Epipolar rectification

Pascal Monasse
Université Paris-Est Marne-la-Vallée

Zhongwei Tang
Huawei Technologies

Epipolar rectification

Pascal Monasse, Jean-Michel Morel, Zhongwei Tang

October 30, 2010

Abstract

Image stereo-rectification is the process by which two images of the same solid scene undergo homographic transforms, so that their corresponding epipolar lines coincide and become parallel with the x-axis of image. A pair of stereo-rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each match to a line parallel to the x-axis. Due to the redundant degrees of freedom, the solution to stereo-rectification is not unique and actually can lead to undesirable distortions or be stuck in a local minimum of the distortion function. Different algorithms have been proposed to solve this problem. Since most of the algorithms are based on multi-view geometry, it is necessary to review the fundamental properties of multi-view geometry before entering the details of rectification.

1 Pinhole camera

Establishing a camera model is the first step to treat the other multi-view geometry problems. In this report, the camera is considered as a pinhole model (In reality, the geometric distortion introduced by camera lens should also be considered). Some basic concepts about the camera are shown in Fig. 1:

- camera center (or optic center, or focal point): the point through which all relevant light rays pass.
- image plane: the camera CCD plane where the image is formed. This plane does *not* contain the camera center.

Here is some standard vocabulary concerning such a model:

- principal axis: the line from the camera center perpendicular to the image plane.
- principal plane: the plane containing the camera center and parallel to the image plane.
- focal length f: the distance from the camera center to the image plane.
- camera frame: the coordinate frame based on camera which has camera center as origin and principal axis as Z-axis.

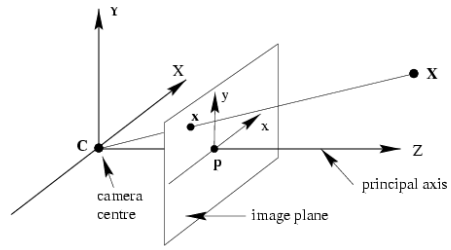

Figure 1: Pinhole camera model.

- world frame: a pre-fixed coordinate frame where any 3D point can be represented.

Note that there is a rotation and translation between the world frame and the camera frame.

1.1 Central projection

In the pinhole camera model, all the relevant light rays pass through the camera center. So it is called central or conic projection. To project a 3D point to camera plane, the first step is to represent it in the camera frame by a translation and a rotation from the world frame:

$$\hat{\mathbf{X}}_c = \mathbf{R}(\hat{\mathbf{X}} - \mathbf{C}) \tag{1}$$

with $\hat{\mathbf{X}} = (X, Y, Z)^T$ and $\hat{\mathbf{X}}_c = (X_c, Y_c, Z_c)^T$ the coordinate of a point in the world frame and in the camera frame respectively; $\mathbf{C} = (X_o, Y_o, Z_o)^T$ represents the camera center in the world frame.

The projection of $\hat{\mathbf{X}}_c$ on camera plane is:

$$x_c = fX_c/Z_c \tag{2}$$

$$y_c = fY_c/Z_c \tag{3}$$

Exercice 1. *Prove these equations. Hint: simply apply Thales's theorem!*

This can also be represented more succinctly in matrix form by using homogeneous coordinates:

$$\mathbf{x}_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{X}}_c \tag{4}$$

In homogeneous coordinates, $\mathbf{x}_c = (fX_c, fY_c, Z_c)^T$ is equivalent to the 2D point $(fX_c/Z_c, fY_c/Z_c)^T$ by dividing the first two coordinates by the third coordinate. We will come back to homogeneous coordinates in Sect. 2.1.

By concatenating the frame change and the central projection, a 3D point is projected to a 2D point:

$$\mathbf{x}_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{X}}_c = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right) \begin{pmatrix} \hat{\mathbf{X}} \\ 1 \end{pmatrix} \tag{5}$$

$$= \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right) \mathbf{X}$$

with $\mathbf{X} = (X, Y, Z, 1)^T$ the homogeneous coordinates of 3D point $\hat{\mathbf{X}} = (X, Y, Z)^T$.

1.2 Internal parameters

The above obtained 2D point $\mathbf{x}_c$ has a meter or millimeter (or inch!) unit. But any digital image is measured in the pixel unit. In addition, the projected point $\mathbf{x}_c$ has the principal point as the origin, while the convention is to take the top-left corner of the image as origin. Due to some manufacture imprecision, the pixel in a CCD array is not necessarily square, but may be a rectangle or even a parallelogram. By considering this deformation and the other mentioned factors, we have:

$$\mathbf{x} = \begin{pmatrix} m_x & 0 & x_0 \\ 0 & m_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\cot\theta & 0 \\ 0 & \frac{1}{\sin\theta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}_c$$

$$= \begin{pmatrix} m_x & 0 & x_0 \\ 0 & m_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -\cot\theta & 0 \\ 0 & \frac{1}{\sin\theta} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right) \mathbf{X}$$

$$= \begin{pmatrix} m_x f & -m_x f \cot\theta & x_0 \\ 0 & \frac{m_y f}{\sin\theta} & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}\left(\mathbf{I} \mid -\mathbf{C}\right) \mathbf{X}$$

$$= \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} = \mathbf{P}\mathbf{X} \tag{6}$$

with α_x and β_y the number of pixels per unit length in the skewed x-axis direction and the skewed y-axis direction in the image plane respectively; f the focal length of camera; x_0 and y_0 the coordinates of the principal point, represented in the skewed image frame in pixels; s the skewness factor which is 0 when the pixel is rectangle; θ the skewness angle between two sides of image CCD plane. $\mathbf{K}$ is called the calibration matrix. It depends only on the camera settings, not on its position:

$$\mathbf{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} m_x f & -m_x f \cot\theta & x_0 \\ 0 & \frac{m_y f}{\sin\theta} & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Observe that the entries in $\mathbf{K}$ are not all positive. θ is generally in the range $[0, \pi]$. The entry $-m_x f \cot\theta$ will be positive if $\theta > 90^\circ$, negative if $\theta < 90^\circ$ and 0 when $\theta = 90^\circ$. The entry $\frac{m_y f}{\sin(\theta)}$ will always be positive. So the determinant of $\mathbf{K}$ will always be positive.

1.3 Projection matrix

In conclusion, the central projection from 3D to 2D performed by a pinhole camera can be represented by a 3×4 matrix $\mathbf{P} = \mathbf{KR}\,[\mathbf{I} \mid -\mathbf{C}]$, which is called the camera projection matrix. This matrix contains all the parameters of the camera: the calibration matrix as *internal parameters*; the camera orientation and camera center as *external parameters*.

Exercice 2. *Prove that* $\mathbf{P}$ *has rank 3 and that a null vector of* $\mathbf{P}$ *is the vector representing the camera center in homogeneous coordinates (see below).*

2 Projective geometry

It is necessary to introduce projective geometry to further investigate the properties of the camera projection matrix. It is easier to understand the 2D projective geometry, which is in fact the geometry of projective transformations of the plane. These transformations model the geometric distortion which arises when a plane is imaged by a pinhole camera. Under perspective imaging certain geometric properties are preserved while others are not. Projective geometry models this imaging and also provides a mathematical representation appropriate for computations.

2.1 Homogeneous coordinates

Homogeneous coordinates are very useful in multi-view geometry, as they represent many fundamental relationships in vector or matrix form and reduce them to linear algebra. We first introduce the homogeneous notation for points and lines on a plane. Then the homogeneous notation for 3D space is just evident. A convention in multi-view geometry is that all the geometric entities are represented by column vectors by default.

A line in the plane can be represented by an equation $ax + by + c = 0$ with $(x, y)^T$ a point on the line. It is natural to represent the equation in vector form: $\mathbf{x}^T\mathbf{l} = 0$ with $\mathbf{x} = (x, y, 1)^T$ and $\mathbf{l} = (a, b, c)^T$. But the vector $(x, y, 1)^T$ and $(a, b, c)^T$ are not the only vectors which satisfy the line equation. Any vector $m(x, y, 1)^T$ and $n(a, b, c)^T$ satisfy also the line equation for any $m \neq 0$ and $n \neq 0$. So two vectors related by an overall non-zero scaling are considered as being equivalent. An equivalence class of vectors under this equivalence relationship is known as a homogeneous vector. For a point in the plane, its homogeneous coordinates have the form $\mathbf{x} = (x_1, x_2, x_3)^T$, representing the point inhomogeneous coordinates $(x_1/x_3, x_2/x_3)^T$ $(x_3 \neq 0)$ in $\mathcal{R}^2$. Even if the homogeneous coordinates of points and lines in the plane are 3-vectors, by homogeneity the real degrees of freedom (DOF) are still 2.

Given two lines $\mathbf{l} = (a, b, c)^T$ and $\mathbf{l}' = (a', b', c')^T$, the homogeneous coordinates of the intersection point are $\mathbf{x} = \mathbf{l} \times \mathbf{l}'$ with $\times$ the cross product. On the other hand, the line passing through two points $\mathbf{x}$ and $\mathbf{x}'$ has the form $\mathbf{l} = \mathbf{x} \times \mathbf{x}'$. Note that the simplicity of the expressions is a direct consequence of the use of the homogeneous vector representation of lines and points.

Exercice 3. *Prove these assertions.*

Now consider two parallel lines $ax + by + c = 0$ and $ax + by + c' = 0$. They are represented by vectors $\mathbf{l} = (a, b, c)^T$ and $\mathbf{l}' = (a, b, c')^T$. Note that the first two coordinates are the same because they are parallel. The intersection point of the two lines is $\mathbf{l} \times \mathbf{l}' = (c' - c)(b, -a, 0)^T$. By ignoring the scale $c' - c$, the intersection point is $(b, -a, 0)$. If we try to compute the inhomogeneous coordinates of this point, we have $(b/0, -a/0)^T$, which makes no sense, except to suggest that this point has infinitely large coordinates. In fact, a point with homogeneous coordinates in the form $(x, y, 0)^T$ does not correspond to any finite point in $\mathcal{R}^2$. This agrees with the usual idea that parallel lines meet at infinity.

Homogeneous vectors $\mathbf{x} = (x_1, x_2, x_3)^T$ such that $x_3 \neq 0$ correspond to finite points in $\mathcal{R}^2$. By augmenting $\mathcal{R}^2$ with points having a last coordinate $x_3 = 0$, the resulting space is the set of all homogeneous 3-vectors, namely the 2D projective space $\mathcal{P}^2$. The points with last coordinate $x_3 = 0$ are called ideal points or points at infinity. Each ideal point represents a direction determined by the ratio $x_1 : x_2$ ($x_2 \neq 0$) or $x_2 : x_1$ ($x_1 \neq 0$). In addition, all the ideal points lie on a line at infinity, denoted by $\mathbf{l}_\infty = (0, 0, 1)^T$ since $(x_1, x_2, 0)(0, 0, 1)^T = 0$, $\forall x_1, x_2$. Each line $\mathbf{l}$ intersects $\mathbf{l}_\infty$ at an ideal point, which corresponds to the direction of $\mathbf{l}$. So the line at infinity $\mathbf{l}_\infty$ can also be considered as the set of all directions of lines in the plane.

2.2 Projective plane

The set of equivalence classes of vectors in $\mathcal{R}^3 - (0, 0, 0)^T$ forms the projective space $\mathcal{P}^2$ (the vector $(0, 0, 0)^T$ makes no sense in projective space). We can also think of $\mathcal{P}^2$ as a set of rays in $\mathcal{R}^3$. The set of vectors $k(x_1, x_2, x_3)^T$ as the scalar k varies forms a ray through the origin. Such a ray may be thought of as representing a single point in $\mathcal{P}^2$. In this model, the lines in $\mathcal{P}^2$ are planes passing through the origin. Two non-identical rays lie on exactly one plane, and any two planes intersect in one ray. This is the analogue of two distinct points uniquely defining a line, and two lines always intersecting in a point. Points and lines may be obtained by intersecting this set of rays and planes by the plane $x_3 = 1$. In Fig. 2, the rays representing ideal points and the plane representing $\mathbf{l}_\infty$ are parallel to the plane $x_3 = 1$.

Notice that in the projective plane, two lines always intersect (at an ideal point if they are parallel).

2.3 Transformations

The most important transformation in the projective plane is the projective transformation (or homography), which simply is a non-singular 3×3 matrix, denoted by $\mathbf{H}$. The 2D planar projective transformation preserves the collinearity: If $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are on the line $\mathbf{l}$, then $\mathbf{H}\mathbf{x}_1$, $\mathbf{H}\mathbf{x}_2$ and $\mathbf{H}\mathbf{x}_3$ are also on the line $\mathbf{H}^{-T}\mathbf{l}$. (This traduces the fact that if three vectors are coplanar, so are their images by a 3D linear transformation).

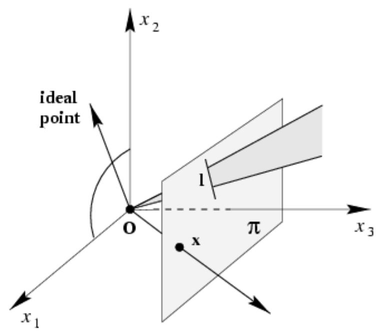

Figure 2: Projective plane model.

Note that all of the computation here is in homogeneous coordinates,

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{8}$$

$\mathbf{H}$ in the above equation may be changed by multiplication by an arbitrary non-zero factor without changing the result. The inhomogeneous coordinate of the point $(x'_1, x'_2, x'_3)^T$ is:

$$\begin{aligned} \tfrac{x'_1}{x'_3} &= \frac{h_{11}x_1 + h_{12}x_2 + h_{13}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \\ \tfrac{x'_2}{x'_3} &= \frac{h_{21}x_1 + h_{22}x_2 + h_{23}x_3}{h_{31}x_1 + h_{32}x_2 + h_{33}x_3} \end{aligned} \tag{9}$$

Note that the multiplication of H itself by a scalar factor yields the same transform. So $\mathbf{H}$ is also a homogeneous geometric entity and it has 8 degrees of freedom.

A point $\mathbf{x}$ is transformed to point $\mathbf{Hx}$ under the homography H, while a line $\mathbf{l}$ is transformed to a line $\mathbf{H^{-T}l}$. More details about projective transformations can be found in [6].

Exercice 4. *Prove that the image by $\mathbf{H}$ of the line passing through $\mathbf{x}_1$ and $\mathbf{x}_2$ is represented by the homogeneous vector $\mathbf{Hx}_1 \times \mathbf{Hx}_2$, so that if $\mathbf{x}_3$ is aligned with $\mathbf{x}_1$ and $\mathbf{x}_2$, then $\mathbf{Hx}_3$ is also on the image line. Hint: use the algebraic property defining the vector product: $(\mathbf{x}_1 \times \mathbf{x}_2)^T \mathbf{y} = |\mathbf{x}_1 \quad \mathbf{x}_2 \quad \mathbf{y}|$ $\forall \mathbf{y}$.*

Exercice 5. *Consider the case $h_{31} = h_{32} = 0 \neq h_{33}$ (affine transform of the plane). Prove that the image of $\mathbf{l}_\infty$ is $\mathbf{l}_\infty$. Conversely, prove that a homography that keeps the $\mathbf{l}_\infty$ line globally invariant is affine. What is the effect of a plane translation on $\mathbf{l}_\infty$? of a zoom? of a rotation in the plane?*

Exercice 6. *Prove that a homography is uniquely determined by the data of the images of 4 points in general position (no more than 2 of them on any line). Hint: consider the case where the 4 points are $\{0,1\} \times \{0,1\}$ and use composition of homographies to handle the general case.*

3 Camera rotation

A particular 2D projective transformation can be induced by a pure camera rotation without changing its optical center (Fig. 3). Given a 3D point $\mathbf{X}$, its projected image by rotating the camera is:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{K}_1\mathbf{R}_1[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \\ \mathbf{x}_2 &= \mathbf{K}_2\mathbf{R}_2[\mathbf{I} \mid -\mathbf{C}]\mathbf{X} \end{aligned} \tag{10}$$

By simple computation, we find that $\mathbf{x}_1$ and $\mathbf{x}_2$ are related by a homography:

$$\mathbf{x}_2 = \mathbf{K}_2\mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{K}_1^{-1}\mathbf{x}_1 = \mathbf{H}\mathbf{x}_1 \tag{11}$$

Camera rotation is not the only situation which induces the homography. When the 3D scene is a plane, the relationship between two images taken by a camera is also a homography. The third situation inducing homography is that the scene is very far away from the camera.

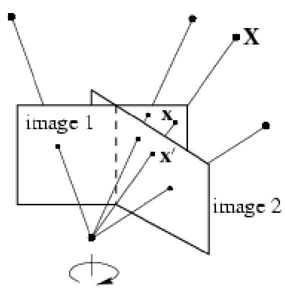

Figure 3: 2-D projective transformation (homography) induced by a pure camera rotation without changing camera center.

4 Fundamental matrix

The epipolar geometry is the intrinsic projective geometry between two views. It is independent of the observed scene structure and of the world frame. It depends only on the camera internal parameters and relative pose. The fundamental matrix $\mathbf{F}$ encapsulates this intrinsic geometry. It is a 3×3 matrix of rank 2. If $\mathbf{x}$ and $\mathbf{x}'$ is a pair of corresponding points in two views, then they satisfy the scalar equation $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$. The fundamental matrix can be computed from correspondences of imaged scene points alone, without requiring knowledge of the camera internal parameters or relative pose.

4.1 Epipolar constraint

In Fig. 4, we can see that a 3D point $\mathbf{X}$ is projected to $\mathbf{x}$ and $\mathbf{x}'$ in two views. These three points $\mathbf{X}$, $\mathbf{x}$ and $\mathbf{x}'$ form the epipolar plane, which intersects the two image planes by two epipolar lines respectively. The line connecting two camera centers is called baseline and intersects the

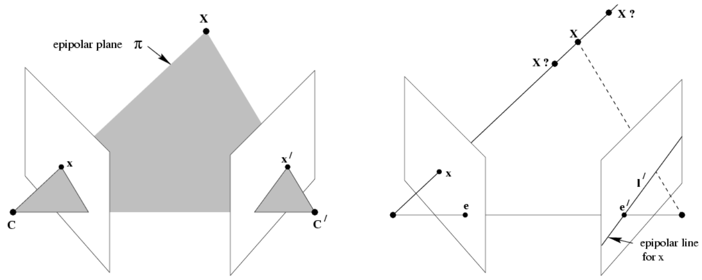

Figure 4: Left: The epipolar plane is formed by two camera centers and the 3-D point $\mathbf{X}$ Right: The 3-D point $\mathbf{X}$ must be on the back-projected ray defined by the left camera center and $\mathbf{x}$. This ray is imaged as a line $\mathbf{l}'$ called epipolar line in the second view and the image of $\mathbf{X}$ must lie on $\mathbf{l}'$.

two image planes at two epipoles respectively. The well-known equation for epipolar geometry writes:

$$\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0 \tag{12}$$

with $\mathbf{F}$ the fundamental matrix which is the algebraic representation of the epipolar geometry. We should remark that this equation is only a necessary condition for two corresponding points since $\mathbf{F}$ projects a point in one image to a corresponding epipolar line in the other image (Fig. 4). In Fig. 4, it is clear that the epipolar line is obtained by projecting the ray, which passes the optic center $\mathbf{C}$ and $\mathbf{x}$, to the other image. Since the position of the 3-D scene point $\mathbf{X}$ is not determined on the ray, the image point $\mathbf{x}'$ could be anywhere on the epipolar line $\mathbf{l}'$. So the mapping from a point to its epipolar line: $\mathbf{x} \mapsto \mathbf{l}'$ is a projection represented by $\mathbf{F}$, which can be written as:

$$\mathbf{l}' = \mathbf{F} \mathbf{x} \tag{13}$$

In geometry, the points in the left image represent a 2-D projective space, and the epipolar lines in the right image represent a 1-D projective space since all the epipolar lines have a common intersection point, the epipole $\mathbf{e}'$. So $\mathbf{F}$ represents a projection from a 2-D projective space to a 1-D projective space. From this viewpoint, it is natural to derive that $\mathbf{F}$ has rank equal to 2. For any point $\mathbf{x}'$ on the epipolar line $\mathbf{l}'$, we have the equation $\mathbf{x}'^T \mathbf{l}' = 0$, which is exactly (12) by using (13).

When the position of the 3-D scene point $\mathbf{X}$ varies, the epipolar plane rotates about the baseline (Fig. 4). The family of planes is known as an epipolar pencil, which intersects the two images at two pencils of epipolar lines. Each pencil of epipolar lines intersects at the corresponding epipole.

To get the explicit form of the $\mathbf{F}$ matrix, we can write the projections of a 3D point $\mathbf{X}$ in the two cameras, expressed in the first camera coordinate frame:

$$\lambda \mathbf{x} = \mathbf{K} [\mathbf{I} \ | \mathbf{0}] \mathbf{X} = \lambda \mathbf{K} \hat{\mathbf{X}} \tag{14}$$

$$\lambda' \mathbf{x}' = \mathbf{K}' [\mathbf{R} \ | \mathbf{T}] \mathbf{X} \tag{15}$$

R is the rotation of the camera 2 frame relative to the camera 1 frame and $\mathbf{T}$ is the coordinate of camera 1's optical center in the camera 2 frame. These 6 scalar equations have 5 parameters depending on the scene structure: $\hat{\mathbf{X}}$, λ and λ'. So it is expected that by eliminating these from the system we shall get one scalar equation.

The first equation, $\hat{\mathbf{X}} = \lambda K^{-1}\mathbf{x}$ substituted in the second equation, yields:

$$\lambda' \mathbf{K}'^{-1}\mathbf{x}' = \lambda \mathbf{R}\mathbf{K}^{-1}\mathbf{x} + \mathbf{T}. \tag{16}$$

The cross-product of each side with vector $\mathbf{T}$ gives:

$$\lambda'[\mathbf{T}]_\times \mathbf{K}'^{-1}\mathbf{x}' = [\mathbf{T}]_\times \mathbf{R}\mathbf{K}^{-1}\mathbf{x} \tag{17}$$

and since the left hand side is orthogonal to $\mathbf{K}'^{-1}\mathbf{x}'$:

$$(\mathbf{K}'^{-1}\mathbf{x}')^T[\mathbf{T}]_\times \mathbf{R}\mathbf{K}^{-1}\mathbf{x} = 0, \tag{18}$$

which is (12) with

$$\mathbf{F} = \mathbf{K}'^{-T}[\mathbf{T}]_\times \mathbf{R}\mathbf{K}^{-1}. \tag{19}$$

Some properties of the fundamental matrix are summarized here (the details could be found in chapter 9 of [6])

- F is a 3×3 rank-2 homogeneous matrix with 7 freedom degrees
- $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$ for a pair of corresponding image points $\mathbf{x}$ and $\mathbf{x}'$
  - Given a point $\mathbf{x}$ in the left image, the corresponding epipolar line in the right image is $\mathbf{l}' = \mathbf{F}\mathbf{x}$
  - Given a point $\mathbf{x}'$ in the right image, the corresponding epipolar line in the left image is $\mathbf{l} = \mathbf{F}^T\mathbf{x}'$
- $\mathbf{F}\mathbf{e} = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}' = 0$: the epipoles are the null vectors of $\mathbf{F}$ and $\mathbf{F}^T$ and all epipolar lines contain the epipoles.
- Correspondence between epipolar lines: $\mathbf{l}' = \mathbf{F}[\mathbf{e}]_\times\mathbf{l}$ and $\mathbf{l} = \mathbf{F}^T[\mathbf{e}']_\times\mathbf{l}'$

$\mathbf{F}$ only depends on projective properties of the cameras $\mathbf{P}$, $\mathbf{P}'$. The camera projection matrices relate 3-space measurements to image measurements and so depend on both the image coordinate frame and the choice of the world coordinate frame. On the contrary $\mathbf{F}$ does not depend on the choice of the world frame. More precisely, if $\mathbf{H}$ is a 4×4 matrix representing a projective transformation of the 3D-space, then the fundamental matrix corresponding to the pairs of camera matrix $(\mathbf{P}, \mathbf{P}')$ and $(\mathbf{P}\mathbf{H}, \mathbf{P}'\mathbf{H})$ are the same. This is because $\mathbf{x} = \mathbf{P}\mathbf{X} = (\mathbf{P}\mathbf{H})\left(\mathbf{H}^{-1}\mathbf{X}\right)$ and $\mathbf{x}' = \mathbf{P}'\mathbf{X} = (\mathbf{P}'\mathbf{H})\left(\mathbf{H}^{-1}\mathbf{X}\right)$. So the matched points $\mathbf{x} \leftrightarrow \mathbf{x}'$ are not changed under the 3-D projective transformation $\mathbf{H}$ even if the pair of camera matrix and the 3-D point are changed. Then the fundamental matrix also remains unchanged.

When $\mathbf{K}$ and $\mathbf{K}'$ are known (called the calibrated case), the constraint (12) but involving $\mathbf{K}^{-1}\mathbf{x}$ and $\mathbf{K}'^{-1}\mathbf{x}'$ can be rewritten:

$$\left(\mathbf{K}'^{-1}\mathbf{x}'\right)^T \mathbf{E}\left(\mathbf{K}^{-1}\mathbf{x}\right) = 0 \tag{20}$$

with

$$\mathbf{E} = [\mathbf{T}]_\times \mathbf{R}. \tag{21}$$

$\mathbf{E}$ is called the essential matrix; its discovery was published in 1981 by H.C. Longuet-Higgins in *Nature*. It predates by 10 years the discovery of the fundamental matrix!

4.2 Computation

Two algorithms are usually used to compute the $\mathbf{F}$ matrix: the 7-point algorithm and the 8-point algorithm. The 7-point algorithm is used when there are only 7 correspondences, which is the minimal number of correspondences needed to determine a unique $\mathbf{F}$. The 8-point algorithm requires 8 or more correspondences. In the rectification we will mention later, the 8-point algorithm is used because there are usually more than 8 correspondences.

4.2.1 The 7-point algorithm

The equation $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$ gives us one linear equation in the unknown entries of $\mathbf{F}$. More explicitly, given $\mathbf{x} = (x, y, 1)^T$, $\mathbf{x}' = (x', y', 1)^T$, write $\mathbf{f} = (\mathbf{F}_{11}, \mathbf{F}_{12}, \mathbf{F}_{13}, \mathbf{F}_{21}, \mathbf{F}_{22}, \mathbf{F}_{23}, \mathbf{F}_{31}, \mathbf{F}_{32}, \mathbf{F}_{33})^T$ the vector made up of entries of $\mathbf{F}$ in row-major order. Then the equation can be written as:

$$\left(x'x, x'y, x', y'x, y', x, y, 1\right)\mathbf{f} = 0 \tag{22}$$

If we have n image point correspondences, the n linear equations can be stacked in a linear system:

$$\mathbf{A}\mathbf{f} = \mathbf{0} \tag{23}$$

with the $\mathbf{A}$ the $n \times 9$ coefficient matrix.

Since $\mathbf{F}$ has 7 degrees of freedom, 7 correspondences are enough to compute $\mathbf{F}$. In this minimum case (7 correspondences), the solution space has dimension 2 of the form $\lambda_1\mathbf{F}_1 + \lambda_2\mathbf{F}_2$ with $\mathbf{F}_1$ and $\mathbf{F}_2$ corresponding to two independent null vectors of $\mathbf{A}$. Since $\mathbf{F}$ is a homogeneous entity, the solution does not change by multiplying a non-zero scalar, for example, $1/(\lambda_1 + \lambda_2)$. So the solution becomes $\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$ ($\alpha = \lambda_1/(\lambda_1 + \lambda_2)$).

Remember that $\det(\mathbf{F}) = \det\left(\alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2\right) = 0$ since $\mathbf{F}$ has rank 2. This leads to a cubic polynomial equation in α. It has either 3 real solutions or 1 real solution and 2 complex conjugate solutions. Only the real solutions make sense for $\mathbf{F}$. So $\mathbf{F} = \alpha\mathbf{F}_1 + (1-\alpha)\mathbf{F}_2$ has 1 or 3 possible solutions. The geometric interpretation of 1 or 3 solutions is in chapter 22 of [6]. From the point of view of critical surfaces, the seven points and two camera centers must lie on a quadric surface (since 9 points lie on a quadric). If this quadric is ruled, then there will be three solutions. On the other hand, if it is not ruled quadric (for instance an ellipsoid) then there will be only one solution. The 7-point algorithm requires the minimum number of correspondences to solve $\mathbf{F}$ and it is often integrated in the outlier detection algorithm to extract the good $\mathbf{F}$ like [1, 10].

4.2.2 The 8-point algorithm

The 8-point algorithm is a simpler computation method of $\mathbf{F}$. In this case, the solution space has only dimension 1 and the $\mathbf{F}$ is uniquely determined up to scale. But because of noise in point coordinates, $\det(\mathbf{F})$ is not equal to 0. The convenient method to enforce the determinant constraint is to use the SVD (Singular Value Decomposition, see Appendix B) and to replace the smallest singular value by 0. The obtained $\mathbf{F}'$ is optimal in the sense of minimizing the Frobenius norm $\|\mathbf{F} - \mathbf{F}'\|$. A key point of the 8-point algorithm is the normalization which makes the points more concentrated around their centroid. In Hartley's original paper [4], he proposed to translate the points so that the origin is at the centroid of the points, and to scale the points so that the average distance from the points to their centroid is equal to $\sqrt{2}$. This normalization improves dramatically the conditioning of the coefficient matrix $\mathbf{A}$ in (22) and make all entries of $\mathbf{F}$ contribute approximately equally to the error term.

Algorithm 1 (8-point normalization algorithm)**.**
Objective Given $n \geqslant 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix $\mathbf{F}$ such that $\mathbf{x}_i'^T \mathbf{F} \mathbf{x}_i = 0$

1. ***Normalization**: Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathbf{T}'\mathbf{x}'_i$, where $\mathbf{T}$ and $\mathbf{T}'$ are normalization transformations consisting of a translation and scaling:*

$$\mathbf{T} = \begin{pmatrix} 1/\alpha & 0 & -u/\alpha \\ 0 & 1/\alpha & -v/\alpha \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{T}' = \begin{pmatrix} 1/\alpha' & 0 & -u'/\alpha' \\ 0 & 1/\alpha' & -v'/\alpha' \\ 0 & 0 & 1 \end{pmatrix}$$

   with $(u, v)^T$ and $(u', v')^T$ the centroid of the points in the first and second image respectively; $1/\alpha$ and $1/\alpha'$ the scale to make average distance from the points to the centroid equal to $\sqrt{2}$.

2. *Find the fundamental matrix $\hat{\mathbf{F}}$ corresponding to the matches $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$ by*

   (a) ***Linear solution**: Determine $\hat{\mathbf{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathbf{A}}$, where $\hat{\mathbf{A}}$ is the coefficient matrix composed from the matches $\{\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i\}$.*

   (b) ***Constraint enforcement:** Replace $\hat{\mathbf{F}}$ by $\hat{\mathbf{F}}'$ such that $det(\hat{\mathbf{F}}') = 0$ by setting the smallest singular value to be 0.*

3. ***Denormalization:** Set $\mathbf{F} = \mathbf{T}'^T \hat{\mathbf{F}}' \mathbf{T}$. Matrix $\mathbf{F}$ is the fundamental matrix corresponding to the original data $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$.*

5 Rectification

This section is dedicated to image rectification, the process of re-sampling pairs of stereo images in order to produce a pair of "matched epipolar" projections. The rectification makes the corresponding epipolar lines coincide and be parallel to the x-axis. Consequently, the disparity between two images is only in the x-direction. A pair of stereo-rectified images is helpful for dense stereo matching algorithms. It restricts the search domain for each match to a line parallel to the x-axis.

5.1 Special form of F

In geometric view, the rectification is achieved when both cameras have their image planes coplanar and the x-axis of the image planes parallel to the baseline. This means that the motion between both cameras is a pure translation with no rotation. One can assume that the rectified camera matrices are: $\mathbf{P} = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = \mathbf{K}[\mathbf{I} \mid -\lambda\mathbf{i}]$ with $\mathbf{i} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$

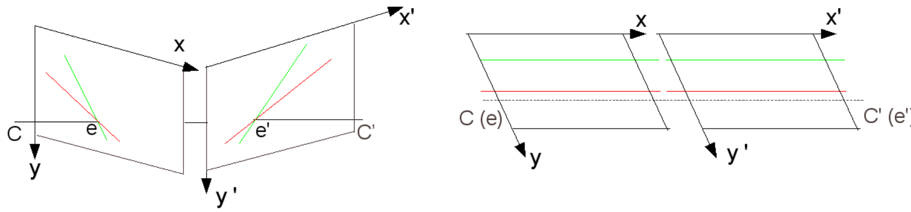

Figure 5: Rectification illustration. Left: original cameras configuration. Right: Camera configuration after rectification. Image planes are coplanar and their x-axis is parallel to the baseline $\mathbf{CC}'$.

(λ is the distance between the camera centers, called the baseline). Using (19) with $\mathbf{R} = \mathbf{I}$ and $\mathbf{T} = -\lambda\mathbf{i}$ we get

$$\mathbf{F} = -\lambda \mathbf{K}^{-T}[\mathbf{i}]_\times \mathbf{K}^{-1} \tag{24}$$

and after simplification (and ignoring the scale $-\lambda$)

$$\mathbf{F} = [\mathbf{i}]_\times \tag{25}$$

Putting this special fundamental matrix in (12), we have $y = y'$, that is, the epipolar lines are corresponding raster lines and the disparity is in the x-direction.

Exercice 7. *Verify that (25) is a consequence of (24) using the fact that $\mathbf{K}^{-1}$ is upper-triangular. Hint: a smarter way than carrying out the multiplications is observing that a) $\mathbf{Fi} = 0$ and b) $\mathbf{F}$ is anti-symmetric.*

5.2 Invariance

Note that the solution to the rectification is not unique. Once the rectification is achieved, we can rotate two cameras together around the baseline and the resulting images remain rectified. But the introduced projective distortion is not different. The ideal is to achieve the rectification by introducing a projective distortion as small as possible.

Exercice 8. *Prove this by computing $(\mathbf{KRK}^{-1})^T[\mathbf{i}]_\times(\mathbf{KRK}^{-1})$ when $R = \begin{pmatrix} 1 & 0 \\ 0 & \hat{\mathbf{R}} \end{pmatrix}$ with $\hat{\mathbf{R}}$ a 2×2 rotation matrix.*

6 Quasi-Euclidean rectification

Since the rectification can be achieved by rotating two cameras without changing their camera center, it is equivalent to apply a pair of homographies on two images respectively. This is the idea used in many different rectification algorithms. The problem is how to find this pair of homographies which introduce a small projective distortion. In [5], the authors first rectify one image and find another "matched" homography to rectify the other image. The distortion is reduced by imposing that one homography is approximately rigid around one point and by minimizing the

x-disparity between both rectified images. In [7], the distortion reduction is improved by decomposing the homographies into three components: homograhy, similarity and shear. A projective transformation is sought, as affine as possible to reduce the projective distortion, but the affine distortion is not treated. In [3], the distortion is interpreted as local loss or creation of pixels in the rectified images. Thus the local area change in the rectified images is minimized. A similar idea is exposed in [9], whose solution is a homography that can be locally well approximated by a rigid transformation through the whole image domain. Quasi-Euclidean rectification [2] proposed a new parametrization of the fundamental matrix based on two rectification homographies to fit the feature correspondences. The rectification problem is formulated as a 6-parameter non-linear minimization problem. This method is very compact but no special attention is paid to the distortion reduction.

6.1 Parameterization of the problem

Given pairs of correspondences $(\mathbf{x}_l, \mathbf{x}_r)$, we have $\mathbf{x}_l^T \mathbf{F} \mathbf{x}_r = 0$. Once the rectification is achieved, we have:

$$\hat{\mathbf{x}}_r^T \mathbf{F}_0 \hat{\mathbf{x}}_l = 0 \tag{26}$$

with

$$\mathbf{F}_0 = [\mathbf{i}]_\times , \tag{27}$$

and $\hat{\mathbf{x}}_l$ and $\hat{\mathbf{x}}_r$ are obtained by applying a pair of rectification homography $(\mathbf{H}_l, \mathbf{H}_r)$ on $(\mathbf{x}_l, \mathbf{x}_r)$ respectively:

$$\hat{\mathbf{x}}_l = \mathbf{H}_l \mathbf{x}_l \tag{28}$$

$$\hat{\mathbf{x}}_r = \mathbf{H}_r \mathbf{x}_r \tag{29}$$

So the original fundamental matrix can also be written as:

$$\mathbf{F} = \mathbf{H}_l^T \, [\mathbf{i}]_\times \, \mathbf{H}_r \tag{30}$$

According to (11), $\mathbf{H}_l$ and $\mathbf{H}_r$ can be parametrized as:

$$\mathbf{H}_l = \mathbf{K}_{nl} \mathbf{R}_l \mathbf{K}_{ol}^{-1} \tag{31}$$

$$\mathbf{H}_r = \mathbf{K}_{nr} \mathbf{R}_r \mathbf{K}_{or}^{-1} \tag{32}$$

with $(\mathbf{K}_{ol}, \mathbf{K}_{or})$ the old calibration matrices, $(\mathbf{K}_{nl}, \mathbf{K}_{nr})$ the new calibration matrices and $(\mathbf{R}_l, \mathbf{R}_l)$ the rotations performed in the rectification of left and right cameras respectively. The old calibration matrix and the rotation are unknown, while the new calibration matrix can be set arbitrarily, provided the second row of $\mathbf{K}_{nl}$ is equal to the second row of $\mathbf{K}_{nr}$. Indeed, under this condition, it can be easily checked that $\mathbf{K}_{nl}^T \, [\mathbf{i}]_\times \, \mathbf{K}_{nr} = [\mathbf{i}]_\times$. Consequently the original fundamental matrix $\mathbf{F}$ can be parameterized as:

$$\mathbf{F} = \mathbf{K}_{ol}^{-T} \mathbf{R}_l^T \, [\mathbf{i}]_\times \, \mathbf{R}_r \mathbf{K}_{or}^{-1} \tag{33}$$

The unknown parameters include 3 rotation angles in $\mathbf{R}_l$, 3 rotation angles in $\mathbf{R}_r$, 5 parameters in $\mathbf{K}_{ol}$ and 5 parameters in $\mathbf{K}_{or}$. Since the rectification is invariant to a common rotation of two cameras around the baseline, one degree of freedom can be eliminated. In [2], the rotation around the X-axis of the left camera is fixed to be zero. The unknown parameters are further reduced by assuming that the calibration matrix are the same for two cameras with zero skewness, aspect ratio equal to 1 and principal point at the center of image.

$$\mathbf{K}_{ol} = \mathbf{K}_{or} = \begin{pmatrix} \alpha & 0 & w/2 \\ 0 & \alpha & h/2 \\ 0 & 0 & 1 \end{pmatrix} \tag{34}$$

where w and h are the image width and height respectively. So for the calibration matrix, there is only the focal length α to be estimated. Since 5 parameters are left for the rotations, there are overall 6 parameters to be estimated.

6.2 Solution

The decomposition of $\mathbf{F}$ in the form of (33) with constraint (34) may be difficult. Instead of that, [2] does not explicitly compute $\mathbf{F}$ from point correspondences but estimates the 6 parameters so as to minimize the distance from points to their epipolar line:

$$\begin{aligned} \min \sum_i \big(& d^2(\mathbf{K}_o \mathbf{R_l} \mathbf{K}_o^{-1} \mathbf{x}_{li}, [\mathbf{i}]_\times \mathbf{K}_o \mathbf{R_r} \mathbf{K}_o^{-1} \mathbf{x}_{ri}) + \\ & d^2(\mathbf{K}_o \mathbf{R_r} \mathbf{K}_o^{-1} \mathbf{x}_{ri}, [\mathbf{i}]_\times \mathbf{K}_o \mathbf{R_l} \mathbf{K}_o^{-1} \mathbf{x}_{li}) \big) \end{aligned} \tag{35}$$

with d^2 the square point-line distance and $\{(\mathbf{x}_{li}, \mathbf{x}_{ri})\}_i$ the set of point correspondences. Instead of using these geometric distances, an algebraic approximation is used, where the term i of the sum is replaced by:

$$E_i^2 = \frac{\big((\mathbf{R}_l \mathbf{K}_0^{-1} x_{li})^T [\mathbf{i}]_\times (\mathbf{R}_r \mathbf{K}_0^{-1} x_{ri})\big)^2}{(\mathbf{K}_0^{-T} \mathbf{R}_r^T [\mathbf{i}]_\times \mathbf{R}_l \mathbf{K}_0^{-1} x_{li})(1:2)^2 + (\mathbf{K}_0^{-T} \mathbf{R}_l^T [\mathbf{i}]_\times \mathbf{R}_r \mathbf{K}_0^{-1} x_{ri})(1:2)^2} \tag{36}$$

The numerator corresponds to the linear constraint and each term of the denominator to the normalization factor of an epipolar line equation.

This problem is a non-linear minimization. An algorithm called Levenberg-Marquardt (see Appendix C) is used to solve it. This requires computing the partial derivatives of E_i with respect to the 6 parameters. Those are expressed as:

$$\frac{\partial E_i}{\partial p} = \frac{\mathbf{x}_{il}^T \mathbf{F}' \mathbf{x}_{ir}}{D} - N \frac{\overline{\mathbf{F}^T \mathbf{x}_{il}}^T \overline{\mathbf{F}'^T \mathbf{x}_{il}} + \overline{\mathbf{F} \mathbf{x}_{ir}}^T \overline{\mathbf{F}' \mathbf{x}_{ir}}}{D^3}, \tag{37}$$

with p any of the parameters, $\overline{\begin{pmatrix} a & b & c \end{pmatrix}^T} = \begin{pmatrix} a & b \end{pmatrix}^T$, and

$$\mathbf{F} = (\mathbf{R}_l \mathbf{K}^{-1})^T [\mathbf{i}]_\times (\mathbf{R}_r \mathbf{K}^{-1}) \qquad \mathbf{F}' = \frac{\partial \mathbf{F}}{\partial p}, \tag{38}$$

$$N = \mathbf{x}_{il}^T \mathbf{F} \mathbf{x}_{ir}, \qquad D = \sqrt{\|\overline{\mathbf{F}^T \mathbf{x}_{il}}\|^2 + \|\overline{\mathbf{F} \mathbf{x}_{ir}}\|^2}. \tag{39}$$

A derivative of $\mathbf{F}$ with respect to one angle parameter is easy to compute, as the angle appears only in one term of the decomposition (33). The left and right rotation matrices are themselves decomposed into rotations along the axes $\mathbf{R} = \mathbf{R}_z\mathbf{R}_y\mathbf{R}_x$ (without the rotation around x for $\mathbf{R}_l$ as explained above) and we have for example:

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \Rightarrow \mathbf{R}'_x(\theta) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\theta & -\cos\theta \\ 0 & \cos\theta & -\sin\theta \end{pmatrix}. \quad (40)$$

Now the 5 angle parameters vary in the range $[-\pi, \pi]$ whereas the focal length α of (34) can take much larger values, typically in the range $[(w+h)/3, (w+h)\times 3]$. Such a disparity in ranges generates a bad condition for the Jacobian matrix J. To avoid that we take as unknown the variable $\beta = \log_3(\frac{\alpha}{w+h})$ and then β is in the range $[-1, 1]$. Deriving with respect to β yields:

$$\frac{\partial \mathbf{F}}{\partial \beta} = (\mathbf{R}_l\mathbf{K}_0^{-'}(\alpha))^T[\mathbf{i}]_\times\mathbf{R}_r\mathbf{K}_0^{-1}(\alpha) + (\mathbf{R}_l\mathbf{K}_0^{-1}(\alpha))^T[\mathbf{i}]_\times\mathbf{R}_r\mathbf{K}_0^{-'}(\alpha) \quad (41)$$

with

$$\mathbf{K}_0^{-1}(\alpha) = \begin{pmatrix} 1/\alpha & 0 & -w/(2\alpha) \\ 0 & 1/\alpha & -h/(2\alpha) \\ 0 & 0 & 1 \end{pmatrix} \text{ and} \quad (42)$$

$$\mathbf{K}_0^{-'}(\alpha) = -\log 3 \begin{pmatrix} 1/\alpha & 0 & -w/(2\alpha) \\ 0 & 1/\alpha & -h/(2\alpha) \\ 0 & 0 & 0 \end{pmatrix}. \quad (43)$$

6.3 Examples

To perform a rectification, the only required input is more than 7 pairs of correspondences. The SIFT method [8] is used to find the correspondences. All bad correspondences should be filtered out before rectification. This can be achieved by a RANSAC procedure, as for example in [1, 10], searching for a fundamental matrix. A typical example of rectification can be seen in Fig. 6.

7 IPOL projects

Three different algorithms are introduced in this section for IPOL projects. Students can freely choose one or two of them to implement. Hartley's and Loop's methods are two of the earliest rectification algorithms, while Gluckman's method has a different idea to treat the distortion introduced by rectification. More details about the *fundamental matrix*, the multi-view geometry and the rectification can be found in the book [6].

7.1 Hartley's method

Hartley's method [5] is one of the earliest methods of image rectification. Unlike quasi-Euclidean rectification which finds two rectification homographies together, Hartley's method first computes one homography to

Figure 6: Rectification example. Top: a pair of original images. Bottom: the rectified images, with some epipolar lines and matching points marked as crosses

rectify one image, then finds the other "matching" homography under epipolar geometry constraints.

It is well-known that for a pair of rectified images, the epipoles are at $(1,0,0)^T$ because the rectified fundamental matrix in (27) has $(1,0,0)^T$ as its left and right null-vector. But this is only a necessary condition, because one can always find a homography which sends the epipole in the left image to $(1,0,0)^T$, and independently the other homography which sends the epipole in the right image to $(1,0,0)^T$. Then the epipolar lines in each image will be horizontal, but one epipolar line in one image will not necessarily have the same ordinate as its corresponding epipolar line in the other image. So the images are not rectified, due to the independence between two homographies.

In [5], a pair of matching homographies $(\mathbf{H}, \mathbf{H}')$ is defined by

$$\mathbf{H}^* \mathbf{l} = \mathbf{H}'^* \mathbf{l}' \tag{44}$$

with $(\mathbf{l}, \mathbf{l}')$ a pair of corresponding epipolar lines before rectification; $\mathbf{H}^*$ and $\mathbf{H}'^*$ the line homographies corresponding to the point map $\mathbf{H}$ and $\mathbf{H}'$ respectively (see section 2.3). The following theorem shows the explicit relationship between $\mathbf{H}$ and $\mathbf{H}'$.

Theorem 1. *Let $\mathbf{I}$ and $\mathbf{I}'$ be images with fundamental matrix $\mathbf{F} = [\mathbf{e}']_\times \mathbf{M}$ ($\mathbf{M}$ non-singular matrix), and let $\mathbf{H}'$ be a projective transformation of $\mathbf{I}'$. A projective transformation $\mathbf{H}$ of $\mathbf{I}$ matches $\mathbf{H}'$ if and only if $\mathbf{H}$ is of the form*

$$\mathbf{H} = \left(\mathbf{I} + \mathbf{H}'\mathbf{e}'\mathbf{a}^T\right) \mathbf{H}'\mathbf{M} \tag{45}$$

for some non-zero vector $\mathbf{a}$.

$\mathbf{H}'$ is first computed by sending the epipole $\mathbf{e}'$ to $(1,0,0)^T$. Suppose $\mathbf{x}_0$ is the origin and the epipole $\mathbf{e}' = (f,0,1)^T$ lies on the x-axis. The following transform:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{pmatrix} \tag{46}$$

sends the epipole $\mathbf{e}'$ to infinity $(f, 0, 0)^T$ as required. A point $(x, y, 1)^T$ is mapped by $\mathbf{G}$ to the point $(\hat{x}, \hat{y}, 1)^T = (x, y, 1-x/f)^T = (\frac{x}{1-x/f}, \frac{y}{1-x/f}, 1)^T$. The Jacobian is

$$\frac{\partial(\hat{x}, \hat{y})}{\partial(x, y)} = \begin{pmatrix} (1-x/f)^{-2} & 0 \\ -y(1-x/f)^{-2}/f & (1-x/f)^{-1} \end{pmatrix} \tag{47}$$

Now at the origin $x = y = 0$ this is the identity map. In other words, $\mathbf{G}$ is approximated (to first order) at the origin by the identity mapping. For an arbitrary placed point of interesting $\mathbf{x}_0$ and epipole $\mathbf{e}'$, the required mapping $\mathbf{H}'$ is a product $\mathbf{H}' = \mathbf{GRT}$ where $\mathbf{T}$ is a translation taking the point $\mathbf{x}_0$ to the origin, $\mathbf{R}$ is a rotation about the origin taking the epipole $\mathbf{e}'$ to a point $(f, 0, 1)^T$ on the x-axis, and $\mathbf{G}$ is the mapping just considered taking $(f, 0, 1)^T$ to infinity $(1, 0, 0)^T$. This composite mapping $\mathbf{G}$ is to first order a rigid transformation in the neighborhood of $\mathbf{x}_0$.

According to (45), the homography $\mathbf{H}$ matched to $\mathbf{H}'$ can be written as:

$$\mathbf{H} = \left(\mathbf{I} + \mathbf{H}'\mathbf{e}'\mathbf{a}^T\right)\mathbf{H}'\mathbf{M} = \left(\mathbf{I} + (1, 0, 0)^T\mathbf{a}^T\right)\mathbf{H}'\mathbf{M} = \mathbf{A}\mathbf{H}'\mathbf{M} = \mathbf{A}\mathbf{H}_0 \tag{48}$$

with any non-zero vector $\mathbf{a} = (a, b, c)^T$, $\mathbf{H}_0 = \mathbf{H}'\mathbf{M}$ and

$$\mathbf{A} = \begin{pmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{49}$$

which represents an affine transformation. The unknown parameters a, b, c can be determined by minimizing the disparity. Writing $\hat{\mathbf{x}}'_i = \mathbf{H}'\mathbf{u}'_i$ and $\hat{\mathbf{x}}_i = \mathbf{H}_0\mathbf{u}_i$, the minimization problem is to find $\mathbf{A}$ such that

$$\sum_i d(\mathbf{A}\hat{\mathbf{x}}_i, \hat{\mathbf{x}}'_i)^2 \tag{50}$$

is minimized. Let $\hat{\mathbf{x}}_i = (\hat{x}_i, \hat{y}_i, 1)^T$ and $\hat{\mathbf{x}}'_i = (\hat{x}'_i, \hat{y}'_i, 1)^T$. Since $\mathbf{H}'$ and $\mathbf{M}$ are known, these vectors can be computed from the matched points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. Then the minimization problem becomes:

$$\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}'_i)^2 + (\hat{y}_i - \hat{y}'_i)^2. \tag{51}$$

Since $(\hat{y}_i - \hat{y}'_i)^2$ is a constant, this is equivalent to minimizing

$$\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}'_i)^2 \tag{52}$$

This is linear least-square problem and can be easily solved. Once a, b, c are computed, $\mathbf{A}$ and $\mathbf{H}$ can also be computed from (49) and (48).

Hartley's method does not pay special attention to the projective distortion introduced by rectification because the first homography $\mathbf{H}'$ is only approximated by a rigid transform to first order around a chosen point.

7.2 Gluckman's method

Gluckman's method [3] pays attention to the distortion. It interprets the distortion as the resampling that can impede stereo matching. The effects they seek to minimize are the loss of pixels due to under-sampling and the creation of new pixels due to over-sampling. To minimize these effects, they parameterize the family of rectification transformations and solve for the one that minimizes the change in local area integrated over the area of the images.

The change in local area is given by the determinant of the Jacobian of the homography. If the determinant is smaller than one, the local area decreases which corresponds to a local down-sampling; for values greater than unity, the local area increases which corresponds to a local over-sampling. Given a homography $\mathbf{H}$:

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \tag{53}$$

A point $(x, y)^T$ is mapped to the point $(\hat{x}, \hat{y})^T$:

$$\hat{x} = \frac{h_1 x + h_2 y + h_3}{h_7 x + h_8 y + h_9} \quad \text{and } \hat{y} = \frac{h_4 x + h_5 y + h_6}{h_7 x + h_8 y + h_9} \tag{54}$$

The Jacobian $\mathbf{J}$ is obtained by taking the partial derivatives of the above equations with respect to x and y:

$$\mathbf{J}(x, y) = \frac{\partial(\hat{x}, \hat{y})}{\partial(x, y)} = \begin{pmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{pmatrix} \tag{55}$$

Then the determinant of the Jacobian of the homography $\det(\mathbf{J}(\mathbf{x}, \mathbf{y}))$ is:

$$\frac{h_9(h_5 h_1 - h_4 h_2) + h_8(h_4 h_3 - h_1 h_6) + h_7(h_2 h_6 - h_5 h_3)}{(h_7 x + h_8 y + h_9)^3} \tag{56}$$

Since the Jacobian $\mathbf{J}(x, y)$ is a function of x and y, the determinant can be less than one in some places and greater than one in others. To find the homography which perturbs the least the local area over the entire image, they choose the metric as the square of the difference between the determinant and one and integrate over the width w and the height h of the image:

$$\int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{w}{2}}^{\frac{w}{2}} \left(\det\left(\mathbf{J}(x, y)\right) - 1\right)^2 dx dy \tag{57}$$

Assume the origin of the coordinate system at the center of each image with the x-axis running along the scan-lines. A rotation and translation have been applied so that a pair of corresponding epipolar lines coincides with the x-axis. Since both epipoles are on the x-axis, the fundamental matrix must be of the form

$$\mathbf{F} = \begin{pmatrix} 0 & f_2 & 0 \\ f_4 & f_5 & f_6 \\ 0 & f_8 & 0 \end{pmatrix} \tag{58}$$

The epipoles can be computed as the left and right null-vectors of $\mathbf{F}$: $\mathbf{e} = \left(1, 0, -\frac{f_4}{f_6}\right)^T$ and $\mathbf{e}' = \left(1, 0, -\frac{f_2}{f_8}\right)^T$. To project $\mathbf{e}$ and $\mathbf{e}'$ to infinity $\mathbf{i} = (1, 0, 0)^T$, the homography $\mathbf{H}$ and $\mathbf{H}'$ must be of the form

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & 0 \\ 0 & h_5 & 0 \\ \frac{f_4}{f_6} & h_8 & 1 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} h'_1 & h'_2 & 0 \\ 0 & h'_5 & 0 \\ \frac{f_2}{f_8} & h'_8 & 1 \end{pmatrix} \quad (59)$$

The third column of $\mathbf{H}$ and $\mathbf{H}'$ are both zero because the origin is assumed to be unchanged under rectification.

Although these conditions ensure that the epipolar lines are parallel, corresponding epipolar lines may lie on different scan-lines. Thus additional constraints should be derived from the equation $\mathbf{F} = \mathbf{H}'^T [\mathbf{i}]_\times \mathbf{H}$, which is the necessary and sufficient condition for rectification. Two independent constraints are obtained: $-f_8 h'_5 = f_6 p_5$ and $h'_8 f_8 + h_8 f_6 = f_5$, that will be satisfied when

$$\mathbf{H} = \begin{pmatrix} h_1 & h_2 & 0 \\ 0 & h_5 & 0 \\ f_4 & p_8 & f_6 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} h'_1 & h'_2 & 0 \\ 0 & -h_5 & 0 \\ f_2 & f_5 - p_8 & f_8 \end{pmatrix}. \quad (60)$$

The unknown parameters are reduced by imposing that the skew of both images is zero ($h_2 = h'_2 = 0$) and the aspect ratio does not change ($h_5 = h_1$ and $h'_1 = h'_5$). So finally there are two free parameters, h_1 and h_8:

$$\mathbf{H} = \begin{pmatrix} h_1 & 0 & 0 \\ 0 & h_1 & 0 \\ f_4 & h_8 & f_6 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} -h_1 & 0 & 0 \\ 0 & -h_1 & 0 \\ f_2 & f_5 - h_8 & f_8 \end{pmatrix}. \quad (61)$$

¿From (57), the error metric when two homographies $\mathbf{H}$ and $\mathbf{H}'$ are applied to a pair of images $\mathbf{I}$ and $\mathbf{I}'$ is

$$\begin{aligned} \varepsilon &= \varepsilon_1 + \varepsilon_2 = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{w}{2}}^{\frac{w}{2}} \left(\det\left(\frac{\partial \mathbf{H}(x,y)}{\partial (x,y)} \right) - 1 \right)^2 dxdy \\ &+ \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{-\frac{w}{2}}^{\frac{w}{2}} \left(\det\left(\frac{\partial \mathbf{H}'(x,y)}{\partial (x,y)} \right) - 1 \right)^2 dxdy \end{aligned} \quad (62)$$

The integral is a 16^{th} degree rational polynomial in h_8 and is quadratic in h_1. Therefore, given a value of h_8 the value of h_1 that minimizes the integral can be found by solving $\frac{\partial \varepsilon}{\partial h_1} = 0$.

Before solving for the optimal value of h_1, an optimal value for h_8 must be obtained. By considering h_1 a known parameter, the error metric is a function of the location of the epipoles and h_8. This function is in fact convex in the range 0 to $\frac{f_5}{2}$. Therefore, an optimal solution can be found using $h_8 = \frac{f_5}{2}$ as an initial estimate and applying a simple iterative technique such as gradient descent. Once h_8 is solved, h_1 can be obtained explicitly. Finally $\mathbf{H}$ and $\mathbf{H}'$ are obtained from (61).

7.3 Loop's method

Another very popular method is proposed by Loop and Zhang [7]. This method decomposes the homography into three parts: a projective transform, a similarity transform and a shear transform. The distortion is minimized at each step.

Assume a pair of rectifying homography $\mathbf{H}$ and $\mathbf{H}'$:

$$\mathbf{H} = \begin{pmatrix} \mathbf{u}^T \\ \mathbf{v}^T \\ \mathbf{w}^T \end{pmatrix} = \begin{pmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & 1 \end{pmatrix} \text{ and } \mathbf{H}' = \begin{pmatrix} \mathbf{u}'^T \\ \mathbf{v}'^T \\ \mathbf{w}'^T \end{pmatrix} = \begin{pmatrix} u'_a & u'_b & u'_c \\ v'_a & v'_b & v'_c \\ w'_a & w'_b & 1 \end{pmatrix} \tag{63}$$

A necessary condition for rectification is that $\mathbf{He} = \mathbf{i}$ and $\mathbf{H}'\mathbf{e}' = \mathbf{i}$. This implies that $\mathbf{v}^T\mathbf{e} = 0$ and $\mathbf{w}^T\mathbf{e} = 0$, which means $\mathbf{w}$ and $\mathbf{v}$ are both epipolar lines for image $\mathbf{I}$. In fact, it can be proven that $(\mathbf{v}, \mathbf{v}')$ and $(\mathbf{w}, \mathbf{w}')$ are corresponding epipolar lines.

$\mathbf{H}$ (similarly for $\mathbf{H}'$) is decomposed into a projective transform $\mathbf{H}_p$, a similarity transform $\mathbf{H}_r$ and a shearing transform $\mathbf{H}_s$.

$$\mathbf{H}_p = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ w_a & w_b & 1 \end{pmatrix} \tag{64}$$

$$\mathbf{H}_r = \begin{pmatrix} v_b - v_c w_b & v_c w_a - v_a & 0 \\ v_a - v_c w_a & v_b - v_c w_b & v_c \\ 0 & 0 & 1 \end{pmatrix} \tag{65}$$

$$\mathbf{H}_s = \begin{pmatrix} s_a & s_b & s_c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{66}$$

7.3.1 Projective transform

The projective transformation $\mathbf{H}_p$ (and $\mathbf{H}'_p$) will send the epipole $\mathbf{e}$ (or $\mathbf{e}'$) to infinity because $\mathbf{w}^T\mathbf{e} = 0$ (or $\mathbf{w}'^T\mathbf{e}' = 0$). $\mathbf{w}$ and $\mathbf{w}'$ are not independent since they are corresponding epipolar lines. So they can be parametrized by a direction vector $z = (\lambda, \mu, 0)^T$ in image $\mathbf{I}$:

$$\mathbf{w} = [\mathbf{e}]_\times \mathbf{z} \text{ and } \mathbf{w}' = \mathbf{Fz}. \tag{67}$$

And the aim is to find the vector $\mathbf{z}$ which introduces the least projective distortion.

For a set of points in image $\mathbf{p}_i = (p_{i,u}, p_{i,v}, 1)^T$, they will be transformed by $\mathbf{H}_p$ to points $(p_{i,u}/w_i, p_{i,v}/w_i, 1)^T$ with weight $w_i = \mathbf{w}^T\mathbf{p}_i$. The projective transformation $\mathbf{H}_p$ is reduced to an affine transformation if all the weights are identical. But in order to map the epipole $\mathbf{e}$ to infinity, $\mathbf{H}_p$ cannot be affine, except if $\mathbf{e}$ is already at infinity. So what can be done is to make $\mathbf{H}_p$ as affine as possible. Based on the idea, the variation of the weights assigned to a collection of points over both images should be minimized. Over one image, the variation of the assigned weights:

$$\sum_i \left(\frac{w_i - w_c}{w_c} \right)^2 \tag{68}$$

with $w_c = \mathbf{w}^T\mathbf{p}_c$ where $\mathbf{p}_c = \frac{1}{n}\sum_{i=1}^{n}\mathbf{p}_i$ is the average of the points. (68) can also be written as:

$$\sum_i \left(\frac{\mathbf{w}^T(\mathbf{p}_i - \mathbf{p}_c)}{\mathbf{w}^T\mathbf{p}_c}\right)^2 = \frac{\mathbf{w}^T\mathbf{P}\mathbf{P}^T\mathbf{w}}{\mathbf{w}^T\mathbf{p}_c\mathbf{p}_c^T\mathbf{w}} \tag{69}$$

with $\mathbf{P}$ the $3 \times n$ matrix:

$$\mathbf{P} = \begin{bmatrix} p_{1,u} - p_{c,u} & p_{2,u} - p_{c,u} & \dots & p_{n,u} - p_{c,u} \\ p_{1,v} - p_{c,v} & p_{2,v} - p_{c,v} & \dots & p_{n,v} - p_{c,v} \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{70}$$

The variation of weights for the other images can be similarly defined with $\mathbf{p}'_c$ and $\mathbf{P}'$. With $\mathbf{w}$ and $\mathbf{w}'$ replaced by $\mathbf{w} = [\mathbf{e}]_\times\mathbf{z}$ and $\mathbf{w}' = \mathbf{F}\mathbf{z}$, the variation of weights of two images should be minimized together:

$$\begin{aligned} &\min \quad \frac{\mathbf{w}^T\mathbf{P}\mathbf{P}^T\mathbf{w}}{\mathbf{w}^T\mathbf{p}_c\mathbf{p}_c^T\mathbf{w}} + \frac{\mathbf{w}'^T\mathbf{P}'\mathbf{P}'^T\mathbf{w}'}{\mathbf{w}'^T\mathbf{p}'_c\mathbf{p}'^T_c\mathbf{w}'} \\ \iff &\min \quad \frac{\mathbf{z}^T \overbrace{[\mathbf{e}]_\times^T\mathbf{P}\mathbf{P}^T[\mathbf{e}]_\times}^{A} \mathbf{z}}{\mathbf{z}^T \underbrace{[\mathbf{e}]_\times^T\mathbf{p}_c\mathbf{p}_c^T[\mathbf{e}]_\times}_{B} \mathbf{z}} + \frac{\mathbf{z}^T \overbrace{[\mathbf{F}]_\times^T\mathbf{P}'\mathbf{P}'^T[\mathbf{F}]_\times}^{A'} \mathbf{z}}{\mathbf{z}^T \underbrace{[\mathbf{F}]_\times^T\mathbf{p}'_c\mathbf{p}'^T_c[\mathbf{F}]_\times}_{B'} \mathbf{z}} \end{aligned} \tag{71}$$

Since $\mathbf{z}$ is defined up to scale, we can set $\mu = 1$. Then it can be shown that the minimization problem is in fact to find the root of a 7-order polynomial in λ.

7.3.2 Similarity transform

The first step of projective transformation sends the two epipoles to infinity. $\mathbf{w}$ and $\mathbf{w}'$ are obtained by minimizing the variation of weights over both images. Yet the epipoples are at infinity, but not in the direction $(1,0,0)^T$. In continuation, a pair of similarity transformations $\mathbf{H}_r$ and $\mathbf{H}'_r$ are found to rotate the images so that the epipoles are at the direction $(1,0,0)^T$. v_a and v_b can be eliminated from (65) by making use of the following:

$$\begin{aligned} \mathbf{F} = &\quad \mathbf{H}'^T[\mathbf{i}]_\times\mathbf{H} \\ = &\quad \begin{pmatrix} v_a w'_a - v'_a w_a & v_b w'_a - v'_a w_b & v_c w'_a - va' \\ v_a w'_b - v'_b w_a & v_b w'_b - v'_b w_b & v_c w'_b - v'_b \\ v_a - v'_c w_a & v_b - v'_c w_b & v_c - v'_c \end{pmatrix} \end{aligned} \tag{72}$$

¿From the last row of the matrix, v_a, v_b and v_c can be written as:

$$v_a = \mathbf{F}_{31} + v'_c w_a \tag{73}$$

$$v_b = \mathbf{F}_{32} + v'_c w_b \tag{74}$$

$$v_c = \mathbf{F}_{33} + v'_c \tag{75}$$

(73-75) are substituted into (65) to get the $\mathbf{H}_r$ matrix:

$$\mathbf{H}_r = \begin{pmatrix} \mathbf{F}_{32} - w_b\mathbf{F}_{33} & w_a\mathbf{F}_{33} - \mathbf{F}_{31} & 0 \\ \mathbf{F}_{31} - w_a\mathbf{F}_{33} & \mathbf{F}_{32} - w_b\mathbf{F}_{33} & \mathbf{F}_{33} + v'_c \\ 0 & 0 & 1 \end{pmatrix} \tag{76}$$

In similar manner and using the relation $\mathbf{F}^T = \mathbf{H}^T[\mathbf{i}]_\times \mathbf{H}'$, we can have the $\mathbf{H}'_r$ matrix:

$$\mathbf{H}'_r = \begin{pmatrix} \mathbf{F}_{23} - w'_b\mathbf{F}_{33} & w'_a\mathbf{F}_{33} - \mathbf{F}_{12} & 0 \\ \mathbf{F}_{13} - w'_a\mathbf{F}_{33} & \mathbf{F}_{23} - w'_b\mathbf{F}_{33} & v'_c \\ 0 & 0 & 1 \end{pmatrix} \tag{77}$$

There remains a translation term involving v'_c in (76) and (77). This shows that translation in y-direction is linked between two images and the offset of $\mathbf{F}_{33}$ is needed to align horizontal scan-lines. v'_c is determined so that minimum y-coordinate in either image is zero. The similarity transform will not introduce any projective distortion.

7.3.3 Shearing transform

$\mathbf{H}_r\mathbf{H}_p$ and $\mathbf{H}'_r\mathbf{H}'_p$ are already sufficient to do the rectification. The images remain rectified by a shearing transformation, which does change x-coordinate, but not the y-coordinate. We can also verify that:

$$\mathbf{F} = \mathbf{H}'^T[\mathbf{i}]_\times\mathbf{H} = \mathbf{H}'^T_p\mathbf{H}'^T_r\mathbf{H}'^T_s[\mathbf{i}]_\times\mathbf{H}_s\mathbf{H}_r\mathbf{H}_p = \mathbf{H}'^T_p\mathbf{H}'^T_r[\mathbf{i}]_\times\mathbf{H}_r\mathbf{H}_p \tag{78}$$

The shearing transformation cannot completely undistort the effect of projective transformation. But an appropriate shearing transform can reduce the distortion. The authors choose a shearing transformation which attempts to preserve perpendicularity and aspect ratio of the middle lines. In detail, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ are four middle points of the edges of the image. And $\hat{\mathbf{a}}$, $\hat{\mathbf{b}}$, $\hat{\mathbf{c}}$ and $\hat{\mathbf{d}}$ are the points after the transformation $\mathbf{H}_r\mathbf{H}_p$. With $\mathbf{x} = \hat{\mathbf{b}} - \hat{\mathbf{d}}$ and $\mathbf{y} = \hat{\mathbf{c}} - \hat{\mathbf{a}}$, the desired shearing transformation will preserve the perpendicularity:

$$\mathbf{H}_s\mathbf{x}^T\mathbf{H}_s\mathbf{y} = 0 \tag{79}$$

and the aspect ratio:

$$\frac{(\mathbf{H}_s\mathbf{x})^T(\mathbf{H}_s\mathbf{x})}{(\mathbf{H}_s\mathbf{y})^T(\mathbf{H}_s\mathbf{y})} = \frac{w^2}{h^2} \tag{80}$$

with w and h the image weight and height respectively and $\mathbf{H}_s = \begin{pmatrix} a & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1. \end{pmatrix}$ $\mathbf{H}_s$ does not contain a translation component because it does not reduce the distortion. $\mathbf{H}'_s$ can be computed in a similar manner independently.

The combined transform $\mathbf{H}_s\mathbf{H}_r\mathbf{H}_p$, and similarly $\mathbf{H}'_s\mathbf{H}'_r\mathbf{H}'_p$ rectify images $\mathbf{I}$ and $\mathbf{I}'$ with minimal distortion.

A Cross products

The 3×3 skew-symmetric (anti-symmetric) matrix are very useful in multi-view geometry. If $\mathbf{a} = (a_1, a_2, a_3)^T$ is a 3-vector, then the corresponding skew-symmetric matrix is defined as follows:

$$[\mathbf{a}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{81}$$

The matrix $[\mathbf{a}]_\times$ is singular, and $\mathbf{a}$ is its null-vector(right and left). The cross product is related to skew-symmetric matrix by:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \left(\mathbf{a}^T [\mathbf{b}]_\times\right)^T . \tag{82}$$

If $\mathbf{M}$ is any 3×3 matrix (invertible or not), and $\mathbf{x}$ and $\mathbf{y}$ are column vectors, then

$$(\mathbf{Mx}) \times (\mathbf{My}) = \mathbf{M}^* (\mathbf{x} \times \mathbf{y}) \tag{83}$$

with $\mathbf{M}^*$ the matrix of cofactors of $\mathbf{M}$, satisfying $\mathbf{M}^*\mathbf{M} = \mathtt{det}(\mathbf{M})$. This equation car be written as:

$$[\mathbf{Mx}]_\times \mathbf{M} = \mathbf{M}^* [\mathbf{x}]_\times . \tag{84}$$

Furthermore, for any vector $\mathbf{t}$ and non-singular matrix $\mathbf{M}$, one has:

$$[\mathbf{t}]_\times \mathbf{M} = \mathbf{M}^* \left[\mathbf{M}^{-1}\mathbf{t}\right]_\times = \mathbf{M}^{-T} \left[\mathbf{M}^{-1}\mathbf{t}\right]_\times . \tag{85}$$

The cross product has the important property:

$$(\mathbf{a} \times \mathbf{b})^T \mathbf{c} = |\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}|, \tag{86}$$

(determinant of the matrix composed of the vectors as columns). This is actually the definition of the cross product as the function

$$\mathbf{c} \to \varphi(\mathbf{c}) = |\mathbf{a} \quad \mathbf{b} \quad \mathbf{c}| \tag{87}$$

is a linear form, which can be expressed as a scalar product with a fixed vector, namely $\mathbf{a} \times \mathbf{b}$.

B Singular Value Decomposition (SVD)

The SVD is a very useful decomposition of a matrix generalizing the diagonalization of a symmetric matrix in an orthonormal frame, but valid for any matrix, even rectangular ones.

First, consider a symmetric matrix $\mathbf{A}$. The quadratic form

$$\mathbf{x} \to \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} \tag{88}$$

is continuous and reaches its maximum when restricted on the sphere $\mathbf{x}^T\mathbf{x} = 1$. The Lagrangian of this optimization problem can be written:

$$L(\mathbf{x}, \lambda) = \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - 1) \tag{89}$$

and has all partial derivatives 0 when reaching the maximum at unit vector $\mathbf{x}_1$. Thus we get

$$\mathbf{A}\mathbf{x}_1 = \lambda \mathbf{x}_1 \tag{90}$$

meaning $\mathbf{x}_1$ is an eigenvector of A.

Now if $\mathbf{x}_1^T \mathbf{y} = 0$, we have

$$\mathbf{x}_1^T (\mathbf{A}\mathbf{y}) = (\mathbf{A}\mathbf{x}_1)^T \mathbf{y} = \lambda \mathbf{x}_1^T \mathbf{y} = 0, \tag{91}$$

the first equality using the symmetry of $\mathbf{A}$. This shows that the subspace orthogonal to $\mathbf{x_1}$ is globally preserved by $\mathbf{A}$ and an easy recursion argument shows that we can get an orthonormal basis of eigenvectors of $\mathbf{A}$.

If $\mathbf{B}$ is an $m \times n$ matrix with $m \geq n$, $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ is symmetric and can be written

$$\mathbf{A} = \mathbf{V} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{pmatrix} \mathbf{V}^T \tag{92}$$

with $\mathbf{V}$ an $n \times n$ orthonormal matrix and all diagonal terms are positive since $\mathbf{A}$ is positive. Define $\mathbf{U}_i = \mathbf{B}\mathbf{V}_i / \|\mathbf{B}\mathbf{V}_i\|$ for all i such that $\mathbf{B}\mathbf{V}_i \neq 0$. For two indices i, j, we have

$$(\mathbf{B}\mathbf{V}_i)^T(\mathbf{B}\mathbf{V}_j) = \mathbf{V}_i^T\mathbf{A}\mathbf{V}_j = s_j\mathbf{V}_i^T\mathbf{V}_j = s_j\delta_{ij}. \tag{93}$$

So the $\mathbf{U_i}$ form an orthonormal family, which can be completed to get n orthonormal vectors of $\mathbb{R}^m$ if necessary, as $m \geq n$. Then we have

$$\mathbf{B} = \mathbf{U} \begin{pmatrix} \sqrt{s_1} & & \\ & \ddots & \\ & & \sqrt{s_n} \end{pmatrix} \mathbf{V}^T \tag{94}$$

since the two sides have the same images when applied to the basis $\mathbf{V}_i$. Indeed, calling $\mathbf{C}$ the right hand side of (94), on the one hand we have

$$\mathbf{C}\mathbf{V}_i = \sqrt{s_i}\mathbf{U}_i. \tag{95}$$

On the other hand,

$$\|\mathbf{B}\mathbf{V}_i\|^2 = \mathbf{V}_i^T\mathbf{A}\mathbf{V}_i = s_i. \tag{96}$$

Therefore, if $s_i = 0$, we have $\mathbf{C}\mathbf{V}_i = 0 = \mathbf{B}\mathbf{V}_i$. If $s_i \neq 0$, we have

$$\mathbf{C}\mathbf{V}_i = \sqrt{s_i}\frac{\mathbf{B}\mathbf{V}_i}{\|\mathbf{B}\mathbf{V}_i\|} = \mathbf{B}\mathbf{V}_i. \tag{97}$$

This proves that $\mathbf{B} = \mathbf{C}$.

Equation (94) is called the SVD of $\mathbf{B}$. U is an $m \times n$ matrix with orthonormal columns, V is an $n \times n$ rotation matrix and the $\sqrt{s_i}$ are called the singular values of B.

Exercice 9. *Show that the dimension of the kernel of* $\mathbf{B}$ *is the number of* s_i *that are* 0 *and that a basis of the kernel is formed by the corresponding* $\mathbf{V}_i$.

C Levenberg-Marquardt algorithm

Suppose a function $\mathbf{X} = \mathbf{f}(\mathbf{P})$ where $\mathbf{X}$ is a measurement vector and $\mathbf{P}$ is a parameter vector in $\mathbb{R}^N$ and $\mathbb{R}^M$ respectively. We want to find the vector $\hat{\mathbf{P}}$ satisfying $\mathbf{X} = \mathbf{f}(\hat{\mathbf{P}}) - \epsilon$ for which ϵ is minimized. If $\mathbf{f}$ is a linear function, this problem is a linear least-square problem. If $\mathbf{f}$ is not linear, we can start with an initial estimated value $\mathbf{P}_0$ and proceed to refine the estimate under the assumption that the function $\mathbf{f}$ is locally linear.

Let $\epsilon_0 = \mathbf{f}(\mathbf{P}_0) - \mathbf{X}$. We assume that the function $\mathbf{f}$ is approximated by the Jacobian matrix $\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{P}}$. We want to find an update step $\boldsymbol{\Delta}$ to $\mathbf{P}_0$ such that with the updated parameter vector $\mathbf{P}_1 = \mathbf{P}_0 + \boldsymbol{\Delta}$, $\mathbf{f}(\mathbf{P}_1) - \mathbf{X} = \mathbf{f}(\mathbf{P}_0) + \mathbf{J}\boldsymbol{\Delta} - \mathbf{X} = \epsilon_0 + \mathbf{J}\boldsymbol{\Delta}$. This problem is in fact to minimize $\|\epsilon_0 + \mathbf{J}\boldsymbol{\Delta}\|$ over $\boldsymbol{\Delta}$, which is a linear minimization problem. The vector $\boldsymbol{\Delta}$ is solution to the normal equation:

$$\mathbf{J}^T\mathbf{J}\boldsymbol{\Delta} = -\mathbf{J}^T\epsilon_0 \tag{98}$$

This normal equation is in fact used in the Gauss-Newton algorithm. In the Levenberg-Marquardt algorithm, this normal equation is replaced by the augmented normal equations:

$$\left(\mathbf{J}^T\mathbf{J} + \lambda\,\mathrm{diag}(\mathbf{J}^T\mathbf{J})\right)\boldsymbol{\Delta} = -\mathbf{J}^T\epsilon \tag{99}$$

for some value of λ that varies from iteration to iteration and diag is an operator replacing non-diagonal elements of its argument by 0. A typical initial value of λ is 10^{-3}.

If the values of $\boldsymbol{\Delta}$ obtained by solving the augmented normal equations leads to a reduction in the error, then the increment is accepted and λ is divided by a factor (typically 10) before the next iteration. On the other hand if the value leads to an increased error, then λ is multiplied by the same factor and the augmented normal equations are solved again, this process continuing until a value of $\boldsymbol{\Delta}$ is found that gives rise to a decreased error. This process of repeatedly solving the augmented normal equations for different values of λ until an acceptable $\boldsymbol{\Delta}$ is found constitutes one iteration of the LM algorithm. When λ is small, the method is essentially the same as a Gauss-Newton iteration; on the other hand when λ is large, $\boldsymbol{\Delta}$ approaches the value given by the gradient descent. Thus the LM algorithm moves seamlessly between a Gauss-Newton iteration, which will cause rapid convergence in the neighborhood of the solution, and a gradient descent approach, which will guarantee a decrease in the cost function when the progress is difficult. Indeed, when λ becomes larger and larger, the length of the increment step $\boldsymbol{\Delta}$ decreases and eventually leads to a decrease of the cost function.

In practice, the LM algorithm is completely specified by setting the initial value of λ to 10^{-3} and by setting the division or multiplication factor of λ in each iteration to be 10.

References

[1] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(3):381–395, 1981.

[2] A. Fusiello and L. Irsara. Quasi-Euclidean uncalibrated epipolar rectification. *International Conference on Pattern Recognition*, pages 1–4, 2008.

[3] J. Gluckman and S.K. Nayar. Rectifying transformations that minimize resampling effects. *IEEE Conf. Computer Vision and Pattern*, 1:111, 2001.

[4] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[5] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):115–127, 1999.

[6] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[7] C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. *Computer Vision and Pattern Recognition*, 1:125–131, 1999.

[8] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[9] J. Mallon and Paul F. Whelan. Projective rectification from the fundamental matrix. *Image and Vision Computing*, 23:643–650, 2005.

[10] L. Moisan and B. Stival. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, 57(3):201–218, 2004.