UDC 514.123.1

PLAUSIBLE RECTIFICATION FROM THE FUNDAMENTAL MATRIX

V. Nechaieva[1, a], V. Krygin[2,3, b]

[1] *Taras Shevchenko National University of Kyiv*

[2] *National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Institute of Physics and Technology*

[3] *International Research and Training Center for Information Technologies and Systems of the National Academy of Sciences of Ukraine and Ministry of Education and Science of Ukraine*

Abstract

Rectification of a stereo pair from the fundamental matrix (also known as *uncalibrated rectification* [1]) is a procedure that results in a new pair of images that have strictly horizontal epipolar lines. Moreover, the corresponding epipolar lines of the two images acquire the same vertical position. This means that the vertical disparity between corresponding pixels becomes small enough (e.g. zero, if we speak about integer pixel coordinates), and efficient stereo matching algorithms can be applied further. This article presents a *plausible* algorithm to achieve this. The plausibility consists in the fact that the algorithm works not for all cases, and these cases are studied in detail in this paper. However, authors claim that the presented algorithm deserves attention because of implementation and explanation simplicity.

Keywords: fundamental matrix, projective rectification, stereo vision, epipolar geometry, computer vision

Introduction

The stereo correspondence problem is complex. One of the possible ways to state it is: how to find the correspondence between pixels of both images so that they are the projections of the same points of observed objects? It means that each pixel of one image is labelled either by the coordinate of the corresponding pixel of another image or by "no correspondence" label. Formally,

$$\overline{d} : I_L \to I_R \cup \{\varnothing\},$$

where I_L is a set of the left image pixels and I_R is a set of the right image pixels. So, we look for an answer in the set of $|I_L|^{|I_R|+1}$ possible ones. A good rectification can collapse the answers set dramatically: given the right image of width w and height h, we can reduce the $|I_L|^{|I_R|+1} = |I_L|^{w \cdot h+1}$ to $|I_L|^{w+1}$ by removing the vertical disparity. The stereo correspondence itself is not a subject of this paper, but the interested reader is referred to [2, 3].

In the first section of this paper, we describe the rectification of the right image, provided by [4]. We're not diving into details of epipolar geometry – just providing the main definitions and relations.

In the second section, we describe our plausible method of rectification of the left image based on the rectification of the right one.

The third section provides an analysis of the domain of the proposed algorithm: the cases when it doesn't work.

1. Right image rectification

We are looking for matrices that transform the images so that their corresponding pixels lie on the same horizontal lines. One instrument that is going to help us is the fundamental matrix $F \in \mathbb{R}^{3\times 3}$.

The fundamental matrix specifies the relation between the points of two images, lying on the corresponding epipolar lines

$$\boldsymbol{x}'^T \cdot F \cdot \boldsymbol{x} = 0.$$

Given a set of corresponding points $C \subset I_L \times I_R$ (can be obtained with the help of [5, 6, 7]), the fundamental matrix can be estimated using the 8-point algorithm, 7-point algorithm [8], or any other.

The fundamental matrix F has rank 2 so its left and right kernels are one-dimensional spaces. A vector $\boldsymbol{e} = \begin{bmatrix} e_x & e_y & 1 \end{bmatrix}^T$ from the right kernel is called the *epipole* of the left image. Geometrically, epipole of an image is a projection of an optical center of another camera. In the same manner, $\boldsymbol{e}' = \begin{bmatrix} e'_x & e'_y & 1 \end{bmatrix}^T$ from the left kernel is called the *epipole* of the right image.

Henceforth, we will use $\boldsymbol{e}$ and $\boldsymbol{e}'$ for the unit vectors, having the same directions as the epipoles. This redefinition simplifies further formulas but doesn't break them.

We assume that we have a pinhole camera with an image center in $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and the camera center in $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$. To make the epipolar lines horizontal, we must move the epipolar point to $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$. This can be performed with a linear mapping called *rectification*. Let us call this mapping R_r, or the *rectification matrix*

$$R_r = G \cdot R,$$

[a] veronika.netch@gmail.com
[b] valeriy.krygin@gmail.com

where R is a rotation matrix – it rotates the image so that the epipole lies on the abscissa of the image, and G is a shear matrix – it makes the epipolar lines parallel to each other, i.e., it moves the epipolar point to the right into the infinity.

Let us look a little further into these components. While there are different ways to represent them [4, 9], we choose the following way. The R matrix consists of orthonormal vectors

$$R = \begin{bmatrix} (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \\ (\boldsymbol{k} \times \boldsymbol{e}')^T \\ \boldsymbol{k}^T \end{bmatrix},$$

where $k = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ is one of the basis vectors of the space. The G matrix is

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-e'_z}{x} & 0 & 1 \end{bmatrix},$$

where

$$x = (R \cdot \boldsymbol{e}')_x = (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \cdot \boldsymbol{e}'. \qquad (1)$$

Thus, R_r is

$$R_r = G \cdot R = \begin{bmatrix} (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \\ (\boldsymbol{k} \times \boldsymbol{e}')^T \\ (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \cdot \frac{-e'_z}{x} + \boldsymbol{k}^T \end{bmatrix}.$$

You can read more about these implications in [4, 8].

You can see a comparison of one of our experimental rectification results with the one achieved with OpenCV [10] on fig. 1. Fig. 1b and fig. 1a are the input images taken by a smartphone. Correspondences and the fundamental matrix were found by an application, written for this paper evaluation. Fig. 1d and fig. 1c are the results of applying `stereoRectifyUncalibrated` function from OpenCV library. Fig. 1f and fig. 1e are the results of applying the plausible rectification method. You can see that both methods give similar results.

The pairs of rectification results are valid stereograms: you can cross your eyes to see the third picture between them and focus on the paper plane to percept the depth. The digital version of the article contains high-quality pictures, so it's recommended to perform the depth perception with it.

2. Plausible rectification of the left image

We will use a matrix

$$[\boldsymbol{e}']_\times = \begin{bmatrix} 0 & -e'_z & e'_y \\ e'_z & 0 & -e'_x \\ -e'_y & e'_x & 0 \end{bmatrix}.$$

It can be trivially checked that

$$[\boldsymbol{e}']_\times \cdot \boldsymbol{x} = \boldsymbol{e}' \times \boldsymbol{x}, \quad \forall \boldsymbol{x} \in \mathbb{R}^3.$$

Let us build the intermediate matrix of rank 2

$$M = R_r \cdot [\boldsymbol{e}']_\times \cdot F$$

and denote its last two rows as $\boldsymbol{m}_y^T$ and $\boldsymbol{m}_z^T$, and construct the new matrix

$$M' = \begin{bmatrix} \frac{(\boldsymbol{m}_y \times \boldsymbol{m}_z)^T}{\sqrt{\|\boldsymbol{m}_y \times \boldsymbol{m}_z\|}} \\ \boldsymbol{m}_y^T \\ \boldsymbol{m}_z^T \end{bmatrix}.$$

Usually, x-coordinates of the left image, resampled with the M' matrix, differ a lot from the corresponding ones of the transformed right image. That's why it's recommended to use one more trick [4] and define the left image rectification matrix as

$$R_l = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot M',$$

where a, b and c are the solutions to the following linear least squares problem:

$$\sum_{\langle \boldsymbol{x}, \boldsymbol{x}' \rangle \in \mathcal{C}} \left(\begin{bmatrix} a \\ b \\ c \end{bmatrix}^T \cdot \frac{M' \cdot \boldsymbol{x}}{(M' \cdot \boldsymbol{x})_z} - \frac{(R_r \cdot \boldsymbol{x}')_x}{(R_r \cdot \boldsymbol{x}')_z} \right)^2 \to \min_{a,b,c \in \mathbb{R}},$$

where $\mathcal{C}$ is a set of corresponding points, found for the fundamental matrix estimation. Don't forget to divide by z-coordinates to normalize coordinates of the pixels.

The next section provides a derivation of the algorithm and analysis of its weaknesses.

3. Analysis of the plausible rectification method

Let M be the intermediate epipolar projection matrix that transforms the left image. The fundamental matrix F can be represented as $F = [\boldsymbol{e}']_\times \cdot S$, where S is a transformation from the space of the right image to the space of the left one. One of the methods to fetch this matrix (up to scale) is [11]

$$S = [\boldsymbol{e}']_\times \cdot F.$$

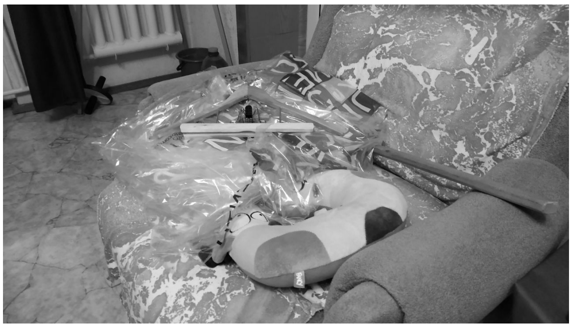
This means that the linear transformation M consists of two parts: teleport the left image to the right image space using S and rectify it there using R_r matrix

$$\begin{aligned} M &= R_r \cdot S \\ &= G \cdot \begin{bmatrix} ((\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')) \times \boldsymbol{e}')^T \\ ((\boldsymbol{k} \times \boldsymbol{e}') \times \boldsymbol{e}')^T \\ (\boldsymbol{k} \times \boldsymbol{e}')^T \end{bmatrix} \cdot F \\ &= G \cdot \begin{bmatrix} (\boldsymbol{k} \times \boldsymbol{e}')^T \\ ((\boldsymbol{k} \times \boldsymbol{e}') \times \boldsymbol{e}')^T \\ (\boldsymbol{k} \times \boldsymbol{e}')^T \end{bmatrix}^T \cdot F. \end{aligned}$$

As we can see, the rank of this matrix is not greater than 2. We want it to become full-rank after we replace the first row of it. For this, we have to make sure that the second and third rows are linearly independent.

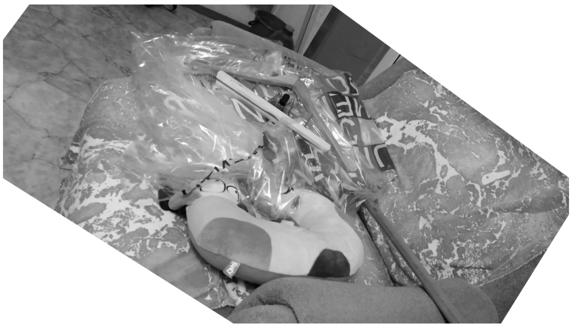
Let us look into $G \cdot R \cdot [\boldsymbol{e}']_\times$ first. Since

$$(\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')) \times \boldsymbol{e}' = \boldsymbol{k} \times \boldsymbol{e}',$$



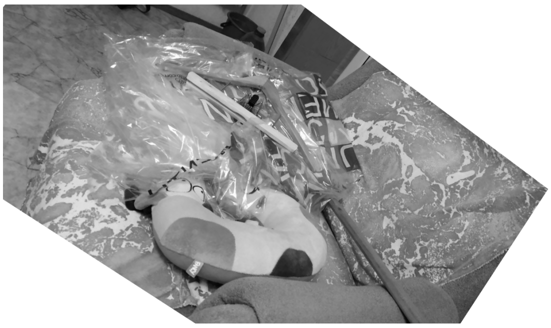
(a) Original right image

(b) Original left image

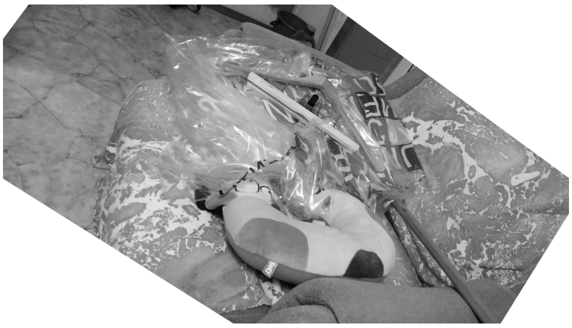


(c) The right image rectified using OpenCV

(d) The left image rectified using OpenCV



(e) The right image rectified using our plausible method



(f) The left image rectified using our plausible method

Fig. 1. Experimental results

we get

$$G \cdot R \cdot [\boldsymbol{e}']_\times = \begin{bmatrix} (\boldsymbol{k} \times \boldsymbol{e}')^T \\ (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \\ (\boldsymbol{k} \times \boldsymbol{e}')^T \cdot \left(\frac{-e'_z}{x} + 1\right) \end{bmatrix}.$$

Since $\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}') \perp (\boldsymbol{k} \times \boldsymbol{e}')$ for any non-zero vectors, the only case that worries us is when one of the rows equals zero. The second line can turn into zero only if $\boldsymbol{k}$ and $\boldsymbol{e}'$ are collinear. That would mean that $\boldsymbol{e}'$ lies exactly in the center of the image. If $\boldsymbol{e}'$ lies within an image, then it is impossible to make all the epipolar lines parallel, and $x = 0$, so G does not even exist; so this case is not a subject of the paper.

The only edge case that does interest us is

$$\frac{-e'_z}{x} = -1.$$

Substitution of x with (1) gives

$$e'_z = (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \cdot \boldsymbol{e}'.$$

Since $\boldsymbol{k}$ is the applicate, $e'_z = \boldsymbol{k} \cdot \boldsymbol{e}'$. Then we get

$$\boldsymbol{k}^T \cdot \boldsymbol{e}' = (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'))^T \cdot \boldsymbol{e}'. \qquad (2)$$

Let us say that α is the angle between $\boldsymbol{e}'$ and $\boldsymbol{k}$. Since $\boldsymbol{k}$, $\boldsymbol{e}'$, and $\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')$ are coplanar, and $\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')$ is perpendicular to $\boldsymbol{k}$, $\boldsymbol{e}'$ can be written as a linear combination of $\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')$ and $\boldsymbol{k}$:

$$\boldsymbol{e}' = \cos\alpha \cdot \boldsymbol{k} + \sin\alpha \cdot \boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')$$

Since $\boldsymbol{e}'$ is not collinear to $\boldsymbol{k}$, we know for sure that $\sin\alpha \neq 0$ and can divide by it

$$\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}') = \frac{\boldsymbol{e}' - \cos\alpha \cdot \boldsymbol{k}}{\sin\alpha}$$

and (2) turns into

$$\boldsymbol{k}^T \cdot \boldsymbol{e}' = \frac{\boldsymbol{e}'^T \cdot \boldsymbol{e}' - \cos\alpha \cdot \boldsymbol{k}^T \cdot \boldsymbol{e}'}{\sin\alpha},$$

implying

$$(\cos\alpha + \sin\alpha) \cdot \boldsymbol{k}^T \cdot \boldsymbol{e}' = \boldsymbol{e}'^T \cdot \boldsymbol{e}'.$$

By definition of the dot product,

$$(\cos\alpha + \sin\alpha) \cdot \|\boldsymbol{k}\| \cdot \|\boldsymbol{e}'\| \cdot \cos a = \|\boldsymbol{e}'\|^2$$

All the vectors are of length 1, meaning

$$(\cos\alpha + \sin\alpha) \cdot \cos\alpha = 1,$$

thus

$$\sin\alpha \cdot (\sin\alpha - \cos\alpha) = 0,$$

which means the following solutions for α:

$$\begin{aligned} \alpha &= \pi \cdot n, \\ \alpha &= \pi \cdot n - \frac{pi}{4}, \end{aligned} \quad \forall n \in \mathbb{Z}.$$

The first case means that $\boldsymbol{e}'$ is collinear to the applicate, but we have already omitted this case. So, the only situation that troubles us is when $\boldsymbol{e}'$ creates a 45° angle with one of these two vectors and a 135° with another one. This is a significant problem with the method.

Now we return to the original formula

$$M = G \cdot R \cdot [\boldsymbol{e'}]_\times \cdot F$$

and see what happens if we unfold it completely.

$$M = \begin{bmatrix} (\boldsymbol{k} \times \boldsymbol{e'})^T \cdot F \\ (\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e'}))^T \cdot F \\ (\boldsymbol{k} \times \boldsymbol{e'}) \cdot F \cdot \left(\frac{-e'_z}{x} + 1\right) \end{bmatrix},$$

where F is the fundamental matrix, which has rank 2. The only unit vector in its left nullspace is $\boldsymbol{e}'$. Consider the decomposition of an arbitrary vector $\boldsymbol{x}$ into

$$\boldsymbol{x} = \boldsymbol{y} + c \cdot \boldsymbol{e}', \quad c \in \mathbb{R},\ \boldsymbol{y} \in \mathbb{R}^3,\ \boldsymbol{y} \perp \boldsymbol{e}'.$$

As long as $\boldsymbol{e}'$ belongs to the left kernel of the fundamental matrix F, meaning $c \cdot \boldsymbol{e}'^T \cdot F = 0$, the following equation applies:

$$\boldsymbol{x}^T \cdot F = (\boldsymbol{y} + c \cdot \boldsymbol{e}')^T \cdot F = \boldsymbol{y}^T \cdot F + c \cdot \boldsymbol{e}'^T \cdot F = \boldsymbol{y}^T \cdot F,$$

Thus, for any 3D vectors $\boldsymbol{x}$ and $\boldsymbol{y}$

$$\boldsymbol{x}^T \cdot F = \boldsymbol{y}^T \cdot F \iff \exists c \in \mathbb{R} : \boldsymbol{x} = \boldsymbol{y} + c \cdot \boldsymbol{e}', \boldsymbol{e}' \perp \boldsymbol{y}.$$

Suppose that multiplying by F makes the second and third rows' vectors collinear. It would mean that

$$\boldsymbol{k} \times \boldsymbol{e}' = \boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}') + c \cdot \boldsymbol{e}'.$$

This case is impossible since $\boldsymbol{k} \times \boldsymbol{e}' \perp \boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}')$ and $\boldsymbol{k} \times \boldsymbol{e}' \perp \boldsymbol{e}'$ if $\boldsymbol{e}' \neq \boldsymbol{k}$. It means that the fundamental matrix does not create any more special cases, and thus the limitations stay the same.

To conclude, we have three different cases where the method fails:

$$\begin{gathered} \angle(\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'), \boldsymbol{e}') = 135° \ \&\ \angle(\boldsymbol{k}, \boldsymbol{e}') = 45°; \\ \angle(\boldsymbol{k} \times (\boldsymbol{k} \times \boldsymbol{e}'), \boldsymbol{e}') = 45° \ \&\ \angle(\boldsymbol{k}, \boldsymbol{e}') = 135°; \\ \angle(\boldsymbol{k}, \boldsymbol{e}') = 0°, \end{gathered} \tag{3}$$

where the center of the image lies in $\boldsymbol{k}$.

4. Further work

Further work can be divided into two parts. First, some cases change the flow of the analysis – for example, if another method of right image rectification is used. Second, it is possible that in the cases where the plausible rectification fails, certain alternations of the method may be used.

Conclusion

The plausible rectification method has a certain elegancy and does not require too many calculations. However, the method is not universal; it fails when the basis vector $\boldsymbol{k}$ of the space creates a 0°, 45° or 135° angle with the epipole (3). The number of such cases is limited but still can affect the method correctness depending on the situation.

References

1. Fusiello Andrea, Irsara Luca. Quasi-Euclidean uncalibrated epipolar rectification. — 2009. — 01. — P. 1 – 4.
2. Savchynskyy Bogdan. Discrete Graphical Models — An Optimization Perspective // Foundations and Trends® in Computer Graphics and Vision. — 2019. — Vol. 11, no. 3-4. — P. 160–429.
3. Szeliski Richard. Computer vision algorithms and applications. — London; New York : Springer, 2011. — Access mode: `http://dx.doi.org/10.1007/978-1-84882-935-0`.
4. Hartley Richard I. Theory and Practice of Projective Rectification // Int. J. Comput. Vis. — 1999. — Vol. 35, no. 2. — P. 115–127. — Access mode: `https://doi.org/10.1023/A:1008115206617`.
5. Lowe David G. Object Recognition from Local Scale-Invariant Features // Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. — ICCV '99. — USA : IEEE Computer Society, 1999. — P. 1150.
6. Speeded-Up Robust Features (SURF) / Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool // Comput. Vis. Image Underst. — 2008. — Jun. — Vol. 110, no. 3. — P. 346–359. — Access mode: `https://doi.org/10.1016/j.cviu.2007.09.014`.
7. ORB: An Efficient Alternative to SIFT or SURF / Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski // International Conference on Computer Vision. — Barcelona, 2011. — 11/2011.
8. Hartley R. I., Zisserman A. Multiple View Geometry in Computer Vision. — Second edition. — Cambridge University Press, ISBN: 0521540518, 2004.
9. Mallon John, Whelan Paul F. Projective rectification from the fundamental matrix // Image and Vision Computing. — 2005. — Vol. 23, no. 7. — P. 643 – 650. — Access mode: `http://www.sciencedirect.com/science/article/pii/S0262885605000399`.
10. Bradski G. The OpenCV Library // Dr. Dobb's Journal of Software Tools. — 2000.
11. Luong Q.-T., Viéville T. Canonical Representations for the Geometries of Multiple Projective Views // Computer Vision and Image Understanding. — 1996. — Vol. 64, no. 2. — P. 193 – 229. — Access mode: `http://www.sciencedirect.com/science/article/pii/S1077314296900557`.