MVA '98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

12—5
Optimal Homography Computation with a Reliability Measure

Kenichi Kanatani*
Department of Computer Science, Gunma University

Abstract

This paper describes an algorithm for optimally computing the homography between two given sets of points on a plane and evaluating the reliability of the resulting mapping. The basic principle is maximum likelihood estimation by a technique called renormalization.

1. Introduction

A *homography* is a mapping that occurs between two perspective images of a planar surface in the scene [4]. The computation of homographies plays an essential role in image registration and mosaicing [10]. By computing the homography between two images of a planar surface, we can obtain information about the 3-D position and orientation of the surface and the camera motion [8, 9, 12]. Homographies also play an essential role in scene understanding [1, 3, 11] and calibration [2].

In the past, the least-squares method has been frequently used, but it has been pointed out that the least-squares solution has statistical bias [5, 8]. We apply the statistical optimization theory of Kanatani [5, 7] to homography estimation and derive an algorithm that can not only compute a homography optimally but also also evaluate the reliability of the computed solution in quantitative terms.

2. Homography Computation

A homography is an image mapping in the form

$$x' = \frac{Ax + By + C}{Px + Qy + R}, \quad y' = \frac{Dx + Ey + F}{Px + Qy + R}. \tag{1}$$

In terms of 3-D vectors $\boldsymbol{x} = (x/f, y/f, 1)^\top$ and $\boldsymbol{x}' = (x'/f, y'/f, 1)^\top$ (the superscript $\top$ designates transpose), eqs. (1) can be written as

$$\boldsymbol{x}' = Z[\boldsymbol{H}\boldsymbol{x}], \quad \boldsymbol{H} = \begin{pmatrix} A & B & C/f \\ D & E & F/f \\ P & Q & R/f \end{pmatrix}, \tag{2}$$

where f is a scale factor defined in such a way that x/f and y/f have order of 1 and $Z[\cdot]$ denotes the normalization to make the Z component 1. In order to remove scale indeterminacy, we adopt the normalization $\|\hat{\boldsymbol{H}}\| = 1$, where the norm of matrix $\boldsymbol{H}$ $= (H_{ij})$ is defined by $\|\boldsymbol{H}\| = \sqrt{\sum_{i,j=1}^{3} H_{ij}{}^2}$.

We regard data points as random variables. Let $\boldsymbol{\Sigma}_\alpha$ and $\boldsymbol{\Sigma}'_\alpha$ be the covariance matrices of points (x_α, y_α) and (x'_α, y'_α), respectively. The covariance matrices of the corresponding vectors $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$ are

$$V[\boldsymbol{x}_\alpha] = \begin{pmatrix} \boldsymbol{\Sigma}_\alpha & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}, \quad V[\boldsymbol{x}'_\alpha] = \begin{pmatrix} \boldsymbol{\Sigma}'_\alpha & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}, \tag{3}$$

We assume that these covariance matrices are known only *up to scale* and write

$$V[\boldsymbol{x}_\alpha] = \epsilon^2 V_0[\boldsymbol{x}_\alpha], \quad V[\boldsymbol{x}'_\alpha] = \epsilon^2 V_0[\boldsymbol{x}'_\alpha]. \tag{4}$$

We call ϵ the *noise level* and the matrices $V_0[\boldsymbol{x}_\alpha]$ and $V_0[\boldsymbol{x}'_\alpha]$ the *normalized covariance matrices*, which specify the relative dependence of noise occurrence on positions and orientations. If no prior knowledge is available for them, we simply assume $V_0[\boldsymbol{x}_\alpha] = V_0[\boldsymbol{x}'_\alpha] = \mathrm{diag}(1, 1, 0)$, where $\mathrm{diag}(\cdots)$ designates the diagonal matrix with diagonal elements $\cdots$ in that order.

The computed matrix $\boldsymbol{H}$ is also a random variable. We consider, in the parameter space of $\boldsymbol{H}$, the orientation along which error is the most likely to occur and on it take two points a standard deviation apart from the mean in both directions. We call the corresponding matrices, $\{\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)}\}$, the *primary deviation pair* [5].

3. Algorithm

The computational technique described below is called *renormalization* and is shown to be optimal in the first order [5, 6].

Input: Two sequences of points represented by vectors $\{\boldsymbol{x}_\alpha\}_{\alpha=1}^N$ and $\{\boldsymbol{x}'_\alpha\}_{\alpha=1}^N$ together with their normalized covariance matrices $\{V_0[\boldsymbol{x}_\alpha]\}_{\alpha=1}^N$ and $\{V_0[\boldsymbol{x}'_\alpha]\}_{\alpha=1}^N$ $(N \geq 4)$.

Output: An optimal estimate $\hat{\boldsymbol{H}}$ and its primary deviation pair $\{\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)}\}$.

Procedure:

*Address: Kiryu, Gunma 376-8515 Japan. E-mail: kanatani@cs.gunma-u.ac.jp

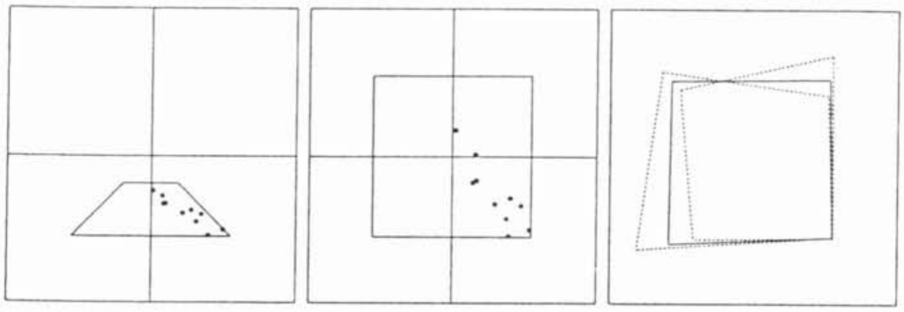


Figure 1: (a) A quadrilateral region (perspective image). (b) A rectangular region (true shape). (c) Boundary image (solid lines) and its standard deviation pair (dotted lines).

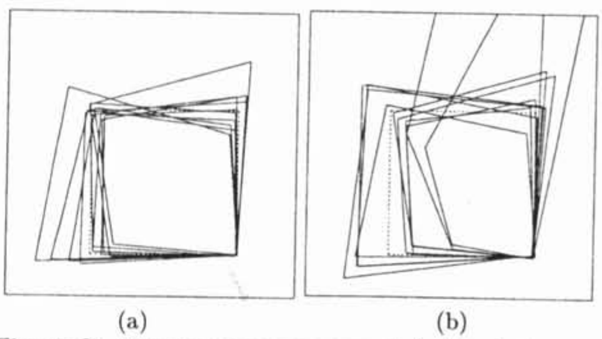


Figure 2: Boundary images mapped by ten instances of the computed homography (solid lines) and the true shape (dotted lines). (a) Our algorithm. (b) Least-squares method.

1. Let $c = 0$ and $\boldsymbol{W}_\alpha = \boldsymbol{I}$, $\alpha = 1, ..., N$.
2. Define the following tensor $\mathcal{M}$:
$$\mathcal{M} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{k,l=1}^{3} W_\alpha^{(kl)}(\boldsymbol{e}^{(k)} \times \boldsymbol{x}'_\alpha \boldsymbol{x}_\alpha^\top) \otimes(\boldsymbol{e}^{(l)} \times \boldsymbol{x}'_\alpha \boldsymbol{x}_\alpha^\top). \tag{5}$$
3. Compute the following tensor $\mathcal{N} = (N_{ijkl})$:
$$N_{ijkl} = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{m,n,p,q=1}^{3} \varepsilon_{imp}\varepsilon_{knq} W_\alpha^{(mn)} \Big(V_0[\boldsymbol{x}_\alpha]_{jl} x'_{\alpha(p)} x'_{\alpha(q)} + V_0[\boldsymbol{x}'_\alpha]_{pq} x_{\alpha(j)} x_{\alpha(l)}\Big). \tag{6}$$
4. Compute the nine eigenvalues $\lambda_1 \geq \cdots \geq \lambda_9$ of tensor
$$\hat{\mathcal{M}} = \mathcal{M} - c\mathcal{N} \tag{7}$$
and the corresponding orthonormal system of eigenmatrices $\{\boldsymbol{H}_1, ..., \boldsymbol{H}_9\}$ of unit norm.
5. If $\lambda_9 \approx 0$, return λ_8, $\boldsymbol{H}_8$, $\boldsymbol{H}_9$, and c.
6. Else, update c and $\boldsymbol{W}_\alpha$ in the following way and go back to Step 2:
$$c \leftarrow c + \frac{\lambda_9}{(\boldsymbol{H}_9; \mathcal{N}\boldsymbol{H}_9)}, \tag{8}$$
$$\boldsymbol{W}_\alpha \leftarrow \Big(\boldsymbol{x}'_\alpha \times \boldsymbol{H}_9 V_0[\boldsymbol{x}_\alpha]\boldsymbol{H}_9^\top \times \boldsymbol{x}'_\alpha + (\boldsymbol{H}_9\boldsymbol{x}_\alpha) \times V_0[\boldsymbol{x}'_\alpha] \times (\boldsymbol{H}_9\boldsymbol{x}_\alpha)\Big)_2^-. \tag{9}$$
7. Estimate the square noise level by
$$\hat{\epsilon}^2 = \frac{c}{1 - 4/N}. \tag{10}$$
8. Let $\hat{\boldsymbol{H}} = \boldsymbol{H}_9$, and compute
$$\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)} = N[\hat{\boldsymbol{H}} \pm \sqrt{\frac{\hat{\epsilon}^2}{\lambda_8 N}}\boldsymbol{H}_8]. \tag{11}$$
9. Return $\hat{\boldsymbol{H}}$ and $\{\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)}\}$.

Symbols and notations:

The letter $\boldsymbol{I}$ denotes the unit matrix. The symbols $x_{\alpha(i)}$ and $x'_{\alpha(i)}$ denote the ith components of $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}'_\alpha$, respectively. The symbol $W_\alpha^{(kl)}$ denotes the (kl) element of $\boldsymbol{W}_\alpha$. For $\boldsymbol{A} = (A_{ij})$ and $\boldsymbol{B} = (B_{ij})$, the product $\mathcal{C} = \boldsymbol{A} \otimes \boldsymbol{B}$ is a tensor with elements $C_{ijkl} = A_{ij}B_{kl}$. We define $\boldsymbol{e}^{(1)} = (1, 0, 0)^\top$, $\boldsymbol{e}^{(2)} = (0, 1, 0)^\top$, and $\boldsymbol{e}^{(3)} = (0, 0, 1)^\top$. The symbols $V_0[\boldsymbol{x}_\alpha]_{ij}$ and $V_0[\boldsymbol{x}'_\alpha]_{ij}$ denote the (ij) elements of the normalized covariance matrices $V_0[\boldsymbol{x}_\alpha]$ and $V_0[\boldsymbol{x}'_\alpha]$, respectively. The symbol ε_{ijk} denotes the Eddington epsilon, taking 1 and -1 if (ijk) is an even and odd permutation, respectively, and 0 otherwise.

The product $\hat{\mathcal{M}}\boldsymbol{H}$ of tensor $\hat{\mathcal{M}} = (\hat{M}_{ijkl})$ and matrix $\boldsymbol{H} = (H_{ij})$ is a matrix whose (ij) element is $\sum_{k,l=1}^{3} \hat{M}_{ijkl}H_{kl}$. We say that $\boldsymbol{H}$ is an *eigenmatrix* of $\hat{\mathcal{M}}$ for *eigenvalue* λ if $\hat{\mathcal{M}}\boldsymbol{H} = \lambda\boldsymbol{H}$. For computing eigenmatrices, we regard the matrix $\boldsymbol{H} = (H_{ij})$ and the tensor $\hat{\mathcal{M}} = (\hat{M}_{ijkl})$ as a nine-dimensional vector and a 9×9 matrix [5]. For $\boldsymbol{A} = (A_{ij})$ and $\boldsymbol{B} = (B_{ij})$, we define $(\boldsymbol{A}; \boldsymbol{B}) = \sum_{i,j=1}^{3} A_{ij}B_{ij}$. For $\boldsymbol{a} = (a_i)$ and $\boldsymbol{A} = (A_{ij})$, we define $\boldsymbol{a}\times\boldsymbol{A}\times\boldsymbol{a}$ to be a symmetric matrix whose (ij) element is $\sum_{i,j=1}^{3} \varepsilon_{ikl}\varepsilon_{jmn}a_k a_m A_{ln}$. The operation $(\cdot)_2^-$ denotes the rank-constrained generalized inverse with rank 2; this operation is necessary for preventing numerical instability [5]. The

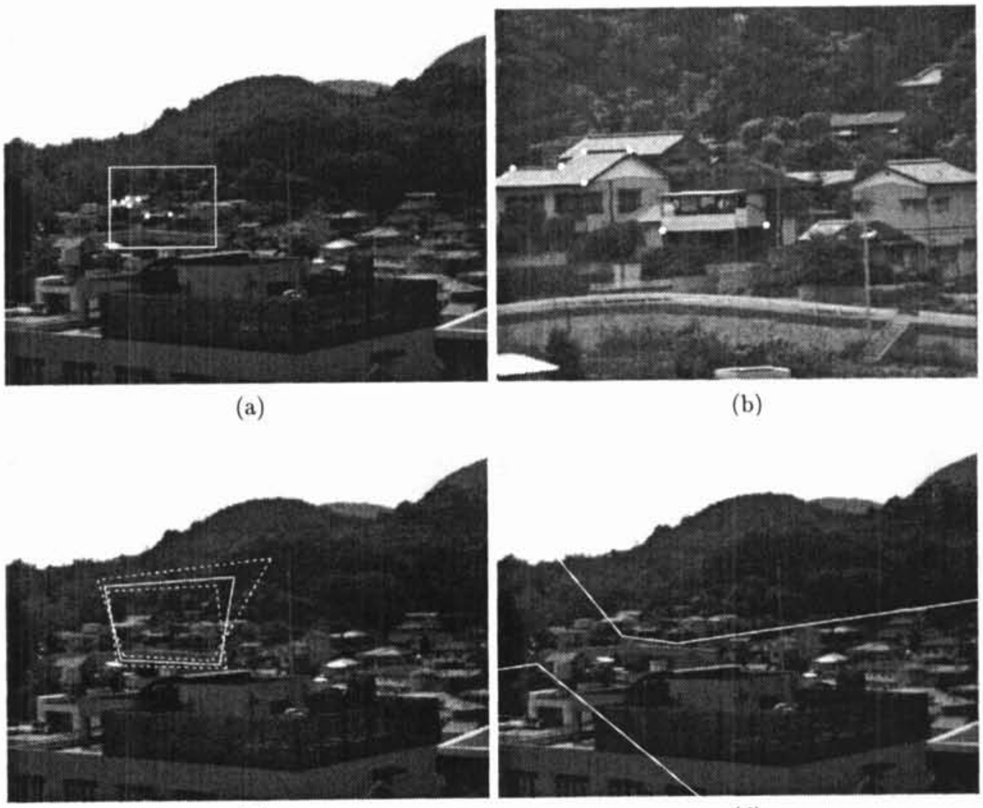


Figure 3: (a) A real image of an outdoor scene and selected feature points. (b) A zoomed image of the same scene; it corresponds to the white frame in (a). (c) The mapped boundary (solid lines) and its primary deviation pair (dashed lines) computed by our algorithm. (d) The mapped boundary computed by the least-squares method.

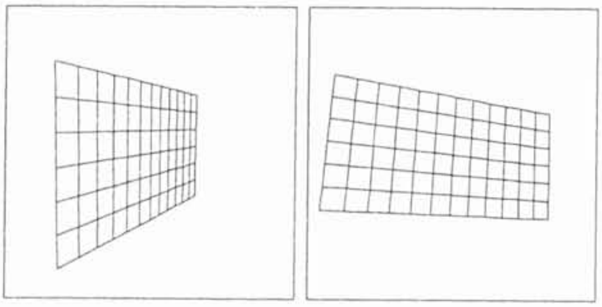


Figure 4: Synthetic images of a grid pattern.

operation $N[\cdot]$ normalizes a matrix to have unit norm.

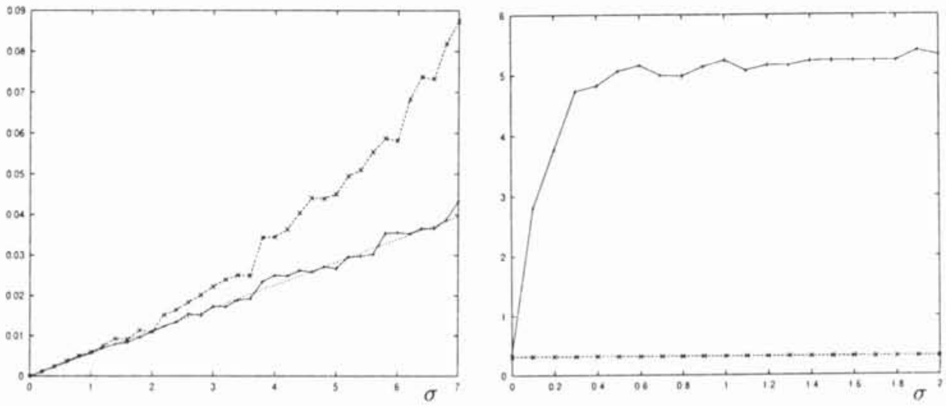
4. Examples

Fig. 1(a) is an image of a rectangular region and ten points in it. Fig. 1(b) is an image of the same region viewed from above. After randomly perturbing the ten points in Figs. 1(a) and (b) independently, we computed the homography $\hat{\boldsymbol{H}}$ between the two images and its standard deviation pair $\{\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)}\}$. The boundary image of the region in Fig. 1(a) mapped by the computed homography $\hat{\boldsymbol{H}}$ and its standard deviation pair $\{\boldsymbol{H}^{(+)}, \boldsymbol{H}^{(-)}\}$ are shown in Fig. 1(c). Fig. 2(a) shows the boundary images for ten instances obtained by changing the random noise. The standard deviation pair in Fig. 1(c) characterizes possible deviations of the solution very well. Fig. 2(b) shows the results computed by the least-squares method, which minimizes

$$\sum_{\alpha=1}^{N} \|\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{x}_\alpha\|^2. \tag{12}$$

Fig. 3(a) is a real image of an outdoor scene, and Fig. 3(b) is its zoomed image. We selected feature points as marked in the images and computed the homography. Mapping the boundary of Fig. 3(b) according to the computed homography, we obtain the frame with its primary deviation pair shown in Fig. 3(c). It is seen that the upper-right part of the mapped frame is the most uncertain. Fig. 3(d) shows the result by the least-squares method; its primary deviation pair is not defined. We observe the upper-right part extends infinitely to the upper right and appears from the lower left. As compared with this, our algorithm yields a reasonable solution even in such an unstable feature configuration.

We evaluated the accuracy and efficiency of our algorithm for the images in Fig. 4(a) by adding random noise of standard deviation σ (pixels) to the co-



(a) (b)

Figure 5: (a) Errors of computation: optimal algorithm (solid line); least-squares method (dashed line); theoretical lower bound (dotted line). (b) Computation time (seconds) for our algorithm (solid lines), the second-order renormalization (dashed lines) and the least-squares method (dotted lines).

ordinates of the grid points independently. Fig. 4(b) plots the root-mean-square error between the computed matrix $\hat{\boldsymbol{H}}$ and the true value $\bar{\boldsymbol{H}}$; we repeated 50 computations with different noise for each σ. The dotted line shows the theoretical lower bound computed by the theory of Kanatani [5, 6]. Our algorithm not only performs better than the least-squares method but indeed also almost attains the theoretical lower bound. Fig. 4(c) plots the average computation time in seconds. The least-squares is much faster than our algorithm. Our algorithm serves as a *benchmark* for evaluating to what extent accuracy is sacrificed for efficiency.

5. Concluding remarks

This paper has described an algorithm for optimally computing the homography between two given sets of points on a plane and evaluating the reliability of the resulting mapping. This algorithm is very accurate and robust and is expected to be used widely for many computer vision applications.

Acknowledgments. The author thanks Naoya Ohta and Yoshiyuki Shimizu of Gunma University for implementing a public-domain program[1] based on this paper. This work was in part supported by the Ministry of Education, Science, Sports and Culture, Japan under a Grant in Aid for Scientific Research C(2) (No. 09680352).

References

[1] A. Criminisi, I. Reid and A. Zisserman, Duality, rigidity and planar parallax, *Proc. 5th Euro. Conf. Comput. Vision*, June 1998, Freiburg, Germany, Vol. 2, pp. 846–861.

[2] D. Q. Huynh, Calibration of a structured light system: A projective approach, *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, June 1997, Puerto Rico, pp. 225–230.

[3] M. Irani, P. Anandan and D. Weinshall, From reference frames to reference planes: Multi-view parallax geometry and applications, *Proc. 5th Euro. Conf. Comput. Vision*, June 1998, Freiburg, Gemrany, Vol. 2, pp. 829–845.

[4] K. Kanatani, *Geometric Computation for Machine Vision*, Oxford University Press, Oxford, 1993.

[5] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*, Elsevier Science, Amsterdam, 1996.

[6] K. Kanatani, Cramer-Rao lower bounds for curve fitting, *Graphical Models Image Process.*, **60**(2), 1998, 93–99.

[7] K. Kanatani, Statistical optimization and geometric inference in computer vision, *Phi. Trans. R. Soc. Lond.*, A**356**-1740 (1998), 1303–1320.

[8] K. Kanatani and S. Takeda, 3-D motion analysis of a planar surface by renormalization, *IEICE Trans. Inf. & Syst.* **E78-D**(8), 1995, 1074–1079.

[9] H. C. Longuet-Higgins, The reconstruction of a plane surface from two perspective projections, *Proc. Roy. Soc. Lond.*, **B227**, 1986, 399–410.

[10] H. S. Sawhney, S. Hsu and R. Kumar, Robust video mosaicing through topology inference and local to global alignment, *Proc. 5th Euro. Conf. Comput. Vision*, June 1998, Freiburg, Gemrany, Vol. 2, pp. 103–119.

[11] T. Viéville, C. Zeller and L. Robert, Using collineations to compute motion and structure in an uncalibrated image sequences, *Int. J. Comput. Vision*, **20**(3), 1996, 213–242.

[12] J. Weng, N. Ahuja and T. S. Huang, Motion and structure from point correspondences with error estimation: Planar surfaces, *IEEE Trans. Patt. Anal. Mach. Intell.*, **39**(12), 1991, 2691–2717.

[1] This program is implemented in the C++ language and can be obtained along with accompanying routine packages on request by e-mail.