Self-calibration with Partially Known Rotations

Ferid Bajramovic and Joachim Denzler

Chair for Computer Vision, Friedrich-Schiller-University Jena
{bajramov,denzler}@informatik.uni-jena.de
http://www4.informatik.uni-jena.de

Abstract. Self-calibration methods allow estimating the intrinsic camera parameters without using a known calibration object. However, such methods are very sensitive to noise, even in the simple special case of a purely rotating camera. Suitable pan-tilt-units can be used to perform pure camera rotations. In this case, we can get partial knowledge of the rotations, e.g. by rotating twice about the same axis. We present extended self-calibration algorithms which use such knowledge. In systematic simulations, we show that our new algorithms are less sensitive to noise. Experiments on real data result in a systematic error caused by non-ideal hardware. However, our algorithms can reduce the systematic error. In the case of complete rotation knowledge, it can even be greatly reduced.

1 Introduction

For many computer vision tasks, the intrinsic camera parameters have to be known. Classic calibration uses a calibration pattern with known geometry and easily detectable features to establish correspondences between known 3D points and 2D image points. However, having to use such a pattern is not very convenient and sometimes impossible. Luckily, there are self-calibration methods, which estimate the intrinsic camera parameters from images taken by a moving camera *without* knowledge about the scene. For an overview, the reader is referred to the literature [1]. An important special case is self-calibration from a purely rotating camera as introduced by Hartley [2,1].

However, most self-calibration methods are very sensitive to noise [3,1]. They work well at low noise levels, but most often have serious problems at higher noise levels. On the other hand, in many practical situations, additional knowledge is available, which can be used to increase the robustness of self-calibration. De Agapito, Hayman and Reid [6] exploit a priori knowledge on the intrinsic parameters by using a MAP estimator. In this paper, we focus on rotation knowledge. Hartley [2] mentions the possibility to incorporate known rotation matrices into the nonlinear refinement step and reported greatly improved self-calibration results. Frahm and Koch [4,5] have presented a linear approach that uses known relative orientation provided by an external rotation sensor.

In practice, however, there are cases in between no and full rotation knowledge. For example, a pan-tilt-unit is often used to perform rotations about one of two physical rotation axes at a time. To the best of our knowledge, using such a priori information to improve self-calibration has not been systematically studied. In this paper, we give an overview of different kinds of partial rotation information with real pan-tilt-units in mind, and show how this knowledge can be incorporated into a nonlinear

F.A. Hamprecht, C. Schnörr, and B. Jähne (Eds.): DAGM 2007, LNCS 4713, pp. 1–10, 2007.

self-calibration procedure. We demonstrate the improvements gained by our new algorithms in systematic simulations and also in experiments with real hardware.

The paper is organized as follows: In Section 2 we give a repetition of self-calibration for a rotating camera. Section 3 describes how partial rotation information can be incorporated into the self-calibration procedure in various situations. Our new algorithms are evaluated in Section 4. Finally, we give conclusions in section 5.

2 Self-calibration of a Rotating Camera

2.1 Camera Model

First of all, we introduce the camera model and some notation. The pinhole camera model [1,7] is expressed by the equation $\lambda \boldsymbol{p} = \boldsymbol{K}\boldsymbol{p}_\mathrm{C}$, where $\boldsymbol{p}_\mathrm{C}$ is a 3D point in the camera coordinate system, $\boldsymbol{p} = (p_\mathrm{x}, p_\mathrm{y}, 1)^T$ is the imaged point in homogeneous 2D pixel coordinates, $\lambda \neq 0$ is a projective scale factor and $\boldsymbol{K} \stackrel{\text{def}}{=} ((f_\mathrm{x}, s, o_\mathrm{x}), (0, f_\mathrm{y}, o_\mathrm{y}), (0, 0, 1))^T$ is the camera calibration matrix, where f_x and f_y are the effective focal lengths, s is the skew parameter and $(o_\mathrm{x}, o_\mathrm{y})$ is the principal point. The relation between a 3D point in camera coordinates $\boldsymbol{p}_\mathrm{C}$ and the same point expressed in world coordinates $\boldsymbol{p}_\mathrm{W}$ is $\boldsymbol{p}_\mathrm{C} = \boldsymbol{R}_\mathrm{o}\boldsymbol{p}_\mathrm{W} + \boldsymbol{t}$, where $\boldsymbol{R}_\mathrm{o}$ is the orientation of the camera and $\boldsymbol{t}$ is the position of its optical center. Thus, $\boldsymbol{p}_\mathrm{W}$ is mapped to the image point $\boldsymbol{p}$ by the equation $\lambda \boldsymbol{p} = \boldsymbol{K}(\boldsymbol{R}_\mathrm{o}\boldsymbol{p}_\mathrm{W} + \boldsymbol{t})$.

2.2 Linear Self-calibration

We will give a very brief repetition of Hartley's linear self-calibration algorithm [2,1] for a purely rotating camera. In this situation, without loss of generality, we can assume $\boldsymbol{t} = 0$. Taking a second image $\boldsymbol{p}' = (p'_\mathrm{x}, p'_\mathrm{y}, 1)^T$ of the point $\boldsymbol{p}_\mathrm{W}$ with camera orientation $\boldsymbol{R}'_\mathrm{o}$ then results in $\lambda' \boldsymbol{p}' = \boldsymbol{K}\boldsymbol{R}'_\mathrm{o}\boldsymbol{p}_\mathrm{W}$, where $\lambda' \neq 0$ is another scale factor. The points $\boldsymbol{p}$ and $\boldsymbol{p}'$ *correspond*. By eliminating $\boldsymbol{p}_\mathrm{W}$, we get (cf. [1]):

$$\lambda'' \boldsymbol{p}' = \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}\boldsymbol{p} \quad \text{with} \quad \boldsymbol{R} \stackrel{\text{def}}{=} \boldsymbol{R}'_\mathrm{o}\boldsymbol{R}_\mathrm{o}^T \quad \text{and} \quad \lambda'' \stackrel{\text{def}}{=} \lambda'/\lambda \ . \tag{1}$$

In this formulation, $\boldsymbol{R}$ is the relative camera rotation. The transformation $\lambda'' \boldsymbol{p}' = \boldsymbol{H}\boldsymbol{p}$ maps $\boldsymbol{p}$ to $\boldsymbol{p}'$, where $\boldsymbol{H} \stackrel{\text{def}}{=} \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{-1}$ is the *infinite homography*. It is related to the *dual image of the absolute conic* $\boldsymbol{\omega}^* \stackrel{\text{def}}{=} \boldsymbol{K}\boldsymbol{K}^T$ by the equation $\boldsymbol{\omega}^* = \boldsymbol{H}\boldsymbol{\omega}^*\boldsymbol{H}^T$. Now, given $n \geq 2$ rotations of the camera (not all about the same axis), the self-calibration problem can be solved linearly by the following algorithm:

Input: A set of point correspondences $\{ (\boldsymbol{p}_{i,j}, \boldsymbol{p}'_{i,j}) \mid 1 \leq i \leq n, 1 \leq j \leq m_i \}$, where $n \geq 2$ is the number of image pairs and m_i is the number of point correspondences for pair i. For numerical reasons [1,8], we normalize pixel coordinates throughout the paper to the range $[-1, 1]$ by applying a translation and an isotropic scaling.

1. For each image pair i, estimate the inter-image homography $\boldsymbol{H}'_i$ from the point correspondences of image pair i and enforce $\det(\boldsymbol{H}_i) = 1$ by setting $\boldsymbol{H}_i = \det(\boldsymbol{H}'_i)^{-\frac{1}{3}}\boldsymbol{H}'_i$.
2. Solve the set of equations $\{ \boldsymbol{\omega}^* = \boldsymbol{H}_i\boldsymbol{\omega}^*\boldsymbol{H}_i^T \mid 1 \leq i \leq n \}$ for $\boldsymbol{\omega}^*$ (linear least squares).
3. Compute $\boldsymbol{K}$ from $\boldsymbol{\omega}^* = \boldsymbol{K}\boldsymbol{K}^T$, e.g. by Cholesky decomposition of $(\boldsymbol{\omega}^*)^{-1}$.

Note that Hartley and Zisserman [1] require the homographies $\boldsymbol{H}_i$ to be expressed with respect to a common reference image. It is obvious that this requirement is *not* necessary, as only the *relative* orientations $\boldsymbol{R}_i$ of pairs of views are required.

2.3 Nonlinear Refinement

With equation (1) in mind, the self-calibration problem for a rotating camera with constant intrinsic parameters $\boldsymbol{K}$ can be defined as the solution of the following optimization problem (similar to Hartley's and Zisserman's nonlinear refinement [1]):

$$\boldsymbol{K} = \underset{\boldsymbol{K}}{\operatorname{argmin}} \min_{(\boldsymbol{R}_i \in \mathrm{SO}(3))_{1 \le i \le n}} \sum_{i=1}^{n} \sum_{j=1}^{m_i} d\left(\boldsymbol{K}\boldsymbol{R}_i\boldsymbol{K}^{-1}\boldsymbol{p}_{i,j}, \lambda''_{i,j}\boldsymbol{p}'_{i,j}\right)^2 \quad , \tag{2}$$

where $\mathrm{SO}(3) = \{\, \boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I} \wedge \det(\boldsymbol{R}) = 1 \,\}$ denotes the rotation group and $d(\cdot,\cdot)$ is the Euclidean distance of 2D points in homogeneous coordinates. There are two advantages of the nonlinear formulation of the problem. First, the distance $d(\cdot,\cdot)$ is a geometrically meaningful measure on the point correspondences. Second, the constraint, that $\boldsymbol{K}$ is constant, will be enforced directly, which is impossible for the homography estimation part of the linear algorithm. The nonlinear optimization problem in equation (2) can be solved by finding a good initial approximation to the solution and refining that using a local nonlinear optimization algorithm. We use a modern second order Trust Region algorithm [9]. The initial solution for $\boldsymbol{K}$ is provided by the linear self-calibration method described above. The rotation matrices can be initialized as follows: compute $\boldsymbol{R}_i = \boldsymbol{K}^{-1}\boldsymbol{H}_i\boldsymbol{K}$ and enforce the constraint $\boldsymbol{R}_i \in \mathrm{SO}(3)$ by setting all singular values of $\boldsymbol{R}_i$ to one. The gradient and the Hessian of the objective function in equation (2) (and all variants that will follow in the rest of the paper) can be gained symbolically in "closed form". We will leave out the details, as automatic differentiation methods can be applied.

2.4 Zero Skew

For modern cameras, we can often assume zero skew $s = 0$ [1]. This assumption can be easily incorporated into the optimization problem by initially setting $s = 0$ and removing s from the set of optimization parameters. In the linear algorithm, the assumption can also be applied [1].

3 Improved Self-calibration with Partially Known Rotations

Rotation information can be available to several different extents. The various cases of partially known rotations are summarized in Figure 1. We will subsequently explain these cases, and show how each kind of a priori knowledge can be incorporated into the nonlinear self-calibration procedure by presenting appropriate variants of equation (2) with modified parameterizations of the rotations $\boldsymbol{R}_i$. Even though the new formulations may look more complicated, decreasing the dimension of the parameter space will (hopefully) reduce the over-adaptation to noise. Despite that, in the cases with known values of some parameters, the algorithms simply cannot introduce errors by misestimating them.

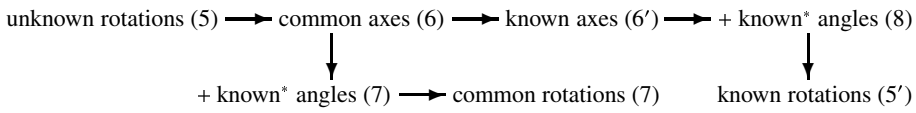

Fig. 1. Cases of partially known rotations. The arrows indicate additional rotation knowledge, and "+ known* angles" means that, additionally, rotation angles are known up to a common scale factor. Numbers refer to equations, where (5′) and (6′) mean that a variation of the equation with fewer optimization parameters is used. Further explanations can be found in the text.

3.1 Unknown Rotations

Our starting point for using partial rotation information is the optimization problem in equation (2). First, we have a closer look at the rotation matrices. Enforcing the constraint $\boldsymbol{R}_i \in \mathrm{SO}(3)$ during the optimization can implicitly be achieved by using a minimal parameterization such as *exponential parameters* as suggested by Ma, Soatto, Kosecka and Sastry [10]. A rotation matrix $\boldsymbol{R}$ can be represented by a vector $\boldsymbol{w} = (w_1, w_2, w_3)^T$ using Rodrigues' formula [10]:

$$\boldsymbol{R} = \mathrm{Rod}(\boldsymbol{w}) \stackrel{\text{def}}{=} \boldsymbol{I} + \frac{\boldsymbol{S}(\boldsymbol{w})}{\|\boldsymbol{w}\|} \sin(\|\boldsymbol{w}\|) + \frac{\boldsymbol{S}(\boldsymbol{w})^2}{\|\boldsymbol{w}\|^2}(1 - \cos(\|\boldsymbol{w}\|)) \ , \tag{3}$$

using the skew symmetric matrix $\boldsymbol{S}(\boldsymbol{w}) \stackrel{\text{def}}{=} ((0, -w_3, w_2), (w_3, 0, -w_1), (-w_2, w_1, 0))^T$. The related *axis-angle* parameterization separates the rotation axis $\boldsymbol{v}$ and angle α explicitly at the cost of one additional parameter and the constraint $\|\boldsymbol{v}\| = 1$:

$$\boldsymbol{R} = \mathrm{Rod}(\boldsymbol{v}, \alpha) \stackrel{\text{def}}{=} \boldsymbol{I} + \boldsymbol{S}(\boldsymbol{v}) \sin(\alpha) + \boldsymbol{S}(\boldsymbol{v})^2(1 - \cos(\alpha)) = \mathrm{Rod}(\alpha\boldsymbol{v}) \ . \tag{4}$$

We prefer the axis-angle parameterization over exponential parameters only in cases of appropriate a priori knowledge, e.g. if $\boldsymbol{v}$ is known. For the following reasons, we do not use unit quaternions, even though they are very popular:

- They do not provide a minimal parameterization, as they have four parameters, and the unit quaternion constraint is required (similar to axis-angle).
- The close relationship between exponential parameters and the axis-angle representation helps pinpoint the precise differences between the various cases of partial rotation knowledge, as will become evident in the rest of the paper.
- There is no clear agreement as to which parameterization is best [11].

Applying exponential parameters to equation (2), we get the following nonlinear optimization problem, which has a total of $3n + 5$ parameters:

$$\boldsymbol{K} = \operatorname*{argmin}_{\boldsymbol{K}} \min_{(\boldsymbol{w}_i)_{1 \le i \le n}} \sum_{i=1}^{n} \sum_{j=1}^{m_i} d\left(\boldsymbol{K}\mathrm{Rod}(\boldsymbol{w}_i)\boldsymbol{K}^{-1}\boldsymbol{p}_{i,j}, \lambda''_{i,j}\boldsymbol{p}'_{i,j}\right)^2 \ . \tag{5}$$

For the initialization of the nonlinear optimization problem, we need to compute the rotation parameters $\boldsymbol{w}_i$ from the rotation matrices $\boldsymbol{R}_i$ [10].

3.2 Rotation About Two Axes Only

If the pan-tilt-unit is actively controlled for the purpose of self-calibration, it is possible to restrict rotations to be about one of the two physical rotation axes of the pan-tilt-unit *at a time*, such that there are only two mathematical rotation axes in total (Figure 1: "common axes"). In this case, rotations only have one parameter each (the rotation angle) and there are two degrees of freedom for each rotation axis. From a theoretical point of view, we can generalize this and allow $r \leq n$ rotation axes instead of only two. This results in $n + 2r + 5$ parameters. To obtain a suitable version of the optimization problem in equation (5), we replace exponential parameters by axis-angle and use only r instead of n rotation axes. Finally, we add constant indices k_i as a priori knowledge, which assign the correct rotation axis $\boldsymbol{v}_{k_i}$ to the rotation with index i, and get:

$$\boldsymbol{K} = \underset{\boldsymbol{K}}{\operatorname{argmin}} \min_{\substack{(\alpha_i)_{1\leq i\leq n},\,(\boldsymbol{v}_k)_{1\leq k\leq r} \\ \|\boldsymbol{v}_k\| = 1 \text{ for all } k}} \sum_{i=1}^{n}\sum_{j=1}^{m_i} d\left(\boldsymbol{K}\mathrm{Rod}(\alpha_i, \boldsymbol{v}_{k_i})\boldsymbol{K}^{-1}\boldsymbol{p}_{i,j}, \lambda''_{i,j}\boldsymbol{p}'_{i,j}\right)^2 \quad . \tag{6}$$

Note that this formulation uses one parameter too much for each axis $\boldsymbol{v}_k$. In this one case we trade minimality for simplicity and ignore the constraints $\|\boldsymbol{v}_k\| = 1$ during the optimization. To initialize the common rotation axes, we take the average over all independent estimates which belong to the same axis. Throughout the paper, an according strategy is applied whenever two or more rotations have common parameters. If the rotation axes are known (Figure 1: "known axes"), the parameters $\boldsymbol{v}_k$ are constant and need not to be optimized, and we have a minimal parameterization with $5 + n$ parameters.

3.3 Rotation Angles Known Up to Scale

Things simplify even more if we have knowledge about the rotation angle. Using a pan-tilt-unit, relative rotation angles are often known in some device specific unit. If the pan-tilt-unit is calibrated and provides a mapping from machine units to radians, we get the actual angles. Otherwise, we assume a linear mapping from angles β_i in machine units to radians α_i: $\alpha_i = \theta_{k_i}\beta_i$, where θ_{k_i} is the unknown scale factor, which may be specific to each of the r rotation axes. This results in a total of $5 + 3r$ parameters for r unknown but fixed rotation axes (Figure 1: "common axes + known* angles"). To formulate an appropriate optimization problem, we begin with equation (5). We encode each axis *and* scale factor by *one* vector $\boldsymbol{u}_k$. For each actual rotation, the appropriate vector $\boldsymbol{u}_{k_i}$ is multiplied by the known angle in machine units β_i to produce the exponential parameters ($\boldsymbol{w}_i$ in equation (5)). We now minimize over the vectors $\boldsymbol{u}_k$:

$$\boldsymbol{K} = \underset{\boldsymbol{K}}{\operatorname{argmin}} \min_{(\boldsymbol{u}_k)_{1\leq k\leq r}} \sum_{i=1}^{n}\sum_{j=1}^{m_i} d\left(\boldsymbol{K}\mathrm{Rod}(\beta_i\boldsymbol{u}_{k_i})\boldsymbol{K}^{-1}\boldsymbol{p}_{i,j}, \lambda''_{i,j}\boldsymbol{p}'_{i,j}\right)^2 \quad . \tag{7}$$

If the rotation axes are known (Figure 1: "known axes + known* angles"), we start with the formulation in equation (6). The angles α_i are replaced by $\theta_{k_i}\beta_i$, and the values θ_k are the new optimization parameters ($5 + r$ parameters in total):

$$\boldsymbol{K} = \underset{\boldsymbol{K}}{\operatorname{argmin}} \min_{(\theta_k)_{1\leq k\leq r}} \sum_{i=1}^{n}\sum_{j=1}^{m_i} d\left(\boldsymbol{K}\mathrm{Rod}(\theta_{k_i}\beta_i, \boldsymbol{v}_{k_i})\boldsymbol{K}^{-1}\boldsymbol{p}_{i,j}, \lambda''_{i,j}\boldsymbol{p}'_{i,j}\right)^2 \quad . \tag{8}$$

3.4 Common Rotations

It may be possible to control the pan-tilt-unit such that all rotations about each physical axis share the same angle (which we do not need to know). In other words, there are only two (or more generally: r) unknown rotation matrices (Figure 1: "common rotations"). Mathematically, this situation is a special case of equation (7) if we set $\beta_i = 1$ for all i. Thus, there is again a total of $5 + 3r$ parameters. This case is also a direct special case of equation (5) with fewer rotation parameters.

3.5 Known Rotations

Finally, rotations can be known completely ("known rotations"). In practice, such data can be provided by a dedicated rotation sensor (Figure 1: "known rotations"). Calibrated pan-tilt-units, as described above, are a further possibility ("known axes + known* angles" plus known scale factors θ_k). To get an appropriate optimization problem, we observe that this is a special case of *each* case presented above. We choose equation (5). The rotation parameters $\boldsymbol{w}_i$ are now constant and need not to be optimized.

This situation with only five parameters has already been briefly mentioned by Hartley [2], who used a similar nonlinear formulation. Frahm and Koch [4,5] investigated this case in more detail and presented a linear algorithm. Note, however, that these approaches cannot benefit from only *partially* known rotations.

3.6 A Note on Real Pan-Tilt-Units

All rotation knowledge in equation (2) and the above mentioned reformulations is expressed in the camera coordinate system. In the cases "common axes" and "common rotations", this seems uncritical at first sight, but really *is* an issue, as will be explained in this section. In the cases "known axes" and "known rotations" the problem is quite obvious: the alignment of the rotation axes with the camera coordinate system needs to be known. The additionally complicating issue, which is relevant to *all* cases of partial rotation knowledge, is that the alignment of the rotation axes with the camera coordinate system is typically *not* constant even though the camera is rigidly mounted onto (or into) the pan-tilt-unit. In most (if not all) pan-tilt-units, one of the rotation axes is placed "on top" of the other one. This means that the camera coordinate system is rigidly mounted relative to one of the rotation axes only. In case of the Directed Perception PTU-46-17.5, the pan mechanism rigidly rotates the tilt axis *and* the camera. However, the tilt mechanism only rotates the camera and does *not* affect the pan mechanism. Thus, tilting changes the alignment of the *pan* axis relative to the camera coordinate system. We can avoid this problem by keeping the tilt setting constant for *all* pan rotations, e.g. by first performing pan rotations and then tilt rotations.

4 Experiments

In the experiments, we investigate all cases of partially known rotations shown in Figure 1. We also include the linear standard algorithm. By simulation, we demonstrate how the influence of noise reduces when using partial rotation knowledge. We also present results for real hardware.

4.1 Simulation

For the simulation, we use a virtual pinhole camera with parameters $f_x^{(GT)} = f_y^{(GT)} = 100$, $s^{(GT)} = 0$, $o_x^{(GT)} = 150$, $o_y^{(GT)} = 100$. In normalized pixel coordinates, the values are: $f_x^{(GT,N)} = f_y^{(GT,N)} = 2/3$, $s^{(GT,N)} = o_x^{(GT,N)} = o_y^{(GT,N)} = 0$. These parameters are of course unknown to the self-calibration algorithms. Point correspondences are generated by projecting 100 3D points into the camera twice – before and after rotating the camera about its X or Y axis. Each resulting 2D point is modified by uniformly distributed, additive noise in the range $[-\phi/2, \phi/2] \times [-\phi/2, \phi/2]$. We systematically perform experiments for different values of the noise parameter ϕ. If one of the resulting 2D points of a corresponding pair lies outside of the image area $[0, 300] \times [0, 200]$, the pair is discarded. The 3D points are randomly generated by a uniform distribution on the cuboid $[-15000, 15000] \times [-10000, 10000] \times [-10000, 10000]$ (in pixel units). As an alternative, more difficult situation, we change the values of some parameters as follows: $f_x^{(GT)} = f_y^{(GT)} = 400$ (normalized: $f_x^{(GT)} = f_y^{(GT)} = 8/3$), 2000 3D points.

We perform a series of ten (relative) rotations about the Y axis followed by another ten rotations about the X axis. The rotation angle is 10° for each rotation. The initial configuration for the first sequence of rotations is $\boldsymbol{R}_o = \mathrm{Rod}((0, -25°, 0)^T)$, and $\boldsymbol{R}_o = \mathrm{Rod}((0, 0, -25°)^T)$ for the second one. We measure the error of the self-calibration result $\boldsymbol{K}^{(N)}$ by computing the Frobenius norm of the difference between $\boldsymbol{K}^{(N)}$ and the ground truth data $\boldsymbol{K}^{(GT,N)}$, both expressed in normalized pixel units: $e_F = \left\| \boldsymbol{K}^{(N)}, \boldsymbol{K}^{(GT,N)} \right\|_2$. If the self-calibration fails, e.g. because $\boldsymbol{\omega}^*$ is not positive definite, we set $e_F = \infty$. Each experimental setup is simulated 100 times with identical parameters. For the final evaluation, we compute the median of e_F over all 100 runs.

Results: Figure 2 (left) shows the results of the simulation in the simple situation. You can clearly see the improvements gained by using partial rotation knowledge. As expected, a greater amount of rotation knowledge leads to better results. The linear algorithm is outperformed by the nonlinear ones, and additional rotation knowledge further improves the results. However, there is a clustering in the plots: "common axes", "common axes + machine angles" and "common rotations" perform almost equally well. The same is the case for "known axes" and "known axes + machine angles". Obviously, in this experiment, knowing rotation angles up to scale does *not* further improve self-calibration. However, there is a pronounced difference between "common axes", "known axes" and completely "known rotations". Knowing only "common axes" is already better than "unknown rotations", although this is clearly visible only for high noise levels. Note how these results agree with the hierarchy in Figure 1.

The results for the difficult situation are shown in Figure 2 (right). Note that the scale of the error axis is more than ten times larger in the right plot revealing that this situation *is* actually a lot more difficult. As far as the ranking of the algorithms is concerned, the main impression is the same. However, there are a few interesting differences to the simple situation. The most striking one is the large improvement gained by completely "known rotations". Beginning with noise level $\phi = 6$, there *is* now also an improvement gained by knowing angles up to scale in the case of "common axes". However, in the case "known axes + known* angles" there is a serious problem. The reason for this might be some bad local minimum, which cannot be overcome

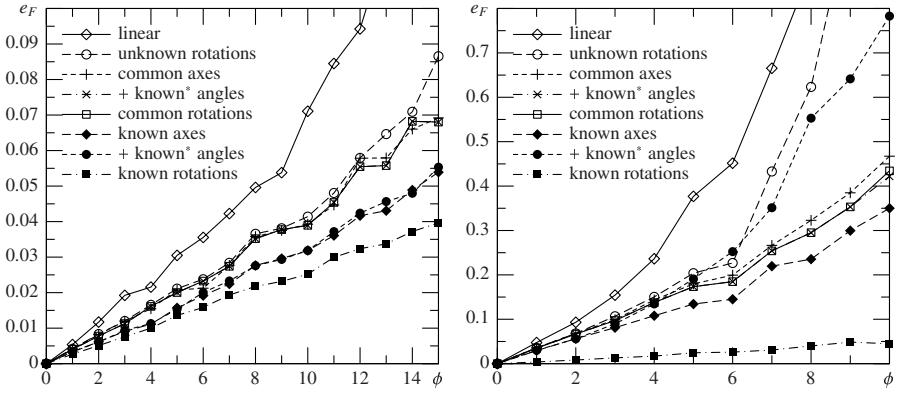

Fig. 2. Median of Frobenius error e_F for various noise levels ϕ. Left: simple situation ($f_x^{(GT)} = f_y^{(GT)} = 100$). Right: difficult situation ($f_x^{(GT)} = f_y^{(GT)} = 400$). Note the different scalings of the Y axes.

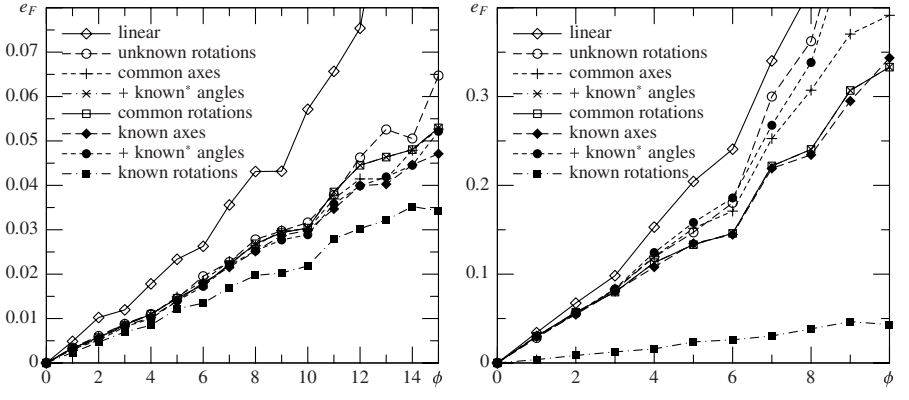

Fig. 3. Median of Frobenius error e_F for various noise levels ϕ with the assumption of zero skew. Left: simple situation ($f_x^{(GT)} = f_y^{(GT)} = 100$). Right: difficult situation ($f_x^{(GT)} = f_y^{(GT)} = 400$). Note the different scalings of the Y axes.

in the restricted seven dimensional space given a bad initialization for the angle scale parameters. A further possible explanation would be problems during the optimization caused by ill-conditioned Hessians, as might occur if the initialization is too bad.

The results for the zero skew variants of the algorithms are shown in Figure 3. The first important observation is that the error is reduced further. However, the improvement gained by (correctly) assuming zero skew is different for the various algorithms. As the good algorithms seem to gain less, the difference between the results for the various types of partial rotation knowledge is much smaller, but still visible. The case of completely "known rotations" distinguishes surprisingly well from the rest.

Table 1. Median Frobenius errors e_F and median relative errors in percent on the normalized focal length $e_r = 100|f_x^{(N)} - f_x^{(GT,N)}|/f_x^{(GT,N)}$ for the experiments on real data ($f_x^{(GT,N)} = 2.4098$)

	base algorithm								zero skew variant							
	t10		t5		o10		o5		t10		t5		o10		o5	
algorithm	e_F	e_r	e_F	e_r	e_F	e_r	e_F	e_r	e_F	e_r	e_F	e_r	e_F	e_r	e_F	e_r
linear	1.11	35	1.13	35	1.29	38	1.24	38	1.11	35	1.13	34	1.26	37	1.23	38
unknown rotation	1.04	36	1.06	35	1.24	44	1.20	40	1.04	35	1.06	35	1.21	44	1.18	40
common axes	1.00	35	1.04	34	1.13	41	1.13	38	0.99	34	1.04	34	1.13	41	1.12	38
+ known* angles	0.99	34	1.01	34	1.13	37	1.15	36	0.98	33	1.00	33	1.13	37	1.14	36
common rotations	0.99	34	1.01	34	1.13	37	1.15	36	0.98	33	1.00	33	1.13	37	1.14	36
known axes	1.17	37	1.15	35	1.34	44	1.22	40	0.97	33	1.03	33	1.05	38	1.08	37
+ known* angles	1.12	35	1.13	35	1.30	38	1.24	38	1.11	35	1.13	34	1.26	37	1.23	38
known rotations	0.12	2	0.12	2	0.17	2	0.15	2	0.12	2	0.12	2	0.17	2	0.15	2

4.2 Real Camera

For the experiments with real hardware, we use a Sony DFW-VL500 progressive scan firewire camera mounted onto a Directed Perception PTU-46-17.5 pan-tilt-unit such that the tilt axis is parallel to the X axis of the camera coordinate system, and the pan axis for tilt setting 0 is parallel to the Y axis. Note that this setup violates the pure rotation assumption, i.e. the translation vector $\boldsymbol{t}$ is *not* zero. Lacking ideal hardware, we nonetheless assume rotation about the X and Y axis of the camera coordinate system. Note that this problem is *not* specific to our approach, but common to the vast majority of the rotational self-calibration literature. We also adopt the further common simplification of ignoring camera distortions.

The pan-tilt-unit performs two rotation subsequences similar to the setup in the simulation. We use two different scenes for the experiments: a wall with artificial texture (t), which is well suited for point tracking, and a typical office environment (o). For each scene, we record one sequence with 10° rotations and another one with 5° rotations. Each of the four sequences (t10, t5, o10, o5) is repeated ten times with a randomly modified initial pan-tilt configuration. The videos are available online at `http://www4.informatik.uni-jena.de/selfcalib`. To get point correspondences, we track up to 200 points using KLT tracking [12]. All points which could be tracked from the beginning to the end of each 5° or 10° subsequence, respectively, are used as point correspondences.

A ground truth estimation of the camera parameters is performed using Zhang's [13] method. As in the simulation, we compute the median (over ten sequences) of the Frobenius error of the self-calibration results in normalized pixel units. The results are listed in Table 1. For all algorithms, except for "known rotations", the error is very large. Actually, the focal lengths f_x and f_y are severely overestimated throughout the experiments. The reason for this is very probably a violation of model assumptions: pure rotation about the optical center and a distortion-free camera. Given the good optics of the test camera, we expect the non-ideal pan-tilt-unit to cause most of the systematic error. Note, however, that our algorithms with partial rotation knowledge are able

to reduce the systematic error in most cases. The algorithm with completely "known rotations" can even greatly reduce it and produce reasonable results.

5 Conclusions

We have presented improvements for rotational self-calibration with partially known rotations, which are available, e.g., when using a pan-tilt-unit to rotate the camera. The knowledge is exploited by restricting the rotation parameterization in a nonlinear self-calibration algorithm. In systematic simulations, we showed that our new algorithms can reduce the sensitivity to noise. The experiments on real data revealed a systematic error, probably caused by non-zero translation. Our algorithms with partial rotation knowledge were able to reduce this error. In case of full rotation knowledge, the remaining error was very small in comparison.

As future research, given the problems with the non-ideal pan-tilt-unit, we plan to extend our approach to be able to deal with the case of constant non-zero translation. We also plan to extend our formulation of partial rotation knowledge such that we do not have to express the rotation axes in the camera coordinate system.

References

1. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
2. Hartley, R.I.: Self-Calibration of Stationary Cameras. International Journal of Computer Vision 22(1), 5–23 (1997)
3. Quan, L., Triggs, B.: A Unification of Autocalibration Methods. In: Proceedings of the Fourth Asian Conference on Computer Vision, pp. 917–922 (2000)
4. Frahm, J.M.: Camera Self-Calibration with Known Camera Orientation. PhD thesis, Institut für Informatik, Universität Kiel (2005)
5. Frahm, J.M., Koch, R.: Camera Calibration with Known Rotation. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2, pp. 1418–1425 (2003)
6. de Agapito, L., Hayman, E., Reid, I.D.: Self-Calibration of Rotating and Zooming Cameras. International Journal of Computer Vision 45(2), 107–127 (2001)
7. Trucco, E., Verri, A.: Introductory Techniques for 3D Computer Vision. Prentice-Hall, Englewood Cliffs (1998)
8. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(6), 580–593 (1997)
9. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. SIAM Society for Industrial and Applied Mathematics, Philadelphia (2000)
10. Ma, Y., Soatto, S., Košecká, J., Sastry, S.: An Invitation to 3D Vision, Springer, Heidelberg (2004)
11. Schmidt, J., Niemann, H.: Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization. In: Proceedings of the Vision Modeling and Visualization Conference, pp. 399–406 (2001)
12. Shi, J., Tomasi, C.: Good Features to Track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600. IEEE Computer Society Press, Los Alamitos (1994)
13. Zhang, Z.: Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 666–673. IEEE Computer Society Press, Los Alamitos (1999)