining scene and autocalibration constraints. In *Proceedings of the International Conference on Pattern Recognition (ICCV)*, volume 1, pages 293–300. IEEE, September 1999.

[14] M. Lourakis and A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Proceedings of the 10th International conference on Computer Vision (ICCV)*, pages 1526–1531, 2005.

[15] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 2008.

[16] G. F. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 17(11):1090–1095, 1995.

[17] R. Pflugfelder. *Self-calibrating Cameras in Video Surveillance*. PhD thesis, Graz University of Technology, May 2008.

[18] C. Rother. A new approach for vanishing point detection in architectural environments. In *Proceedings of the British Machine Vision Conference*, volume 20, pages 647–656, 2002.

[19] C. Rother. *Multi-View Reconstruction and Camera Recovery using a Real and Virtual Reference Plane*. PhD thesis, Royal Institute of Technology, January 2003.

[20] G. Schindler and F. Dellaert. Atlanta world: An expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 203–209, 2004.

[21] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, chapter Bundle Adjustment: A modern synthesis, pages 298–375. Springer, 2000.

[22] F. A. Van den Heuvel. Vanishing point detection for architectural photogrammetry. *International archives of photogrammetry and remote sensing*, 32(5):652–659, 1998.

[23] F. A. Van den Heuvel. Estimation of interior orientation parameters from constraints on line measurements in a single image. In *International archives of photogrammetry and remote sensing*, volume 32, pages 81–88, 1999.