A Three-Point Minimal Solution for Panoramic Stitching with Lens Distortion

Hailin Jin
hljin@adobe.com

Advanced Technology Labs
Adobe Systems Incorporated
345 Park Avenue, San Jose, CA 95110

Abstract

We present a minimal solution for aligning two images taken by a rotating camera from point correspondences. The solution particularly addresses the case where there is lens distortion in the images. We assume to know the two camera centers but not the focal lengths and allow the latter to vary. Our solution uses a minimal number (three) of point correspondences and is well suited to be used in a hypothesis testing framework. It does not suffer from numerical instabilities observed in other algebraic minimal solvers and is also efficient. We validate our solution in multi-image panoramic stitching on real images with lens distortion.

1. Introduction

This paper deals with panoramic image alignment, which is the problem of computing geometric relationships among images for the purpose of stitching them into composites. In particular, we focus on feature-based techniques [3, 4, 24] which have been shown to be capable of handling large scene motions without initialization. Most feature-based methods are typically done in two stages: pairwise alignment and multi-image alignment. The pairwise stage starts from feature (point) correspondences, which are obtained through a separate feature extraction and matching process, and returns an estimate of the alignment parameters and a set of point correspondences that are consistent with the parameters. Robust methods, such as RANSAC [6], are often used to handle outliers in point correspondences. The multi-image stage uses nonlinear optimization techniques to further refine the alignment parameters, jointly over all the images, based on the consistent point correspondences retained in the pairwise stage. It is known that the convergence of the multi-image stage depends on how close the initial alignment parameters are to the optimal values. However, an equally important fact, which is often overlooked, is that the quality of the final result from the multi-image stage depends on the number of consistent point correspondences retained in the pairwise stage. When the number of consistent point correspondences is low, the multi-image alignment will still succeed but the quality of the final result may be poor.

In the pairwise stage, it is commonly assumed that the imaging system satisfies an ideal pinhole model. As a result, most methods only estimate either 3×3 homographies or "rotation + focal lengths" [3]. However, real imaging systems have some amount of lens distortion. Moreover, wide-angle lenses that are commonly used for shooting panoramic images introduce larger distortions than regular lenses. Modeling lens distortion is critical for obtaining high-quality alignment. One may think that it is sufficient to model lens distortion at the multi-image alignment stage. This strategy may work if all the correct correspondences are kept at the pairwise alignment. However, without modeling lens distortion at the pairwise stage, one would not be able to retain all the correct correspondences. Among those correct correspondences rejected by the model without lens distortion, most are the ones close to image borders because lens distortion effects are more pronounced for the points close to image borders than those close to image centers. Correspondences that have points close to image borders are, on the other hand, more important for estimating lens distortion, for the same reason that lens distortion effects are larger there. Losing them at the pairwise stage makes it difficult for the multi-image stage to correctly estimate lens distortion. As a result, misalignment will show up when images are stitched together, particularly along image borders. Therefore, it is crucial to estimate the lens distortion jointly with other alignment parameters at the pairwise stage.

Solving lens distortion jointly with the geometry is not a new idea. For instance, [7] proposed a linear algorithm for estimating a homography and radial distortion from point correspondences between two images which, in principle, can be used for our problem with a subsequent autocalibration stage. However, the algorithm in [7] requires five point correspondences. This increases considerably the number of trials needed in RANSAC to maintain the same level of confidence compared to the standard four correspondences for homographies or three for "rotation + focal lengths" [3]. We show that using three point correspon-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

dence, which is minimal, we can estimate "rotation + focal lengths + radial distortion."

1.1. Relations to previous work

This work falls in the category of image stitching, which is now considered as a mature topic in computer vision. There exists an extensive literature on this topic. We will only discuss closely related work here and refer the reader to [10, 24] for more thorough expositions. [19] is a pixel-based work that estimates lens distortion. The authors reported better results when lens distortion was estimated. [3, 4] represent the state-of-the-art in feature-based techniques, but neither of them estimates lens distortion at the pairwise stage. This work can be considered as an extension of [3] to imaging systems with lens distortion.

This work is built upon a series of papers that modeled lens distortion. [26] was the classical paper on camera calibration, in which a five-parameter lens distortion model was discussed. [27] argued that for practical applications, it was often sufficient and sometimes advantageous to consider only the radial components. [7] was the first to introduce the so-called "division" model for radial distortion. The advantage of this model is that it leads to simple linear methods for recovering distortion parameters along with other geometry parameters. This model was later adopted by various researchers for related problems [2, 11, 13, 20] and generalized for omnidirectional cameras [16]. In this work, we use this model for panoramic image alignment.

This work is also related to recent advances in solving minimal problems using techniques from Algebraic Geometry [3, 11, 14, 18, 22]. Since we use a minimal number of point correspondences, this work can be considered as a new addition to the family of minimal solvers. However, our solver is not based on Algebraic Geometry but instead based on nonlinear optimization. It does not suffer from numerical instabilities observed in [5, 12, 22]. Minimal solvers with lens distortion were also considered before in [8, 11], but not for the panoramic image alignment problem in which we are interested.

2. The core two-view problem

In this section, we consider the core problem in the pairwise alignment stage, which is how to relate lens distortion to point correspondences along with other geometric parameters. We will show that it is possible to derive a constraint from only three point correspondences.

We consider two cameras with coincident optical centers viewing three points P_1, P_2 and P_3. Let $\mathbf{X}_1 \in \mathbb{R}^3$ be the coordinates of P_1 with respect to the reference frame of the first camera. We model the imaging process as an ideal pinhole projection plus radial distortion. To be more precise, the pinhole model says that the projection of P_1 on the imaging plane of the first camera, $\mathbf{q}_1 \in \mathbb{R}^2$, is related to $\mathbf{X}_1$ by a perspective projection:

$$\mathbf{q}_1 = \pi(\mathbf{X}_1) \doteq [\mathbf{X}_{11}/\mathbf{X}_{13}, \mathbf{X}_{12}/\mathbf{X}_{13}]^T, \tag{1}$$

where $\mathbf{X}_1 = [\mathbf{X}_{11}, \mathbf{X}_{12}, \mathbf{X}_{13}]^T$. We model the radial distortion with the model proposed by Fitzgibbon in [7]:

$$\mathbf{q}_1 = \frac{\mathbf{p}_1}{1 + \kappa_1 \|\mathbf{p}_1\|^2}, \tag{2}$$

where $\mathbf{p}_1 \in \mathbb{R}^2$ is the radially distorted point and $\kappa_1 \in \mathbb{R}$ is the radial distortion coefficient. Finally, our measurement $\mathbf{x}_1 \in \mathbb{R}^2$, in image coordinates, is related to $\mathbf{p}_1$ through a linear transformation K_1 (intrinsic calibration):

$$\mathbf{x}_1 = K_1 \circ \mathbf{p}_1 \doteq \begin{bmatrix} f_1 & \sigma_1 \\ 0 & s_1 f_1 \end{bmatrix} \mathbf{p}_1 + \mathbf{c}_1, \tag{3}$$

where f_1 is the focal length, $\mathbf{c}_1$ is the camera center, s_1 is the aspect ratio, and σ_1 is the skew of the pixel. K_1 is invertible and its inverse K_1^{-1} is given by

$$\mathbf{p}_1 = K_1^{-1} \circ \mathbf{x}_1 \doteq \begin{bmatrix} f_1 & \sigma_1 \\ 0 & s_1 f_1 \end{bmatrix}^{-1} (\mathbf{x}_1 - \mathbf{c}_1). \tag{4}$$

Combining equations (1), (2), and (3) together, we obtain

$$\mathbf{X}_1 \sim \begin{bmatrix} K_1^{-1} \circ \mathbf{x}_1 \\ 1 + \kappa_1 \|K_1^{-1} \circ \mathbf{x}_1\|^2 \end{bmatrix}, \tag{5}$$

where $\sim$ indicates similarity relationship, i.e. the quantities are equal up to a scale. Let $\mathbf{X}_2$ be the coordinates of P_1 with respect to the reference frame of the second camera and $\mathbf{x}_2$ be the radially distorted projection. We have

$$\mathbf{X}_2 \sim \begin{bmatrix} K_2^{-1} \circ \mathbf{x}_2 \\ 1 + \kappa_2 \|K_2^{-1} \circ \mathbf{x}_2\|^2 \end{bmatrix}, \tag{6}$$

where κ_2 and K_2 are the radial distortion coefficient and the intrinsic calibration of the second camera respectively. Since the two cameras are related by a rotation, $R \in SO(3)$, we have $\mathbf{X}_1 = R\mathbf{X}_2$.

Now let us consider a second point P_2 which has coordinates $\mathbf{Y}_1$ and $\mathbf{Y}_2$ with respect to the two reference frames. The key idea for eliminating the rotation is to notice that rotations preserve angles between vectors:

$$\theta_{\mathbf{X}_1\mathbf{Y}_1} = \theta_{\mathbf{X}_2\mathbf{Y}_2}, \tag{7}$$

where $\theta_{\mathbf{X}_1\mathbf{Y}_1}$ measures the angle between $\mathbf{X}_1$ and $\mathbf{Y}_1$. Using equations (5) and (6), we can express angles using distorted projections as

$$\begin{aligned}
\theta_{\mathbf{X}_1\mathbf{Y}_1} &= \frac{\langle \mathbf{X}_1, \mathbf{Y}_1 \rangle}{\|\mathbf{X}_1\| \cdot \|\mathbf{Y}_1\|} = \\
&= \frac{\left\langle \begin{bmatrix} K_1^{-1} \circ \mathbf{x}_1 \\ 1 + \kappa_1 \|K_1^{-1} \circ \mathbf{x}_1\|^2 \end{bmatrix}, \begin{bmatrix} K_1^{-1} \circ \mathbf{y}_1 \\ 1 + \kappa_1 \|K_1^{-1} \circ \mathbf{y}_1\|^2 \end{bmatrix} \right\rangle}{\left\| \begin{bmatrix} K_1^{-1} \circ \mathbf{x}_1 \\ 1 + \kappa_1 \|K_1^{-1} \circ \mathbf{x}_1\|^2 \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} K_1^{-1} \circ \mathbf{y}_1 \\ 1 + \kappa_1 \|K_1^{-1} \circ \mathbf{y}_1\|^2 \end{bmatrix} \right\|} = \\
\theta_{\mathbf{X}_2\mathbf{Y}_2} &= \frac{\langle \mathbf{X}_2, \mathbf{Y}_2 \rangle}{\|\mathbf{X}_2\| \cdot \|\mathbf{Y}_2\|} = \\
&= \frac{\left\langle \begin{bmatrix} K_2^{-1} \circ \mathbf{x}_2 \\ 1 + \kappa_2 \|K_2^{-1} \circ \mathbf{x}_2\|^2 \end{bmatrix}, \begin{bmatrix} K_2^{-1} \circ \mathbf{y}_2 \\ 1 + \kappa_2 \|K_2^{-1} \circ \mathbf{y}_2\|^2 \end{bmatrix} \right\rangle}{\left\| \begin{bmatrix} K_2^{-1} \circ \mathbf{x}_2 \\ 1 + \kappa_2 \|K_2^{-1} \circ \mathbf{x}_2\|^2 \end{bmatrix} \right\| \cdot \left\| \begin{bmatrix} K_2^{-1} \circ \mathbf{y}_2 \\ 1 + \kappa_2 \|K_2^{-1} \circ \mathbf{y}_2\|^2 \end{bmatrix} \right\|}
\end{aligned} \tag{8}$$

$$
\begin{cases}
& \big(\langle \bar{\mathbf{x}}_1, \bar{\mathbf{y}}_1\rangle + F_1(1+\lambda_1\|\bar{\mathbf{x}}_1\|^2)(1+\lambda_1\|\bar{\mathbf{y}}_1\|^2)\big)^2\big(\|\bar{\mathbf{x}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{x}}_2\|^2)^2\big)\big(\|\bar{\mathbf{y}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{y}}_2\|^2)^2\big) \\
= & \big(\langle \bar{\mathbf{x}}_2, \bar{\mathbf{y}}_2\rangle + F_2(1+\lambda_2\|\bar{\mathbf{x}}_2\|^2)(1+\lambda_2\|\bar{\mathbf{y}}_2\|^2)\big)^2\big(\|\bar{\mathbf{x}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{x}}_1\|^2)^2\big)\big(\|\bar{\mathbf{y}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{y}}_1\|^2)^2\big) \\
& \big(\langle \bar{\mathbf{y}}_1, \bar{\mathbf{z}}_1\rangle + F_1(1+\lambda_1\|\bar{\mathbf{y}}_1\|^2)(1+\lambda_1\|\bar{\mathbf{z}}_1\|^2)\big)^2\big(\|\bar{\mathbf{y}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{y}}_2\|^2)^2\big)\big(\|\bar{\mathbf{z}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{z}}_2\|^2)^2\big) \\
= & \big(\langle \bar{\mathbf{y}}_2, \bar{\mathbf{z}}_2\rangle + F_2(1+\lambda_2\|\bar{\mathbf{y}}_2\|^2)(1+\lambda_2\|\bar{\mathbf{z}}_2\|^2)\big)^2\big(\|\bar{\mathbf{y}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{y}}_1\|^2)^2\big)\big(\|\bar{\mathbf{z}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{z}}_1\|^2)^2\big) \\
& \big(\langle \bar{\mathbf{z}}_1, \bar{\mathbf{x}}_1\rangle + F_1(1+\lambda_1\|\bar{\mathbf{z}}_1\|^2)(1+\lambda_1\|\bar{\mathbf{x}}_1\|^2)\big)^2\big(\|\bar{\mathbf{z}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{z}}_2\|^2)^2\big)\big(\|\bar{\mathbf{x}}_2\|^2 + F_2(1+\lambda_2\|\bar{\mathbf{x}}_2\|^2)^2\big) \\
= & \big(\langle \bar{\mathbf{z}}_2, \bar{\mathbf{x}}_2\rangle + F_2(1+\lambda_2\|\bar{\mathbf{z}}_2\|^2)(1+\lambda_2\|\bar{\mathbf{x}}_2\|^2)\big)^2\big(\|\bar{\mathbf{z}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{z}}_1\|^2)^2\big)\big(\|\bar{\mathbf{x}}_1\|^2 + F_1(1+\lambda_1\|\bar{\mathbf{x}}_1\|^2)^2\big)
\end{cases}
\tag{9}
$$

where $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{R}^2$ are the radially distorted projections of P_2 in the two respective cameras.

To further simplify the problem, we make the following assumptions: the two camera centers are known and coincide with the respective image centers; there is no pixel skew and the pixel aspect ratio is 1, i.e. pixels are square; the focal lengths for the two cameras may vary but the radial distortion coefficients are the same. While the assumption of known camera centers and square pixels are typical for image stitching algorithms (see [24] for a discussion), one may think the assumption of varying focal lengths contradicts that of constant distortion coefficients. Indeed, it is true that the distortion coefficient changes when a lens zooms. However, when a lens does not zoom or the zoom amount is small, the distortion coefficient approximately stays constant, which is the most common scenario for panoramic stitching: people do not zoom when they shoot panoramas. Note that we cannot assume the focal lengths to stay the same because they vary when the camera focuses on objects with different depths even under the same zoom. Under these assumptions, $K_i^{-1} \circ \mathbf{x}_i$ reduces to $\frac{1}{f_i}\bar{\mathbf{x}}_i$ where $\bar{\mathbf{x}}_i \doteq \mathbf{x}_i - \mathbf{c}_i$. We can rewrite equation (8) as

$$
\frac{\frac{1}{f_1^2}\langle \bar{\mathbf{x}}_1, \bar{\mathbf{y}}_1\rangle + (1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2)(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{y}}_1\|^2)}{\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{x}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2\big)^2}\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{y}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{y}}_1\|^2\big)^2}} = \frac{\frac{1}{f_2^2}\langle \bar{\mathbf{x}}_2, \bar{\mathbf{y}}_2\rangle + (1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{x}}_2\|^2)(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{y}}_2\|^2)}{\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{x}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{x}}_2\|^2\big)^2}\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{y}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{y}}_2\|^2\big)^2}} \tag{10}
$$

where $\kappa = \kappa_1 = \kappa_2$. An additional point P_3 yields two more equations:

$$
\frac{\frac{1}{f_1^2}\langle \bar{\mathbf{y}}_1, \bar{\mathbf{z}}_1\rangle + (1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{y}}_1\|^2)(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{z}}_1\|^2)}{\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{y}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{y}}_1\|^2\big)^2}\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{z}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{z}}_1\|^2\big)^2}} = \frac{\frac{1}{f_2^2}\langle \bar{\mathbf{y}}_2, \bar{\mathbf{z}}_2\rangle + (1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{y}}_2\|^2)(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{z}}_2\|^2)}{\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{y}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{y}}_2\|^2\big)^2}\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{z}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{z}}_2\|^2\big)^2}} \tag{11}
$$

$$
\frac{\frac{1}{f_1^2}\langle \bar{\mathbf{z}}_1, \bar{\mathbf{x}}_1\rangle + (1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{z}}_1\|^2)(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2)}{\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{z}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{z}}_1\|^2\big)^2}\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{x}}_1\|^2 + \big(1+\frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2\big)^2}} = \frac{\frac{1}{f_2^2}\langle \bar{\mathbf{z}}_2, \bar{\mathbf{x}}_2\rangle + (1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{z}}_2\|^2)(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{x}}_2\|^2)}{\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{z}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{z}}_2\|^2\big)^2}\sqrt{\frac{1}{f_2^2}\|\bar{\mathbf{x}}_2\|^2 + \big(1+\frac{\kappa}{f_2^2}\|\bar{\mathbf{x}}_2\|^2\big)^2}} \tag{12}
$$

where $\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2 \in \mathbb{R}^2$ are the radially distorted projections of P_3. There are three unknowns (f_1, f_2 and κ) in equations (10- 12). These three equations are generally independent and sufficient to determine the unknowns. On the other hand, we would not be able to derive three equations from less than three point correspondences. Therefore, three is the minimal number of point correspondences.

3. Minimal solvers

In this section, we will discuss two different methods for solving equations (10-12). In particular, we will first present a potential solver based on a Gröbner basis and discuss its pros and cons. We will then present our special solver based on nonlinear optimization.

3.1. Gröbner basis

We can rewrite equations (10-12) into a set of polynomials equations by squaring both sides and re-arranging the terms. The result is equations (9) (top of this page), where $F_i \doteq f_i^2$ and $\lambda_i \doteq \kappa/f_i^2$, $i = 1, 2$. F_i and λ_i are related by

$$
\lambda_1 F_1 = \lambda_2 F_2. \tag{13}
$$

Using Macaulay 2 [9], we verified that equations (9) and (13) are indeed sufficient to determine all four unknowns, F_1, F_2, λ_1 and λ_2. It is possible to further constrain the problem by noticing the following relationship:

$$
\frac{[\mathbf{X}_1, \mathbf{Y}_1, \mathbf{Z}_1]}{\|\mathbf{X}_1\| \cdot \|\mathbf{Y}_1\| \cdot \|\mathbf{Z}_1\|} = \frac{[\mathbf{X}_2, \mathbf{Y}_2, \mathbf{Z}_2]}{\|\mathbf{X}_2\| \cdot \|\mathbf{Y}_2\| \cdot \|\mathbf{Z}_2\|}, \tag{14}
$$

where $[\mathbf{X}, \mathbf{Y}, \mathbf{Z}]$ denotes the *scalar triple product*: $\langle \mathbf{X}, \mathbf{Y} \times \mathbf{Z}\rangle$, for any vectors $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^3$. This triple-product based constraint is *not* algebraically independent but can be used to remove extraneous solutions nevertheless. To be more precise, there are 96 solutions, both real and complex, to equations (9) and (13), out of which 54 satisfy (14).

It is possible to construct a Gröbner basis from equations (9) and (13) and solve for the unknowns. We refer the reader to [11, 21] for details on how to solve algebraic equations using Gröbner basis. In this work we decided not to pursue this route because we found that Gröbner basis-based methods suffer from considerable numerical instabilities for problems of high degree when they were implemented numerically. This was also observed recently by

other computer vision researchers. For instance, [23] used high-precision floating-point numbers and [12] used exact rational arithmetic for this very reason. The price paid was speed which we do not want to sacrifice. [5] proposed an interesting method to improve numerical accuracy for Gröbner basis-based solvers. But its effectiveness is yet to be demonstrated for problems of high degree.

3.2. Optimization

In addition to suffering from numerical instability issues, Gröbner basis-based methods make no use of prior knowledge in a given problem. For instance, in the absence of any prior knowledge, we still know that the two focal lengths are real and positive and the distortion coefficient is a small real number around 0. In practice, we can often obtain known ranges for the focal lengths and distortion coefficients from EXIF data in the images. The inspiration for our new solver comes from the belief that we should have a more efficient solver if we can make use of the prior knowledge.

We cast the root-seeking problem into an optimization framework. In particular, we propose to minimize the following objective function:

$$\left(\theta_{\mathbf{X}_1\mathbf{Y}_1} - \theta_{\mathbf{X}_2\mathbf{Y}_2}\right)^2 + \left(\theta_{\mathbf{Y}_1\mathbf{Z}_1} - \theta_{\mathbf{Y}_2\mathbf{Z}_2}\right)^2 + \left(\theta_{\mathbf{Z}_1\mathbf{X}_1} - \theta_{\mathbf{Z}_2\mathbf{X}_2}\right)^2. \tag{15}$$

It is obvious that the roots to equations (9) and (13) are the minima. Note that cost (15) is not an arbitrary algebraic quantity, but is geometrically meaningful. In fact, it measures the cumulative difference between corresponding angles. Since cost (15) is in a form of nonlinear least squares, we use Levenberg-Marquardt [17] with analytical derivatives to perform the optimization. The initial values for the unknowns are obtained as follows: We use our prior knowledge for κ as the initial value (κ^0) since the distortion coefficient usually does not vary significantly. In the absence of prior knowledge, we just use $\kappa^0 = 0$. We then solve equations (9), assuming κ is known, to obtain initial values for (f_1, f_2). Given $\kappa = \kappa^0$, equations (9) reduce to

$$\begin{cases} & \left(\langle\tilde{\mathbf{x}}_1, \tilde{\mathbf{y}}_1\rangle + \tilde{F}_1\right)^2 \left(\|\tilde{\mathbf{x}}_2\|^2 + \tilde{F}_2\right)\left(\|\tilde{\mathbf{y}}_2\|^2 + \tilde{F}_2\right) \\ = & \left(\langle\tilde{\mathbf{x}}_2, \tilde{\mathbf{y}}_2\rangle + \tilde{F}_2\right)^2 \left(\|\tilde{\mathbf{x}}_1\|^2 + \tilde{F}_1\right)\left(\|\tilde{\mathbf{y}}_1\|^2 + \tilde{F}_1\right), \\ & \left(\langle\tilde{\mathbf{y}}_1, \tilde{\mathbf{z}}_1\rangle + \tilde{F}_1\right)^2 \left(\|\tilde{\mathbf{y}}_2\|^2 + \tilde{F}_2\right)\left(\|\tilde{\mathbf{z}}_2\|^2 + \tilde{F}_2\right) \\ = & \left(\langle\tilde{\mathbf{y}}_2, \tilde{\mathbf{z}}_2\rangle + \tilde{F}_2\right)^2 \left(\|\tilde{\mathbf{y}}_1\|^2 + \tilde{F}_1\right)\left(\|\tilde{\mathbf{z}}_1\|^2 + \tilde{F}_1\right), \\ & \left(\langle\tilde{\mathbf{z}}_1, \tilde{\mathbf{x}}_1\rangle + \tilde{F}_1\right)^2 \left(\|\tilde{\mathbf{z}}_2\|^2 + \tilde{F}_2\right)\left(\|\tilde{\mathbf{x}}_2\|^2 + \tilde{F}_2\right) \\ = & \left(\langle\tilde{\mathbf{z}}_2, \tilde{\mathbf{x}}_2\rangle + \tilde{F}_2\right)^2 \left(\|\tilde{\mathbf{z}}_1\|^2 + \tilde{F}_1\right)\left(\|\tilde{\mathbf{x}}_1\|^2 + \tilde{F}_1\right), \end{cases} \tag{16}$$

where

$$\tilde{\mathbf{x}}_i = \frac{\bar{\mathbf{x}}_i/f_i^p}{1 + \frac{\kappa^0}{f_i^p}\|\bar{\mathbf{x}}_i\|^2}, \tilde{\mathbf{y}}_i = \frac{\bar{\mathbf{y}}_i/f_i^p}{1 + \kappa^0\|\bar{\mathbf{y}}_i/f_i^p\|^2}, \tilde{\mathbf{z}}_i = \frac{\bar{\mathbf{z}}_i/f_i^p}{1 + \kappa^0\|\bar{\mathbf{z}}_i/f_i^p\|^2}, \tag{17}$$

and f_1^p and f_2^p are given by the prior knowledge (we use $f_1^p = f_2^p = 0$ in the absence of prior knowledge). $\tilde{F}_1$ and $\tilde{F}_2$ can be solved using the three-point algorithm in [3]. Finally, the initial values for f_1 and f_2 are given by

$$f_i^o = f_i^p\sqrt{\tilde{F}_i}, \quad i = 1, 2. \tag{18}$$

We note that the Levenberg-Marquardt part is a fairly small problem (three unknowns and three squared terms) and can be implemented very efficiently.

4. Solving for the rotation

Once we know the focal lengths and the distortion coefficient, it is straightforward to compute the rotation. Using equation (5), we can compute $\frac{\mathbf{X}_1}{\|\mathbf{X}_1\|}$ as follows:

$$\frac{\mathbf{X}_1}{\|\mathbf{X}_1\|} = \frac{1}{\sqrt{\frac{1}{f_1^2}\|\bar{\mathbf{x}}_1\|^2 + (1 + \frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2)^2}} \begin{bmatrix} \bar{\mathbf{x}}_1/f_1 \\ 1 + \frac{\kappa}{f_1^2}\|\bar{\mathbf{x}}_1\|^2 \end{bmatrix}. \tag{19}$$

Similarly, we can compute $\frac{\mathbf{X}_2}{\|\mathbf{X}_2\|}$, $\frac{\mathbf{Y}_1}{\|\mathbf{Y}_1\|}$, $\frac{\mathbf{Y}_2}{\|\mathbf{Y}_2\|}$, $\frac{\mathbf{Z}_1}{\|\mathbf{Z}_1\|}$, and $\frac{\mathbf{Z}_2}{\|\mathbf{Z}_2\|}$. We can then invoke the solver of [3] to obtain the rotation.

5. Robust solutions and bundle adjustment

The core solver presented in Sections 3 and 4 are not intended to be used directly on point correspondences because it can neither make use of more than three point correspondences nor deal with outliers or noise in point correspondences. Instead, the solver is best used in a hypothesis testing framework, such as RANSAC [6], to handle outliers and noise. We refer the reader to [4, 10] for more details on how to use RANSAC for panoramic image stitching.

It is often necessary to further refine the parameters obtained by the robust solutions for better results. This step is known as bundle adjustment [25]. In general, there are two bundle adjustments involved: pairwise and multi-image bundle adjustments. We will briefly discuss the multi-image bundle adjustment here, since the pairwise one can be considered as a special case. Our multi-image bundle adjustment optimizes the following geometric cost function:

$$\sum_{i=1}^{M}\sum_{j=1}^{N} w_{ij}\|\hat{\mathbf{x}}_{ij}(\theta_j, \phi_j; R_i, f_i, \mathbf{k}_i|\mathbf{c}_i) - \mathbf{x}_{ij}\|^2, \tag{20}$$

where M is the number of images and N is the number of chains of consistent point correspondences. Consistent means that all the points are projections of the same point in space. This point is denoted as $\mathbf{X}_j$ which is parameterized by spherical coordinates (θ_j, ϕ_j) with respect to a chosen reference frame, i.e. $\mathbf{X}_j = [\cos(\theta_j)\cos(\phi_j), \cos(\theta_j)\sin(\phi_j), \sin(\theta_j)]^T$. $\mathbf{x}_{ij}$ is the measured projection of $\mathbf{X}_j$ in the i-th image and w_{ij} is the associated weight. $w_{ij} = 0$ if $\mathbf{X}_j$ does not appear in the i-th image; otherwise, it is a positive number. R_i, f_i, $\mathbf{k}_i$, and $\mathbf{c}_i$ are the rotation, focal length, radial distortion coefficient and image center of the i-th image respectively. $\hat{\mathbf{x}}_{ij}$ is the measurement equation given by

$$\hat{\mathbf{x}}_{ij}(\theta_j, \phi_j; R_i, f_i, \mathbf{k}_i|\mathbf{c}_i) = f_i\hat{\mathbf{k}}_i(\pi(R_i\mathbf{X}); \mathbf{k}_i) + \mathbf{c}_i \tag{21}$$

where $\hat{\mathbf{k}}_i(\mathbf{q}; \mathbf{k}_i) = \mathbf{q}(1 + \mathbf{k}_{i1}\|\mathbf{q}\|^2 + \mathbf{k}_{i2}\|\mathbf{q}\|^4)$ for any $\mathbf{q} \in \mathbb{R}^2$ where $\mathbf{k}_i = [\mathbf{k}_{i1}, \mathbf{k}_{i2}]^T$. Note that we have switched the

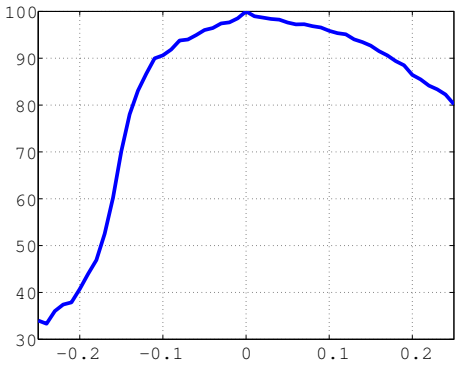

Figure 1. This plot shows the convergence rate of the proposed solver against the distortion coefficient on random geometry. Our solver is able to converge correctly over 80% time for distortion coefficients ranging from -0.14 to 0.25. We refer the reader to Section 6 for details of the plot.

distortion model from [7] to [27] because the latter has two parameters and better represents the distortion effects. It is easy to go from the model of [7] to that of [27] by noticing the following relationship in equation (2):

$$\mathbf{p} = \mathbf{q}(1+\kappa\|\mathbf{p}\|^2) = \mathbf{q}(1+\kappa\|\mathbf{q}\|^2+2\kappa^2\|\mathbf{q}\|^4+O(\|\mathbf{q}\|^6)). \quad (22)$$

The unknowns in equation (20) are θ_j, ϕ_j, $j = 1, \ldots, N$, and $R_i, f_i, \mathbf{k}_i$, $i = 1, \ldots, M$. Observing that cost (20) is in a nonlinear least squares form, we optimize it using Levenberg-Marquardt [17], which can be implemented efficiently using sparse techniques [25].

6. Experiments

The first experiment we did was to test the convergence rate of our optimization-based two-view solver presented in Section 3.2. To that end, we used synthetic data where we had ground truth. For a given distortion coefficient, we generated three noise-free point correspondences from random geometry according to equation (2). In particular, we randomly generated three points in space whose projections in one image were uniformly distributed in $[-0.5, 0.5] \times [-0.5, 0.5]$ and whose depths were uniformly distributed in $[1.3, 1.7]$; The axis of the rotation between two images was randomly sampled within a 30^o cone around the y-axis and the magnitude of the rotation was randomly sampled in $[-\frac{\pi}{6}, \frac{\pi}{6}]$; The two focal lengths are randomly sampled in $[0.5, 1.5]$ which corresponds to a range from 17 mm to 50 mm for 35-mm film cameras. These settings are typical for panoramas. We fed the point correspondences into our solver and recorded if the algorithm found the correct solution. For each distortion coefficient, we repeated the test $10,000$ times and the whole process was repeated for 51 values of the distortion coefficient ranging uniformly from -0.25 to 0.25. The results are presented in Figure 1. As one can see, our solver is able to get over 80% time correct for distortion coefficients ranging from -0.14 to 0.25. The

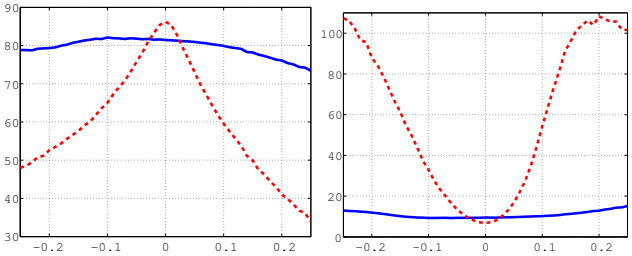

Figure 2. The left plot shows the percentage of correct correspondences that our proposed solver (blue solid line) and the three-point algorithm in [3] (red dashed line) can retain for distortion ranging from -0.25 to 0.25. Our solver averages at above 75% while that of [3] is considerably lower for most distortion coefficients. The right plot shows the number of trials needed to obtain a RANSAC confidence of 0.995. Our solver (blue solid line) needs only 15 trials on average while [3] (red dashed line) needs many more trials on average. We refer the reader to Section 6 for details on how these two plots are generated.

performance degrades for distortion coefficients lower than -0.14. It is interesting to notice that our solver works better for pincushion distortion (positive κ) than barrel distortion (negative κ). We remark that it is not necessary to have a convergence rate of 100% because the solver is intended to be used in a hypothesis testing framework.

The second experiment we did was to check if our solver was able to retain more correct correspondences than an algorithm that did not estimate lens distortion. The algorithm we compared with is the three-point one in [3] which is the state-of-the-art for varying focal lengths. Again, we used synthetic data for the sake of ground truth. Both our solver and that of [3] were wrapped in a RANSAC framework. For each distortion coefficient, we generated 200 *noisy* point correspondences from random geometry which is the same as in the first test. The noise we added to point correspondences was zero-mean Gaussian with standard deviation set to 0.1% of the image width. The maximum number of trials for RANSAC was set to 500 and the desired confidence was set to 0.995. For each distortion coefficient, we repeated the test $10,000$ times. The results are presented in Figure 2 where the blue solid lines are our solver and the red dashed lines are that of [3]. Our solver outperforms that of [3]. In particular, our solver is able to retain over 75% correct point correspondences in less than 15 trials on average. One important implication of these two plots is that although our solver is more expensive than the three-point algorithm in [3], the entire RANSAC process with our solver on images with lens distortion may be significantly faster because of fewer trials and a higher inlier ratio.

Figure 3 shows a comparison on real images with and without lens distortion estimation. The top row is two input images. We used SIFT features [15]. The middle row is the result obtained without lens distortion estimation. The composition mode is cylindrical. We alpha-blended the two

Figure 3. Comparison on real images with and without lens distortion estimation: The top row was two input images. The middle row was the result obtained without lens distortion estimation. We can see visible mis-alignments in the crosswalk region. The bottom row was the result obtained with lens distortion estimation. The improvement in alignment quality is clear.

images with equal weights in the overlapping regions. One can see visible mis-alignments in the crosswalk region. The bottom row is the result obtained with lens distortion estimation. Again, the two images were alpha-blended with equal weights in the overlapping regions. The improvement in alignment quality is clear.

Figure 4 shows the importance of multi-image bundle adjustment. The first composite is created with pairwise bundle adjustment but without multi-image bundle adjustment while the second is created with both. Lens distortion is estimated in both cases. Images are simply stacked one onto another without alpha-blending. One can observe that the alignment is better in the second composite that uses multi-image bundle adjustment.

Finally, in Figure 5 we show several examples of our entire pipeline on real images. Again, features were extracted using SIFT [15]. The top row is a stitch made from 6 images. The middle row is stitched from 35 images. The bottom row is a full 360^o panorama stitched from 23 images. Blending is done using the algorithm in [1].

7. Conclusions

This paper has extended the three-point solver in [3] to include a correction for lens distortion. The main contribution is a minimal solver for simultaneous estimation of a single radial distortion coefficient, a rotation and two focal lengths. Our solver uses only three point correspondences and is well suited for use in a hypothesis testing framework. Although it is possible to use a Gröbner basis to solve the resulting polynomial equations, we have chosen to solve the problem in an optimization framework. The advantages are being able to make use of prior knowledge and being free from numerical instability issues. The cost we optimize is a geometric one instead of an algebraic one. Although our solver is more expensive than the three-point algorithm in [3], it is potentially much faster when the entire RANSAC process is considered for images with lens distortion because it is able to get more correct correspondences in fewer RANSAC trials. In future work, we plan to improve the convergence rate of the proposed solver for large barrel distortion and explore the case where the distortion coefficients in two images are different.

Acknowledgments

I wish to thank Gavin Miller for his valuable comments.

References

[1] A. Agarwala. Efficient gradient-domain compositing using quadtrees. In *Proc. of ACM SIGGRAPH*, 2007.

[2] J. P. Barreto and K. Daniilidis. Fundamental matrix for cameras with radial distortion. In *Proc. of Intl. Conf. on Computer Vision*, 2005.

[3] M. Brown, R. I. Hartley, and D. Nister. Minimal solutions for panoramic stitching. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[4] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. of Computer Vision*, 74(1):59–73, 2006.

[5] M. Byröd, K. Josephson, and K. Åström. Improving numerical accuracy of gröbner basis polynomial equation solvers. In *Proc. of Intl. Conf. on Computer Vision*, 2007.

[6] M A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

[7] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

Figure 4. Importance of multi-image bundle adjustment: The first composite is created with pairwise bundle adjustment but without multi-image bundle adjustment while the second is created with both types of bundle adjustments. One can observe that the alignment is better in the second composite.

[8] C. Geyer and H. Stewénius. A nine-point algorithm for estimating para-catadioptric fundamental matrices. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[9] D. R. Grayson and M. E. Stillman. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2.

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[11] Z. Kukelova and T. Pajdla. A minimal solution to the autocalibration of radial distortion. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[12] Z. Kukelova and T. Pajdla. Two minimal problems for cameras with radial distortion. In *Proc. of Seventh Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, 2007.

[13] H. Li and R. Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. In *Proc. of Workshop on Omnidirectional Vision, Camera Networks and Non-classical cameras*, 2005.

[14] H. Li and R. Hartley. Five-point motion estimation made easy. In *Proc. of Intl. Con. on Pattern Recognition*, volume 1, pages 630–633, 2006.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60(2):91–110, November 2004.

[16] B. Micusik and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 485–490, June 2003.

[17] J. J. Moré. *The Levenberg-Marquardt algorithm: Implementation and theory*. Lecture Notes in Mathematics. Springer

Figure 5. Real-image examples of multi-image stitching with lens distortion: the top row is a stitch of the Golden Gate bridge from 6 images; The middle row is a stitch of the Copacababa beach from 35 images; The bottom row is a full 360^o panorama stitched from 23 images. Blending is done using [1].

Berlin / Heidelberg, 1978.

[18] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(6):756–777, June 2004.

[19] H. S. Sawhney and R. Kumar. True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(3):235–243, March 1999.

[20] R. M. Steele and C. Jaynes. Overconstrained linear estimation of radial distortion and multi-view geometry. In *Proc. of Europ. Conf. on Computer Vision*, pages 253–264, 2006.

[21] H. Stewénius. *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Lund University, April 2005.

[22] H. Stewénius, F. Kahl, D. Nister, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 789–794, June 2005.

[23] H. Stewénius, F. Schaffalitzky, and D. Nister. How hard is 3-view triangulation really? In *Proc. of Intl. Conf. on Computer Vision*, 2005.

[24] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1), 2006.

[25] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proc. Intl. Workshop on Vision Algorithms: Theory and Practice*, pages 298–372, 1999.

[26] R. Y. Tsai. Versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Trans. of Robotics and Automation*, 3(4):323–344, August 1987.

[27] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, November 2000.