HAL
open science

The Polar Epipolar Rectification

François Darmon, Pascal Monasse

To cite this version:

François Darmon, Pascal Monasse. The Polar Epipolar Rectification. Image Processing On Line, 2021, 10.5201/ipol.2021.328 . hal-03157244

HAL Id: hal-03157244

https://enpc.hal.science/hal-03157244v1

Submitted on 3 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

2015/06/16 v0.5.1 IPOL article class

Published in Image Processing On Line on 2021–03–02.
Submitted on 2020–02–02, accepted on 2021–02–05.
ISSN 2105–1232 © 2021 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2021.328

The Polar Epipolar Rectification

François Darmon[1,2], Pascal Monasse[1]

[1] Université Paris-Est, LIGM (UMR CNRS 8049), ENPC, F-77455 Marne-la-Vallée, France
({francois.darmon, pascal.monasse}@enpc.fr)
[2] Thales LAS France

Abstract

Epipolar rectification of a stereo pair is the process of resampling a pair of stereo images so that the apparent motion of corresponding points is horizontal. This is an important preliminary step in depth estimation, substituting depth by disparity estimation. Most methods rely on a perspective transform of both images, which has the advantage to simulate a different attitude of the pinhole cameras. A limitation is that when an epipole is inside the image domain, it has to be sent to infinity by the perspective transform, producing a strong distortion. On the contrary, relying on a polar transform centered at the epipole provides a method applicable universally to a pair of pinhole camera views. We present in detail the algorithm, filling in the information important for its implementation and missing in published articles.

Source Code

The ANSI C++ 03 implementation of the code that we provide is the one which has been peer reviewed and accepted by IPOL. The source code, the code documentation, and the online demo are accessible at the IPOL web part of this article[1]. Compilation and usage instructions are included in the `README.txt` file of the archive.

Keywords: epipolar geometry; stereopsis; polar transform

1 Introduction

1.1 Position of the Problem

One of the key tasks in computer vision is to recover the depth information that is lost during the image acquisition due to two-dimensional projection of the 3-D scene. Though depth is impossible to recover from a single snapshot, a stereo pair with parallax allows in principle to infer that information, following well-studied geometric principles. Assuming the standard pinhole camera model, if the camera attitudes satisfy specific conditions, the apparent motion of points is horizontal between the two images. This has the distinct advantage that the apparent horizontal motion, the disparity, is

[1] https://doi.org/10.5201/ipol.2021.328

inversely proportional to the depth. The recovery of the depth is then translated into a constrained (horizontal displacement) optical flow task. Unless a well calibrated stereo rig is used, such conditions are not fulfilled in practice. Fortunately, it is always possible to change artificially the attitudes and generate *stereo rectified* images. A sufficient number of point correspondences (8 in general) is sufficient to achieve that task.

The relation between corresponding points x_L and x_R in the stereo pair is all summed up in the scalar equation

$$x_L^\top F x_R = 0. \tag{1}$$

Here x_L and x_R are 3-vectors, obtained by taking homogeneous coordinates of the points (appending a 1 as last component to the Euclidean coordinates) and the 3×3 fundamental matrix F depends on the respective camera attitudes and internal parameters. The two views are arbitrarily labelled "left" and "right" images. Inspired by the influential book by Hartley and Zisserman [5], most publications change the order and have x_L on the *right* of F. This unfortunate choice just amounts to change F to its transpose $F^\top$. In the rectified case, F takes the simple form

$$F = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{2}$$

up to some non-zero scale (replacing F by λF in (1) yields the same equation). Applying adequate homographies H_L and H_R (3×3 invertible matrices) to the images yields

$$(H_L x_L)^\top \left(H_L^{-\top} F H_R^{-1}\right) (H_R x_R) = 0, \tag{3}$$

and the resulting correspondences $(H_L x_L, H_R x_R)$ will be in rectified situation if

$$H_L^\top \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} H_R = F, \tag{4}$$

which yields 9 scalar equations on the 18 coefficients of H_L and H_R. While some invariance remains in those equations, since there are families of pairs (H'_L, H'_R) such that

$$H_L'^\top \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} H'_R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{5}$$

and $(H'_L H_L, H'_R H_R)$ will also satisfy (4), there still remain degrees of freedom in choosing the pairs of homographies.

1.2 Previous Work

The best homographies are considered as the ones that distort the least the images. Depending on how this distortion is measured, several algorithms were proposed to find the optimal pairs. Most rely foremost on a first transformation sending the epipoles e_L and e_R to the point at infinity $(1, 0, 0)$. Epipoles represent image points in homogeneous coordinates and can be computed as non-trivial vectors in the null space of $F^\top$ and F

$$F^\top e_L = F e_R = 0. \tag{6}$$

This defines e_L and e_R uniquely up to scale since F has rank 2. All epipolar lines in the left (resp. right) image represented by a vector $F x_R$ (resp. $F^\top x_L$) contain e_L (resp. e_R). Remember that a 3-vector ℓ represents the line of equation $x^\top \ell = 0$.

The method of Hartley [4] achieves the first step by finding homographies that are equal to first order to a rotation-translation in the neighborhood of a given point (typically the image center). Loop and Zhang [7] strive for initial homographies as close as possible to affine, measured by the variance of the third homogeneous coordinate of transformed points of the images. Gluckman and Nayar [3] try instead to adjust as much as possible the determinant of the Jacobian of the homography (as a measure of local resampling area) to the value 1 overall the image. Instead of relying on arbitrary homographies, Fusiello and Irsara [2] assume that the camera intrinsics are known, except the common focal length, allowing to have a quasi-Euclidean rectification process. Notably, the method does not require to compute F but uses directly the point correspondences. A limitation of the method is that both images should come from the same camera with the same settings and in practice the principal point is assumed to be close to the image center (a reasonable assumption unless the images have been cropped). An in-depth analysis of the algorithm with peer-reviewed source code was done by Monasse in [9]. An extension of this method that has less risk of being trapped in a local minimum in the minimization process was proposed by Monasse et al. [10].

Breaking with the homography-based rectification paradigm, an algorithm of Pollefeys et al. [12] has the distinct advantage to not break down when one epipole is inside the image, being applicable in all configurations. It relies on polar transforms around the epipoles. The idea has been reused by Lee et al. [6], simplifying the angular sampling, and by Palander and Brandt [11], using a logarithmic sampling along level lines instead of linear (log-polar transform). The present work revisits the method and gives detailed procedures for its practical implementation.

2 Rectification by Polar Transform

2.1 High-Level Algorithm

The limitation of traditional methods of epipolar rectification is obvious in certain geometric configurations, when one of the epipoles is inside the image. Since rectification involves sending the epipole to infinity, a full straight line in the image is sent to infinity, yielding huge deformations around the line and the two parts of the image separated by this line being mapped to disconnected regions of the Euclidean plane. Still, the line-to-line apparent motion of the points remains valid: whereas the fundamental matrix is generally presented as a point-to-line transform, from $\mathbb{P}^2 \setminus \{e\}$ (e being the epipole), where the vectors are interpreted as points, to $\mathbb{P}^2$, where the vectors are interpreted as line equations, it is also true that

$$F(\lambda x + \mu e_R) = \lambda F x, \tag{7}$$

with $\lambda, \mu \in \mathbb{R}$ and $x \in \mathbb{R}^3 \setminus \{e_R\}$. This shows that any point on the line through x and e_R is mapped on the line on the left image represented by vector Fx. Therefore, pairs of corresponding epipolar lines could be extracted independently, put horizontally, and arranged as wished to yield a correct rectification, meaning that corresponding points move horizontally. This leads to the idea that a polar transform of the left image around e_L and the extraction of corresponding epipolar lines in the right image to construct the right rectified image is appropriate, and could never fail, even when the epipole is inside the image. The high level process of rectification is described in Algorithm 1 and illustrated in Figure 1. While the process looks simple, lots of details must be added.

2.2 Chirality

Taking into account the orientation of lines, the correspondence between epipolar lines can be refined to a correspondence of epipolar *half* lines. Indeed, the epipolar constraint (1) is actually a scalar

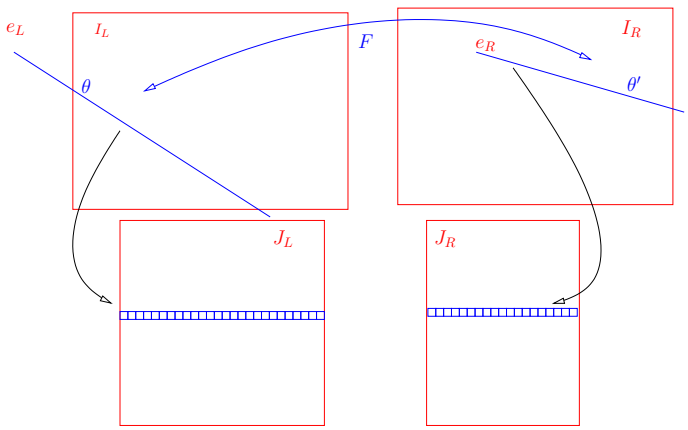


Figure 1: Overview of the rectification process. Here the left epipole is outside the image I_L while the right epipole is inside I_R. Corresponding epipolar lines indexed by θ and $\theta' = \theta'(F, \theta)$ are sampled and copied into the same row of J_L and J_R.

Algorithm 1: High-level polar rectification algorithm.

Input: A stereo pair of images, (I_L, I_R), and their fundamental matrix F, an angular sampling step δ.

Output: Rectified images (J_L, J_R), meaning that any point on J_L has an apparent horizontal motion on J_R.

```
Extract epipoles e_L and e_R from F
for θ_L ∈ [0, 2π] (step δ) do
    Add as new line to J_L the sampling of image I_L along the line issued from e_L at angle θ_L
    Compute angle θ_R of corresponding epipolar line from F and θ_L
    Add as new line to J_R the sampling of image I_R along the line issued from e_R at angle θ_R
```

equation derived from (38) in [8]:

$$e_L \times x_L \overset{+}{\sim} F x_R, \tag{8}$$

meaning proportionality up to a positive factor. Here, e_L, x_L and x_R are vector representations of points in $\mathbb{P}^2$ with a positive third coordinate, typically 1, assuming they are all finite points. The left hand side can be thought as the half line from e_L passing through x_L, that is, oriented from e_L to x_L. The other half line has for equation its opposite, $x_L \times e_L$, oriented from x_L to e_L. This can be used to fix the orientation of F, meaning taking $-F$ instead of F if the positivity in (8) is not satisfied for a pair (x_L, x_R); whereas a single pair could be used for the orientation of F, it should then be valid for all matching pairs. However, since we will be transferring line back and forth between images, we also need to ensure the dual equation

$$e_R \times x_R \overset{+}{\sim} F^\top x_L. \tag{9}$$

It may not be possible to satisfy both (8) and (9) by tuning a single parameter, the sign of F. However, we may play with the signs of e_L and e_R: we keep F as given, but replace e_L by $-e_L$ if (8) is not satisfied, and e_R by $-e_R$ if (9) is not satisfied. In other words, we can have a third coordinate of -1 in e_L or e_R instead of 1 to encode the orientation. In such case, the epipolar line equation $F^T x_L$ and/or $F x_R$ must be negated to get the correct orientation. Taking into account the chirality explains why we can sample along angles θ_L in $[0, 2\pi]$ instead of $[0, \pi]$. The actual computation of oriented vectors e_L and e_R is described in Algorithm 2. Notice that we must have $x_L \neq e_L$ and $x_R \neq e_R$ (actually not even proportional) to be able to orient e_L and e_R.

Algorithm 2: Computation of *oriented* epipoles from F (line 1 of Algorithm 1).

Input: Fundamental matrix F, pair of corresponding points (x_L, x_R).
Output: Epipoles e_L and e_R (3-vectors), such that (8) and (9) are satisfied when x_L and x_R are represented by vectors with 1 as third coordinate.

1 Compute e_L as cross product of two columns of F. `// Algorithm 2 in [8] for details`
2 Compute e_R as cross product of two rows of F.
3 **if** $(e_L \times x_L)^\top F x_R < 0$, *i.e.,* (8) *not satisfied* **then** $e_L \leftarrow -e_L$
4 **if** $(e_R \times x_R)^\top F^\top x_L < 0$, *i.e.,* (9) *not satisfied* **then** $e_R \leftarrow -e_R$

2.3 Transferring Epipolar Lines

Given an epipolar half line at angle θ_L from e_L, the computation of the angle θ_R of the corresponding epipolar half line from e_R (line 4 of Algorithm 1) is recommended by [12] as the application $H\ell_L$ of a homography H to the line equation vector ℓ_L. Such a homography is specified in Appendix A for completeness. However, we will not be using that procedure, as we do not start from a line equation ℓ_L, but from an angle θ_L. It is actually simpler to find a point $x_L = e_L + k(\cos\theta_L, \sin\theta_L, 0)$ with any $k > 0$ on this line, apply F and extract orientation, a procedure detailed in Algorithm 3. The line equation is then

$$\ell_R^1 x + \ell_R^2 y + \ell_R^3 = 0, \tag{10}$$

and a direction vector is $(\ell_R^2, -\ell_R^1)$. Finally, at line 4, we use the standard `C` function `atan2`, where `atan2`(y, x) computes the angle $\theta \in [-\pi, \pi]$ that satisfies both equations

$$\sqrt{x^2+y^2}\cos\theta = x \qquad\qquad \sqrt{x^2+y^2}\sin\theta = y. \tag{11}$$

Any real number $k > 0$ at line 1 selects a point on the epipolar line and is theoretically valid. In practice, the value should not be neither too small nor too large in order to have a well defined point on the line and far enough from the epipole. A good choice is $k = \varrho$, the maximal distance from the epipole to the image domain, defined in Section 2.4.

Transferring in the other direction, from right to left image, works similarly, but the multiplication at line 2 is done with F, not $F^\top$.

Algorithm 3: Computation of θ_R, angle of corresponding epipolar half line (in right image) to the one at angle θ_L in left image.

Input: Fundamental matrix F, its epipoles e_L and e_R, angle θ_L
Output: Corresponding angle θ_R, performing line 4 of Algorithm 1.

1 $p \leftarrow \begin{pmatrix} e_L^1/e_L^3 + k\cos\theta_L & e_L^2/e_L^3 + k\sin\theta_L & 1 \end{pmatrix}^T$ `// any` $k > 0$ `will do, super-indexes are the elements of` e_L
2 $\ell_R \leftarrow F^T p$
3 **if** $e_R^3 < 0$ **then** $\ell_R \leftarrow -\ell_R$
4 $\theta_R \leftarrow \mathrm{atan2}(-\ell_R^1, \ell_R^2)$

2.4 Limitation of Radius and Angular Span

The bounds of sampling of the angle θ_L at line 2 of Algorithm 1 are written as $[0, 2\pi]$, which may be too much, since if the epipole e_L is outside the image, many epipolar half lines will not meet the image at all, producing useless blank lines in the rectified image. Similarly, at lines 3 and 5, the sampling along radius goes from 0 to an unspecified bound, which must actually be precised.

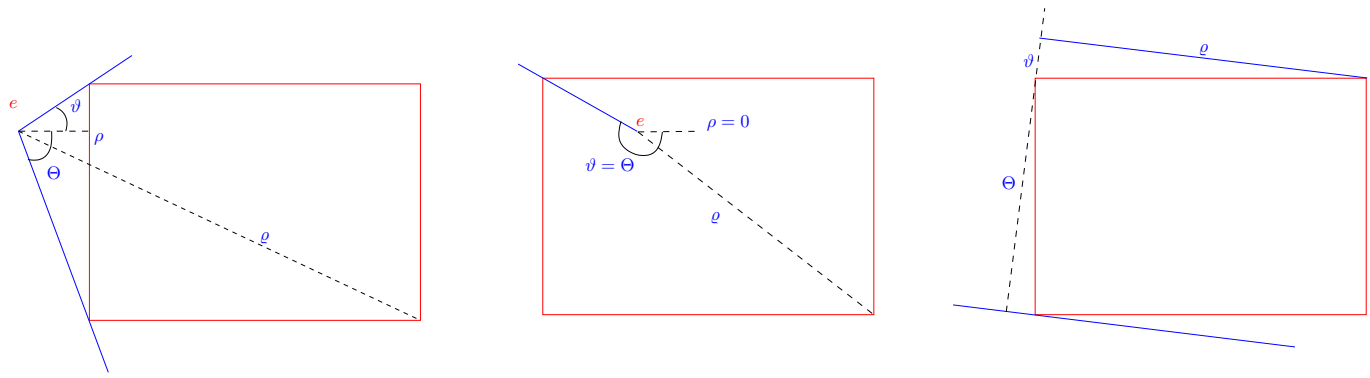


Figure 2: Limits of radius, ρ and ϱ, and limits of θ, ϑ and Θ in different configurations. Left: epipole outside image. Middle: epipole inside image, $\rho = 0$ and $\vartheta = \Theta$. Right: epipole at infinity (here $\rho = 0$); in any case either $\rho = 0$ or $\vartheta = 0$.

The bounds of radius should be the smallest and largest distance from the epipole e to the filled rectangle of the image, $[0, w] \times [0, h]$, with w and h the width and height of the image. The largest distance ϱ, is reached at one of the four corners of the image. The smallest distance ρ is the distance from $e = (e^1, e^2)$ to its projection p on the filled rectangle

$$p = (C_{0,w}(e^1), C_{0,h}(e^2)), \tag{12}$$

where the clamp function is defined as

$$C_{a,b}(x) = \max\left(a, \min(b, x)\right). \tag{13}$$

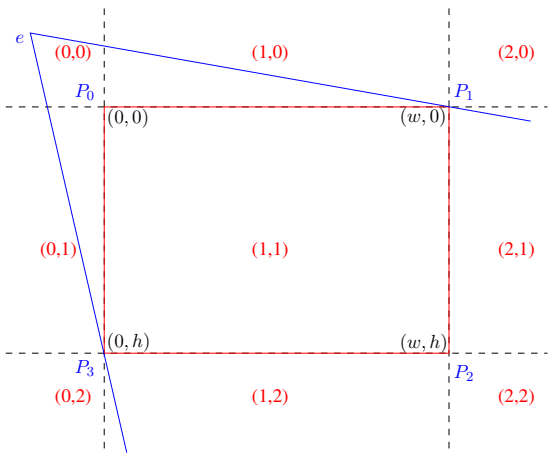
It can be noted that if e is inside the image, $0 \leq e_1 \leq w$ and $0 \leq e_2 \leq h$, we get $p = e$ and the lower bound ρ of the radius is 0, while it is always positive otherwise, see Figure 2.

The minimum ϑ and maximum Θ of angle span obviously a 2π angular interval when e is inside the image, whereas they are given by the angles from e to two of the four corners when e is outside the image. Which of these corners depends on the region of e with respect to the image area and is summarized in Figure 3. Algorithm 4 sums up the computation of the bounds. This is applied independently to both images of the stereo pair with their respective epipole. Notice that line 6 seems useless, as ρ was already set and can never be improved. The explanation for its presence will become clearer when we consider the possibility of e at infinity. The relative position of the corner with respect to the image is encoded by a vector in $\{0, 1, 2\}^2$, found by applying the function $R_{0,w}$ and $R_{0,h}$ to the coordinates

$$R_{a,b}(x) = \begin{cases} 0 & \text{if } x < a, \\ 2 & \text{if } x > b, \\ 1 & \text{otherwise.} \end{cases} \tag{14}$$

Then the two extreme corners determining the bounds of θ are looked up in table $\mathcal{T}$ of Figure 3. Notice that for e inside the image, the "extreme" corners are identical, P_0, so that $\vartheta = \Theta$, which must be interpreted as the full circle and thus we add 2π to Θ.

Actually, the angular bounds can be restricted by computing the common visible area of the two images: the extreme half lines in the right image (joining the epipole to the extreme corners) can be transferred to the left image, defining an angular interval, whose bounds are the mapped extreme angles, see Figure 4. As shown in Appendix B, the correct interval has length less than π, and the intersection can be computed according to Algorithm 5. The angles from the epipole to the extreme corners in the right image are mapped to the left image following Algorithm 3 (but from right to left). This yields two angles ϑ'_L and Θ'_L, which are in $[-\pi, \pi]$, as the result of function `atan2`.



$reg^2 \backslash reg^1$	0	1	2
0	(1,3)	(1,0)	(2,0)
1	(0,3)	(0,0)	(2,1)
2	(0,2)	(3,2)	(3,1)

Table $\mathcal{T}$. Depending on the region reg containing the epipole, the indices i of the two extreme corners P_i are given by $\mathcal{T}(reg)$. The indices are ordered such that the angle between first and second half-lines is less than π (in clockwise orientation).

Figure 3: Regions of the plane w.r.t. the image domain $[0, w] \times [0, h]$. The look-up table $\mathcal{T}$ determines the two extreme corners depending on the region containing the epipole e. When reg is a pair in $\{0, 1, 2\}^2$, $\mathcal{T}_{reg} = (\mathcal{T}^1_{reg}, \mathcal{T}^2_{reg})$ is a pair of indices in $\{0, 1, 2, 3\}^2$.

Algorithm 4: Computation of the bounds of radius and angle for polar sampling

Input: Epipole $e = (e^1, e^2)$, image dimensions $[0, w] \times [0, h]$.
Output: Bounds of polar sampling: $[\rho, \varrho]$ for radius, $[\vartheta, \Theta]$ for angle.

$reg \leftarrow (R_{0,w}(e^1), R_{0,h}(e^2))$ `// Position of` e `w.r.t. image, see` (14)
$p \leftarrow (C_{0,w}(e^1), C_{0,h}(e^2))$ `// See` (13)
$\rho, \varrho \leftarrow d(e, p)$ `//` d `is the Euclidean distance function`
for $i = 0, 1, 2, 3$ **do**
  $(r, \theta) \leftarrow$ polar transform of vector $P_i - e$. `//` P_i `is an image corner`
  $\rho \leftarrow \min(\rho, r)$
  $\varrho \leftarrow \max(\varrho, r)$
  if $i = \mathcal{T}^1(reg)$ **then** $\vartheta \leftarrow \theta$ `// Table` $\mathcal{T}$ `defined in Figure 3`
  if $i = \mathcal{T}^2(reg)$ **then** $\Theta \leftarrow \theta$
  if $\vartheta \geq \Theta$ **then** $\Theta \leftarrow \Theta + 2\pi$

Algorithm 5: Finding the angular span of the common visible region in left image.

Input: $\vartheta_L \in [-\pi, \pi]$, $\Theta_L \in (\vartheta_L, \vartheta_L + 2\pi]$, defining the angular span $[\vartheta_L, \Theta_L]$ in the left image from the epipole; angles $\vartheta'_L, \Theta'_L \in [-\pi, \pi]$, mapped from the extreme angles in the right image.
Output: Angular span of common region $[\vartheta''_L, \Theta''_L] \subset [\vartheta_L, \Theta_L]$

if $\Theta'_L < \vartheta'_L$ **then** $(\vartheta'_L, \Theta'_L) \leftarrow (\Theta'_L, \vartheta'_L)$
if $\Theta'_L > \vartheta'_L + \pi$ **then**
  $(\vartheta'_L, \Theta'_L) \leftarrow (\Theta'_L, \vartheta'_L + 2\pi)$
if $\Theta'_L < \vartheta_L$ **then** $(\vartheta'_L, \Theta'_L) \leftarrow (\vartheta'_L + 2\pi, \Theta'_L + 2\pi)$
if $\Theta_L < \vartheta'_L$ **then** $(\vartheta'_L, \Theta'_L) \leftarrow (\vartheta'_L - 2\pi, \Theta'_L - 2\pi)$
$\vartheta''_L \leftarrow \max(\vartheta_L, \vartheta'_L)$
$\Theta''_L \leftarrow \min(\Theta_L, \Theta'_L)$

2.5 Epipole(s) at Infinity

It may happen that an epipole is already at infinity, so that a rotation in the corresponding image is sufficient to have horizontal epipolar lines. In that case, the polar transform makes no sense, since all epipolar lines are parallel. The angle θ, used to identify epipolar lines, must be replaced by another

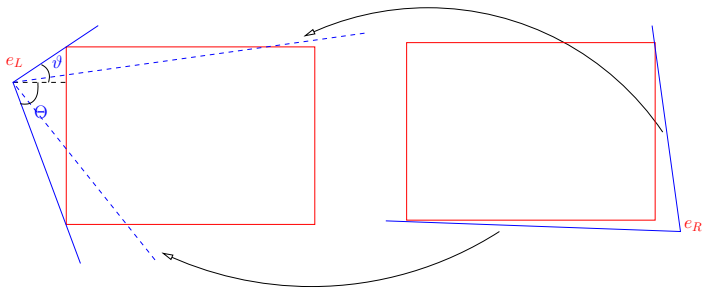


Figure 4: Mapping the extreme epipolar lines of the right view to the left view allows restricting the span of θ, $[\vartheta, \Theta]$, to the common visible region.

value. We propose to use the signed distance from the origin to the line, that is, if the line equation is $\begin{pmatrix}\ell_1 & \ell_2 & \ell_3\end{pmatrix} x = 0$, we define

$$\theta = \frac{\ell_3}{\sqrt{\ell_1^2 + \ell_2^2}}. \tag{15}$$

The trouble with that definition is that, despite its invariance to the scale of ℓ, its sign depends on the orientation of ℓ. It is shown in Appendix C that a correct definition is

$$\theta = \frac{\ell_3}{b\,\ell_1 - a\,\ell_2}, \tag{16}$$

with epipole $e = (a, b, 0)$, normalized such that $a^2 + b^2 = 1$.

In the same manner, the radius ρ of the sampled point must be redefined. We define it as the signed distance along line θ given by the orthogonal projection of the origin on the line. In other words, the "polar" transform of a point (x, y) becomes the pair (ρ, θ), which are the coordinates of the points with respect to the new frame given by the unit line direction vector and its orthogonal. Therefore the transition matrix is orthogonal.

More precisely, (a, b) is a unit direction vector of epipolar lines, which can be completed by its direct orthogonal $(-b, a)$ to form a basis. The transition matrix M can be used to transfer between Cartesian and "polar" coordinates

$$M = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}, \qquad \begin{pmatrix}x & y\end{pmatrix}^\top = M \begin{pmatrix}\rho & \theta\end{pmatrix}^\top, \qquad \begin{pmatrix}\rho & \theta\end{pmatrix}^\top = M^\top \begin{pmatrix}x & y\end{pmatrix}^\top. \tag{17}$$

2.6 Fixing Orientation

As the standard coordinate system for images has the y-coordinate oriented downward (contrary to trigonometric convention), the natural orientation for angles is clockwise. For an epipole in zone $(2, 0)$, $(2, 1)$ or $(2, 2)$, the minimum angle would be given by corner $P_2 = (w, h)$ in the first two cases and $P_3 = (0, h)$ in the last case. That means that the first sampled epipolar line is toward the bottom of the image. Also the sampling step progresses from right to left, and the rectification results in a 180° rotated image. It is generally considered better to modify as little as possible the original image, and thus it is advised to rotate back the rectified image.

The left image is responsible for θ sampling, and it should dictate the orientation. If the left epipole is in one of the above regions, the rotation must be performed in the left image. The rotation can be decomposed into an y-mirror and an x-mirror. For the right image, epipolar lines must follow the movement for the y-mirror. However, the x-mirror can be chosen freely in each image, it does not change the rectified nature of the output. The natural rule is to preserve the orientation of the original image. Specifically, we take three points in the rectified image, $(w/3, h/3)$, $(2w/3, h/3)$ and $(w/3, 2h/3)$. We can map back these points to the original image, call them p_1, p_2, p_3 and

$v_{12} = p_2 - p_1$, $v_{13} = p_3 - p_1$. Vectors v_{12} and v_{13} in that order should be directly orthogonal. If we observe a change of orientation, that is

$$v_{12}^1 \, v_{13}^2 - v_{12}^2 \, v_{13}^1 < 0, \tag{18}$$

it means that an x-mirror must be performed. Each image undergoes the procedure of Algorithm 6. Notice that the condition at line 5 is automatically satisfied for $J = J_L$, which means that the y-mirror must be completed by an x-mirror to yield a rotation.

Algorithm 6: Choose the best image orientation

Input: Rectified image $J = J_L$ or $J = J_R$
Output: Possibly rotated image J

1 **if** *epipole* e_L *in zone* $(2,0)$, $(2,1)$, $(2,2)$ *or* $(1,2)$ **then**
2 | Perform y-mirror on J
3 Map back points $(w/3, h/3)$, $(2w/3, h/3)$ and $(w/3, 2h/3)$ to original image I (I_L or I_R): p_1, p_2, p_3
4 Compute vectors $v_{12} = p_2 - p_1$, $v_{13} = p_3 - p_1$
5 **if** *condition* (18) *is satisfied* **then**
6 | Perform x-mirror on J

3 Implementation

3.1 Normalization of Epipoles

After Algorithm 2 is applied, the scales of the epipoles are still arbitrary, in the sense that $\lambda_L e_L$ and $\lambda_R e_R$ would also fit, provided $\lambda_L, \lambda_R > 0$. The most convenient is to have Cartesian coordinates of the epipoles, that is $e_3 = 1$. However, no positive λ allows that if $e_3 < 0$. The next best thing is to have $e_3 = \pm 1$. A convenient feature of this normalization is that the epipole is at position $(e_1/e_3, e_2/e_3) = (e_1\, e_3, e_2\, e_3)$ since $1/e_3 = e_3$. We have replaced a division by a multiplication, which is more efficient by an order of magnitude for modern computer architectures.

This is valid only for a finite epipole, $e_3 \neq 0$. For an infinite epipole, it is convenient to have a unit vector, so that (e_1, e_2) is a unit vector in the direction of epipolar lines. Their orientation can be chosen arbitrarily, but it is best to have $e_1 \geq 0$, so that the lines are oriented "toward the right" and the rectified image does not require rotation. In the case where $e_1 = 0$, we put $e_2 > 0$ for the same reason (orientation toward bottom of image).

A delicate question is when to judge that the epipole is "at infinity": the test $e_3 = 0$ would never happen because of noise or numerical imprecision, except on synthetic data. Just ignoring a non-null but small (absolute) value of e_3 is not wise, as the minimum and maximum radii ρ and ϱ may become huge, and we can be in a situation where $r \oplus 1 = r$ (floating point addition), leading to trouble when sampling epipolar lines.[2] Therefore, it should be better to force $e_3 = 0$ when that happens. A reasonable threshold must decide when it has a small impact on epipolar lines. The epipolar line through $P_0 = (0,0,1)$ has a direction vector $v = (e_1, e_2)$ whatever e_3 (provided $e \neq P_0$), therefore forcing $e_3 = 0$ has no effect on this line. Assuming that typical dimensions of the image are 10^3, we want at worst a shift of 1 pixel when forcing a line through P_i (another corner) to have direction (e_1, e_2). Considering similar triangles illustrated in Figure 5, we find that the condition is satisfied when $10^3/d < 1/10^3$, that is $d > 10^6$. That translates to $d^2 < 10^{12}$, or equivalently $e_1^2 + e_2^2 > 10^{12}\, e_3^2$, which avoids any division by e_3. The normalization process is summarized in Algorithm 7.

[2] Though it is unlikely to happen in double precision, requiring $r > 2^{53} \sim 10^{16}$.

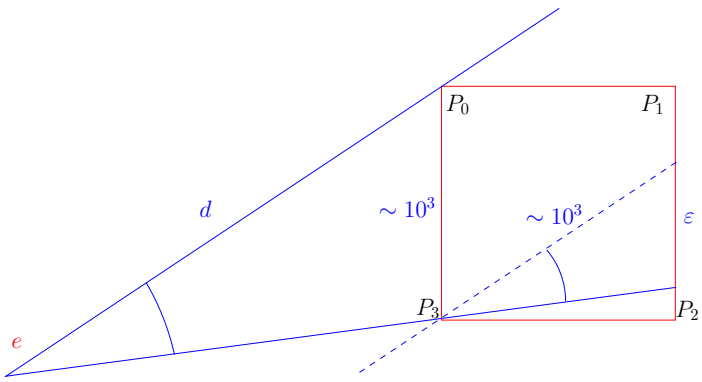


Figure 5: Deciding when to force epipole to infinity by setting $e_3 = 0$. The error made should satisfy $\varepsilon \leq 1$. Considering similar triangles, the distance d of epipole to origin P_0 should be at least 10^6 with typical image dimensions of 10^3.

Algorithm 7: Normalization of epipole, following Algorithm 2.

Input: Epipole $e = (e_1, e_2, e_3)$ as a 3-vector.
Output: Normalized epipole e.

```
if 10^12 e_3^2 < e_1^2 + e_2^2 then                    // e at infinity
    e_3 ← 0
    if e_1 < 0 or (e_1 = 0 and e_2 < 0) then e ← −e
    e ← e/‖e‖
else e ← e/|e_3|
```

3.2 Rectified Images Domain

Since following Algorithm 4 the angular span is synchronized between both images, each row of the generated images has its counterpart in the other image, therefore $h_{J_L} = h_{J_R}$. The choice is made to sample this interval uniformly in the left image. This is in contrast to [12], where the angular interval between consecutive epipolar lines is variable, inversely proportional to the largest distance from the epipole to the image domain in left and right images. Such a choice makes tracking the mapping between input and rectified images complex. We use a sampling step δ between ϑ_L and Θ_L defined by

$$\delta = \begin{cases} \frac{1}{\varrho_L} & \text{if } e_L^3 \neq 0, \\ 1 & \text{if } e_L^3 = 0. \end{cases} \tag{19}$$

In the case of infinite epipole, θ is just a distance, so a natural step of 1 is chosen. In the case of a finite epipole, the rectification is a true polar transform and the choice of $1/\varrho_L$ ensures that no aliasing is introduced (in the left image) since the arc length is upper bounded by 1. Therefore, the common number of rows of rectified images is

$$h_{J_L} = h_{J_R} = \left\lceil \frac{\Theta_L - \vartheta_L}{\delta} \right\rceil . \tag{20}$$

The horizontal sampling, along epipolar lines, is always 1 pixel. Therefore, the horizontal dimensions of the rectified images are

$$w_{J_L} = \lceil \varrho_L - \rho_L \rceil , \qquad w_{J_R} = \lceil \varrho_R - \rho_R \rceil . \tag{21}$$

They need not be the same, even if images I_L and I_R have the same dimension, since they also depend on the position of epipoles.

3.3 Line sampling

The most convenient way to sample along an epipolar line is to have the origin o and a unit vector v. For a finite epipole e, $o = e$ is fixed and the variable unit vector $v = (\cos\theta, \sin\theta)$ depends on θ. For an infinite epipole, the orientation is fixed $v = (e_1, e_2)/\|e\|$ and the center o is variable, orthogonal projection of the point $(0, 0)$ on the line, indexed by θ, see Figure 6. This is summed up in Table 1(a), where it is assumed that e has been normalized so that $\|e\| = 1$ if the epipole is at infinity.

When transferring the line in the other image of the pair, we do not have explicitly θ, but the vector ℓ defining the line equation

$$\ell_1 x + \ell_2 y + \ell_3 = 0. \tag{22}$$

When $e_3 = 0$, knowing that the line direction is (e_1, e_2), θ given by (16) is the signed distance from the origin

$$\theta = \frac{\ell_3}{e_2\,\ell_1 - e_1\,\ell_2}. \tag{23}$$

The point o is then given in Cartesian coordinates by

$$o = \frac{\ell_3}{e_2\,\ell_1 - e_1\,\ell_2}(-e_2, e_1). \tag{24}$$

All cases of computation of o and v from ℓ are in Table 1(b).

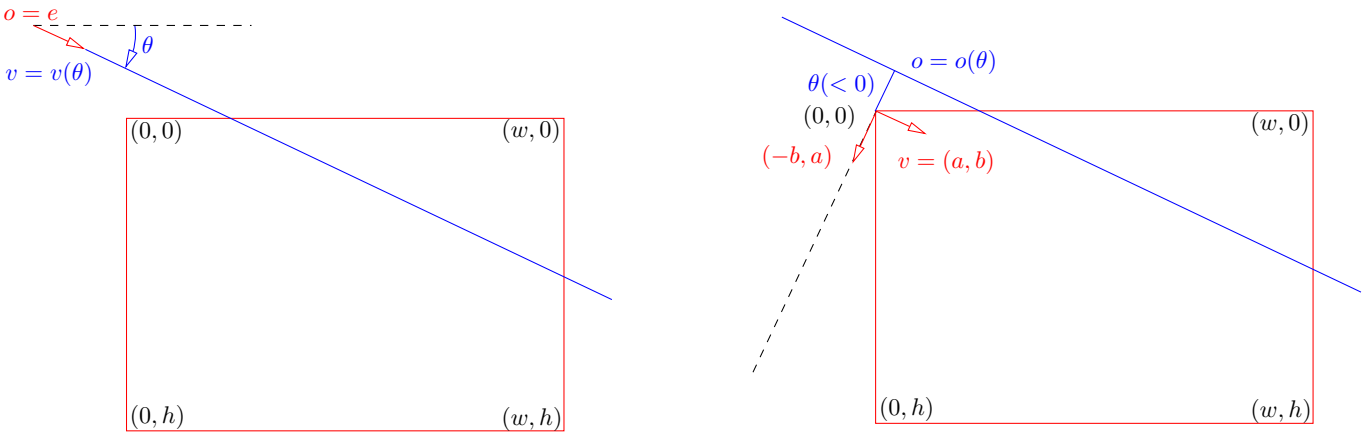


Figure 6: Identification of epipolar line as a function of θ: for a finite epipole (left), the origin o of sampling is the epipole and θ the angle of the line with the horizontal; for an infinite epipole $e = (a, b, 0)$ (right), θ (negative in this case) is the signed distance of the origin and o the orthogonal projection of the origin. In each case, v is a unit direction vector of the line. The polar transform of a point p is (r, θ), with r the (signed) distance to o and θ the parameter of the line through o and p.

Only the diagonal coefficients of these tables changing with each epipolar line, the function `transfer_theta` returns this variable 2-vector instead of the resulting θ in Algorithm 3. If θ is needed afterward, it can be recovered as

$$\theta = \begin{cases} e_1\,o_2 - e_2\,o_1 & \text{if } e_3 = 0, \\ \operatorname{atan2}(v_2, v_1) & \text{if } e_3 \neq 0. \end{cases} \tag{25}$$

The function works as follows

1. Compute o and v from Table 1(a);
2. Find a point p on line ($p = o$ if $e_3 = 0$, $p = o + \rho v$ otherwise);

Epipole	o	v
$e_3 = 0$	$(-\theta e_2, \theta e_1)$	(e_1, e_2)
$e_3 \neq 0$	$(e_1/e_3, e_2/e_3)$	$(\cos\theta, \sin\theta)$

(a) Computation of o and v from θ. The case where the epipole is at infinity ($e_3 = 0$) has a different interpretation of θ (signed distance to origin) than the usual case with a finite epipole (angle of line w.r.t. horizontal).

Epipole	o	v
$e_3 = 0$	$\frac{\ell_3}{e_2\ell_1 - e_1\ell_2}(-e_2, e_1)$	(e_1, e_2)
$e_3 \neq 0$	$(e_1/e_3, e_2/e_3)$	$(\ell_2, -\ell_1)/\sqrt{\ell_1^2 + \ell_2^2}$

(b) Computation of o and v from line equation vector ℓ. The formulas for θ from ℓ are (16) $\theta = \frac{\ell_3}{e_2\ell_1 - e_1\ell_2}$ if $e_3 = 0$ and $\theta = \operatorname{atan2}(-\ell_1, \ell_2)$ otherwise.

Table 1: Parameters for line sampling: o is the origin of sampling, v is a unit vector of the direction of the line. It is assumed that if the epipole e is at infinity ($e_3 = 0$), $\|e\|^2 = e_1^2 + e_2^2 + 0^2 = 1$. Values depending on the epipolar line are in red.

3. Compute $\ell = Fp$; (Take the opposite if $e_3 < 0$)

4. Compute and return variable vector from Table 1(b).

It should be noted that at step 1 the epipole is in the source image (where θ is measured) and at step 3 in the target image. Sometimes, we just need the variable vector from θ but without transfer. If no fundamental matrix is input to the function, it just stops after step 1.

3.4 Pushforward and Pullback

Since input images I_L and I_R are output as J_L and J_R, we need to keep track of the change of coordinates between I_L and J_L, as well as I_R and J_R. The process of changing coordinates from I to J is called pushforward, while the inverse transformation from J to I is called pullback. Actually, to generate J itself, we use a pullback map: for each pixel of J, determine from which location in I to fetch the color. This allows us to separate the rectification from the image sampling and interpolation, which in our implementation relies on splines [1]. The map is an application

$$\{0, \ldots, w_J - 1\} \times \{0, \ldots, h_J - 1\} \to \mathbb{R}^2. \tag{26}$$

The domain is discrete (pixels) but the co-domain can include locations outside the image domain of I. It is more efficient to tabulate the complete discrete function once, instead of computing for each pixel: all pixels of J on the same row share the same θ. Transforming to Cartesian coordinates implies the usage of trigonometric functions, which are costly. Doing that once per line of J is more economical than for each pixel.

We also need the pushforward, transforming from I to J coordinates: this is used to draw in J the positions of points having a corresponding point in the other image of the stereo pair. This is essentially the polar transform. There is no advantage in precomputing the map if we call it once per pixel. An application of the pushforward is to resample the depth map[3] in the original image, once it is computed in the rectified image.

Both pullback and pushforward are more direct when applied to the left image, because it is the one guiding the regular sampling of θ. In the right image, the transfer of θ must be performed. The pullback procedure is detailed in Algorithm 8. The sampling step δ for θ is 1 if the epipole e_L is at infinity, $1/\varrho$ otherwise. The possible mirrors at the end are explained in Subsection 2.6.

The pushforward performs the inverse process, detailed in Algorithm 9.

[3]It actually does not make much sense to resample the *disparity* map, because disparity is defined only in the rectified setup.

Algorithm 8: Generation of the pullback map of rectified image J

Input: Image $J \in \{J_L, J_R\}$, fundamental matrix F if $J = J_R$
Output: Pullback map P^{-1} giving for each pixel of J its origin in corresponding image I

```
for y = 0, ... h_J − 1 do
    θ ← ϑ + y δ
    Compute o and v (Table 1)        // Function transfer_theta, using F iff J = J_R
    for x = 0, ... w_J − 1 do
        r ← ρ + x
        P^{-1}(x, y) ← o + r v
if J_L must rotate then
    Apply y-mirror to P^{-1}
if orientation of P^{-1} incorrect then
    Apply x-mirror to P^{-1}
```

Algorithm 9: Pushforward.

Input: (x, y) Cartesian coordinates in image $I \in \{I_L, I_R\}$, fundamental matrix F if $I = I_R$
Output: (x', y') corresponding point in rectified image J

```
(r, θ) ← polar transform in I of point (x, y)
if I = I_R then
    ℓ ← F (x  y  1)^⊤
    Get new value of θ from ℓ according to Table 1(b)
// Transform (r, θ) into Cartesian coordinates in J
x' ← r − ρ_I                                  // ρ_I smallest radius of image I
y' ← (θ − ϑ_L)/δ                              // δ = 1 if e_L at infinity, 1/ϱ_L otherwise
if J must rotate then x' ← w_J − 1 − x'
if J_L must rotate then y' ← h_J − 1 − y'
```

3.5 Source Code Exploration

We use one class object `Polarizer` per image. This structure records the position of the epipole as a homogeneous 3-vector, computes the bounds of r and θ, and whether the image requires rotation. The method `polar` performs the map $(x, y) \mapsto (r, \theta)$ and `restrict_angles` takes the bounds of θ of the other image, transfers them to the current image via `transfer_theta`, and computes the intersection of intervals (`inter_mod_2pi` for angular intervals).

An object of class `Polarectifyer` is used to manage the rectification process. It controls the object `Polarizer` for each image, stores the fundamental matrix, and generates the pullback map, which can be read by the method `pullback_map`, taking as argument left or right. The inverse process is provided by `pushforward`.

4 Experiments

Figure 7 shows the rectification of two photographs with the epipoles outside the image, though not very far. Traditional rectification methods are able to handle such case, so as the polar rectification. Notice how the left image is cut at the top and the right image at the bottom: their missing parts correspond to epipolar lines that fall outside the image when transferred. SIFT point correspondences being used to compute the fundamental matrix are marked in red in the rectified images. The

program first computes SIFT point correspondences using the implementation of [13], then computes the fundamental matrix by a RANSAC procedure with an adaptive threshold estimation [8] before proceeding to the rectification.

The SIFT algorithm finds keypoints in both images and associates to each one a feature descriptor, a vector in $\mathbb{R}^{128}$. Points P and Q in left and right images are considered to match if

$$\|f_P - f_Q\| \leq C \min_{Q' \neq Q} \|f_P - f_{Q'}\|, \tag{27}$$

with vectors f the descriptors of the points and the minimum spans all keypoints Q' of the right image, except Q. The positive SIFT ratio C is tunable and determines how ambiguous correspondences can be. A low value of C (typically 0.6) keeps only unambiguous correspondences; a higher value means the matching is less demanding. When $C = 1$, each point P matches the closest point in the right image in terms of descriptor distance. This parameter can be changed by the demo user.

The estimation of the fundamental matrix relies on the set of SIFT correspondences. A subset of 7 correspondences can give an estimated fundamental matrix, which can be voted on by all other correspondences according to how they fit this model. After a sufficient number of candidate fundamental matrices has been drawn, the best one is selected. The algorithm of [8] does not require a threshold to distinguish inlier from outlier. It can also decide to fail if all matrix candidates are considered to be due to chance. In that case, the rectification cannot proceed.

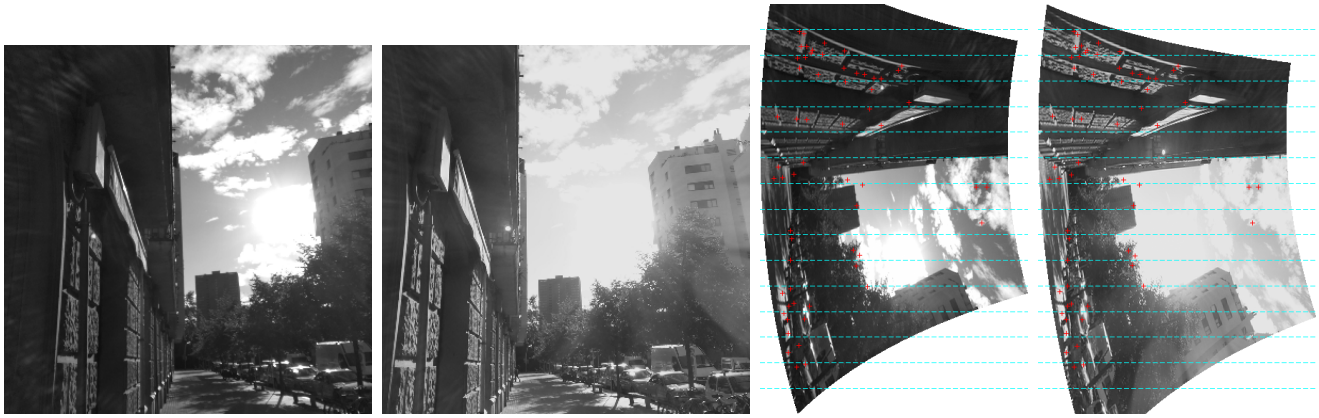


Figure 7: Polar rectification of an urban scene, with epipoles slightly outside the image

Figures 8 and 9 show examples with epipoles inside the images. Usual rectification algorithms working with homography cannot handle such cases correctly, as they need to send the epipoles to infinity. The four corners of the image can be seen as spikes in the rectified images. The drawback of this rectification is that the polar transform can distort strongly the images, even locally. This makes the disparity estimation harder for local algorithms.

Another example of rectification for epipoles inside the image is in Figure 10. In a zone near the epipoles, it can be noticed that the lower SIFT point seems to have an important vertical shift. This is due to the fact that near the center of the polar transform errors are amplified, because the radial sampling is fine enough to accommodate points at large distance of the center. Whereas the maximum error from a point to its epipolar line was 3 pixels after estimation of the fundamental matrix, the angular sampling can increase it noticeably.

Figure 8: Polar rectification with epipoles inside the image. Rectified images are presented rotated for layout purposes, so that points move vertically.

Figure 9: Polar rectification with epipoles inside the image. Rectified images are presented rotated for layout purposes, so that points move vertically.

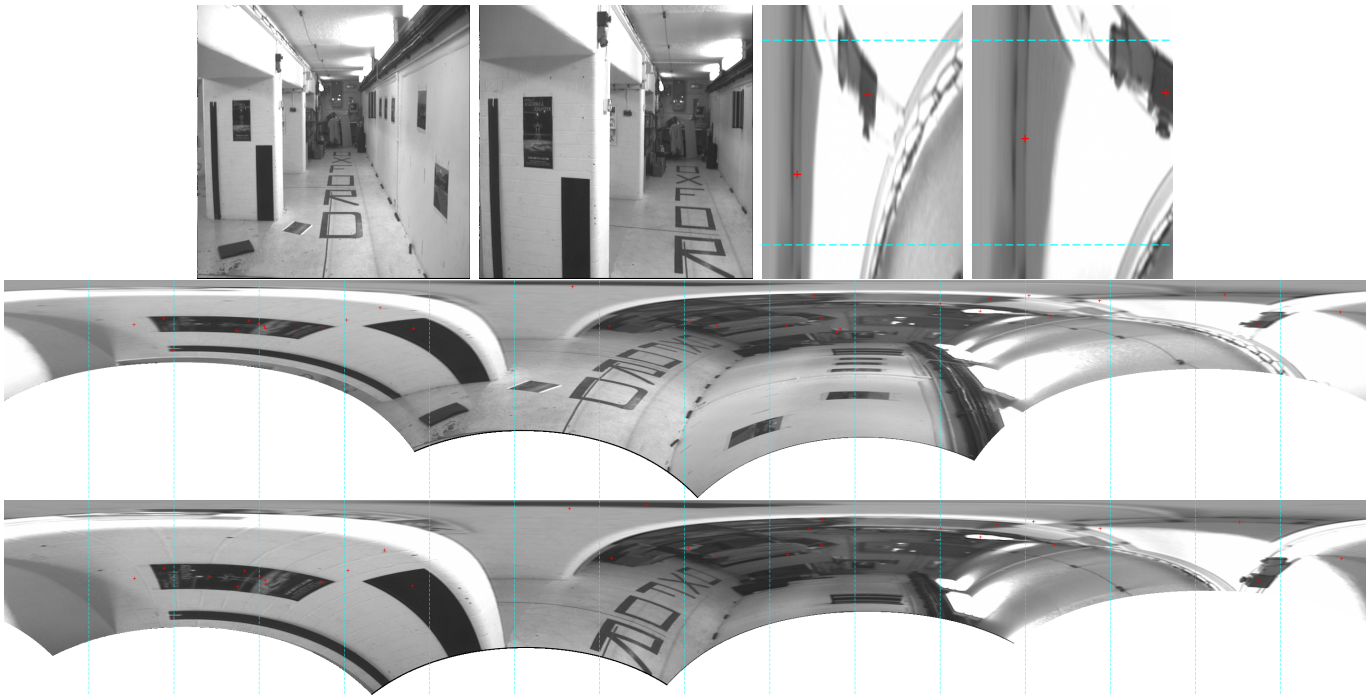


Figure 10: Polar rectification with epipoles inside the image. Rectified images are presented rotated for layout purposes, so that points move vertically. Top, detail of a zone near the epipole.

5 Conclusion

The advantage of the polar transform for rectification is that it is valid for any configuration of epipoles, including when one or both are inside the image domain. A drawback is that the resulting images are distorted, in the sense that straight lines in the original image become curved. That does not facilitate the stereo matching. A better algorithm would mix both modes: when both epipoles are outside the image, rectify with homographies; otherwise, rectify with the polar transform. When only one epipole is inside the image, it is possible to apply a polar transform to this image and a homography to the other. It remains to be seen if such mixed rectification helps a disparity estimation algorithm to get better results, compared to the rectification with two polar transforms.

Image Credits

Schemas by the authors (license CC-BY-SA-4.0). Photograph sources:

VGG, Univ. of Oxford (https://www.robots.ox.ac.uk/~vgg/data/mview/)

Archives of demo system [8]:

http://demo.ipol.im/demo/147/archive?key=7F40D86EB04DBCC07822A1B2DBB0A7B9

http://demo.ipol.im/demo/147/archive?key=170230E6033D7C91167DB11EDF3C01CD

http://demo.ipol.im/demo/147/archive?key=6BE0B64461CD7A097B49D06035B69B8C

A Homography Mapping Epipolar Lines

We derive the form of the homography H mapping an epipolar line from the left image ℓ_L to its corresponding epipolar line in the right image $\ell_R = H\ell_L$. Let us note e_L and e_R the epipoles, $F^\top e_L = Fe_R = 0$.

Proposition 1. *A homography H maps left epipolar lines to their corresponding right epipolar lines if and only if it can be written*

$$H = F^\top [e_L]_\times + v e_L^\top, \tag{28}$$

(up to factor) with $v \in \mathbb{R}^3$ an arbitrary vector such that $v^\top e_R \neq 0$.

Notice that $e_L^\top \ell_L = 0$, expressing that the epipole e_L is on the epipolar line ℓ_L, hence $H\ell_L = F^T[e_L]_\times \ell_L$, independent on v. For the transfer of epipolar lines, we may choose $v = 0$, even though the resulting H is not a homography since it is not invertible.

Proof. We have $x := e_L \times \ell_L \neq 0$, and $x^\top \ell_L = 0$, showing x is a point on this epipolar line different from e_L. Hence $\ell_R = F^\top x = F^\top [e_L]_\times \ell_L$. For any v, we then have

$$F^\top [e_L]_\times \ell_L + v e_L^\top \ell_L = \ell_R + 0, \tag{29}$$

since $e_L^\top \ell_L = 0$. This proves that (28) is a sufficient condition. For the sake of completeness, we also show its necessity.

Let H be a homography mapping left epipolar lines to their corresponding right epipolar lines and $x \in \mathbb{R}^3$. If x is not linearly dependent on e_L, $\ell_L = e_L \times x$ is the epipolar line through x, and we can write

$$\ell_R = F^\top x = H(e_L \times x), \tag{30}$$

(up to scale), the latest equality remaining true even if x is a multiple of e_L, the terms being null. Therefore, $\lambda F^\top = H[e_L]_\times$. Using the double cross product formula, $[a]_\times[b]_\times = ba^\top - (a^\top b)Id$, yields

$$\lambda F^\top [e_L]_\times = He_L e_L^\top - \|e_L\|^2 H. \tag{31}$$

Writing $v := -\frac{1}{\lambda} He_L$, we get the announced result,

$$-\frac{\|e_L\|^2}{\lambda} H = F^T [e_L]_\times + v e_L^\top. \tag{32}$$

The last condition of the theorem is necessary and sufficient to ensure that H defined by (28) is invertible: choose v_1 and v_2 two non-colinear vectors of $\mathbb{R}^3$ orthogonal to e_L. (e_L, v_1, v_2) is a basis of $\mathbb{R}^3$, hence H is invertible if and only if (He_L, Hv_1, Hv_2) is also a basis. We get $Hv_i = F^\top(e_L \times v_i)$ $(i = 1, 2)$. We argue that $e_L \times v_1$ and $e_L \times v_2$ are two distinct epipolar lines

$$(e_L \times v_1) \times (e_L \times v_2) = \left(e_L^\top (v_1 \times v_2)\right) e_L, \tag{33}$$

which is not the null vector since $v_1 \times v_2$ is proportional to e_L. Vectors v_1 and v_2 can represent epipolar lines since $e_L^\top v_i = 0$, and we deduce that Hv_1 and Hv_2 are the corresponding epipolar lines, which are also distinct. Now $He_L = \|e_L\|^2 v$ is not in the span of Hv_1 and Hv_2 if and only if

$$v^\top (Hv_1 \times Hv_2) \neq 0, \tag{34}$$

and $Hv_1 \times Hv_2$ is proportional to e_R, as the intersection of two epipolar lines in the right image, hence the condition becomes $v^\top e_R \neq 0$. □

B Angular Intervals

Given $\theta \in \mathbb{R}$, let us define its class modulo 2π

$$\bar{\theta} = \{\theta + 2k\pi : k \in \mathbb{N}\}. \tag{35}$$

We define an angular interval $[\bar{\theta}, \bar{\vartheta}]$ as

$$[\bar{\theta}, \bar{\vartheta}] = \{\bar{t} : \theta \leq t \leq \vartheta + 2k\pi < \theta + 2\pi, k \in \mathbb{N}\}. \tag{36}$$

It is easy to check that this definition is independent of the representants $\theta \in \bar{\theta}$ and $\vartheta \in \bar{\vartheta}$. Notice that an angular interval is never empty. The index k involved in the definition is uniquely defined by

$$k = \left\lfloor \frac{\theta - \vartheta}{2\pi} \right\rfloor. \tag{37}$$

Finally, this leads to the definition of the length of an interval

$$\left|[\bar{\theta}, \bar{\vartheta}]\right| = \vartheta + 2k\pi - \theta \in [0, 2\pi). \tag{38}$$

Given $\bar{\theta} \neq \bar{\vartheta}$, we can define the two intervals $[\bar{\theta}, \bar{\vartheta}]$ and $[\bar{\vartheta}, \bar{\theta}]$, and we have

$$\left|[\bar{\theta}, \bar{\vartheta}]\right| + \left|[\bar{\vartheta}, \bar{\theta}]\right| = 2\pi, \tag{39}$$

so that if one has length smaller than π, the other length is larger than π. We will also use a simple result whose proof is left as an exercise.

Lemma 1. *Let α be an angular interval. The length $|\alpha| < \pi$ if and only if there is some θ such that $\bar{\theta} \notin \alpha$ and $\overline{\theta + \pi} \notin \alpha$.*

The following result allows finding the correct angular span bounded by $\bar{\theta}$ and $\bar{\vartheta}$ given by the mapping of epipolar half lines.

Proposition 2. *Let F be the fundamental matrix of a stereo pair. Assume that e_R is outside the right image and that neither e_L nor e_R is at infinity ($e_L^3 \neq 0 \neq e_R^3$). The mapping (in the left image) of the right epipolar half lines spans an angular interval of length less than π.*

Proof. The domain of the image being convex, thanks to the Hahn-Banach theorem we can find a line ℓ through e_R such that ℓ does not meet the image. Applying the adequate rotation and translation in the right image and without loss of generality, we can assume that

$$e_R = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top, \qquad \ell = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\top. \tag{40}$$

Given $\theta \in [0, 2\pi)$, let us define with a slight abuse of notation

$$F(\theta) = F \begin{pmatrix} \cos\theta & \sin\theta & 0 \end{pmatrix}^\top, \tag{41}$$

that is, the mapping of the epipolar half line in the right image at angle θ, which is an epipolar half line in the left image. Since the line ℓ does not meet the image, all points P of the right image domain define from e_R an angular span $[\bar{\theta}, \bar{\vartheta}]$ of length smaller than π, and $\bar{0} \notin [\bar{\theta}, \bar{\vartheta}]$, $\bar{\pi} \notin [\bar{\theta}, \bar{\vartheta}]$. Using the fact that $\theta \in [0, 2\pi) \mapsto F(\theta)$ is bijective (expressing the fact that the mapping between epipolar half lines is one-to-one), it is monotonous, so that the image of the angular span $[\bar{\theta}, \bar{\vartheta}]$ is whether $[F(\bar{\theta}), F(\bar{\vartheta})]$ or $[F(\bar{\vartheta}), F(\bar{\theta})]$ depending on whether F is increasing or decreasing.[4] Moreover, neither $F(0)$ nor $F(\pi) = F(0) + \pi$ (epipolar half line complementary of $F(0)$) is in this interval, implying that its length is less than π. □

C Signed Distance from Origin of Epipolar Line for Epipole at Infinity

We show that (16) is the signed distance of the epipolar line ℓ from the origin when the epipole is at infinity $(a, b, 0)$, normalized such that $a^2 + b^2 = 1$. Since ℓ is an epipolar line, we have $(\ell_1, \ell_2) = \lambda(-b, a)$, the latter being directly orthogonal to (a, b), a unit vector directing the epipolar lines. The factor λ can be recovered by

$$\lambda = \begin{pmatrix} -b \\ a \end{pmatrix}^\top \begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix}. \tag{42}$$

The intersection of ℓ and its orthogonal line through the origin can be written in homogeneous coordinates

$$p = \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \times \ell = p_3 \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \tag{43}$$

with (x, y) the Cartesian coordinates of the point represented by p. The signed distance is therefore

$$\theta = \begin{pmatrix} -b \\ a \end{pmatrix}^\top \begin{pmatrix} x \\ y \end{pmatrix}. \tag{44}$$

[4] We have done an abuse of notation here, because $F(\theta)$ is not an angle, but has to be understood as the direction angle of the half line defined by $F(\theta)$.

It is easy to check that $p_3 = \lambda$ (using the normalization of the epipole), and then we find

$$\lambda\,\theta = \begin{pmatrix} -b \\ a \\ 0 \end{pmatrix}^\top \left[\begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \times \ell \right] = \ell^\top \left[\begin{pmatrix} -b \\ a \\ 0 \end{pmatrix} \times \begin{pmatrix} a \\ b \\ 0 \end{pmatrix} \right] = \ell^\top \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} = -\ell_3. \tag{45}$$

Combined with (42), we deduce (16).

References

[1] T. Briand and P. Monasse, *Theory and practice of image B-spline interpolation*, Image Processing On Line, 8 (2018), pp. 99–141. https://doi.org/10.5201/ipol.2018.221.

[2] A. Fusiello and L. Irsara, *Quasi-Euclidean uncalibrated epipolar rectification*, in International Conference on Pattern Recognition (ICPR), IEEE, 2008, pp. 1–4. https://doi.org/10.1109/ICPR.2008.4761561.

[3] J. Gluckman and S.K. Nayar, *Rectifying transformations that minimize resampling effects*, in Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2001, pp. I–I. https://doi.org/10.1109/CVPR.2001.990463.

[4] R.I. Hartley, *Theory and practice of projective rectification*, International Journal of Computer Vision, 35 (1999), pp. 115–127. https://doi.org/10.1023/A:1008115206617.

[5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2nd ed., 2004. ISBN 978-0521540513.

[6] Y. Lee, K-A. Toh, and S. Lee, *Stereo image rectification based on polar transformation*, Optical Engineering, 47 (2008), p. 087205. https://doi.org/10.1117/1.2969124.

[7] C. Loop and Z. Zhang, *Computing rectifying homographies for stereo vision*, in Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 1999, pp. 125–131. https://doi.org/10.1109/CVPR.1999.786928.

[8] L. Moisan, P. Moulon, and P. Monasse, *Fundamental matrix of a stereo pair, with a contrario elimination of outliers*, Image Processing On Line, 6 (2016), pp. 89–113. https://doi.org/10.5201/ipol.2016.147.

[9] P. Monasse, *Quasi-Euclidean epipolar rectification*, Image Processing On Line, 1 (2011), pp. 187–199. https://doi.org/10.5201/ipol.2011.m_qer.

[10] P. Monasse, J-M. Morel, and Z. Tang, *Three-step image rectification*, in British Machine Vision Conference, BMVA Press, 2010, pp. 89.1–89.10. http://dx.doi.org/10.5244/C.24.89.

[11] K. Palander and S.S. Brandt, *Epipolar geometry and log-polar transform in wide baseline stereo matching*, in International Conference on Pattern Recognition (ICPR), IEEE, 2008, pp. 1–4. https://doi.org/10.1109/ICPR.2008.4761515.

[12] M. Pollefeys, R. Koch, and L. Van Gool, *A simple and efficient rectification method for general motion*, in International Conference on Computer Vision (ICCV), vol. 1, IEEE, 1999, pp. 496–501. https://doi.org/10.1109/ICCV.1999.791262.

[13] I. Rey Otero and M. Delbracio, *Anatomy of the SIFT method*, Image Processing On Line, 4 (2014), pp. 370–396. https://doi.org/10.5201/ipol.2014.82.