Towards smooth generic camera calibration

Alexey Pak

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
alexey.pak@ies.uni-karlsruhe.de

Technical Report IES-2014-06

Abstract: In the common approach to the generic camera calibration (GCC), one uses dense ray coding (with e.g. active grids displayed on an LCD screen) in order to find the ray origin and direction for each camera pixel independently. While applicable to many types of imaging sensors, the GCC fails to describe the local differential properties of ray bundles that are important in e.g. studying the geometry of infinitesimal scene changes via optical flow. In this report, we investigate the alternative approach to the GCC where the camera ray origins and directions are assumed to be differentiable functions of the sensor position. In particular, we present a novel calibration technique based on finite element method that unites the ray update and bundle adjustment stages of the common GCC and accommodates arbitrary anisotropic coding uncertainties and non-planar coding surfaces. The accuracy and the stability of the resulting smooth generic camera calibration (sGCC) algorithm are verified based on some non-trivial synthetic examples.

1 Introduction

The geometric camera model is a mapping $(u, v) \to \{\vec{o}_c, \vec{r}_c\}$ that assigns to each sensor pixel with cooridnates (u, v) a three-dimensional ray that originates in point $\vec{o}_c$ and is directed along the vector $\vec{r}_c$ (both defined in the local camera's system of coordinates). It is assumed that any point on that ray projects to the same pixel[1].

The simplest model that is widely adopted in computer graphics and in computer vision is a pinhole camera. It assumes a single projection center $\vec{o}_c$ for all pixels and some parameterization of $\vec{r}_c(u, v)$ which in the simplest case is linear:

$$\vec{o}_c(u, v) = (0, 0, 0)^T, \;\; \vec{r}_c(u, v) = (a_1 u + a_2 v, a_3 u + a_4 v, 1)^T. \qquad (1.1)$$

[1] In this report we ignore any finite sharpness effects.

The coefficients a_1, ..., a_4 may describe the "anisotropic magnification" and the "skewness" of the camera. (From here on, we assume that the center of the sensor is at $(u, v) = (0, 0)$.) Note that provided the coefficients a_1, ..., a_4 and the extrinsic camera parameters, one can easily find the inverse mapping that determines a sensor projection for any given 3D point.

When high-precision results are needed and/or when complex cameras and lenses are used, the linear pinhole model becomes too inaccurate and leads to systematic errors both in image generation and exploitation. One possible way to fix Eq. (1.1) is to introduce a few higher-order polynomial corrections into the second equation.

The calibration for the linear model as well as for the model with a few correction terms can be performed with the convenient algorithm developed by Zhang [Zha00] that is implemented e.g. in the OpenCV library [Bra00]. One needs only a few images of a flat static textured pattern (such as a checkerboard) taken from at least three different camera poses. The algorithm uses a sparse subset of features recognizable in all images (e.g. corners) in order to determine the camera poses and the model parameters.

However, even the corrected pinole model cannot accurately describe cameras with multiple projection centers or those with a wide-angle ($> 180^\circ$) field of view. It is also insufficiently accurate for the metrological tasks where more higher-order corrections need to be compensated for, than what is allowed by a fixed-order model. The more accurate model-independent technique of the generic camera calibration (GCC) [SR03, RSL05] attempts to produce a large look-up table with two vectors $\{\vec{o}_c[i], \vec{r}_c[i]\}$ per each camera pixel i. This model allows one to describe different visual sensors, including arbitrary multi-camera or catadioptric systems; it is also used in precision metrology (see e.g. [RLBB14] and references therein).

Unfortunately, the look-up table concept is notoriously inconvenient for the solution of the inverse problem. Indeed, given a 3D point, one needs to search the entire table for the "respective" pixel that minimizes the re-projection error. The lack of the neighborhood information for each pixel also complicates the analysis of the 3D motion that results in small image displacements over the sensor. Since the pixel coordinates u and v are not explicitly used, the derivatives such as $\partial\vec{o}_c/\partial u$ cannot be evaluated efficiently. Even if all pixels in some neighborhood have been found, the independent processing of camera pixels in the GCC algorithm leads to uncorrelated noise in $\vec{o}_c[i]$ and $\vec{r}_c[i]$ that necessitates the use of higher-order schemes to evaluate the partial derivatives numerically.

In practical implementation, the GCC requires a dense map of correspondences that cannot be obtained with static patterns. Instead, one uses active screens that project coding patterns in synchronization with the camera. For each camera pose, the screen displays a series of patterns and the camera records images so that the

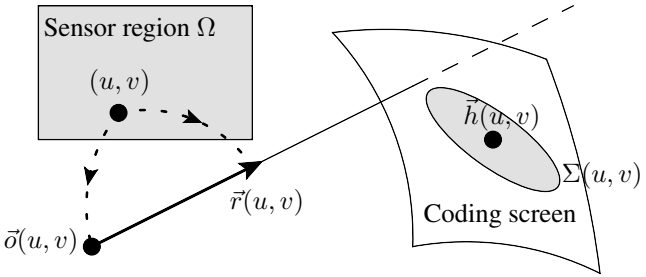

Figure 2.1: Geometry of camera calibration with active coding screen

3D position of the screen pixel that projects to a camera pixel can be decoded from the unique series of the recorded values.

In this report we present the formulation of the variation of the GCC that we call smooth GCC (sGCC) that uses the same input data as the regular GCC in order to recover the smooth functions $\vec{o}_c(u, v)$ and $\vec{r}_c(u, v)$ that alleviate the problems outlined above.

2 Single ray consistency metric

In the calibration setup shown in Fig. 2.1 we consider one camera ray that hits the known arbitrary active coding surface. The global and the camera frames are related via translation vector $\vec{t}$ and the rotation matrix R (six parameter in total). In the camera's own frame, the ray origin is $\vec{o}_c(u, v)$ and its direction is $\vec{r}_c(u, v)$ where $(u, v) \in \Omega$ denote the corresponding sensor position within the limits of some region Ω. For the purpose of further discussion we assume that the coordinates u and v are continuous and that the derivatives of $\vec{o}_c(u, v)$ and $\vec{r}_c(u, v)$ are finite in every point inside Ω.

The respective coding point $\vec{h}(u, v)$ on the screen can only be determined with some finite accuracy. Typically, if a flat LCD screen is used, the lateral uncertainty is not much better than its pixel size. The depth, however, is usually known more accurately (it may e.g. be due to the non-planarity of the screen and the refraction of light in the glass). We thus may only establish the position of the coding point up to some uncertainty ellisoid around $\vec{h}$ characterized by the covariance matrix $\Sigma(u, v)$. The latter is either known a priori or can be determined directly from the coding data and the photometric parameters of the camera [FPT12].

The agreement of the calibrated camera ray with the decoded screen position must be characterized with respect to this ellipsoid. Combining all the pieces together,

we suggest the following scalar ray consistency metric:

$$\Delta = \min_{\alpha} \left(\vec{o} + \alpha\vec{r} - \vec{h}\right)^T \cdot \Sigma^{-1} \cdot \left(\vec{o} + \alpha\vec{r} - \vec{h}\right), \tag{2.1}$$

where the ray origin and the direction in global frame are given by $\vec{o} = \vec{t} + R \cdot \vec{o}_c$ and $\vec{r} = R \cdot \vec{r}_c$ and we suppressed the arguments (u, v) everywhere for brevity.

The minimum in Eq. (2.1) can be found in closed form. Indeed, every symmetric positive-definite matrix Σ can be decomposed as $\Sigma^{-1} = \Lambda^T \cdot \Lambda$, and we find

$$\Delta = \vec{\delta}^{\,T} \cdot \vec{\delta} \text{ with } \vec{\delta} = \vec{o}\,' - \vec{r}\,' \frac{\left(\vec{r}\,'^T \cdot \vec{o}\,'\right)}{\left(\vec{r}\,'^T \cdot \vec{r}\,'\right)} \tag{2.2}$$

$$\text{where } \vec{o}\,' = \Lambda \cdot \left(\vec{o} - \vec{h}\right) \text{ and } \vec{r}\,' = \Lambda \cdot \vec{r}.$$

In particular, if $\Lambda = I$, then the metric of Eq. (2.2) is equivalent to the common Euclidean distance between the line and the point $\vec{h}$ in 3D.

It shoud be noted that the ray definition above is not unique. Indeed, any change $\vec{o}_c \to \vec{o}_c + \alpha\vec{r}_c$ or $\vec{r}_c \to \beta\vec{r}_c$ for any α and β results in the identical imaging geometry and leaves Δ invariant. Such trivial freedom can be removed by introducing two additional constraints which in general may depend on the camera type. In what follows, we limit ourselves with single-sensor cameras whose field of view is less than 180°. In this case, one can use the following simple constraints:

$$\left(\vec{o}_c(u,v)\right)_3 = 0 \text{ and } \left(\vec{r}_c(u,v)\right)_3 = 1 \;\forall (u,v) \in \Omega. \tag{2.3}$$

3 Calibration as optimization problem

The ideal calibration means that $\Delta(u, v)$ vanishes over the entire sensor. Indeed, if the decoded data are available for three or more camera poses and we neglect the inter-pixel correlations, the regular GCC may determine $\vec{o}_c(u_i, v_i)$ and $\vec{r}_c(u_i, v_i)$ for all sensor pixels $i = 1, ..., N$ without any relation to the decoding error $\Sigma(u, v)$. However, the independently found origins and directions will likely contain high-frequency noise that may mask any real differential behaviour of these functions.

Instead, in the suggested sGCC framework we attempt to find *smooth* functions $\vec{o}_c(u, v)$ and $\vec{r}_c(u, v)$ that minimize the discrepancy $\Delta(u, v)$ *integrated over the sensor*. The smoothness of these functions is defined a priori and provides the separation between the variations perceived as noise and those assumed to be instrinsic features of the camera.

The practical implementation is formulated in terms of the finite element method (FEM) as follows. First, we search for the calibration functions as linear combinations of smooth kernels $\psi_i(u,v)$, $i = 1, ..., P$. Their "size" and smoothness control the properties of the solution. The noise in Δ is filtered out by the convolution with some smooth probe functions $\phi_k(u,v)$, $k = 1, ..., M$. The entire problem then is discretized. For a single camera pose, we have:

$$\vec{o}_c\left(u,v|\vec{C}\right) = \sum_{i=1}^{P}\left(c_i^{(o1)}\psi_i(u,v), c_i^{(o2)}\psi_i(u,v), c_i^{(o3)}\psi_i(u,v)\right)^T, \tag{3.1}$$

$$\vec{r}_c\left(u,v|\vec{C}\right) = \sum_{i=1}^{P}\left(c_i^{(r1)}\psi_i(u,v), c_i^{(r2)}\psi_i(u,v), c_i^{(r3)}\psi_i(u,v)\right)^T,$$

$$\vec{C}^* = \mathrm{argmin}_{\vec{C}}\left\{\sum_{k=1}^{M}\|\vec{D}_k\|^2\right\}, \quad \vec{D}_k = \int_\Omega \phi_k(u,v)\,\vec{\delta}\left(u,v|\vec{C}\right)\,du\,dv,$$

where $\vec{\delta}\left(u,v|\vec{C}\right)$ is defined in Eq. 2.2 (trivially adapted for the given camera ray parameterization), $\vec{C} = \left(\vec{c}^{\,(o1)}, \vec{c}^{\,(o2)}, \vec{c}^{\,(o3)}, \vec{c}^{\,(r1)}, \vec{c}^{\,(r2)}, \vec{c}^{\,(r3)}, \vec{t}, \vec{\theta}\right)^T$ is the vector of concatenated model parameters, including the camera translation $\vec{t}$ and the three Euler angles $\vec{\theta}$ that determine the camera rotation matrix R.

Per se, Eq. (3.1) does not lead to a unique solution. First, as in the regular GCC, one needs data from several independent camera poses. The respective changes to the metric are straightforward, each pose adding six parameters to $\vec{C}$ and $3M$ terms to the sum under argmin. Second, we need to enforce the uniqueness conditions Eq. (2.3). Finally, we must fix the camera position and the orientation with respect to the view rays. We again resort to the simplest suitable conditions:

$$\vec{o}_c(0,0) = (0,0,0)^T,\ \vec{r}_c(0,0) = (0,0,1)^T,\ \text{and}\ \frac{\partial(\vec{r}_c)_2}{\partial u}(0,0) = 0.$$

This fixes all six degrees of freedom of the camera in its own frame with respect to the ray bundle near the central pixel $(0,0)$.

Due to the linearity of $\vec{o}_c$ and $\vec{r}_c$ with respect to the coefficients $\vec{c}^{\,(oi)}$ and $\vec{c}^{\,(ri)}$, all these constraints can be succintly represented in matrix form as $A \cdot \vec{C} = \vec{b}$. We thus recognize Eq. (3.1) as a constrained non-linear least squares problem which can be efficiently solved by iterative methods.

For the practical implementation, we chose to define the kernel functions ψ_i and probe functions ϕ_k as the uniform cubic B-splines with control points on a regular

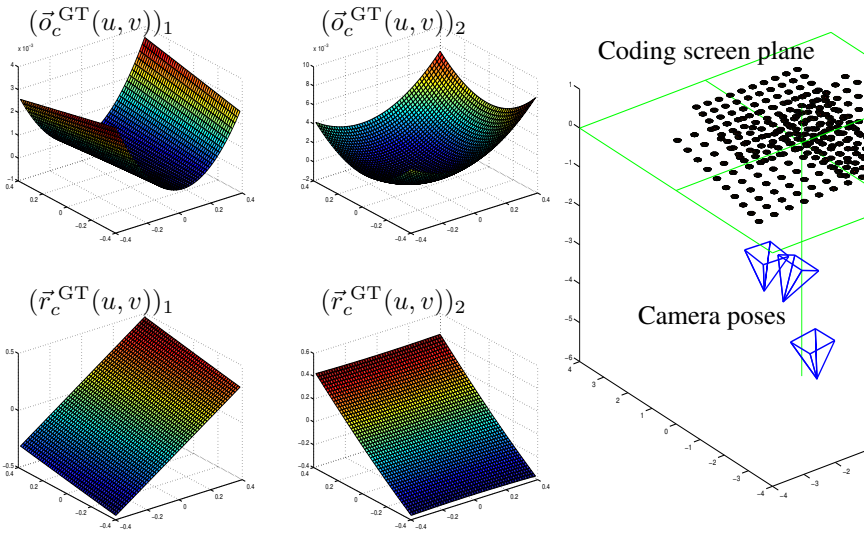

Figure 3.1: Ground truth calibration functions $\vec{o}_c^{\,\mathrm{GT}}(u,v)$ and $\vec{r}_c^{\,\mathrm{GT}}(u,v)$, actual camera poses and a selection of decoded screen points (black spots represent the uncertainty ellipsoids exaggerated by a factor of ten).

rectangular mesh in (u,v)-space. In order to solve to the optimization problem, we used the `levmar` library [Lou04].

With such choices, the dimensionality of the problem is controlled by the mesh dimensions. For instance, a mesh of 4×4 finite elements has 49 basic functions, which for three poses translates to 312 parameters in $\vec{C}$. After accounting for the linear constraints, 209 independent degrees of freedom are left, with the least squares solution to be found in 441-dimensional space (all components of $\vec{\delta}$ are considered separately). Such problem can only be efficiently solved if the respective Jacobian matrix is known. Fortunately, all components of Eq. (3.1) are known analytically and the respective derivatives $\partial\vec{D}_k/\partial(\vec{C})_j$ can be easily found manually or using the automated differentiation techniques.

4 Numerical experiment

In order to verify the feasibility and the correctness of the sGCC algorithm, we have carried out the following synthetic experiment. For some pre-defined non-trivial functions $\vec{o}_c^{\,\mathrm{GT}}(u,v)$ and $\vec{r}_c^{\,\mathrm{GT}}(u,v)$ and the three camera poses (Fig. 3.1)

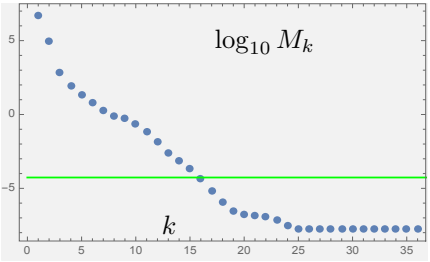

Figure 4.1: The optimization metric during the run, k is the iteration. The green line corresponds to the direct fit of splines to the ground truth functions.

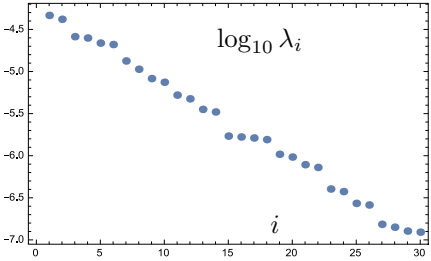

Figure 4.2: The largest eigenvalues of the covariance matrix estimated during the optimization.

we have performed the camera calibration using a 4×4 mesh in a window $(u, v) \in \Omega = [-0.4, 0.4]^2$. The typical distance from the camera to the screen was 4 units, the typical variation of $\vec{o}_c$ was 10^{-3} units, and the $\vec{r}_c$ was a smooth perturbation of order 10^{-3} units on top of a linear pinhole camera.

In order to initialize the numerical optimization, we assumed that the results of a simpler calibration (e.g., with the Zhang's method) were available. In particular, we initialize the intrinsic parameters with some perfect pinhole model, and choose the starting camera positions to deviate by about 0.1 units from the actual points. The starting camera orientations we also inaccurate by about 0.1 radians.

The 40 optimization iterations took about twenty minutes on a laptop (Intel Core i7 CPU, one thread). The optimization converged relatively fast (in about 25 iterations), the corresponding change in the optimization metric of Eq. (3.1) is shown in Fig. 4.1. It is remarkable that the found optimum provides a three orders of magnitude smaller metric value than the best fit of splines directly to the ground truth functions (the respective value denoted by the green line in Fig. 4.1).

The resulting accuracy of the intrinsic calibration functions $\vec{o}_c^{\,*}$ and $\vec{r}_c^{\,*}$ is of the order of 10^{-6} to 10^{-4} units (Fig. 4.3). The final camera position accuracy is about 10^{-4} units, and the error in its orientation is of the order of 10^{-6} radians.

In order to study the shape of the optimization function near the optimum, we also computed a few largest eigenvalues of the covariance matrix estimated by `levmar` (Fig. 4.2). The lack of large "steps" in the magnitudes of eigenvalues can be considered the sign of the "uniqueness" of the found solution.

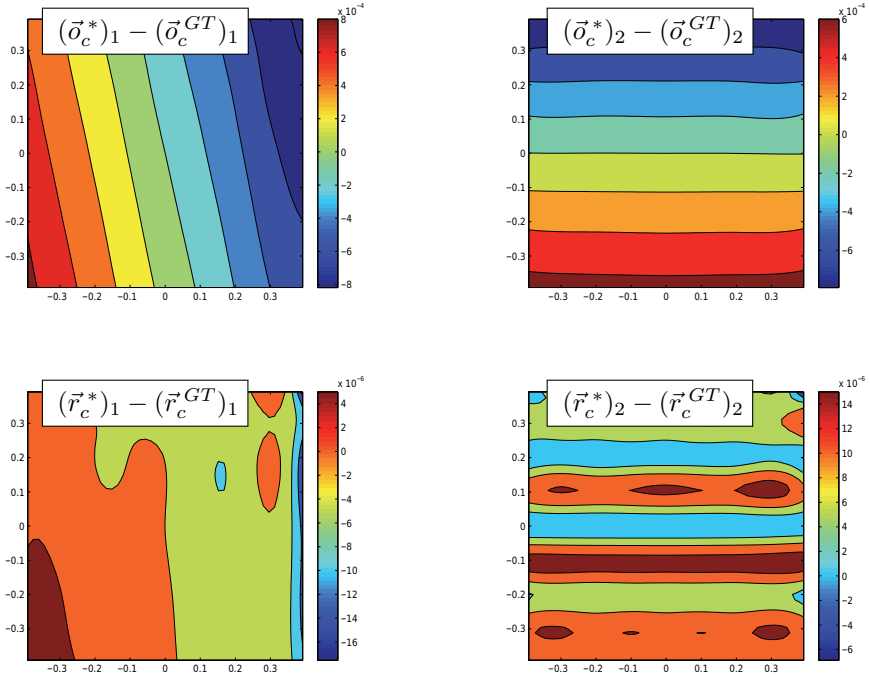

Figure 4.3: Reconstruction errors in calibration functions

Of course, one synthetic experiment is not a decisive argument for the method, and a more elaborate investigation is necessary. We consider the present results encouraging and expect similar behaviour in further synthetic and real experiments.

5 Summary

In this report we suggested a novel model-independent camera calibration algorithm (sGCC) suitable for the precision simulation and computer vision applications. We presented the theoretical background and the first numerical results characterizing the FEM-based implementation of the method for "traditional" cameras. In a synthetic experiment, the method has been able to reproduce a non-trivial intrinsic geometry of a camera together with its extrinsic parameters. The achieved re-projection error is 1000 times smaller than that when the calibrated functions are directly fitted to the ground truth. The solution is shown to be numerically stable and non-ambiguous.

The present report does not specify how exactly the sGCC simplifies the inverse (projection) problem. We note here that based on the sGCC results, it is possible to find the locally "best fitting" pinhole camera. Due to the differentiability of the calibration functions, this can be done in closed form (the details are to be presented in further publications). Iteratively projecting the point to those local pihnole cameras, one can find the projection pixel with an any accuracy.

In the future we plan to study the robustness of the sGCC with respect to noise and determine the applicability limits of the technique. We also plan to develop a practical calibration toolbox and study the real and simulated cameras under various conditions in order to compare sGCC against the existing methods.

Bibliography

[Bra00] G. Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.

[FPT12] Marc Fischer, Marcus Petz, and Rainer Tutsch. Vorhersage des Phasenrauschens in optischen Messsystemen mit strukturierter Beleuchtung. *Technisches Messen*, 79:451–458, 2012.

[Lou04] M. I. A. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, 2004. [Accessed on 13 Nov 2014].

[RLBB14] T. Reh, W. Li, Jan Burke, and R. B. Bergmann. Improving the Generic Camera Calibration technique by an extended model of calibration display. *J. Europ. Opt. Soc. Rap. Public.*, 9:14044, 2014.

[RSL05] S. Ramalingam, Peter Sturm, and S. K. Lodha. Theory and experiments towards complete generic calibration. Technical Report 5562, INRIA, 2005.

[SR03] Peter Sturm and Srikumar Ramalingam. A generic calibration concept - theory and algorithms. Research Report 5058, INRIA, 2003.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Machine Intell.*, 22:1330–1334, 2000.