A FAST AND ACCURATE RE-CALIBRATION TECHNIQUE FOR MISALIGNED STEREO CAMERAS

Mihail Georgiev¹, Atanas Gotchev¹, Miska Hannuksela²

Tampere University of Technology, Tampere, Finland¹; Nokia Research Center, Tampere, Finland²

ABSTRACT

In this paper, we propose a practical approach for robust rectification of stereo camera setups without use of calibration pattern. Our solution simplifies the process to a non-general case of rectification to avoid explicit use of Fundamental Matrix estimation. The solution shows better or comparable robustness than some of recent solutions, but for much lower computational cost and code complexity.

Index Terms— Rectification, Camera calibration, stereo setup, Fundamental matrix, feature matching

1. INTRODUCTION

Camera calibration and rectification are required preprocessing steps in stereo-camera capturing applications. Such applications included depth-from-stereo matching, 3D scene reconstruction, and virtual view synthesis. Stereo calibration refers to the way of finding relative orientations of cameras in a stereo-camera setup, while rectification refers to the way of finding projective transformations called also homographies \( H_L, H_R \) for both cameras which transform captured stereo images of the scene to “row-to-row” image alignment (Fig. 1). Such row data alignment will simplify the subsequent search for stereo-matching correspondences among a captured stereo pair of images, which is then done in horizontal direction only.

Methods for fast and robust camera calibration and rectification have been an active area of research for some time. Most of the solutions are concentrated mainly on estimation of epipolar relations of camera setup described by so-called “Fundamental Matrix” \( F_{[3x3]} \). The matrix can be estimated unambiguously for a certain number of correctly found corresponding features among stereo pairs (e.g. 8-point algorithm [1]). Having \( F \) well-estimated, one can obtain all parameters needed to rectify a stereo pair. The quality and unambiguity of the matrix estimation strongly depend on the location precision of the used correspondences, their number, and the amount of outliers.

A general robust solution for estimation of \( F \) requires some of the following steps: feature normalization, extensive search of outlier-free set of correspondences by Random Consensus Search-type approaches (e.g. RanSAC, MapSAC), non-linear optimization search (e.g. Levenberg-Marquardt iteration method), Singular Value Decomposition Analysis (SVD), etc.

The theory of general rectification approaches based on

\( F \) is described in details in [2]. Same source, analyses various aspects how to improve robustness of estimation (e.g. normalized 7-point algorithm). Currently, many other methods provide enhancing approaches to solve the problem of robustness of rectification [3-10]. Authors in [6], enforce the orthogonality of \( H_{LR} \), others propose a stratified decomposition of \( F \) to rigid projective transformation [5, 7]. Authors in [9] provide further enhancement by constraining the element relations of \( F \) to be estimated with even less number of correspondences. An example of a direct non-linear optimization search of \( H_{LR} \) is given in [8].

We propose a practical approach that works for calibrating roughly-aligned cameras in a stereo setup by avoiding the explicit use of \( F \) for estimating \( H_{LR} \). In our solution, the rectification is applied as compensation post-process of camera misalignments that we assume to vary for some known degrees of freedom (or misalignment) – DoF. We apply a standard linear optimization search procedure for finding the levels of these misalignments that minimize a cost function for a collection of rectification quality metrics. This effectively results in a very robust solution compared to others for a very low-complexity implementation cost.

2. ALGORITHM FRAMEWORK

Our solution works for a case of rectification scenario where a given stereo-setup is made of horizontally mounted cameras of similar models and capturing settings. Another possible case of stereo setup is the one consisting of consecutive capturing of same camera device, which is roughly translated by hand from left-to-right direction between shots.

In our approach we assume that a stereo pair of images is captured by an “arbitrary chosen stereo setup” of left \((L)\) and right camera \((R)\) that is misaligned to some known expected extent and levels. The possible misalignments and levels are defined by set of DoF. Then, the setup is rectified by compensation post-process of misalignments and the quality of result is measured. Such feedback is used for a repeated linear optimization search function that selects the best “arbitrary setup” in terms of rectification quality. The quality
is measured by a collection of rectification cost metrics evaluated on a set of matched feature correspondences. Finally, we calculate $H_{L,R}$ for the compensation of best found misaligned stereo setup and rectify image pairs. The process-flow stages of general application framework are shown in Fig. 2.

### 2.1. Model of Misaligned Stereo Setup

In our application, we describe camera devices by so called “pinhole model” as defined in [2]. A “pinhole” camera is described by its intrinsic parameters: optical center – $c$, focal length – $f_{(x,y)}$, Field-of-view angle (FoV) – $v_{(x,y)}$, principle point – $p_{(x,y)}$, sensor size(width, height) in pixels – $s_{(x,y)}$. A visual depiction of those parameters is given in Fig.3. The parameters are calculated and embedded into Camera Matrix – $K$ as follows:

$$
K_{[3x3]} = \begin{bmatrix}
f_{(x,y)} & 0 & p_{(x,y)} \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}, f_{(x,y)} = s_{(x,y)} \tan\left(\frac{v_{(x,y)}}{2}\right) \tag{1}
$$

In our implementation we assume cameras with similar capturing parameters, which for example can be cameras of same factory model. The similarity between both camera models according to $K_{L,R}$ case can be written as follows:

$$
K_{L} \approx K_{R} \rightarrow f_{L} = \alpha f_{R}, s_{L} = s_{R}, p_{L} = \beta(p_{R}), (\alpha, \beta) \approx 1 \tag{2}
$$

The relative position and pose of cameras in a misaligned stereo setup are defined by Baseline Vector – $B$, Rotation Matrix – $R$ and Projection Distortion Matrix – $A$ (Fig. 4) with a reference to left camera. The first one represents the “horizontal”, “vertical” and “forward/backward” displacement of camera optical centers - $c_{L}$, $c_{R}$ in terms of world coordinate axes – $X$, $Y$, $Z$ and second one describes pose shifts by axial rotation angles - $\phi$, $\theta$, $\psi$. The matrix $A$ defines possible “skew”-ness or “aspect/zoom” distortions in camera sensors, due to some mechanical damage or factory defect. Such imperfections will introduce projective distortions for the captured data, but high influence is rather rare for current technology. We confine our projective distortion model to parameters of an affine transformation – $d_{1}$, $d_{2}$, $d_{3}$. The matrix notations of $B$, $R$ [9], and $A$ are defined as:

$$
R_{[3x3]} = \begin{bmatrix}
1 & -\theta & \phi \\
\theta & 1 & -\psi \\
\phi & \psi & 1
\end{bmatrix}, B_{[3x3]} = \begin{bmatrix}X, Y, Z\end{bmatrix}, A_{[3x3]} = \begin{bmatrix}d_{1} & d_{2} & d_{3} \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

An example of misaligned camera setup by $B$, $R$, and $A$ is depicted in Fig. 4.

#### 2.2. Camera Rectification

We modify an approach given in [7] to calculate $H_{L,R}$ according to given misaligned model. In our case, we extend it to include misalignments of projective distortions defined by $A$. The algorithm is explained briefly as follows:

0. Define $R$, $B$, and $A$ (Eq. 3)
1. Define Projection Matrices – $P_{L,R}^{[3x4]}$: 

$$
P_{L} = (K_{L} \begin{bmatrix}1 & 0 & 0 \end{bmatrix}) \begin{bmatrix}1 & 0 & 0 \end{bmatrix}^{T}, P_{R} = (K_{R} \begin{bmatrix}1 & 0 & 0 \end{bmatrix}) \begin{bmatrix}1 & 0 & 0 \end{bmatrix}^{T} \tag{4}
$$

where $I$, $0$ are respectively Identity and Null Matrix

2. Calculate camera optical centers – $c_{L,R}$:

$$
c_{L} = -(K_{L}^{-1}(P_{L}))^{-1} I, c_{R} = -(K_{R}^{-1}(P_{R}))^{-1} R^{T} \tag{5}
$$

3. Define Rotation Matrix that rectifies stereo pairs – $R'$:

$$
w_{1} = c_{L} - c_{R}, w_{2} = R_{(x,y)}w_{1}, w_{3} = w_{1} \times w_{2}, R' = [w_{1}', w_{2}', w_{3}'] \tag{6}
$$

4. Estimate projective matrices $P_{L,R}'$ of rectified setup:

$$
P_{L}' = (K_{L} \begin{bmatrix}1 & 0 & 0 \end{bmatrix})(R'c_{L}), P_{R}' = (K_{R} \begin{bmatrix}1 & 0 & 0 \end{bmatrix})(R'c_{R}) \tag{7}
$$

6. Find rectification homographies – $H_{L,R}$:

$$
H_{L} = P_{L}'(P_{L})^{-1}, H_{R} = P_{R}'(P_{R})^{-1} A \tag{8}
$$

#### 2.3. Feature Point Estimation & Matching

We estimate feature correspondences by approach based on “Scale Invariant Feature Transform” (SIFT) [10]. The output of feature estimation process is a set of image points – $m_{L,R}$ in image coordinates and correspondences are indexed element-wisely. We have implemented a pre-filtered stage for detection and masking of corresponding outliers that provides masking quality good enough for robust performance of optimization search step (Fig. 5).

The filtering steps are described briefly as follows:

1. **Boundary filter** – excludes feature points that are close to image edges. Removes possible outliers, due to non-mutual image data among stereo pairs
2. **Local region filter** – discards points that are situated very close to each other in both of image pairs
3. **Mirror filter** – discards the differences between matched feature pairs when search left-to-right and right-to-left
4. **One-to-many filter** – removes matches that have same indices in matched pairs or similar matching descriptors
5. **Distance match filter** – removes matched pairs that are situated on big distances relative to image size
6. Intersection filter – removes intersecting pairs of matched points connected with lines
7. Global F filter – applies a global least-squares approach for estimation of $F$ [2] and discards those pairs that are situated relatively very far from epipolar lines
8. Threshold filter – selects best matched pairs for according to matching cost of descriptor similarity

2.4. Rectification Metrics

We utilize a collection of metrics for measuring quality of rectification as optimization criteria in rectification search stage. For every proposed “arbitrary setup”, we calculate $H_{L,R}$ and rectify feature sets $m_{L,R}$ as follows:

$$m_{L}^* = (H_{L}) m_{L}, \quad m_{R}^* = (H_{R}) m_{R}$$

(9)

Our collection includes several metrics used in benchmark tests proposed in [6] and will be briefly described:

**Sample Distance Error (ES)** [2] –

$$E_s = \frac{1}{N} \sum_{i=1}^{N} \left| \left( m_{L}^* \right)_i - \left( m_{R}^* \right)_i \right|^2$$

(10)

**Vertical disparity error (ES)** measures amount of vertical shifts of rectified pairs in terms of estimated feature matches:

$$E_v = \frac{1}{N} \sum_{i=1}^{N} \left| \left( m_{L} - m_{R} \right)_i \right|^2$$

(11)

**Projective Distortions** – measure the amount of introduced projective distortions in image structure of rectified pairs in terms of skew-ness and scale changes. Two metrics are utilized – $E_O, E_d$ that measure respectively orthogonality and proportional sizes of the transformed images. The metrics are measured for the rectified version of middle edge points $a, b, c, d$ for each of image pairs:

$$a = H \left( \frac{s_x}{2}, 0, 0 \right), \quad b = H \left( s_x, \frac{s_x}{2}, 0 \right), \quad c = H \left( \frac{s_x}{2}, s_y, 0 \right), \quad d = H \left( 0, \frac{s_x}{2}, 0 \right)$$

$$E_o = \cos^{-1} \left( \frac{(b-d) \cdot (a-c)}{|b-d| \cdot |a-c|} \right), \quad E_d = \sqrt{\left( \frac{(b-d) \cdot (b-d)}{(a-c) \cdot (a-c)} \right)}$$

(12)

2.5. Optimization Approach

As an example of minimization function we consider any of those global optimization search methods based on “constrained gradient-descent minimization with regularization” [11]. For the minimization criterion is used $E_S$. If it exceeds some defined quality threshold, then a collection of other metrics (e.g. $E_O, E_d, E_s$) are post-optimized, but for the constraint of $E_S$. Our implementation was tested successfully for the constraint of misalignments $\varphi, \psi, \theta, Y, Z, d_{r, z}$ and could be further extended to more others if needed (e.g. optical distortions). Each of the misalignment set is limited to certain search range that is expected or practical for post-rectification. For example, cameras are expected to be roughly (visually) rectified where angles of misalignment $\varphi, \psi, \theta << 45^0$ and cameras to be aligned more in horizontal direction than in vertical (or forward/backward) one $(Y, Z<<X)$. Another possible limitation could be applied on projective distortions levels, which in practical cases are unlikely to change more than a very narrow range. For those limitation examples, our search optimization cost function $C$ is defined as follows:

$$C(E_i, E_s, E_d, E_o) = \arg_{\Delta F} \min \left\{ \| H_{L} m_{L} - H_{R} m_{R} \| \right\}$$

where $C$ subject to condition:

$$(\varphi, \psi, \theta) \in \left[ -\frac{\pi}{4}, \frac{\pi}{4} \right], X = 1, \quad Y, Z \in [-X, X],$$

$$d_{r, z} \in \left[ 1 - \epsilon, 1 + \epsilon \right], \epsilon << 1$$

(13)

The optimization search framework consists of several steps which are envisioned in Fig. 6. First one initializes a set of DoF for expected levels. Next step scans the set and consecutively selects a single DoF from the set and saves result of the one that gives the best rectification among. If result is worse than the one of initial (non-rectified) setup, then the optimization search is canceled. Otherwise, the best performed DoF is added to the search subset and optimization process is repeated for scanning of best performing combination of DoF values in the current subset with additionally chosen DoF from the main set. The process is repeated until all subset combinations are checked or $C$ exceeds some quality threshold.

2.6. Computation Analysis

The code realization shows convergence for the optimization search function for $<2000$ iters., and an example for Ground Truth (GT) data is plotted in Fig. 7. The calculations of cost function (Eq. 13) for $H_{L,R}$ (Eq. 4-8) include few matrix operations for a very small element sizes – $3 \times 3, 3 \times 4$, which totals in $\sim 120$ multiplications $(O)$. The whole optimization procedure totals in $\sim (2000) \times 1200O$. The RansAC approaches [1, 4, 5, 6, 9, 10] require approximately the following minimum multiplication costs: 1. Solution for $F$ by SVD for square matrix($nnn$) [13]:

$$2 \left( n^4 - \frac{n^3}{3} \right) + \left( n^4 - \frac{n^3}{3} + \frac{2n^2}{3} \right)$$

(14)

Fig.5. Result of matching filter process for “Arch” pair(left-to-right): a) non-filtered, b) filtered

Fig.6. Optimization search process flow diagram
if $n=8 \rightarrow \sim 9000O$, if $n=7\rightarrow \sim 2800O$, $n=6\rightarrow 1512O$; 2.

Transform feature sets and estimate outliers by distance to epipolar lines: $\sim 54O$; 3. Point normalization: $\sim 256O$.

The calculations of RanSAC sample iterations for different proportions of outliers is given as [2]:

$$N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^{T})}, \text{ where} \quad (15)$$

$\varepsilon$ – proportion of outliers, $p$ – confidence probability for no selected outliers, $s$ – sample size, and $T$ – RanSAC consensus set, calculated by: $T=(1-\varepsilon) \times M$. We have pre-calculated according to Eq. 15 several cases of outliers percentage for a typical case of benchmark feature point set ($M=40$ samples)[6] and calculations are given in Table 1. For the ideal case of small percentage of outliers, the whole RanSAC search procedure totals at minimum of $\sim (1822) \times 395O$ or $\sim 2000 \times 359O$. Compared the latter to our computation cost, we make a claim that our approach performs faster for simpler code implementation, but without using specific software (e.g. SVD).

### Table 1

<table>
<thead>
<tr>
<th>Method</th>
<th>Sample Set</th>
<th>Proportion of Outliers $\varepsilon$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-point F</td>
<td>8</td>
<td>484</td>
</tr>
<tr>
<td>7-point F</td>
<td>7</td>
<td>438</td>
</tr>
<tr>
<td>6-point F</td>
<td>6</td>
<td>395</td>
</tr>
</tbody>
</table>

3. EXPERIMENTS AND RESULTS

We have experimented with dataset provided by Mallon and Whelan in [6]. We compare our results to those of other rectification methods, which had been tested against same dataset [3, 5, 6, 9] for same rectification metrics (Table 2). For the sake of consistency, we apply our calibration application on provided feature matching points without any additional outlier pre-filtering. The results of [2, 3, 9] were directly transferred from [9], but those in [6] were computed by our metric code realization. The numbers in Table 2 show that we obtain performance comparable to methods in [6, 9] (Mallon, Zilly) and provide far better one than those of [3, 2] (Loop, Hartley). For the estimated quality, we could claim that our results are always close or even in most cases better than those against which we compare (e.g. $E_{R \text{RMEAN}}$ is always better than others). In all cases, we achieve sub-pixel precision for $E_{R}$ for a very small amount of projective distortions ($E_{O}=\pm 0.1^\circ$). Thus, we could infer that our results show very good and balanced performance. The visual results are shown in Fig. 8.

Performance could be even more improved, if outlier pre-filtering step is triggered or projective distortions are excluded. The last one will force results of $E_{O}$ and $E_{I}$ to be always perfect, but will slightly decrease those of $E_{R}$. On the other hand, if the range of $d_{i,j}$ is increased, then $E_{R}$ is very close to best possible, but visible projective distortions will be easily observed ($E_{O}=\pm 1^\circ$).
4. REFERENCES