

Slimane LARABI

Computer Vision

From Bidimensional Images to Three
Dimensional Scene

January 02, 2025

Springer Nature

Chapter 1

The Homography: Computation and Applications

1.1 Introduction

The concept of homography plays a fundamental role in computer vision and image processing, providing a mathematical framework to model transformations between two-dimensional planes. Homography enables various powerful applications, from aligning images in panoramic stitching to correcting distortions and mapping perspectives. A homography is a 3×3 matrix that maps points from one plane to another, preserving straight lines, and it has many applications in computer vision.

We begin by briefly giving some applications of homography in computer vision. In the following, we explore the mathematical formulation of 2D transformations, introducing homogeneous coordinates and affine transformations. These concepts help to understand homography. Key properties of homographies are discussed, along with techniques to calculate them efficiently. The chapter ends with a practical example that illustrates the calculation and usefulness of homography in image stitching. It allows creating a panoramic image from several images taken by a rotating camera.

1.2 Applications of Homography

1.2.1 Perspective Correction

Homographies can correct perspective distortions, making objects in an image appear as if viewed from a frontal angle. For example, if we capture an image of a rectangular object at an angle, such as the book shown on the left in Figure 1.1, applying a homography can "undo" the perspective effect and restore its rectangular appearance. In this case, the transformation is applied only to the region bounded by the four red corners.



Fig. 1.1 Locating the region bounded by four red points and warping it into a rectangular region.

1.2.2 Object Tracking

Homography is widely used in tracking objects across frames in a video. Since it captures the relationship between planes, it can be used to estimate the motion of flat surfaces or objects. For example, tracking a logo on a soccer field across multiple frames relies on homographies to account for camera motion and field distortion [3].

As is well known, the fundamental matrix defines the relationship between a point in one image and the corresponding epipolar line in another image, whereas the homography establishes a one-to-one correspondence between points in two images.

This distinction is used by the authors in [3] to minimize the error in player location estimation when using multiple cameras. By employing homography, tracking is performed more reliably and stably compared to methods based on the fundamental matrix [3].

1.2.3 Augmented Reality (AR)

Homography help anchor virtual objects onto surfaces in real-world scenes by mapping a flat image onto a plane. This allows for convincing overlays in AR, where virtual objects appear to interact realistically with real-world surfaces. Figure 1.2 shows two examples of mapping flat image onto a plane [2].



Fig. 1.2 Mapping flat image onto a plane [2].

1.2.4 Camera Calibration and Stereo Vision

In stereo vision systems, homography is used to align images from two cameras. By calculating homographies, we can rectify stereo images, making it easier to extract depth information and produce disparity maps, which are essential for 3D reconstruction and depth perception (see figures 1.3, 1.4, 1.5).

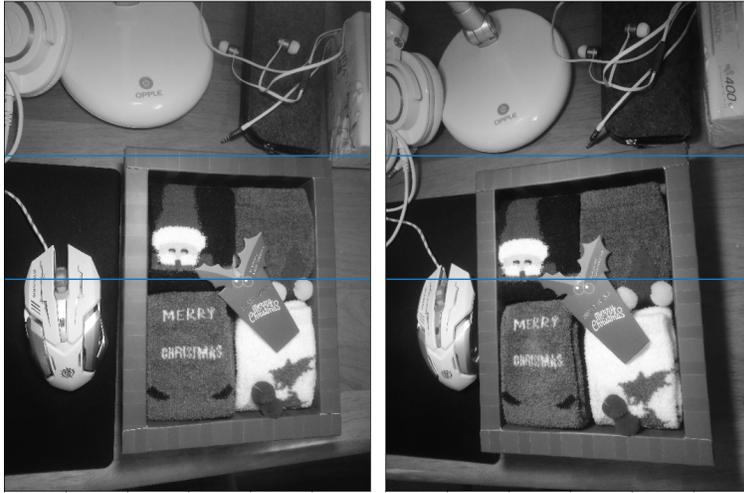


Fig. 1.3 The initial stereo images.

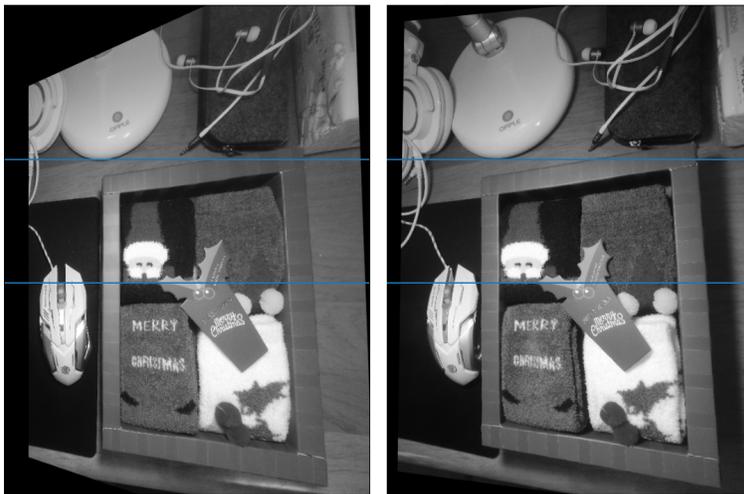


Fig. 1.4 The rectified stereo images. Note that for each point in the left image, its match belongs to the horizontal epipolar line (indicated with blue line).

1.2.5 Image Rectification, Alignment and Stitching

When two images of a scene are taken from slightly different viewpoints, homography can rectify the images, making them appear as if taken from the same viewpoint. This is especially useful in stereo vision and for improving image analysis accuracy.



Fig. 1.5 The computed depths using the rectified images based on the horizontal disparity of matched points .

We assume that we take a set of images of the scene from the same point of view by rotating the camera slightly (see Figure 1.6). Following this rotation of the camera, the fields of view of the captured images overlap. The objective is to create a panoramic image from this set of images. By finding the homography matrix indicating the mapping from one plane to another, the points of one image can be mapped to the corresponding points of another, allowing the creation of the panoramic image (see figure 1.7).

1.3 Mathematical writing of 2D transformation of images.

Different image transformations can be represented using matrices. The transformation of a pixel $p_1(x_1, y_1)$ to a new pixel $p_2(x_2, y_2)$ can be expressed using 2×2 matrix:



Fig. 1.6 Three successive images of my desk.



Fig. 1.7 The panoramic image made from the three images.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (1.1)$$

For example, scaling, rotation, horizontal skew, and vertical skew transformations correspond to the matrices T_s , T_r , T_{hk} , T_{vk} , respectively, as defined by the following equations. Here, θ is the rotation angle, m_x , m_y denotes the scale factors.

$$T_s = \begin{bmatrix} e_{11} & 0 \\ 0 & e_{22} \end{bmatrix}, T_r = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, T_{hk} = \begin{bmatrix} 1 & m_x \\ 0 & 1 \end{bmatrix}, T_{vk} = \begin{bmatrix} 1 & 0 \\ m_y & 1 \end{bmatrix} \quad (1.2)$$

Figure 1.8 shows examples of such transformations.



Fig. 1.8 From left to right: Origin image, application of scale, rotation, horizontal and vertical skew transformations.

These 2D transformations have some properties, we can cite:

- The origin is mapped to the origin,
- Lines are mapped to lines,
- Parallel lines remain parallel,
- The 2D transformation is closed under composition. This means that the composition of two transformations, the result is still a transformation of the same type (e.g., a 2D transformation). Example, if we perform a rotation followed by a translation, this is identical to the case where we perform a 2D transformation such that the corresponding matrix is equal to the multiplication of the two 2D transformation matrices.

1.3.1 Homogeneous coordinates

It is not possible, using a 2×2 matrix, to describe a 2D translation which is also a 2D transformation. To remedy this limitation, representing the 2D transformation with an additional dimension is a solution.

To do this, **homogeneous coordinates** are used where a 2D point $p(x, y)$ is represented by a 3D point $\tilde{p}(\tilde{x}, \tilde{y}, \tilde{z})$ where the third coordinate is fictitious such that:

$$x = \frac{\tilde{x}}{\tilde{z}}, y = \frac{\tilde{y}}{\tilde{z}} \quad (1.3)$$

In the geometric illustration shown in Figure 1.9, all points on the line (L) are equivalent to one another. Each point $\tilde{p}(\tilde{x}, \tilde{y}, \tilde{z})$ on (L), except the origin, can be

mapped to $p(x, y)$ by dividing its coordinates of by a scaling factor \tilde{z} (see figure 1.9). This implies that the homogeneous points $\tilde{p}(\tilde{x}, \tilde{y}, \tilde{z})$ and $\tilde{p}'(\lambda\tilde{x}, \lambda\tilde{y}, \lambda\tilde{z})$ correspond to the same point in the Cartesian coordinate system Oxy .

$$p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\tilde{z} \\ y\tilde{z} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \tilde{p} \quad (1.4)$$

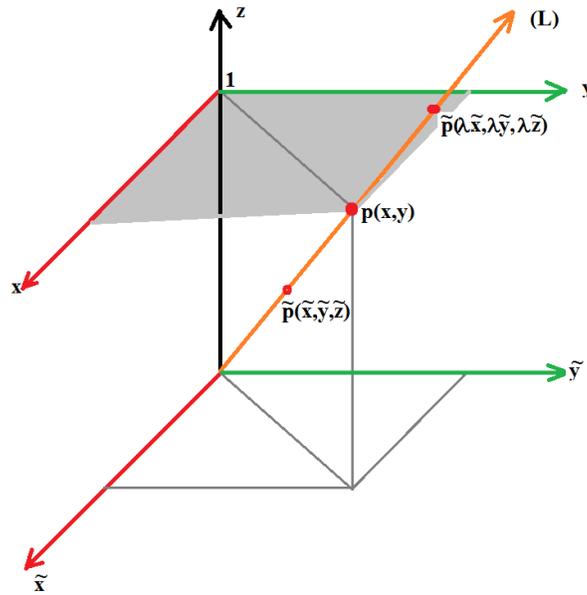


Fig. 1.9 Homogeneous coordinate of the point p .

Using homogeneous coordinates, we can easily write the different 2D transformation, as example, equations 1.5 and 1.6 describe the scaling and translation transformations.

$$\text{Scaling} : \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1.5)$$

$$\text{Translation : } \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (1.6)$$

Remark:

To combine multiple transformations (e.g., rotation, scaling, and translation) into a single transformation, the individual transformations can be multiplied to obtain a (3×3) transformation matrix.

1.3.2 Affine transformation

We define an affine space as a geometric structure consisting of points and vectors, where vectors describe displacements between points, but no single point is distinguished as the origin.

To model transformations and geometric relationships in computer vision we need to use affine transformation which is a function that maps an object from an affine space to an other and which preserve structures. Indeed, an affine transformation preserves lines or distance ratios but changes the orientation, size or position of the object [33].

The set of affine transformation is composed of various operations. Translations which modify object position in the image. Homothetic Transformations composed of the contraction and dilatation of an object, both scaling operations. The transvection (shear mapping) which modify position of an object. Rotation which allows to rotate an object according to it's axis. And a whole set of transformation produced by combining all of the previous [33].

Any transformation affine is written as described by equation 1.7.

$$\begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} \quad (1.7)$$

Affine transformation have some properties, some ones are:

- Origin does not necessarily maps to the origin,
- Lines map to lines,
- Parallel lines remain parallel,
- Closed under composition.

1.4 Homography

1.4.1 Definition

Homography is the transformation matrix that maps one plane to another through a point of projection, as described by equation 1.8. Figure 1.10 shows an example of homography applied to an image.

$$\tilde{p}_2 = \begin{bmatrix} \tilde{x}_2 \\ \tilde{y}_2 \\ \tilde{z}_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \tilde{z}_1 \end{bmatrix} = H\tilde{p}_1 \quad (1.8)$$

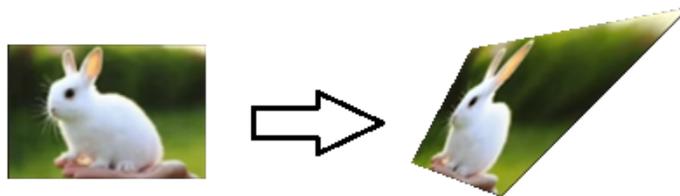


Fig. 1.10 Applying an homography to the image in the left.

1.4.2 Properties of Homography

Homography can only be defined up to scale. Suppose that $\tilde{p}_2 = H\tilde{p}_1$ as illustrated by figure 1.11. Then by elementary properties of matrix multiplication, we have:

$$(kH)\tilde{p}_1 = H(k\tilde{p}_1) = H\tilde{p}'_1 = \tilde{p}_2.$$

so when $k \neq 0$, $H\tilde{p}_1$ and $(kH)\tilde{p}_1$ represent the same point.

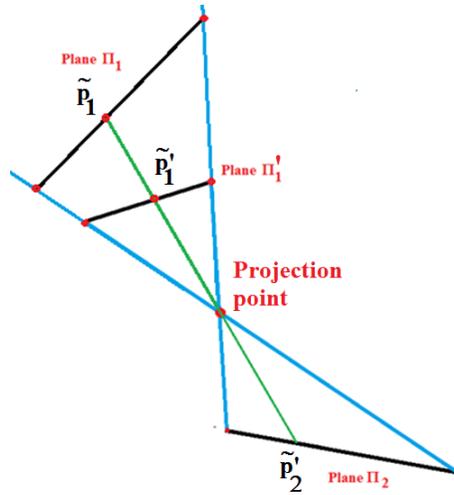


Fig. 1.11 Example of Homography producing infinite points using the same transformation matrix.

If we fix h_{33} or we set to 1 the root of the sum of square of all parameters of H , then 8 parameters are needed to estimate the matrix H .

Some properties are verified for Homography, we can cite:

- Origin does not necessarily maps to the origin,
- Lines map to lines,
- Parallel lines does not necessarily remain parallel,
- Closed under composition.

1.4.3 Computing Homography

Hypothesis:

Images are acquired from the same view point or the scene points should lie on a same plane, or the scene is really far away (scene is a plane at infinity).

In order to find the Homography that map one image to another, we can use the pairs of matched SIFT descriptors ($p_s(x_s, y_s), p_d(x_d, y_d)$) in both images (see figure 1.12).

Each pair of matched points ($p_s(x_s, y_s), p_d(x_d, y_d)$) allow to write the expression given by equation 1.9 and give us two linear equations (1.10, 1.11).

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{x}_d \\ \tilde{y}_d \\ \tilde{z}_d \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \quad (1.9)$$

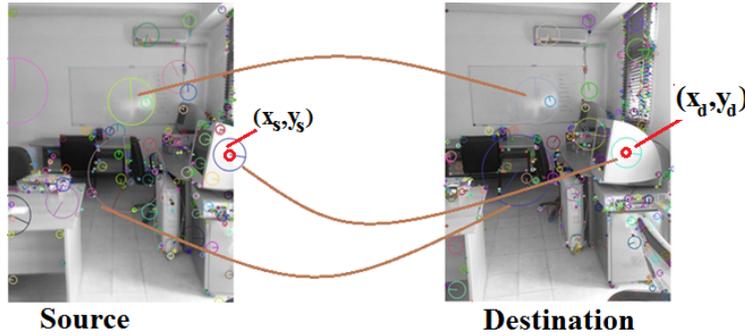


Fig. 1.12 SIFT key points of two images (source and destination).

As we need to determine 8 parameters, 4 pairs of matched points are sufficient to compute the homography.

$$x_d = \frac{(h_{11}x_s + h_{12}y_s + h_{13})}{(h_{31}x_s + h_{32}y_s + h_{33})} \quad (1.10)$$

$$y_d = \frac{(h_{21}x_s + h_{22}y_s + h_{23})}{(h_{31}x_s + h_{32}y_s + h_{33})} \quad (1.11)$$

We can write these equations as follow:

$$h_{31}x_s \cdot x_d + h_{32}y_s \cdot x_d + h_{33}x_d - h_{11}x_s - h_{12}y_s - h_{13} = 0 \quad (1.12)$$

$$h_{31}x_s \cdot y_d + h_{32}y_s \cdot y_d + h_{33}y_d - h_{21}x_s - h_{22}y_s - h_{23} = 0 \quad (1.13)$$

Using 4 pairs of matched points, we obtain the following system where (x_{si}, y_{si}) , (x_{di}, y_{di}) are the coordinates of the pair of matched points:

$$\begin{pmatrix} x_{s1} & y_{s1} & 1 & 0 & 0 & 0 & -x_{s1}x_{d1} & -x_{d1}y_{s1} & -x_d \\ 0 & 0 & 0 & x_{s1} & y_{s1} & 1 & -x_{s1}y_{d1} & -y_{s1}y_{d1} & -y_{d1} \\ x_{s2} & y_{s2} & 1 & 0 & 0 & 0 & -x_{s2}x_{d2} & -x_{d2}y_{s2} & -x_d \\ 0 & 0 & 0 & x_{s2} & y_{s2} & 1 & -x_{s2}y_{d2} & -y_{s2}y_{d2} & -y_{d1} \\ x_{s3} & y_{s3} & 1 & 0 & 0 & 0 & -x_{s3}x_{d3} & -x_{d3}y_{s3} & -x_d \\ 0 & 0 & 0 & x_{s3} & y_{s3} & 1 & -x_{s3}y_{d3} & -y_{s3}y_{d3} & -y_{d1} \\ x_{s4} & y_{s4} & 1 & 0 & 0 & 0 & -x_{s4}x_{d4} & -x_{d4}y_{s4} & -x_d \\ 0 & 0 & 0 & x_{s4} & y_{s4} & 1 & -x_{s4}y_{d4} & -y_{s4}y_{d4} & -y_{d1} \end{pmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (1.14)$$

The form of the equation 1.14 is $Ah = 0$ with the constraint $\|h\|^2 = 1$. The solution may be obtained solving the problem $Min_h(\|Ah\|^2)$ such that $\|h\|^2 = 1$. The solution consists to choose the eigen vector h with smallest value of λ of $A^T A$ which minimize the loss function $L = \|h^T A^T A h\|^2$ [6].

1.4.4 Dealing with outliers

The used four pair of matched key points may present one or more wrong matches. In order to robustly compute transformation in presence of these wrong matches, we apply RanSAC algorithm.

RANSAC Algorithm [4]

- 1- Randomly choose S samples. Typically, S is the minimum samples to fit a model.
- 2- Fit the model to the randomly chosen samples.
- 3- Count the number M of data points (inliers) that fit the model within a measure of error.
- 4- Repeat actions (1, 2, 3) N times.
- 5- Choose a model that has the largest number M of inliers.

In practice, if we assume that we have $m - 4$ pairs of matched pairs of key points, for each four pairs of matched key points, we compute the homography and for all $m - 4$ pairs of matched points (p_i, p'_i) we compute the sum of errors $|H(p_i) - p'_i|$. At the end, we select the best basis of used four pairs of matched key points.

1.4.5 Examples of Homography computing

Example 1

Using homography, we can map a selected region using four points from one image into a destination image whose width and length are known. At the left of Figure 1.13, we select a region by clicking on four points (shown with red color). The destination image is a rectangular region. The homography is computed such that each red point will correspond to a corner of the destination region. The next step is to perform a warping and the obtained result is shown at the right side of the same figure.

Using OpenCV and Python, these two steps are coded as follow:

```
tform, mask = cv2.findHomography(PtsSource, PtsDestination)
```

where **mask** is an array of same length as input points, indicates inliers (which points were actually used in the best computation of the homography **tform**).

```
ImageDestination = cv2.warpPerspective(ImageSource, tform, (width, height))
```

warpPerspective: applies a perspective transformation to an image. ImageSource is the source image (left in figure 1.13).

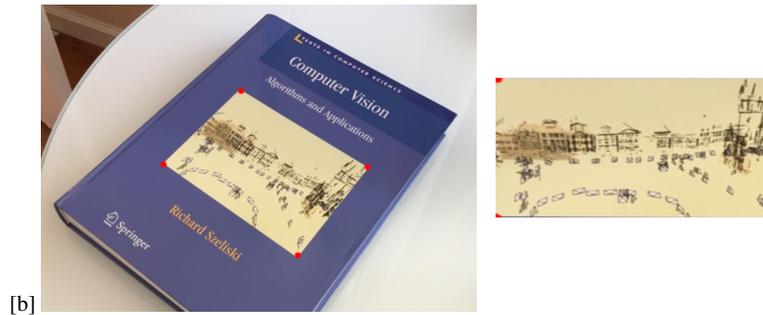


Fig. 1.13 Locating a first region and warping it into a rectangular region.

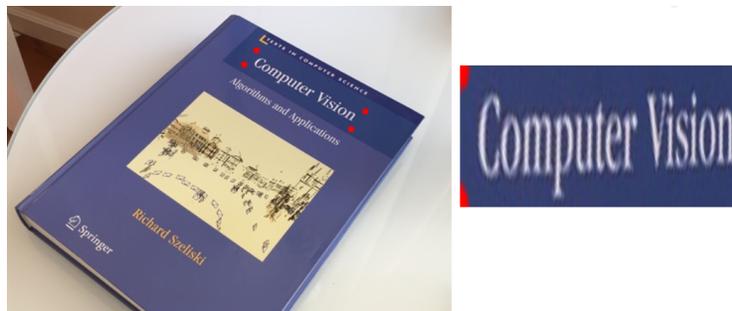


Fig. 1.14 Locating a second region and warping it into a rectangular region.

Example 2

The second example allows to map an image source into an area made up of four points selected on the destination image by the user.



Fig. 1.15 Locating a first region and warping it into a rectangular region.

1.4.6 Images stitching using Homography

The process of image stitching is performed as follow:

- First, two images are taken by rotating a camera at the same position (see figure 1.16).
- The next step is to locate key points in the two images. Figure 1.17 shows the SIFT points located and matched.
- Compute the homography which allow to coincide the SIFT key points of the second image with the matched ones in the first image.
- Apply the homography and warp the second image in order to obtain this overlapping of matched key points as shown by figure 1.18.
- Concatenate the warped image with the reference image (first one) as shown by figure 1.19.

1.5 Conclusion

In this chapter, we presented the fundamentals of homography and its applications in computer vision. The main result of this section is that, to map one plane to another, we need at least four pairs of matched points. This allows us to calculate the



Fig. 1.16 Two images taking at USTHB University campus by rotating a camera.

homography matrix, enabling the transformation of the first plane to align with the second.

In practical applications, matched points are often derived from features like SIFT. To account for possible mismatches, we apply the RANSAC algorithm, which selects the best homography based on the optimal set of four matched points.

We will see in the following chapters that homography is also useful for many other problems, such as fundamental matrix computation and image rectification for depth map computation. The next chapter is devoted to camera calibration and its applications.



Fig. 1.17 Locating and matching SIFT points.



Fig. 1.18 Warping the second image.



Fig. 1.19 Result of images stitching

References

1. Seymour Papert. The Summer Vision Project. Artificial Intelligence Group, MIT, 1966.
<https://dspace.mit.edu/handle/1721.1/11589>
2. Bob Sumner. Augmented Creativity, TEDxZurich, 19th of January, 2015.
<https://www.youtube.com/watch?v=AJJOWemfOYI>
3. Sachiko Iwase and Hideo Saito. Tracking soccer players based on homography among multiple views”, Proc. SPIE 5150, Visual Communications and Image Processing 2003, (23 June 2003);
<https://doi.org/10.1117/12.502967>
4. Martin A. Fischler, Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, Volume 24, Issue 6, pp 381–395, <https://doi.org/10.1145/358669.358692>
5. ”The Eye of a Robot: Studies in Machine Vision at MIT” and ”TX-O Computer”. Courtesy of MIT Museum.
6. Richard Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.
7. D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), volume 1, pages 195-202, Madison, WI, June 2003.
8. Zhengyou Zhang. A Flexible New Technique for Camera Calibration. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 22, NO. 11, NOVEMBER 2000
9. A computer algorithm for reconstructing a scene from two projections. Nature volume 293, pages 133–135 (1981)
10. OD Faugeras, QT Luong, SJ Maybank. Camera self-calibration: Theory and experiments. Second European Conference on Computer Vision ECCV’92, 1992
11. Olsson, Carl; Enqvist, Olof, Stable Structure from Motion for Unordered Image Collections Scandinavian Conference on Image Analysis, 2011.
12. Marc Pollefeys and Luc Van Gool. 3-D Modeling from Images. COMMUNICATIONS OF THE ACM July 2002/Vol. 45, No. 7.
13. Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Xin ; O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis , July 2017. ACM Transactions on Graphics, 36(4):1-11
14. M. Michalkiewicz, J. K. Pontes, D. Jack, M. Baktashmotlagh, A. Eriksson. Deep Level Sets: Implicit Surface Representations for 3D Shape Inference. 2019. arXiv:1901.06802 [cs.CV].
<https://arxiv.org/pdf/1901.06802.pdf>

15. A. Podlozhnyuk, S. Pirker, C. Kloss. Efficient implementation of superquadric particles in Discrete Element Method within an open-source framework. *Computational Particle Mechanics* 4(1), 2016. DOI: 10.1007/s40571-016-0131-6
16. M.A. Turk et al. Face recognition using eigenfaces, CVPR 1991.
17. S.K.Nayar et al., Real-Time 100 Object Recognition System, ICRA, 1996
18. C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3dr2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016
19. R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016
20. D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems*, pages 4996–5004, 2016
21. S. R. Richter and S. Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018
22. C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 1(2):4, 2017.
23. Y. Liao, S. Donne, and A. Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018
24. H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, volume 2, page 6, 2017
25. Mateusz Michalkiewicz et al. Deep Level Sets: Implicit Surface Representations for 3D Shape Inference. arXiv:1901.06802v1 [cs.CV], 21 Jan 2019.
26. Roberts, Lawrence G. 1963. Machine perception of three-dimensional solids. Outstanding PhD dissertations in the computer sciences. Garland Publishing, New York. 1963.
27. Adolfo Guzman-Aréñas, Computer Recognition of Three-Dimensional Objects In a Visual Scene. PhD Dissertations in the computer sciences, MIT 1968.
28. M.B. Clowes. On seeing things. *Artificial Intelligence*, Volume 2, Issue 1, Spring 1971, Pages 79-116.
29. David L. Waltz. GENERATING SEMANTIC DESCRIPTIONS FROM DRAWINGS OF SCENES WITH SHADOWS. MIT Artificial Intelligence Laboratory. Technical Report 271, November 1972.

30. Kemp, M. Julesz's joyfulness. *Nature* 396, 419 (1998). <https://doi.org/10.1038/24753>
31. Julesz B. *Foundations of Cyclopean Perception*. Chicago University Press; Chicago, IL, USA: 1971.
32. David Marr. *Vision, A Computational Investigation into the Human Representation and Processing of Visual Information*. Originally published: San Francisco : W. H. Freeman, c1982.
33. Arthur Coste. *Affine Transformation, Landmarks registration, Non linear Warping*. https://www.sci.utah.edu/~acoste/uou/Image/project3/ArthurCOSTE_Project3.pdf October 2012.
34. Sid Yingze Bao, Mohit Bagra, Yu-Wei Chao, Silvio Savarese. *Semantic Structure From Motion with Points, Regions, and Objects*. CVPR 2012.
35. Marco Crocco, Cosimo Rubino, Alessio Del Bue. *Structure from Motion with Objects*, CVPR 2016.
36. D.P. Robertson and R. Cipolla. *Structure from Motion*. In Varga, M., editors, *Practical Image Processing and Computer Vision*, John Wiley, 2009.
37. C. Tomasi and T. Kanade. *Shape and motion from image streams under orthography: A factorization method*. *International Journal of Computer Vision*, 9(2):137–154, 1992.
38. P. F. Sturm and W. Triggs. *A factorization based algorithm for multi-image projective structure and motion*. In *European Conference on Computer Vision (ECCV'96)*, pages 709–720, 1996.
39. F. Schaffalitzky, A. Zisserman, R. I. Hartley, and P. H. S. Torr. *A six point solution for structure and motion*. In *European Conference on Computer Vision (ECCV'00)*, pages 632–648, 2000.
40. W. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. *Bundle adjustment: A modern synthesis*. In W. Triggs, A. Zisserman, and R Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. Springer Verlag, 2000.
41. D. C. Brown. *The bundle adjustment - progress and prospects*. *International Archives of Photogrammetry*, 21(3), 1976.