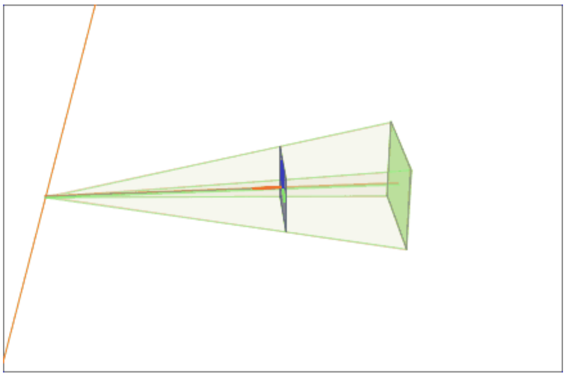
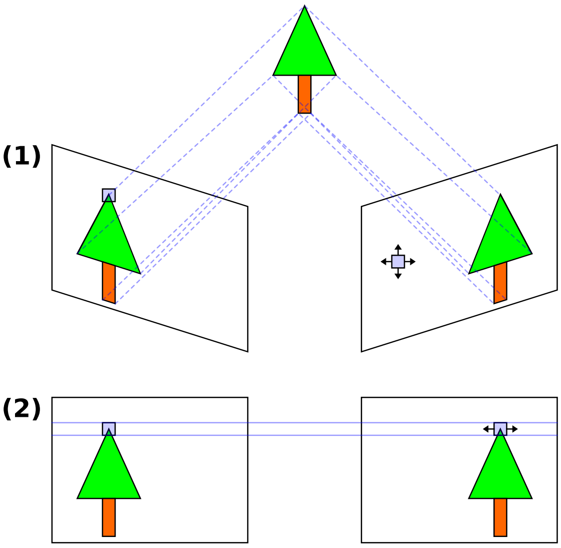
Image rectification

Image rectification is a transformation process used to project two-or-more images onto a common image plane. This process has several degrees of freedom and there are many strategies for transforming images to the common plane.

- It is used in computer stereo vision to simplify the problem of finding matching points between images (i.e. the correspondence problem).
- It is used in geographic information systems to merge images taken from multiple perspectives into a common map coordinate system.



A camera (red) rotates about the blue axis by 5° to 90° (green), as the images are rectified by projection to the virtual image plane (blue). The virtual plane must be parallel to the stereo baseline (orange) and for visualization is located in the center of rotation. In this case, rectification is achieved by a virtual rotation of the red and green image planes, respectively, to be parallel to the stereo baseline.



the search space before (1) and after (2) rectification

1 In computer vision

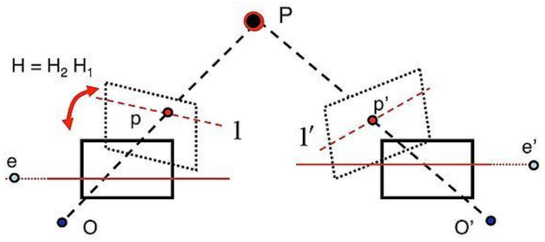
Stereo vision uses triangulation based on epipolar geometry to determine distance to an object. More specifically, binocular disparity is the process of relating the depth of an object to its change in position when viewed from a different camera, given the relative position of each camera is known.

With multiple cameras it can be difficult to find a corresponding point viewed by one camera in the image of the other camera (known as the correspondence problem). In most camera configurations, finding correspondences requires a search in two-dimensions. However, if the two cameras are aligned correctly to be coplanar, the search is simplified to one dimension - a horizontal line parallel to the line between the cameras. Furthermore, if the location of a point in the left image is known, it can be searched for in the right image by searching left of this location along the line, and vice versa (see binocular disparity). Image rectification is an equivalent (and more often used[1]) alternative to perfect camera alignment. Even with high-precision equipment, image rectification is usually performed because it may be impractical to maintain perfect alignment between cameras.

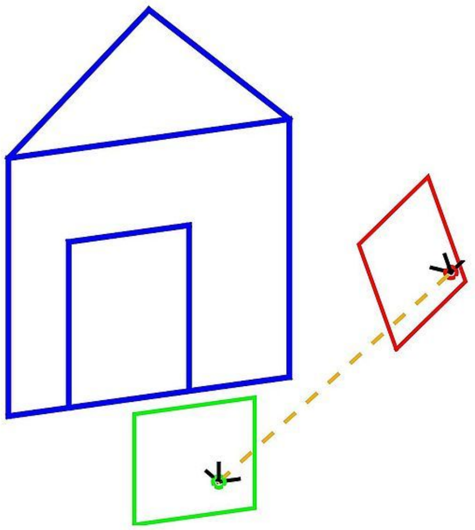
1.1 Transformation

If the images to be rectified are taken from camera pairs without geometric distortion, this calculation can easily be made with a linear transformation. X & Y rotation puts the images on the same plane, scaling makes the image frames be the same size and Z rotation & skew adjustments make the image pixel rows directly line up. The rigid alignment of the cameras needs to be known (by calibration) and the calibration coefficients are used by the transform.[2]

In performing the transform, if the cameras themselves are calibrated for internal parameters, an essential matrix provides the relationship between the cameras. The more general case (without camera calibration) is represented by the fundamental matrix. If the fundamental matrix is not known, it is necessary to find preliminary point correspondences between stereo images to facilitate its extraction.[2]



Model used for image rectification example. (image source Silvio Savarese)



3D view of example scene. The first camera's optical center and image plane are represented by the green circle and square respectively. The second camera has similar red representations.

1.2 Algorithms

There are three main categories for image rectification algorithms: planar rectification,[3] cylindrical rectification[1] and polar rectification.[4][5][6]

1.3 Implementation details

All rectified images satisfy the following two properties:[7]

- All epipolar lines are parallel to the horizontal axis.
- Corresponding points have identical vertical coordinates.

In order to transform the original image pair into a rectified image pair, it is necessary to find a projective transformation *H*. Constraints are placed on *H* to satisfy the two properties above. For example, constraining the epipolar lines to be parallel with the horizontal axis means that epipoles must be mapped to the infinite point *[1,0,0]*^{T} in homogeneous coordinates. Even with these constraints, *H* still has four degrees of freedom.[8] It is also necessary to find a matching *H'* to rectify the second image of an image pair. Poor choices of *H* and *H'* can result in rectified images that are dramatically changed in scale or severely distorted.

There are many different strategies for choosing a projective transform *H* for each image from all possible solutions. One advanced method is minimizing the disparity or least-square difference of corresponding points on the horizontal axis of the rectified image pair.[8] Another method is separating *H* into a specialized projective transform, similarity transform, and shearing transform to minimize image distortion.[7] One simple method is to rotate both images to look perpendicular to the line joining their collective optical centers, twist the optical axes so the horizontal axis of each image points in the direction of the other image's optical center, and finally scale the smaller image to match for line-to-line correspondence.[9] This process is demonstrated in the following example.

1.4 Example

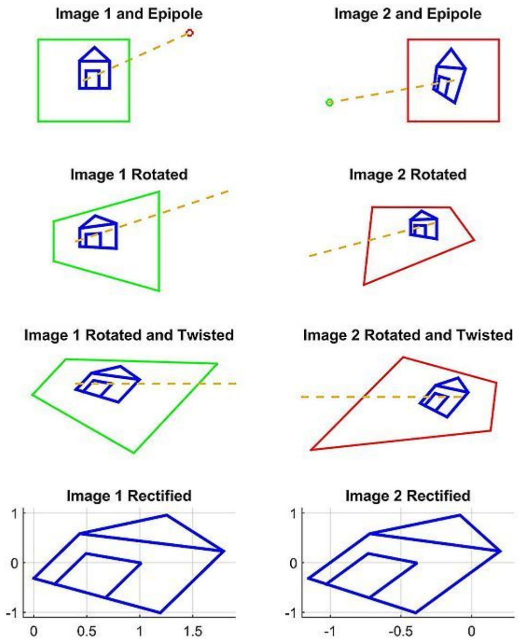
Our model for this example is based on a pair of images that observe a 3D point *P*, which corresponds to *p* and *p'* in the pixel coordinates of each image. *O* and *O'* represent the optical centers of each camera, with known camera matrices $M = K[I\ 0]$ and $M' = K'[R\ T]$ (we assume the world origin is at the first camera). We will briefly outline and depict the results for a simple approach to find a *H* and *H'* projective transformation that rectify the image pair from the example scene.

First, we compute the epipoles, *e* and *e'* in each image:

$$e = M\begin{bmatrix}O'\\1\end{bmatrix} = M\begin{bmatrix}-R^TT\\1\end{bmatrix} = K[I\,0]\begin{bmatrix}-R^TT\\1\end{bmatrix} = -KR^TT$$

$$e' = M'\begin{bmatrix}O\\1\end{bmatrix} = M'\begin{bmatrix}0\\1\end{bmatrix} = K'[R\,T]\begin{bmatrix}0\\1\end{bmatrix} = K'T$$

Second, we find a projective transformation H_1 that rotates our first image to be perpendicular to the baseline connecting *O* and *O'* (row 2, column 1 of 2D image set). This rotation can be found by using the cross product between the original and the desired optical axes.[9] Next, we find the projective transformation H_2 that takes the rotated image and twists it so that the horizontal axis aligns with the baseline. If calculated correctly, this second transformation should map the *e* to infinity on the x axis (row 3, column 1 of 2D image set). Finally, define $H = H_2H_1$ as the projective transformation for rectifying the first image.



Set of 2D images from example. The original images are taken from different perspectives (row 1). Using systematic transformations from the example (rows 2 and 3), we are able to transform both images such that corresponding points are on the same horizontal scan lines (row 4).

Third, through an equivalent operation, we can find H' to rectify the second image (column 2 of 2D image set). Note that H'_1 should rotate the second image's optical axis to be parallel with the transformed optical axis of the first image. One strategy is to pick a plane parallel to the line where the two original optical axes intersect to minimize distortion from the reprojection process.[10] In this example, we simply define H' using the rotation matrix R and initial projective transformation H as $H' = HR^T$.

Finally, we scale both images to the same approximate resolution and align the now horizontal epipoles for easier horizontal scanning for correspondences (row 4 of 2D image set).

Note that it is possible to perform this and similar algorithms without having the camera parameter matrices M and M' . All that is required is a set of seven or more image to image correspondences to compute the fundamental matrices and epipoles.[8]

2 Geographic information system

Image rectification in GIS converts images to a standard map coordinate system. This is done by matching ground control points (GCP) in the mapping system to points in the image. These GCPs calculate necessary image transforms.[11]

Primary difficulties in the process occur

- when the accuracy of the map points are not well known
- when the images lack clearly identifiable points to correspond to the maps.

The maps that are used with rectified images are non-topographical. However, the images to be used may contain distortion from terrain. Image orthorectification additionally removes these effects.[11]

Image rectification is a standard feature available with GIS software packages.

3 Reference implementations

This section provides external links to reference implementations of image rectification.

- stereoRectify and stereoRectifyUncalibrated (part of OpenCV), open source for Windows, Linux/Mac, Android, and iOS
- Rectification Kit and Uncalibrated Rectification Kit, MATLAB source code with tutorial by Andrea Fusiello[2]
- rectification, part of a MATLAB package by Du Huynh with a short tutorial[12]
- rectifyStereoImages, native MATLAB function in Computer Vision System Toolbox

4 See also

- Binocular disparity
- Correspondence problem
- Epipolar geometry
- Georeferencing
- Homography
- Point set registration
- Image registration
- Geographic information system
- Rubbersheeting
- Stereo camera
- Stereo vision
- Structure from motion

5 References

[1] Oram, Daniel (2001). "Rectification for Any Epipolar Geometry".

[2] Fusiello, Andrea (2000-03-17). "Epipolar Rectification". Retrieved 2008-06-09.

[3] Fusiello, Andrea; Trucco, Emanuele; Verri, Alessandro (2000-03-02). "A compact algorithm for rectification of stereo pairs" (PDF). *Machine Vision and Applications*. Springer-Verlag. **12**: 16–22. doi:10.1007/s001380050120. Retrieved 2010-06-08.

[4] Pollefeys, Marc; Koch, Reinhard; Van Gool, Luc (1999). "A simple and efficient rectification method for general motion" (PDF). *Proc. International Conference on Computer Vision*: 496–501. Retrieved 2011-01-19.

[5] Lim, Ser-Nam; Mittal, Anurag; Davis, Larry; Paragios, Nikos. "Uncalibrated stereo rectification for automatic 3D surveillance" (PDF). *International Conference on Image Processing*. **2**: 1357. Retrieved 2010-06-08.

[6] Roberto, Rafael; Teichrieb, Veronica; Kelner, Judith (2009). "Retificação Cilíndrica: um método eficente para retificar um par de imagens" (PDF). *Workshops of Sibgrapi 2009 - Undergraduate Works* (in Portuguese). Retrieved 2011-03-05.

[7] Loop, Charles; Zhang, Zhengyou (1999). "Computing rectifying homographies for stereo vision" (PDF). *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on*. Retrieved 2014-11-09.

[8] Richard Hartley and Andrew Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[9] Richard Szeliski (2010). *Computer vision: algorithms and applications*. Springer.

[10] David A. Forsyth and Jean Ponce (2002). *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference.

[11] Fogel, David. "Image Rectification with Radial Basis Functions". Retrieved 2008-06-09.

[12] Huynh, Du. "Polar rectification". Retrieved 2014-11-09.

1. R. I. Hartley (1999). "Theory and Practice of Projective Rectification". *Int. Journal of Computer Vision*. **35** (2): 115–127. doi:10.1023/A:1008115206617.

2. Pollefeys, Marc. "Polar rectification". Retrieved 2007-06-09.

3. Linda G. Shapiro and George C. Stockman (2001). *Computer Vision*. Prentice Hall. p. 580. ISBN 0-13-030796-3.

6 Further reading

- Computing Rectifying Homographies for Stereo Vision by Charles Loop and Zhengyou Zhang (April 8, 1999) Microsoft Research

7 Text and image sources, contributors, and licenses

7.1 Text

- **Image rectification** *Source:* https://en.wikipedia.org/wiki/Image_rectification?oldid=758255801 *Contributors:* Komap, MathMartin, Intgr, DaGizza, Realkyhick, SmackBot, KYN, Kolossos, Pascal.Tesson, Alaibot, User A1, Lgrove, Laoris, Lourakis, PbBot, Mxbuck, SchreiberBike, Addbot, Bagriffin, Fgnievinski, Aortezitron, Legobot, Gidoca, Tbbaron, Citation bot 1, Manusmad, RjwilmsiBot, Onoes, Timbobel, Jonpatterns, BG19bot, Tonyxc600, Ttnysetvold, BartVanAndel, Me, Myself, and I are Here, Pogojotz, Aaichert and Anonymous: 12

7.2 Images

- **File:2DRectificationBAG.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/7/70/2DRectificationBAG.jpg *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Bagriffin
- **File:3DRectificationBAG.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/4/4a/3DRectificationBAG.jpg *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Bagriffin
- **File:Image_rectification.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/9a/Image_rectification.svg *License:* CC BY-SA 3.0 *Contributors:* Inkscape *Original artist:* Bart van Andel
- **File:Lecture_1027_stereo_01.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/0/02/Lecture_1027_stereo_01.jpg *License:* Public domain *Contributors:* Lecture presentation for computer vision. Course lecturer has seen draft of wikipedia article with this image included and has encouraged me to upload to wikipedia. *Original artist:* Silvio Savarese
- **File:Planar_rectification_for_a_rotating_camera..gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/89/Planar_rectification_for_a_rotating_camera..gif *License:* CC BY-SA 4.0 *Contributors:* Own work *Original artist:* Aaichert

7.3 Content license

- Creative Commons Attribution-Share Alike 3.0