Rectification and Distortion Correction

Hagen Spies

March 12, 2003

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University, Sweden

Contents

- Distortion Correction
- Rectification by projecting on a 3D plane
- Rectification that minimises image distortion

Objectives:

- How to undo image distortion.
- Learn two ways to do a rectification.

Distortion Correction

Often we will want to undo the effects of image distortion:

- To get the correct image coordinates of a measured point that are then used in a subsequent 3D reconstruction. In this case the image does not have to be warped but feature extraction is performed on the original image. This might be beneficial as warping can introduce aliasing and generally implies a low-pass filtering step.

- Other tasks such as the extraction of straight lines or the grouping of features is better done on the warped images. This is also the case for simple stereo correlation methods that seeks to establish dense correspondences by searching along horizontal scan lines - such algorithms also require rectification.

Distortion Correction

How the correction of the distortion is done obviously depends on the used distortion model. If $\boldsymbol{x} = (x, y, 1)^T$ denote the (measured) pixel coordinates we first transform into normalised coordinates:

$$\hat{\boldsymbol{x}} = \boldsymbol{K}^{-1}\boldsymbol{x}$$

Then the distortion is applied:

$$\hat{\boldsymbol{x}}^c = \hat{\boldsymbol{x}} - \boldsymbol{\delta}(\hat{\boldsymbol{x}})$$

where $\delta(\hat{\boldsymbol{x}})$ models the distortion in terms of radial and tangential coefficients. Corrected pixel coordinates are then obtained by applying the camera matrix again:

$$\boldsymbol{x}^c = \boldsymbol{K}\hat{\boldsymbol{x}}^c = \boldsymbol{K}(\hat{\boldsymbol{x}} - \boldsymbol{\delta}(\hat{\boldsymbol{x}})) = \boldsymbol{x} - \boldsymbol{K}\boldsymbol{\delta}(\hat{\boldsymbol{x}})$$

Using two radial and two tangential distortion parameters the displacement becomes:

$$\boldsymbol{\delta}(\hat{\boldsymbol{x}}) = \left[\begin{array}{c} \hat{x}(r_1 R^2 + r_2 R^4) + 2\, t_1 \hat{x}\hat{y} + t_2\, (R^2 + 2\hat{x}^2) \\ \hat{y}(r_1 R^2 + r_2 R^4) + t_1\, (R^2 + 2\hat{y}^2) + 2\, t_2 \hat{x}\hat{y} \end{array} \right]$$

Improved Distortion Correction

The problem with the above scheme is that we are using the disturbed image coordinates in $\hat{\boldsymbol{x}} = \boldsymbol{K}^{-1}\boldsymbol{x}$ to evaluate the displacement where we should really use the undistorted (but unknown) normalised image coordinates $\hat{\boldsymbol{x}}^c = \boldsymbol{K}^{-1}\boldsymbol{x}^c$.

However we can get a better approximation by reinserting:

$$\boldsymbol{x}'^c = \boldsymbol{x} - \boldsymbol{K}\boldsymbol{\delta}(\boldsymbol{K}^{-1}\boldsymbol{x}^c) = \boldsymbol{x} - \boldsymbol{K}\boldsymbol{\delta}(\hat{\boldsymbol{x}} - \boldsymbol{\delta}(\hat{\boldsymbol{x}}))$$

Depending on the amount of distortion this needs to be iterated a few times.

For more simple distortion models there might exist analytical solutions.

A general approach is to numerically invert (e.g using Powell's method) the distortion function:

$$\boldsymbol{x} = f(\boldsymbol{x}^c, \boldsymbol{p}) \quad \rightarrow \quad \boldsymbol{x}^c = f^{-1}(\boldsymbol{x}, \boldsymbol{p})$$

in the sense that we seek the corrected image points that best predict the measured image points given the internal camera parameters.

Rectification

The purpose of rectification is to transform the general stereo geometry to that of the standard stereo setup. In this simple geometry the image planes are parallel and coordinate systems aligned. Then the epipoles are at infinity and the epipolar lines are parallel and along the x-direction (scan lines).

After rectification the problem of stereo matching reduces to a one-dimensional search along the rows of the images. The depth of a 3D point becomes inversely proportional to the disparity, i.e. the horizontal distance between corresponding image points.

Another way to express this geometry is to say that both cameras have the same internal parameters and that the motion between the cameras is a pure translation along the x-axis. Then we can choose projection matrices as:

$$\boldsymbol{P} = \boldsymbol{K}\,[\mathbb{1} \mid \mathbf{0}] \quad \text{and} \quad \boldsymbol{P}' = \boldsymbol{K}\,[\mathbb{1} \mid \boldsymbol{T}] \quad \text{with} \quad \boldsymbol{T} = [T_x,\ 0,\ 0]^T$$

Rectified Fundamental Matrix

Recall that the fundamental matrix for the calibrated case is given by:

$$\boldsymbol{F} = [\boldsymbol{e}']_\times \boldsymbol{K}' \boldsymbol{R} \boldsymbol{K}^{-1}$$

In the rectified case the epipole is at infinity: $\boldsymbol{e}' = [1, 0, 0]^T$, with $\boldsymbol{R} = \mathbb{1}$ we also have $\boldsymbol{K}' \boldsymbol{R} \boldsymbol{K}^{-1} = \mathbb{1}$. Hence for rectified images the fundamental matrix takes the simple form:

$$\boldsymbol{F}_r = [\boldsymbol{e}']_\times = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

The epipolar line to a point in the first image does indeed become identical to the scan-line:

$$\boldsymbol{l}' = \boldsymbol{F}_r \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -y \end{bmatrix}$$

Mapping the Epipole

If the epipolar lines are parallel to the x-axis the the epipoles should be mapped to a special point at infinity: $\boldsymbol{i} = (1,0,0)^T$. The process of rectification is then to find a homography for each image that maps the epipole to this point. Let

$$\bar{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{x} \quad \text{and} \quad \bar{\boldsymbol{x}}' = \boldsymbol{H}'\boldsymbol{x}'$$

then we obtain

$$\bar{\boldsymbol{x}}'^T \boldsymbol{F}_r \bar{\boldsymbol{x}} = \boldsymbol{x}'^T \boldsymbol{H}'^T \boldsymbol{F}_r \boldsymbol{H}\boldsymbol{x} = 0$$

thus we look for homographies that satisfy:

$$\boldsymbol{H}'^T \boldsymbol{F}_r \boldsymbol{H} = \boldsymbol{F}$$

Note that $\boldsymbol{H}$ and $\boldsymbol{H}'$ are not uniquely defined by this constraint.

Using 3D Construction

A possible rectification technique projects the images onto on common plane Π using the original camera centres. For the epipoles to be at infinity this plane has to be parallel to the baseline, i.e. the line connecting the camera centres. Let's look in more detail at an explicit algorithm proposed by Fusiello et al. [2000].

As we use the same camera centre the projection matrices of the rectified images can be expressed using the same rotation matrix. Furthermore the camera matrices have to be identical by definition, thus:

$$\boldsymbol{P}_r = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{R}\boldsymbol{C}] \quad \text{and} \quad \boldsymbol{P}'_r = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{R}\boldsymbol{C}']$$

For the common camera matrix we might choose that of the first camera (this is arbitrary) and then compute the new projection matrices once we have determined the rotation matrix.

Fixing the Rotation

The rotation matrix can be specified by its row vectors: $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]^T$. Those row vectors can be computed noting:

1. The new X-axis is parallel to the baseline: $\boldsymbol{r}_1 = \frac{(C - C')}{\|C - C'\|}$
2. The new Y-axis is orthogonal to the X-axis and to the old Z-axis: $\boldsymbol{r}_2 = \frac{\boldsymbol{Z} \times \boldsymbol{r}_1}{\|\boldsymbol{Z} \times \boldsymbol{r}_1\|}$. Where $\boldsymbol{Z}$ is given by the third row of the rotation matrix in the the first camera.
3. The new Z-axis is orthogonal to the XY-plane: $\boldsymbol{r}_2 = \boldsymbol{r}_1 \times \boldsymbol{r}_2$

We can thus compute the new projection matrices $\boldsymbol{P}_r$ and $\boldsymbol{P}'_r$.

Finding the Homographies

We decompose the original projection matrices as $\boldsymbol{P}_o = [\boldsymbol{Q}_o \mid \boldsymbol{t}_o]$ and the rectified version as $\boldsymbol{P}_r = [\boldsymbol{Q}_r \mid \boldsymbol{t}_r]$. Than we can determine the optical ray to any point in the original image $\boldsymbol{x}_o$ and the rectified image $\boldsymbol{x}_r$:

$$\boldsymbol{X} = \boldsymbol{C} + \lambda_o \boldsymbol{P}_o^+ \boldsymbol{x}_o \quad \text{and} \quad \boldsymbol{X} = \boldsymbol{C} + \lambda_r \boldsymbol{P}_r^+ \boldsymbol{x}_r$$

Here $\boldsymbol{C}$ denotes the optical centre. Because the optical centre remains unchanged and the image points correspond to the same optical ray we can identify the affine part to obtain:

$$\boldsymbol{x}_r = \lambda \boldsymbol{Q}_r \boldsymbol{Q}_o^{-1} \boldsymbol{x}_o$$

hence the sought homography is given by $\boldsymbol{H} = \boldsymbol{Q}_r \boldsymbol{Q}_o^{-1}$. For the second image this becomes $\boldsymbol{H}' = \boldsymbol{Q}'_r \boldsymbol{Q}'^{-1}_o$.

While this method is relatively easy to implement and does produce rectified images it does not impose any limitation on the amount of distortion applied to the images.

Improved Rectification

A more involved algorithm to determine the homographies seeks to minimise the associated image distortions. We will now discuss the method given in Hartley and Zisserman [2000]. Here knowledge of the fundamental matrix and a number of corresponding points is required - a complete calibration is not needed.

First the epipoles should be mapped to the point $\boldsymbol{i}$. This is done in three steps:

1. Translate the considered point $\boldsymbol{x}_i$ to the origin using a transform $\boldsymbol{T}$.
2. Rotate with $\boldsymbol{R}$ about the origin to place the epipole on the x-axis.
3. Map the epipole to infinity using a projective transform $\boldsymbol{G}$.

Detailed Transforms

The translation is defined by the origin (x_0, y_0). If known this is given by the principal point, otherwise the centre of the image is a good choice:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The rotation takes the epipole $\boldsymbol{e} = (e_1, e_2, 1)^T$ to a point on the x-axis $(e_x, 0, 1)^T$:

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{with} \quad \tan\theta = \frac{e_2}{e_1}$$

If the epipole is already at infinity $\boldsymbol{e} = (e_1, e_2, 0)^T$ we can still rotate it into the x-axis to obtain $(e_x, 0, 0)^T$.

The x-coordinate of the transformed epipole becomes: $e_x = \cos\theta\, e_1 + \sin\theta\, e_2$.

The Projective Transform

Finally we want to map the new epipole $\boldsymbol{e} = (e_x, 0, 1)^T$ to the point $\boldsymbol{i} = (1, 0, 0)^T$. This can be achieved using the projective transform:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{-1}{e_x} & 0 & 1 \end{bmatrix}$$

This maps the epipole to infinity as required. An arbitrary point $(x, y, 1)^T$ is mapped to $(x, y, 1 - \frac{x}{e_x})^T$. If $|\frac{x}{e_x}| < 1$ the mapping becomes the identity in first order.

The overall homography is then obtained as:

$$\boldsymbol{H} = \boldsymbol{G}\,\boldsymbol{R}\,\boldsymbol{T}$$

which is to the first order a rigid transformation and thus results in a small projective distortion.

Matching Transforms

The above scheme can be used to obtain individual homographies $\boldsymbol{H}$ and $\boldsymbol{H'}$. However these two are not independent and we seek to find a corresponding pair that also minimise the image distortions.

We are looking for homographies that match epipolar lines in both images. In other words the transformed lines shall be identical. Recall that lines are transformed using the dual $\boldsymbol{H}^{-T}$, then the requirement can be written as:

$$\boldsymbol{H}^{-T}\boldsymbol{l} = \boldsymbol{H'}^{-T}\boldsymbol{l'}$$

Consider a point $\boldsymbol{x}$ in the first image, then the epipolar line in the first image is given by $\boldsymbol{l} = [\boldsymbol{e}]_\times \boldsymbol{x}$. That in the second image is given by $\boldsymbol{l'} = \boldsymbol{F}\boldsymbol{x} = [\boldsymbol{e'}]_\times \boldsymbol{M}\boldsymbol{x}$. This holds for all $\boldsymbol{x}$ hence we have:

$$\boldsymbol{H}^{-T}[\boldsymbol{e}]_\times = \boldsymbol{H'}^{-T}[\boldsymbol{e'}]_\times \boldsymbol{M}$$

Matching Transforms

We use the following rule for commuting skew-symmetric matrices with non-singular matrices:

$$[\boldsymbol{t}]_\times \boldsymbol{A} = \boldsymbol{A}^{-T}[\boldsymbol{A}^{-1}\boldsymbol{t}]_\times$$

Inserting this gives the following relation:

$$[\boldsymbol{H}\boldsymbol{e}]_\times \boldsymbol{H} = [\boldsymbol{H}'\boldsymbol{e}']_\times \boldsymbol{H}'\boldsymbol{M}$$

this implies $\boldsymbol{H} = (\mathbb{1} + \boldsymbol{H}'\boldsymbol{e}'\boldsymbol{a}^T)\boldsymbol{H}'\boldsymbol{M}$ with an arbitrary vector $\boldsymbol{a}$.

In the our special case we have $\boldsymbol{H}'\boldsymbol{e}' = (1, 0, 0)^T$, hence:

$$(\mathbb{1} + \boldsymbol{H}'\boldsymbol{e}'\boldsymbol{a}^T) = \boldsymbol{H}_a = \begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

becomes an affine transformation.

Minimising Distortions

Given a previously computed homography $\boldsymbol{H}'$ we want to find a matching homographies $\boldsymbol{H}$ that minimise the sum of squared distances between transformed points:

$$\sum_i d(\boldsymbol{H}\boldsymbol{x}_i, \boldsymbol{H'}\boldsymbol{x'}_i)^2$$

We've seen before that $\boldsymbol{H} = \boldsymbol{H}_a \boldsymbol{H}_o$ with $\boldsymbol{H}_o = \boldsymbol{H'}\boldsymbol{M}$. Thus we can set $\hat{\boldsymbol{x}}'_i = \boldsymbol{H'}\boldsymbol{x}'_i = (\hat{x}_i, \hat{y}_i, 1)^T$ and $\hat{\boldsymbol{x}}_i = \boldsymbol{H}_o \boldsymbol{x}_i = (\hat{x}'_i, \hat{y}'_i, 1)^T$. The minimisation can then be written as:

$$\sum_i d(\boldsymbol{H}_a \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}'_i)^2 = \sum_i (a_1 \hat{x}_i + a_2 \hat{y}_i + a_3 - \hat{x}'_i)^2 + (\hat{y}_i - \hat{y}'_i)^2$$

because $(\hat{y}_i - \hat{y}'_i)^2$ is constant this reduces to the minimisation of:

$$\sum_i (a_1 \hat{x}_i + a_2 \hat{y}_i + a_3 - \hat{x}'_i)^2$$

Least Squares Solution

The minimisation is a standard least squares problem and can be written in the form:

$$\boldsymbol{A}\tilde{\boldsymbol{a}} = \mathbf{0} \quad \text{with} \quad \boldsymbol{A} = \begin{bmatrix} \hat{x}_1 & \hat{y}_1 & 1 & -\hat{x}'_1 \\ \vdots & \vdots & \vdots & \vdots \\ \hat{x}_n & \hat{y}_n & 1 & -\hat{x}'_n \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{a}} \sim \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix}$$

This is easily solved using the least squares techniques discussed in previous lectures.

Algorithm Summary

The complete algorithm to rectify a stereo image pair can be decomposed as follows:

1. Establish correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}'_i$.
2. Compute the fundamental matrix.
3. Determine homography $\boldsymbol{H}'$ that maps the epipole to infinity: $\boldsymbol{e}' \mapsto \boldsymbol{i}$.
4. Find matching homography $\boldsymbol{H}$ that minimises the sum of least squares distances.
5. Warp the first image (or point coordinates) with $\boldsymbol{H}$ and the second image using $\boldsymbol{H}'$.

Because the images are not treated entirely symmetrical, first $\boldsymbol{H}'$ is chosen, this algorithm might not always give optimal results. However it seems to work well in practise.

Optimal Rectification

Loop and Zhang [1999] propose an algorithm that explicitly minimises image distortions when building the homographies. Here only the fundamental matrix needs to be known, no corresponding points are needed. The minimisation is obtained by minimising the distortions over all image pixel. We won't discuss this method here but only state the general idea.

The homography is decomposed into subsequent projective and affine transformations, with the latter again being decomposed into a similarity and a shearing transform:

$$\boldsymbol{H} = \boldsymbol{H}_a \boldsymbol{H}_p = \boldsymbol{H}_s \boldsymbol{H}_r \boldsymbol{H}_p$$

Those individual transformations can then be chosen such as to minimise image distortions.

Next Lecture

- Matrix Representation of Perspective Geometry
- Reconstruction
- Error Propagation

References

References

A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

R. I. Hartley and A. Zisserman. *Multiple View geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. Technical Report MSR-TR-99-21, Microsoft Research, April 1999.