Camera Calibration with Lens Distortion from Low-rank Textures

Zhengdong Zhang
Microsoft Research Asia
v-kelviz@microsoft.com

Yasuyuki Matsushita
Microsoft Research Asia
yasu@microsoft.com

Yi Ma
Microsoft Research Asia
mayi@microsoft.com

Abstract

We present a simple, accurate, and flexible method to calibrate intrinsic parameters of a camera together with (possibly significant) lens distortion. This new method can work under a wide range of practical scenarios: using multiple images of a known pattern, multiple images of an unknown pattern, single or multiple image(s) of multiple patterns etc. Moreover, this new method does not rely on extracting any low-level features such as corners or edges. It can tolerate considerably large lens distortion, noise, error, illumination and viewpoint change, and still obtain accurate estimation of the camera parameters. The new method leverages on the recent breakthroughs in powerful high-dimensional convex optimization tools, especial those for matrix rank minimization and sparse signal recovery. We will show how the camera calibration problem can be formulated as an important extension to principal component pursuit, and solved by similar techniques. We characterize to exactly what extent the parameters can be recovered in case of ambiguity. We verify the efficacy and accuracy of the proposed algorithm with extensive experiments on real images.

1. Introduction

Camera calibration is arguably one of the most classic and fundamental problems in computer vision (and photogrammetry), which has been studied extensively for decades. It is fundamental because not only every newly produced camera must run calibration to correct its radial distortion and intrinsic parameters, but also it is the first step towards many important applications in vision, such as reconstructing 3D structures from multiple images (structure from motion, photometric stereo, structured lights etc).

Existing methods have provided us with many choices to solve this problem in different settings. To the best of our knowledge, almost all of calibration methods rely on extraction of certain local features first, such as corners, edges, and SIFT features, and then assemble them to establish correspondences, calculate vanishing points, infer lines or conic curves for calibration. It is well-known that in practice it is difficult to accurately and reliably extract all wanted features in all images in the presence of noise, occlusion, image blur, and change of illumination and viewpoint. Large noise, outliers, missing features, and mismatches all could render the calibration result inaccurate and even invalid. Today arguably the only reliable way to obtain accurate calibration and distortion still relies on *manually* labeling the precise location of points in multiple images of a pre-designed pattern, as required by most standard calibration toolboxes (e.g. [1]). Not only does the use of a pre-designed pattern limit the use of such methods to restricted (laboratory) conditions, but also the careful manual input makes camera calibration a time-consuming task.

(a) Image from a fisheye camera (b) Distortion automatically corrected

Figure 1. Distortion in an image of a building taken by a fisheye camera automatically corrected by our method.

Recently, breakthroughs in high-dimensional convex optimization have enabled people to correct global geometric distortion of images directly using image intensity values. In particular, the recent work [17] has shown that for an image of a plane whose texture, as a matrix, is very low-rank, one can efficiently and accurately recover the low-rank texture from its perspectively distorted version via convex rank minimization techniques. Inspired by that work, in this paper, we show how such optimization techniques can help solve the camera calibration problem in a more convenient and flexible way. A representative result of our method is given in Fig. 1, in which the lens distortion of a fisheye camera is corrected based on an image itself.

Contributions. In this paper, we will show that this new approach leads to *a simple and accurate solution to camera calibration or self-calibration without requiring extracting, labeling, or matching any low-level features such as points and edges*. The new algorithm directly works with raw image intensity values and can accurately estimate the camera intrinsic parameters and lens distortion under a broad practical conditions: from a single or multiple images, from a known or unknown pattern, even with possible noise, saturations, occlusion, and under different illuminating conditions. It can be used either for pre-calibrating the camera from a known pattern or for performing automatic self-calibration from images of structured scenes. It requires few, inaccurate initialization, and thus is very convenient to use. Also, as it relies on scalable optimization techniques, with proper implementation, the speed can be very fast. As we will verify with extensive experiments, the algorithm achieves comparable performance to the standard toolbox, but with more flexible initialization and working under broader realistic conditions.

1.1. Prior work

During the past several decades, researchers have studied many different approaches for developing more convenient, practical, and accurate algorithms for camera calibration.

One important class of these solutions require a specially designed calibration object, with 3-D geometric information known explicitly [2, 6, 7, 14, 15]. The calibration objects include 3-D [14], 2-D plane [15], and 1-D [16] line targets. By observing these targets from different viewpoints, these techniques recover the camera intrinsic parameters. The 3-D calibration object usually consists of two or three planes orthogonal to each other, and it gives the most accurate calibration with a simple algorithm; however, the setup is more complicated and expensive. The 2-D plane-based calibration requires observing a planar pattern from different viewpoints. The technique is implemented in Camera Calibration Toolbox [1], and it gives accurate results with less complicated settings. The 1-D line-based calibration uses a set of collinear points with known distances. Because it can better avoid occlusion problems, it is often used for multi-camera calibration.

Unlike above methods, *camera self-calibration* [11, 8] avoids the use of known calibration pattern and aims at calibrating a camera by finding intrinsic parameters that are consistent with the geometry of a given set of images. It is understood that sufficient point correspondences among three images are sufficient to recover both intrinsic and extrinsic parameters. Because self-calibration relies on point correspondences across images, it is important for these approaches to extract accurate feature point locations and it normally does not handle lens distortion.

Calibration based on *vanishing points* are also investigated by researchers [3, 10, 13, 4, 9]. These approaches utilize parallelism and orthogonality among lines in the 3-D space. For example, certain camera intrinsics with the rotation matrix can be estimated from three mutually orthogonal vanishing points. While useful, these approaches strongly rely on a process of edge detection and line fitting for accurately determining vanishing points. Methods that use line features like done by Devernay and Faugeras [5] share similar processes, and the accuracy and robustness are too susceptible to noisy and faulty low-level feature extraction.

All in all, almost all calibration methods share one thing in common, *i.e.*, almost exclusively relying on whether points or lines can be reliably obtained from local corner or edge features. Feature extraction or labeling often becomes a bottleneck of the process, affecting robustness, accuracy, and convenience. The proposed new method naturally avoids this problem by a new formulation that does not require any low-level feature extraction.

2. Camera Model with Lens Distortion

We first briefly describe the common mathematical model used for camera calibration and introduce notation used in this paper. We use a vector $M = (X_0, Y_0, Z_0)^T \in \mathbb{R}^3$ to denote the 3D coordinates of a point in the world coordinate frame, use $m_n = (x_n, y_n)^T \in \mathbb{R}^2$ to denote its projection on the canonical image plane in the camera coordinate frame. For convenience, we always denote the homogeneous coordinate of a point m as $\tilde{m} = \left[\begin{smallmatrix} m \\ 1 \end{smallmatrix}\right]$.

Lens distortion model. If the lens of the camera is distorted, on the image plane, the coordinates of a point m_n may be transformed to a different one, denoted as $m_d = (x_d, y_d)^T \in \mathbb{R}^2$. A very commonly used general mathematical model for this distortion $D : m_n \mapsto m_d$ is given by a polynomial distortion model [2] by neglecting any higher-order terms as below:

$$\begin{aligned} r &\doteq \sqrt{x_n^2 + y_n^2}, \\ f(r) &\doteq 1 + k_c(1)r^2 + k_c(2)r^4 + k_c(5)r^6, \\ m_d &= \begin{bmatrix} f(r)x_n + 2k_c(3)x_n y_n + k_c(4)(r^2 + 2x_n^2) \\ f(r)x_n + 2k_c(4)x_n y_n + k_c(3)(r^2 + 2y_n^2) \end{bmatrix}. \end{aligned} \tag{1}$$

Notice that this model has a total of five unknowns $k_c(1), \ldots, k_c(5) \in \mathbb{R}$. If there is no distortion, simply set all $k_c(i)$ to be zero, and then it becomes $m_d = m_n$.

Intrinsic parameters. To transform a point into the pixel coordinates, we use the usual pin-hole model parametrized by an intrinsic matrix $K \in \mathbb{R}^{3\times3}$, which also have five unknowns; the focal length along x and y-axes f_x and f_y, skew parameter θ, and coordinates of the principle point

(o_x, o_y). In the matrix form, it is described as

$$K \doteq \begin{bmatrix} f_x & \theta & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{3\times 3}. \quad (2)$$

Extrinsic parameters. Finally, we use $R = [r_1, r_2, r_3] \in \mathbb{SO}(3)$ and $T \in \mathbb{R}^3$ to denote the Euclidean transformation from the world coordinate frame to the camera frame – so-called extrinsic parameters. The rotation R can be parameterized by a vector $\omega = (\omega_1, \omega_2, \omega_3)^T \in \mathbb{R}^3$ using the Rodrigues formula [7]: $R(\omega) = I + \sin\|\omega\|\frac{\hat{\omega}}{\|\omega\|} + (1 - \cos\|\omega\|)\frac{\hat{\omega}^2}{\|\omega\|^2}$, where $\hat{\omega}$ denotes the 3×3 matrix form of the rotation vector ω, defined as $\hat{\omega} = [0, -\omega_3, \omega_2; \omega_3, 0, -\omega_1; -\omega_2, \omega_1, 0] \in \mathbb{R}^{3\times 3}$.

With all the notation, the overall imaging process of a point M in the world to the camera pixel coordinates m by a pinhole camera can be describe as:

$$\tilde{m} = K\tilde{m}_d = KD(\tilde{m}_n); \quad \lambda\tilde{m}_n = [R\ T]\tilde{M}, \quad (3)$$

where λ is the depth of the point. If there is no lens distortion ($m_d = m_n$), the above model reduces the typical pin-hole projection with an uncalibrated camera: $\lambda\tilde{m} = K[R\ T]\tilde{M}$.

For compact presentation, later in this paper, we will let τ_0 denote the intrinsic parameters and lens distortion parameters all together. We use τ_i $(i = 1, 2, \ldots)$ to denote the extrinsic parameters R_i and T_i for the i-th image. By a slight abuse of notation, we will occasionally use τ_0 to represent the combined transformation of K and D acting on the image domain, *i.e.*, $\tau_0(\cdot) = KD(\cdot)$, and use τ_i $(i = 1, 2, \ldots)$ to represent the transforms from the world to individual image planes.

3. Calibration from Low-rank Textures

Our method estimates camera parameters from low-rank textures. The pattern can be unknown, but is sufficiently *structured*, *i.e.*, as a matrix it is sufficiently low-rank (*e.g.*, the normally used checkerboard is one such pattern). We describe our method in two cases; multiple-image and single-image cases. From multiple observations of the low-rank textures, our method can fully recover lens distortion, intrinsics, and extrinsics. In the case of a single image as input, our method can estimate lens distortion as well as intrinsics with additional yet reasonable assumptions.

By default we choose the origin of the world coordinate to be the top-left corner of the image and let the image lie in the plane $Z = 0$ and X and Y be the horizontal and vertical direction, respectively.

3.1. Multiple Images of the Same Low-Rank Pattern

Suppose we have images of a certain pattern $I_0 \in R^{m_0\times n_0}$ taken from N different viewpoints $R(\omega_i)$ and T_i (in brief τ_i), with the same intrinsic matrix K and lens distortion k_c (in brief τ_0). In practice, the observed images are not direct transformed versions of I_0, each may have contained some background or partially occluded regions (say due to limited field of view of the camera). We use E_i to model such error between the original pattern I_0 and the ith observed image I_i with the transformations undone. Then mathematically we have:

$$I_i \circ (\tau_0, \tau_i)^{-1} = I_0 + E_i, \quad (4)$$

where the operator $\circ$ denotes the geometric transformations. The task of camera calibration is then to recover τ_0 and probably τ_i $(1 \le i \le N)$, too, from these images.

In general, we assume that we do not know I_0 in advance.[1] So, we do not have any ground-truth pattern to compare or correspond with for the images taken. Our goal is to fully recover the distortion and calibration by utilizing only the low-rankness of the texture I_0 and by establishing precise correspondence among the N images I_i themselves.

Rectifying deformation via rank minimization. We draw inspiration from two previous work. Since we know the pattern is low-rank, from the work on transform invariant low-rank textures (TILT) [17], we can estimate the deformation of each image I_i from I_0 by solving the following robust rank-minimization problem:

$$\min \|A_i\|_* + \lambda\|E_i\|_1, \ \text{ s.t. } \ I_i \circ (\tau_0, \tau_i)^{-1} = A_i + E_i, \quad (5)$$

with A_i, E_i, τ_i and τ_0 as unknowns. The work [17] has shown that if there is no radial distortion in τ_0, the above optimization recovers the low-rank pattern I_0 up to a translation and scaling in each axis, *i.e.*,

$$A_i = I_0 \circ \tau, \quad \text{where} \quad \tau = \begin{bmatrix} s_x & 0 & m_x \\ 0 & s_y & m_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

However, in our problem, both intrinsic parameters and distortion are present in the deformation. Therefore, a single image can no longer recover all the unknowns (and we will discuss in the next section exactly what can be recovered from a single image of low-rank patterns.)

Our hope is that multiple images give us additional information for all the unknown parameters. For that, we need to establish precise point-to-point correspondence among all the N images. Again, robust rank-minimization techniques offer a good guideline for solving this problem. In the previous work of RASL [12], the authors have proposed that multiple images can be precisely and efficiently aligned by solving a robust rank-minimization problem similar to Eq. (5). However, the resulting aligned images could

[1] This is where our method deviates from the classical camera calibration setting and it makes our method works under broader conditions. We will discuss by the end of the section what if we do know the pattern in advance.

still differ from the canonical view I_0 by an arbitrary linear transformation, and each individual image as a matrix does not need to be low-rank.

Simultaneous alignment and rectification. For calibration, we need to align all the N images point-wise, and at the same time each resulting image should be rectified as a low-rank texture. Or more precisely, we want to find the transformation τ_0', τ_i' such that $I_i, 1 \leq i \leq N$ can be expressed as

$$I_i \circ (\tau_0' \circ \tau_i')^{-1} = A_i + E_i,$$

where all A_i are low-rank and equal to each other $A_i = A_j$. Therefore, the natural optimization problem associated with this problem becomes

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \|A_i\|_* + \|E_i\|_1, \\ \text{s.t.} \quad & I_i \circ (\tau_0' \circ \tau_i')^{-1} = A_i + E_i,\ A_i = A_j. \end{aligned} \tag{7}$$

One can use optimization techniques similar to that of TILT and RASL to solve the above optimization problem, such as the Alternating Direction Method (ADM) used in [17]. However, having too many constraining terms affects the convergence of these algorithms. In addition, in practice, due to different illumination and exposure time, the N images could differ from each other in intensity and contrast. Hence, in this paper, we propose an alternative, more effective and efficient way to align the images in the desired way. The idea is to concatenate all the images as submatrices of a joint low-rank matrix:

$$\begin{aligned} & D_1 \doteq [A_1, A_2, \ldots, A_N],\ D_2 \doteq [A_1^T, A_2^T, \ldots, A_N^T], \\ & E \doteq [E_1, E_2, \ldots, E_N]. \end{aligned} \tag{8}$$

We try to simultaneously align the columns and rows of A_i and minimize its rank by solving the following problem:

$$\begin{aligned} \min \quad & \|D_1\|_* + \|D_2\|_* + \lambda\|E\|_1, \\ \text{s.t.} \quad & I_i \circ (\tau_0 \circ \tau_i)^{-1} = A_i + E_i, \end{aligned} \tag{9}$$

with A_i, E_i, τ_0, τ_i as unknowns. Notice that, by comparing to Eq. (7), which introduces in $N + \frac{N(N-1)}{2}$ constraints, the new optimization has just N constraints and hence is easier to solve. In addition, it is insensitive to illumination and contrast change across different images. One may view the above optimization as a generalization for both TILT and RASL: When $N = 1$, it reduces to TILT; and when there is no D_2, this reduces to something similar to RASL.

To deal with the nonlinear constraints in Eq. (9), we linearize the constraints $I_i \circ (\tau_0, \tau_i)^{-1} = A_i + E_i$ w.r.t all the unknown parameters τ_0, τ_i. To reduce the effect of change in illumination and contrast, we normalize $I_i \circ (\tau_0, \tau_i)^{-1}$ by its Frobenius norm to $\frac{I_i \circ (\tau_0 \circ \tau_i)^{-1}}{\|I_i \circ (\tau_0 \circ \tau_i)^{-1}\|_F}$. Let $J_i^0 = \frac{\partial}{\partial \tau_0}\left(\frac{I_i \circ (\tau_0 \circ \tau_i)^{-1}}{\|I_i \circ (\tau_0 \circ \tau_i)^{-1}\|_F}\right)$ be the Jacobian of the normalized image w.r.t. shared intrinsic and distortion parameters τ_0 and $J_i^1 = \frac{\partial}{\partial \tau_i}\left(\frac{I_i \circ (\tau_0 \circ \tau_i)^{-1}}{\|I_i \circ (\tau_0 \circ \tau_i)^{-1}\|_F}\right)$ be the Jacobian w.r.t extrinsic parameters τ_i for each image. The local linearized version of Eq. (9) becomes

$$\begin{aligned} \min \quad & \|D_1\|_* + \|D_2\|_* + \lambda\|E\|_1, \\ \text{s.t.} \quad & I_i \circ (\tau_0, \tau_i)^{-1} + J_i^0 \Delta\tau_0 + J_i^1 \Delta\tau_i = A_i + E_i, \end{aligned} \tag{10}$$

with $\Delta\tau_0, \Delta\tau_i, A_i, E_i$ as unknowns. Notice that this linearized problem is a convex optimization problem and can be efficiently solved by some of the modern high-dimensional optimization methods such as the ADM method mentioned earlier. To find the global solution to the original nonlinear problem Eq. (9), we only have to incrementally update τ_0 and τ_i by $\Delta\tau_0, \Delta\tau_i$ and iteratively rerun the above program until convergence. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 (Align Low-rank Textures for Calibration).

Input: A rectangular window $I_i \in \mathbb{R}^{m_i \times n_i}$ in each image, initial extrinsic parameter τ_i, common intrinsic and lens distortion parameters τ_0, and weight $\lambda > 0$.

While not converged **Do**

step 1: for each image, normalize it and compute the Jacobian w.r.t. unknown parameters:

$$I_i \circ (\tau_0, \tau_i)^{-1} \leftarrow \frac{I_i \circ (\tau_0, \tau_i)^{-1}}{\|I_i \circ (\tau_0, \tau_i)^{-1}\|_F};$$

$$J_i^0 \leftarrow \frac{\partial}{\partial \zeta_0}\left(\frac{I_i \circ (\zeta_0, \zeta_i)^{-1}}{\|I_i \circ (\zeta_0, \zeta_i)^{-1}\|_F}\right)\Big|_{\zeta_0=\tau_0, \zeta_i=\tau_i};$$

$$J_i^1 \leftarrow \frac{\partial}{\partial \zeta_i}\left(\frac{I_i \circ (\zeta_0, \zeta_i)^{-1}}{\|I_i \circ (\zeta_0, \zeta_i)^{-1}\|_F}\right)\Big|_{\zeta_i=\tau_i, \zeta_0=\tau_0};$$

step 2: solve the linearized convex optimization:

$$\begin{aligned} & \min \|D_1\|_* + \|D_2\|_* + \lambda\|E\|_1, \\ & \text{s.t. } I_i \circ (\tau_0, \tau_i)^{-1} + J_i^0 \Delta\tau_0 + J_i^1 \Delta\tau_i = A_i + E_i; \end{aligned}$$

step 3: update: $\tau_0 \leftarrow \tau_0 + \Delta\tau_0$, $\tau_i \leftarrow \tau_i + \Delta\tau_i$;

End While

Output: Converged solution τ_i, τ_0.

Now the remaining question is, under what conditions the correct intrinsic parameters K and the extrinsic parame-

[2] although a rigorous mathematical proof for this fact is beyond the scope of this paper.

ters (R_i, T_i) are the global minimum to the problem Eq. (9), and whether there is still some ambiguity.

Proposition 1. *Given $N \geq 5$ images of the low-rank pattern I_0 taken by a camera with the same intrinsic parameters K under generic viewpoints $\tau_i = (R_i, T_i)$: $I_i = I_0 \circ (\tau_0 \circ \tau_i), i = 1, \ldots, N$. Then the optimal solution (K', τ_i') to problem Eq. (9) must satisfy $K' = K$ and $R_i' = R_i$.*

That is, all the distortion and intrinsic parameters τ_0 can be recovered and so is the rotation R_i of each image. There is only ambiguity left in the recovered translation T_i of each image. The proof is rather routine based on existing work on cameral calibration, but for completeness, a detailed derivation is given in Appendix A as supplementary material to this paper.

With a known pattern. If the ground-truth I_0 is given and its metric is known, then we may want to align I_i to I_0 directly or indirectly. One possible solution is to slightly modify Algorithm 1 by appending D_1, D_2, E with A_0, A_0^T, E_0, respectively, and adding the constraint $I_0 = A_0 + E_0$. Another possible solution would be to align the already rectified textures A_i to I_0 by maximizing the correlation.

In both situations, with knowledge about the metric of I_0, we can uniquely determine T_i and get exactly the full set of intrinsic and extrinsic parameters. Technical justification is given in Appendix B of the supplementary material.

3.2. Self-Calibration from a Single Image

With a single plane. For most everyday usage of a camera, people normally do not need to know the full intrinsic parameters of the camera. For instance, for webcam users, it normally suffices if we can simply remove the annoying lens distortion. For such users, asking them to take multiple images and conduct a full calibration might be too much trouble. Sometimes, we need to remove the radial distortion of an image but without any access to the camera itself.

Therefore, it would be desirable if we can calibrate the lens distortion of a camera from a single image. Normally this would be impossible for a generic image. Nevertheless, if the image contains a plane with low-rank pattern rich with horizontal and vertical lines, then the lens distortion k_c can be correctly recovered using our method.

Given a single image with a single low-rank pattern, since we cannot expect to obtain all the intrinsic parameters correctly, we can make the following simplifying assumptions about K: No skew $\theta = 0$, principal point known (say set at the center of the image), and pixel being square ($f_x = f_y = f$). Although these seem to be somewhat restrictive, they approximately hold for many cameras made today. In this circumstance, if the viewpoint is not degenerate, applying the algorithm to the image of this single pattern correctly recovers the lens distortion parameters k_c and the focal length f.

With two orthogonal planes. Very often, an image contains more than one planar low-rank textures, and they satisfy additional geometric constraints. For instance, in a typical urban scene, an image often contains two (orthogonal) facades of a building. Each facade is full of horizontal and vertical lines and can be considered as a low-rank texture. In this case, the image encodes much richer information about the camera calibration: Both the focal length and the principal point can be recovered from such an image, given that the pixel of the camera is assumed to be square, *i.e.*, $f_x = f_y = f$, and there's no skew, *i.e.*, $\theta = 0$.

For simplicity, we let the intersection of these two orthogonal planes be the z-axis of the world frame, and the two planes are $x = 0$ and $y = 0$, each with a low-rank texture $I_0^{(i)}, i = 1, 2$. We take a photo of the two planes by a camera with intrinsic parameters K and lens distortion k_c, from the viewpoint (R, T). Denote the photo as I, which contains two mutually orthogonal low-rank patterns.

Let $M_L = [0 \;\; Y_1 \;\; Z_1]^T \in \mathbb{R}^3$ be a point on the left facade, and $M_R = [X_2 \;\; 0 \;\; Z_2]^T \in \mathbb{R}^3$ be a point on the right facade, and let $m_L, m_R \in \mathbb{R}^2$ be the corresponding images on I. Then we have:

$$\lambda_1 \tilde{m}_L = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} [r_2 \; r_3 \; T_1] \begin{bmatrix} Y_1 \\ Z_1 \\ 1 \end{bmatrix}, \tag{11}$$

and

$$\lambda_2 \tilde{m}_R = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} [r_1 \; r_3 \; T_2] \begin{bmatrix} X_2 \\ Z_2 \\ 1 \end{bmatrix}. \tag{12}$$

Here we have used a different translation T_1 or T_2 for each plane, mainly because otherwise we must exactly find the position of the intersection of the two planes, which is beyond the scope of this paper. So in this circumstance let $\tau_0 = [f, o_x, o_y, k_c(1:5), \omega]$ and $\tau_i = [T_i]$ and the optimization problem we need to solve to recover them is:

$$\begin{aligned} \min_{A_i, E_i, \tau_0, \tau_i} \quad & \|A_1\|_* + \|A_2\|_* + \lambda(\|E_1\|_1 + \|E_2\|_1), \\ \text{subject to} \quad & I \circ (\tau_0, \tau_i)^{-1} = A_i + E_i. \end{aligned} \tag{13}$$

With similar normalization and linearization techniques, we can solve this problem with slight modification to Algorithm 1.

Proposition 2. *Given one image of two orthogonal planes with low-rank textures, taken by a camera from a generic viewpoint (R, T) with intrinsic parameters K with zeros skew $\theta = 0$ and square pixels($f_x = f_y$). If K', R', T_1', T_2' are solutions to problem Eq. (13), then $K' = K$, $R' = R$.*

By an argument similar to the multiple-image case, to recover τ_0, we only need to rectify the left and right textures with a joint parameterization of the rotation. For completeness, we leave detailed analysis in Appendix C of the supplementary material and shows that the solution for τ_0 is indeed the correct one.

3.3. Implementation

Detection of the pattern. The initialization of our algorithm is extremely simple and flexible. The location of the initial window can be obtained from any segmentation methods that approximately detect the region of the pattern. Or it can be easily specified by a human. There is no need for the location of the initial window to be exact or even cover the pattern region. The proposed method is very robust and can converge precisely to the pattern.

Initialization. We can first run the TILT on each initial window to approximately extract the homography H_i for the ith image. Then we can obtain a rough estimate of K, R, T as from the vanishing points given by the first two columns of τ_i.[3] For lens parameters, even if large lens distortion is present, we set their initial values to be zero.

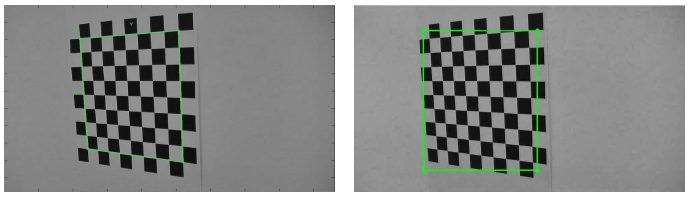
Multi-resolution implementation. To make the convergence region of our algorithm large and to accelerate the algorithm, we employ the conventional multi-resolution implementation with a proper blurring and pyramid scheme, similar to that described in the work on TILT [17].

4. Simulations and Experiments

4.1. Calibration from Multiple Images

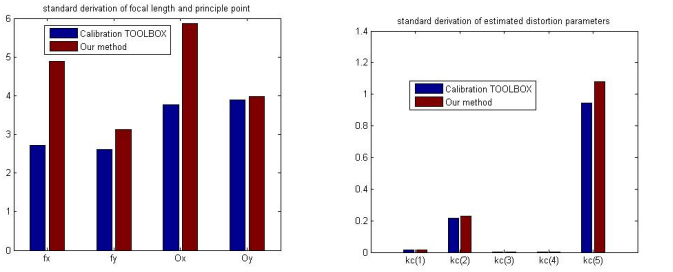
A. Calibration using a known pattern. In this experiment, we compare our proposed method with the standard camera calibration toolbox [1]. Normally, the error of calibration can be evaluated by reprojection error of extracted feature points. But since our method does not involve any feature extraction and uses only the raw image pixels, this measurement of error is no longer suitable here. Instead, we try to compare the accuracy of estimated camera parameters against the average estimates. More precisely, we run multiple experiments with different images by the same camera and compute the standard deviation for every parameter we estimate. The smaller the deviation, the more stable the estimates are.

In this experiment, we take 50 photos of a known checker-board pattern using the same camera and setting, from different viewpoints. In each experiment, we randomly select 20 out of the 50 images. With these selected images, we calibrate the camera with both the proposed method and the standard toolbox. Note that we need to manually click the precise location of the four corners of the checker-board for the toolbox. But for our method, the initialization needs not to be exact at all (several pixels away). See Figure 2 for examples. We repeat the experiment 20 times, and calculate the standard derivation of each parameter[4] for each method. The result is shown in Figure 3.



(a) Initialization for the Toolbox (b) Initialization of Our Method

Figure 2. Representative examples of initialization for the two methods. Notice that ours can be very flexible.



(a) Comparison of intrinsic parameters (b) Comparison of distortion parameters

Figure 3. Comparison with the standard calibration toolbox. Standard deviation in the estimated parameters, in pixels.

From the figure, we can see that our method is more sensitive in the estimation of focal length and principle points than toolbox, but the performance is comparable.[5] The estimation of lens-distortion parameters of our method is almost the same as the toolbox.

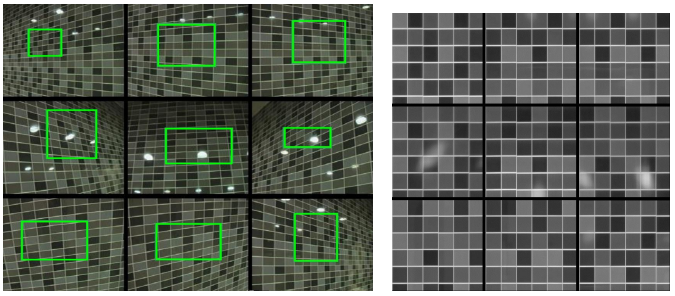
So to conclude from this experiment, under noise-free, well-controlled conditions, the performance of our method is quite comparable to the toolbox. But our method does not require exact initialization of the point location. In later experiments, we will see that our method can work under much broader conditions: with an unknown pattern, from a single image, and even when significant lens distortion exists, such as fish-eye images.

In order to verify the accuracy of the remaining experiments, unless otherwise stated, we use the same Panasonic camera with the same image resolution $[960, 540]$ (directly

[3] It is easy to see that the first two columns of τ_i correspond to the vertical and horizontal directions of the low-rank textures.

[4] By default the calibration toolbox disables the estimation of skew. When turned on, it gives an error. So in this comparison, we do not estimate skew either although our method does not have this limitation.

[5] No serious attempt has been made to improve the numerical stability of our method. We believe this sensitivity can be easily addressed by a more careful numerical implementation in the future.



(a) Input images and windows (b) Rectified and aligned textures

Figure 4. Camera calibration from images of an unknown pattern. The algorithm aligns the low-rank textures precisely despite specularities in the images. Although aligned and rectified, corresponding pixels do not have to correspond to the same 3D point.

down-sampled from its full $[1920, 1080]$ resolution). The camera parameters estimated from this experiment is:

$$K = \begin{bmatrix} 1142.0 & 0 & 453.7 \\ 0 & 1136.5 & 301.2 \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

The camera should have the same set of parameters, except for focal length which may change from experiment to experiment.

B. Calibration from an unknown pattern. In this experiment, we take multiple photos of an unknown mosaic wall from different viewpoints. By rectifying and aligning the mosaic images pixel-wise into a common canonical frame, as shown in Figure 4, we obtain the camera calibration. The recovered intrinsic matrix is:

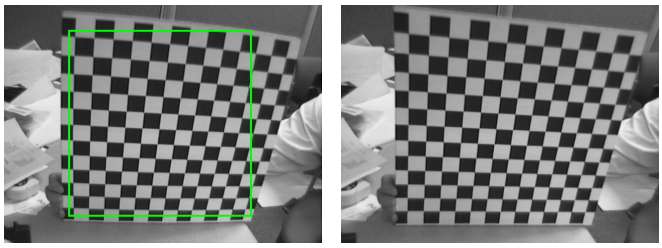
$$\hat{K} = \begin{bmatrix} 1138.6 & 0 & 482.3 \\ 0 & 1127.8 & 267.7 \\ 0 & 0 & 1 \end{bmatrix}. \quad (15)$$

4.2. Calibration from a Single Image

C. Calibration from a single pattern. Given just a single image or a regular pattern, to calibrate the camera, we have to work with fairly strong assumptions, say that the principal point is known (and simply set as the center of the image) and the pixel is square. Then from the image, one can calibrate the focal length as well as eliminating the lens distortion. Figure 5 shows an example with an image given in the standard toolbox. The estimated intrinsic parameters $\hat{K}$ and the ground-truth K (provided by the calibration toolbox) are respectively:

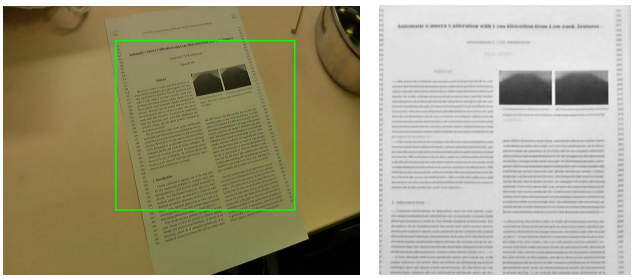
$$\begin{bmatrix} 677.1812 & 0 & 319.5000 \\ 0 & 677.1812 & 239.5000 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 661.6700 & 0 & 306.0959 \\ 0 & 662.8285 & 240.78987 \\ 0 & 0 & 1 \end{bmatrix}.$$

The small error in focal length is mainly due to that the principle point is approximated with the center of the image. Nevertheless, we see in Figure 5(b), the radial distortion is completely removed by our algorithm.



(a) Input image and window (b) Radial distortion removed

Figure 5. Calibration from a single image in the Toolbox.



(a) Input image (b) Rectified texture

Figure 6. Calibration of our camera using an image of this paper.

To show the flexibility of our method, we further show another example in Figure 6 where we took with the Panasonic camera an image of the frontal page of this paper. With this image as input, the recovered calibration matrix is:

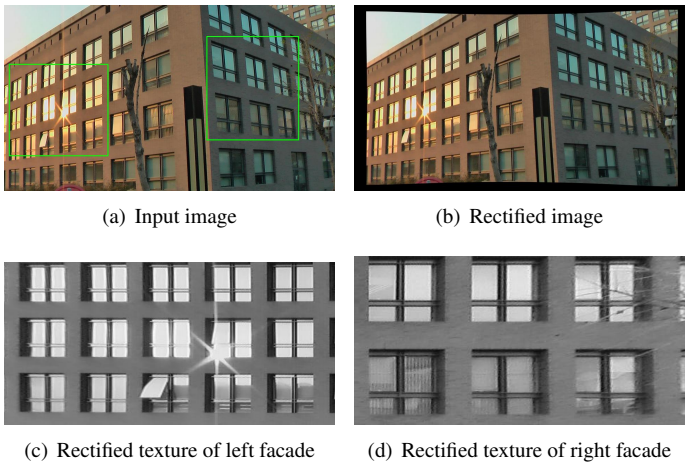
$$\hat{K} = \begin{bmatrix} 1229.1 & 0 & 479.5 \\ 0 & 1229.1 & 269.5 \\ 0 & 0 & 1 \end{bmatrix}. \quad (16)$$

D. Calibration from two orthogonal planes. In this section, we present the results of calibrating a camera from observing two orthogonal facades of a building. The result is shown in Figure 7 with the estimated calibration parameters:

$$\hat{K} = \begin{bmatrix} 1189.7 & 0 & 474.3 \\ 0 & 1189.7 & 273.4 \\ 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

E. Rectifying fisheye images. Note that the image used in Figure 5 is taken from a standard example in the MATLAB Calibration Toolbox [1], presumably the one with the largest radial distortion among all the examples. Our method can actually handle distortion far beyond that, as we show in this section with images taken by a typical fisheye camera (not the Panasonic anymore).

There have been many parametric models proposed for this kind of images, our method should apply as long as the model is known. Here to illustrate the basic idea, we make the simplifying assumptions that there is only 1-D



(a) Input image (b) Rectified image

(c) Rectified texture of left facade (d) Rectified texture of right facade

Figure 7. Calibration from two orthogonal facades of a building.

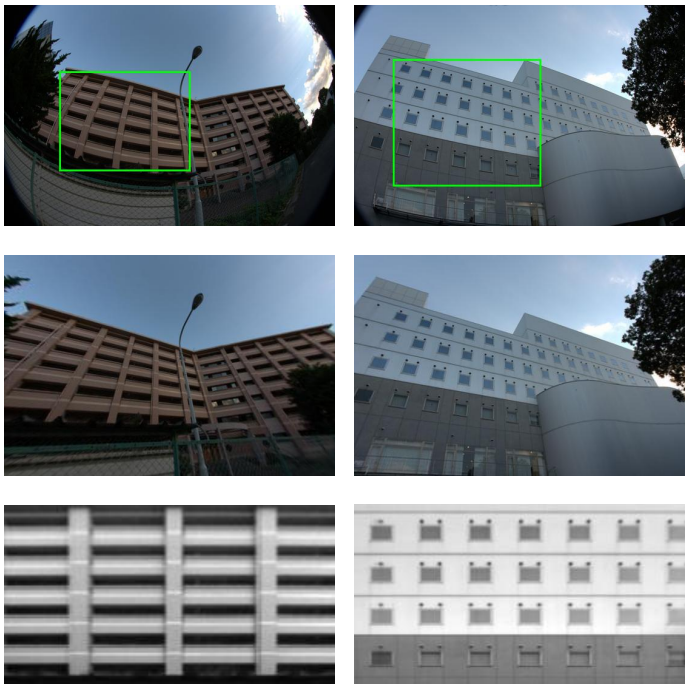


Figure 8. Rectify fisheye images with significant lens distortion. **Top:** input images with a selected window (red); **Middle:** rectified images; **Bottom:** rectified low-rank textures (from green window).

distortion along the radial direction and hence try to estimate the mapping along the radius before and after distortion $r = f(r_d)$. We approximate $f(\cdot)$ by polynomials up to degree 4. In addition, we assume the center of the distortion is the center of the image. Moreover, since no prior knowledge about K, R, T is available, we model the transformation from the pattern to the image plane as a general homography transformation $H \in \mathbb{R}^{3\times 3}$. Some representative results are shown in Figure 8.

References

[1] J.-Y. Bouguet. Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. 1, 2, 6, 7

[2] D. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971. 2

[3] B. Caprile and V. Torre. Using vanishing points for camera calibration. *IJCV*, 4(2):127–140, Mar 1990. 2

[4] R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *BMVC*, volume 2, pages 382–391, 1999. 2

[5] F. Devernay and O. D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *SPIE Conference on Investigative and Trial Image Processing*, volume 2567, pages 62–72, 1995. 2

[6] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12):1479–1486, 1975. 2

[7] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993. 2, 3

[8] O. Faugeras, Q. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *ECCV*, pages 321–334, 1992. 2

[9] L. Grammatikopoulos, G. Karras, E. Petsa, and I. Kalisperakis. A unified approach for automatic camera calibration from vanishing points. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, XXXVI(5), 2006. 2

[10] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *CVPR*, pages 482–488, June 1998. 2

[11] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *IJCV*, 8(2):123–152, Aug. 1992. 2

[12] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *CVPR*, 2010. 3

[13] P. Sturm and S. J. Maybank. A method for interactive 3D reconstruction of piecewise planar objects from single images. In *In BMVC*, 1999. 2

[14] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, Aug. 1987. 2

[15] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *ICCV*, 1999. 2

[16] Z. Zhang. Camera calibration with one-dimensional objects. *PAMI*, 26(7):892 –899, 2004. 2

[17] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma. TILT: Transform invariant low-rank textures. In *ACCV*, 2010. 1, 3, 4, 6

Supplementary Material

A. Proof of Proposition 1: Ambiguities in Calibration with an Unknown Pattern

Proof. Suppose by solving Eq. (9), we have aligned all the images up to translation and scaling of I_0. To be more specific we have managed to find $\tau_i' = (R_i', T_i'), \tau_0' = (K', k_c')$ such that

$$I_i \circ (\tau_0' \circ \tau_i')^{-1} = I_0 \circ \tau, \text{ with } \tau = \begin{bmatrix} s_x & 0 & m_x \\ 0 & s_y & m_y \\ 0 & 0 & 1 \end{bmatrix}.$$

As all the lines have become straight in the recovered images A_i, the radial distortion parameters k_c' should be exact $k_c' = k_c$.

Here s_x, s_y are scaling in the x and y directions of the aligned images A_i w.r.t. the original low-rank pattern I_0. m_x and m_y are the translations between A_i and I_0. Now let us consider the mapping between a point M_0 on I_0 (notice that the Z-coordinate is zero by default) and its image $\tilde{m} \in \mathbb{R}^3$ (in homogeneous coordinates): $\lambda \tilde{m} = K[r_1, r_2, T]M_0$. As the recovered parameters are consistent with all constraints, the same point and its image satisfy:

$$\lambda' \tilde{m} = K'[r_1', r_2', T'] \begin{bmatrix} s_x & 0 & m_x \\ 0 & s_y & m_y \\ 0 & 0 & 1 \end{bmatrix} M_0.$$

So the matrix $K[r_1, r_2, T]$ must be equivalent to $K'[s_x r_1', s_y r_2', m_x r_1' + m_y r_2' + T']$ (*i.e.*, up to a scale factor), so we have

$$\begin{cases} Kr_1 = \xi s_x K' r_1', \\ Kr_2 = \xi s_y K' r_2'. \end{cases} \Rightarrow \begin{cases} K'^{-1} K r_1 = \xi s_x r_1', \\ K'^{-1} K r_2 = \xi s_y r_2'. \end{cases} \quad (18)$$

Since $r_1'^T r_2' = 0$, we have

$$(Kr_1)^T K'^{-T} K'^{-1} (Kr_2) = 0. \quad (19)$$

This gives one linear constraint for $B = K'^{-T} K'^{-1}$. Such a symmetric B has six degrees of freedom. Since each image gives one constraint on B, we need only five general images (not in degenerate configurations) to recover B up to a scale. Since $K^{-T} K^{-1}$ is a solution too, thus we must have $K' = K$ as the unique solution of the form Eq. (2). Further from Eq. (18), we have $r_1' = r_1$, $r_2' = r_2$, and $s_x = s_y$. That is, once all the images are aligned and rectified, they only differ from the original pattern I_0 by a global scale $s = s_x = s_y$ and a translation (m_x, m_y). In addition, we recovered rotation R_i' is the correct $R_i' = R_i$. But since we still do not know the exact values of s_x, m_x, and m_y, the recovered T_i' is not necessarily the correct T_i.

With a similar analysis, we can show that in fact if we individually rectify the images, we still can obtain the correct K and R_i. The only difference is that s_x, s_y, m_x and m_y are all different for different images, thus the translations T_i are even less constrained. □

B. Determine Translation from Ground-Truth

If the low-rank pattern I_0 is given, we can directly or indirectly align I_i to I_0. From a derivation similar to the above, one can show that we can recover s_x, m_x and m_y with respect to the ground truth metric of I_0. Then for each image the ground-truth translation can be recovered by

$$T = \frac{m_x r_1 + m_y r_2 + T'}{s_x}. \quad (20)$$

C. Proof of Proposition 2: Ambiguities in Calibration with Two Orthogonal Planes

Proof. Suppose a low-rank texture lies on the left plane $X = 0$ and another lies on the right plane $Y = 0$. $M_L = (0, Y_1, Z_1)$ is the point on the left plane, and $M_R = (X_2, 0, Z_2)$ is a point on the right plane. Similarly we have the image point $m_L = (x_L, y_L)$ of the left point and $m_R = (x_R, y_R)$ of the right point. Then

$$\lambda_1 m_L = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} [r_2\ r_3\ T] \begin{bmatrix} Y_1 \\ Z_1 \\ 1 \end{bmatrix}, \quad (21)$$

and

$$\lambda_2 m_R = \begin{bmatrix} f & 0 & o_x \\ 0 & f & o_y \\ 0 & 0 & 1 \end{bmatrix} [r_1\ r_3\ T] \begin{bmatrix} X_2 \\ Z_2 \\ 1 \end{bmatrix}. \quad (22)$$

For convenience, we use (x, y) to represent points both on $Y = 0$ and on $X = 0$. Suppose the rectified image A_i differs from the ground truth $I_0^{(i)}$ by scaling and translation: $s_x^{(i)}, s_y^{(i)}, m_x^{(i)}, m_y^{(i)}$. Then the ground truth K, $R = [r_1\ r_2\ r_3]$ and T, and the recovered parameters K', $R' = [r_1'\ r_2'\ r_3']$ and T_1', T_2' are related through the following formulae:

$$\begin{aligned} &\left[s_x^{(1)} K r_2 \ \ s_y^{(1)} K r_3 \ \ K(m_x^{(1)} r_2 + m_y^{(1)} r_3 + T) \right] \\ &= \xi_1 \left[K' r_2' \ \ K' r_3' \ \ K' T_1' \right], \\ &\left[s_x^{(2)} K r_1 \ \ s_y^{(2)} K r_3 \ \ K(m_x^{(2)} r_1 + m_y^{(2)} r_3 + T) \right] \\ &= \xi_2 \left[K' r_1' \ \ K' r_3' \ \ K' T_2' \right]. \end{aligned} \quad (23)$$

This gives

$$K'^{-1} K \left[\frac{s_x^{(2)}}{\xi_2} r_1, \ \frac{s_x^{(1)}}{\xi_1} r_2, \ \frac{s_y^{(1)}}{\xi_1} r_3 \right] = [r_1', \ r_2', \ r_3']. \quad (24)$$

Knowing that r_1', r_2', r_3' are orthogonal to each other, we derive three linear constraints for $B = K'^{-T} K'^{-1}$, which has three unknowns. So in general, we can extract unique solution K' from B. Note that $K' = K$ is one solution too, hence the recovered is the correct solution.

Also, from Eq. (24) we can see that $R' = R$, leaving only T_i being ambiguous.

□