Electronics and Communications in Japan, Part 3, Vol. 86, No. 1, 2003
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J85-A, No. 2, February 2002, pp. 231–239

Do We Really Have to Consider Covariance Matrices for Image Feature Points?

Yasushi Kanazawa[1] and Kenichi Kanatani[2]

[1]Department of Knowledge-Based Information Engineering, Toyohashi University of Technology, Toyohashi, Aichi, 441-8580 Japan

[2]Department of Information Technology, Okayama University, Okayama, 700-8535 Japan

SUMMARY

We first describe in a unified way how to compute the covariance matrix from the gray levels of the image. We then experimentally investigate whether or not the computed covariance matrix actually reflects the accuracy of the feature position by doing subpixel correction using variable template matching. We also test if the accuracy of the homography and the fundamental matrix can really be improved by optimization using the covariance matrix computed from the gray levels. © 2002 Wiley Periodicals, Inc. Electron Comm Jpn Pt 3, 86(1): 1–10, 2003; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ecjc.10042

Key words: feature extraction; covariance matrix; template matching; homography; fundamental matrix.

1. Introduction

The authors have presented various optimization techniques for feature-based problems, such as 3D reconstruction, camera calibration, and image mosaicking, using the covariance matrix of a feature point as the measure of its uncertainty [4, 6, 8, 9]. In most cases, we have assumed isotropic homogeneous noise for the covariance matrix.

In the past, various methods have been proposed for computing the covariance matrix from the image gray levels [1, 10, 12, 13]. It has not been clear, however, whether or not the computed covariance matrix actually reflects the "accuracy" of the feature position. It has also not been clear if a better result can be obtained using such covariance matrices rather than simply assuming "isotropic homogeneous noise."

It is true that many authors have demonstrated by simulation the effectiveness of using feature covariance, but they have generated the noise *according to the assumed covariance*. Can we really relate the accuracy of the feature position in the actual image to the covariance matrix derived from the gray levels? This question has frequently been posed, but so far no satisfactory answer has been given.

In this paper, we first describe in a unified way how to compute the covariance matrix from the gray levels of the image. We then compare the accuracy of the feature position located by template matching with its covariance matrix computed from the gray levels. We also test if the accuracy of the homography and the fundamental matrix can really be improved by optimization using such covariance information.

2. Covariance of Feature Position

The position of a feature point has a certain degree of uncertainty, whether it is extracted manually using a mouse or automatically using a feature detector such as SUSAN [14] and the Harris operator [2]. Let $(\overline{x}, \overline{y})$ be its true position, and (x, y) its observed position. Regarding $\Delta x =$

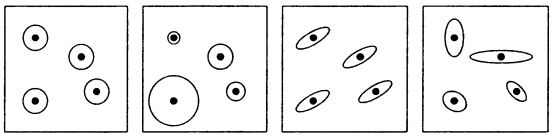

Fig. 1. Noise modeling: (a) isotropic homogeneous; (b) isotropic inhomogeneous; (c) anisotropic homogeneous; (d) anisotropic inhomogeneous.

$x - \bar{x}$ and $\Delta y = y - \bar{y}$ as random variables, we define the covariance matrix of this observation by

$$\begin{pmatrix} E[\Delta x^2] & E[\Delta x \Delta y] \\ E[\Delta y \Delta x] & E[\Delta y^2] \end{pmatrix} = \sigma^2 \Sigma^0 \tag{1}$$

where $E[\cdot]$ denotes expectation. The constant σ, which we call the *noise level*, represents the absolute magnitude of the noise, while the matrix Σ^0, which we call the *normalized covariance matrix*, describes the relative magnitude of the noise and its directional dependence [3].

The reason why we divide the covariance matrix into the noise level and the normalized covariance matrix is that, as will be shown in the next section, the covariance matrix computed from the gray levels is defined only up to a constant multiplier. Another reason is that multiplying the covariance matrix by a constant does not affect the optimization solution [3].

In general, the noise distribution is classified as in Fig. 1 according to its dependence on the position and direction. If the noise characteristics do not depend on the position or direction (*isotropic homogeneous noise*), we can use the following default value for the normalized covariance matrix Σ^0:

$$\Sigma^0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2}$$

3. Covariance Matrix Computation

The computation of the covariance matrix from the image gray levels is roughly classified into the residual-based approach [10, 12] and the derivative-based approach [1, 13].

3.1. Residual-based approach

Let $\mathcal{N}_p$ be a rectangular grid centered on a feature point p. The (*self-*)*residual*[*] is defined by

$$J(x,y) = \frac{1}{2} \sum_{(i,j) \in \mathcal{N}_p} w_{ij} \Big(I(i+x, j+y) - I(i,j) \Big)^2 \tag{3}$$

where x and y are real numbers and w_{ij} is an appropriate (e.g., Gaussian) weight (Fig. 2). The gray level $I(i, j)$ is regarded as a continuous function via an appropriate interpolation. Since $J(x, y)$ is a nonnegative function which takes its minimum at $x = y = 0$, it can be approximated by the following quadratic function over a neighborhood $\mathcal{X}$ of the origin (0, 0):

$$\begin{aligned} g(x,y) &= \frac{1}{2}(n_1 x^2 + 2n_2 xy + n_3 y^2) \\ &= \frac{1}{2} \begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned} \tag{4}$$

The *Hessian H* is defined by

$$H = \begin{pmatrix} n_1 & n_2 \\ n_2 & n_3 \end{pmatrix} \tag{5}$$

The elements n_1, n_2, and n_3 are determined by the following least-squares minimization:

$$\iint_{\mathcal{X}} w(x,y) \Big(J(x,y) - g(x,y) \Big)^2 dx dy \rightarrow \min \tag{6}$$

Here, $w(x, y)$ is an appropriate weight, typical examples being the *Gaussian weight* $w(x, y) = e^{-(x^2+y^2)/\sigma^2}$ and the *Gibbs weight* $w(x, y) = e^{-J(x,y)/\sigma^2}$ with an appropriate constant σ.

The solution $\boldsymbol{n} = (n_1, n_2, n_3)^\top$ of the minimization (6) can be obtained by solving the following normal equation, which is obtained by differentiating Eq. (6) with respect to x and y and equating the result to 0:

$$\frac{1}{2} \boldsymbol{A} \boldsymbol{n} = \boldsymbol{b} \tag{7}$$

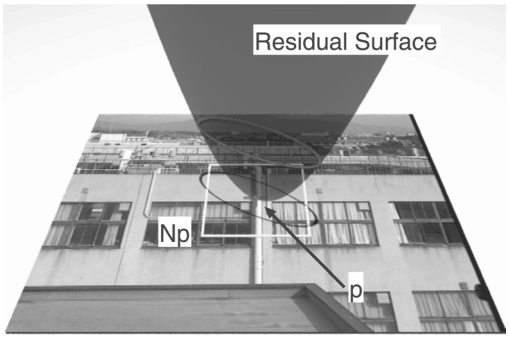

Fig. 2. The residual surface.

[*] The third term $\Sigma_{(i,j) \in \mathcal{N}_p} w_{ij} I(i, j)^2/2$ in the expansion of the right-hand side of Eq. (3) is a constant. If the first term $\Sigma_{(i,j) \in \mathcal{N}_p} w_{ij} I(i + x, j + y)^2/2$ does not depend on x or y, it equals the (*weighted*) *autocorrelation* except for the sign.

The matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ are defined by

$$\boldsymbol{A} = \iint_{\mathcal{X}} w(x,y)\boldsymbol{m}(x,y)\boldsymbol{m}(x,y)^\top dxdy$$
$$\boldsymbol{b} = \iint_{\mathcal{X}} w(x,y)J(x,y)\boldsymbol{m}(x,y)dxdy \quad (8)$$

where $\boldsymbol{m}(x, y) = (x^2, 2xy, y^2)^\top$. The integral $\iint_{\mathcal{X}} dxdy$ is evaluated by numerical sampling in the region $\mathcal{X}$. The solution $\boldsymbol{n}$ of Eq. (7) determines the Hessian H in the form of Eq. (5). Its inverse is identified with the normalized covariance matrix Σ^0 [10, 12]:

$$\Sigma^0 = H^{-1} \quad (9)$$

3.2. Derivative-based approach

If the correction terms x and y in Eq. (3) are small, the Taylor expansion of $I(i + x, j + y)$ yields the following first-order approximation:

$$J = \frac{1}{2}\sum_{(i,j)\in\mathcal{N}_p} w_{ij}(I_i x + I_j y)^2 = \frac{1}{2}\begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix} \quad (10)$$

Here, I_i and I_j are the partial derivatives of $I(i, j)$ with respect to i and j (regarded as real variables), respectively. They are numerically evaluated using a smooth differentiation filter (Appendix 1). The Hessian H in Eq. (10) has the form

$$H = \begin{pmatrix} \sum_{(i,j)\in\mathcal{N}_p} w_{ij} {I_i}^2 & \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_i I_j \\ \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_j I_i & \sum_{(i,j)\in\mathcal{N}_p} w_{ij} {I_j}^2 \end{pmatrix} \quad (11)$$

Its inverse is identified with the normalized covariance matrix Σ^0 [1, 13]:

$$\Sigma^0 = H^{-1} \quad (12)$$

3.3. Theoretical background

In both the residual- and derivative-based approaches, the inverse of the Hessian of the residual surface is identified with the normalized covariance matrix; the only difference is whether it is approximated over the neighborhood $\mathcal{N}_p$ of the feature point p or evaluated via the Taylor expansion around p.

Since the Hessian H represents the curvature of the residual surface at p, a large Hessian means that the gray level changes rapidly as we slightly move from p in its neighborhood (Fig. 2). This implies that the feature position can easily be located precisely by template matching. If the Hessian H is small, on the other hand, the feature position is difficult to locate precisely, because the gray level varies only slightly in its neighborhood. Thus, it is intuitively clear that the Hessian H is related to the uncertainty of the feature position. Mathematically, this is formulated as follows.

It is known in statistics that the covariance matrix of a maximum likelihood estimate can be approximated by its *Cramer–Rao lower bound*, which is given by the inverse of the Hessian (or the *Fisher information matrix*) of the logarithmic likelihood function. If locating a feature point by template matching is regarded as a statistical estimation problem, Eqs. (11) and (12) respectively correspond to the Fisher information matrix and the Cramer–Rao lower bound (Appendix 2). Consequently, the inverse of the Hessian H can be identified with the covariance matrix that measures the uncertainty of the feature position.

The above observation amounts to approximating the distribution of the deviation (x, y) from the true value by the Gaussian distribution,

$$p(x,y) \approx \frac{1}{2\pi\sigma^2} e^{-\begin{pmatrix} x & y \end{pmatrix} H \begin{pmatrix} x \\ y \end{pmatrix}/2\sigma^2} \quad (13)$$

of mean 0 and standard deviation σ. From the definition of the Hessian H, Eq. (13) can also be written in the form

$$p(x,y) \approx \frac{1}{2\pi\sigma^2} e^{-J(x,y)/\sigma^2} \quad (14)$$

In many computer vision applications, the uncertainty of a quantity is frequently modeled by this type of *Gibbs distribution* associated with the residual J, which is often called the *energy* by physical analogy [7]. In fact, Singh [12] approximated the normalized covariance matrix by the sample covariance matrix obtained by discrete sampling of $e^{-J(x,y)/\sigma^2}$.

On the other hand, the derivative-based approach can be interpreted as detecting *optical flow* using the *gradient constraint* [3, 11] (Appendix 3). In fact, the inverse of Eq. (11) is used as the covariance matrix that measures the uncertainty of the computed flow [11]. It can be seen from Eq. (11) that the Hessian H has determinant 0 when the gray level is constant in some direction, causing what is known as the *aperture problem*.

3.4. Real image examples

In Fig. 3(a), the covariance matrix evaluated at each vertex of a 3×3 grid arbitrarily placed in the image is displayed as an ellipse that represents the standard deviation in every orientation [3]. Since the absolute size of the ellipse is indeterminate, we adjusted the scale for the ease of visualization. The ellipses in solid lines are obtained by the residual-based approach; those in dashed lines are obtained by the derivative-based approach.

We can see that the ellipses are large in a region in which the gray levels have small variations but are small in the parts where the gray levels have large variations, such

(a) (b) (c)

Fig. 3. Feature covariance evaluated from the residual surface (solid lines) and the gray-level derivatives (dashed lines). Feature points are (a) generated regularly, (b) chosen by hand, and (c) detected by SUSAN.

as at the corners of objects. We can also see that ellipses on an object boundary are elongated along that boundary. These agree with our intuition.

In Fig. 3(b), on the other hand, the feature points were manually selected using a mouse. It is seen that the uncertainty is almost isotropic and homogeneous. The reason seems to be that for choosing image features, humans unconsciously avoid points in regions in which gray levels have small variations or on object boundaries. Instead, humans tend to choose *easy-to-detect* points, such as corners or isolated points, around which the gray levels have large variations in all directions.

In Fig. 3(c), the feature points were extracted by SUSAN [14]. Again, the uncertainty is almost isotropic and homogeneous. This is because most feature detectors including SUSAN, whatever algorithm is used, effectively compute the covariance internally and output those points which have large gray level variations around them in all directions. The Harris operator [2], for instance, directly computes the Hessian in Eq. (11). The Kanade–Lucas–Tomasi feature tracker [15] also uses the same computation internally.

From these observations, we can conclude that as long as feature points are extracted manually or by a feature detector, it suffices to assume the isotropic homogeneous model of Eq. (2). It is, therefore, when feature points are specified randomly or independently of the image content, as in Fig. 3, that we need to evaluate the covariance matrix from the gray levels.

4. Feature Point Matching

4.1. Variable template matching

The next question is: When a feature point is specified randomly or independently of the image content, does the covariance matrix computed from the gray levels really express the accuracy of locating it? To investigate this, we conduct the following experiments.

Using two images of the same scene, we match a randomly specified point p in one image to the other. Cutting out a neighborhood $\mathcal{N}_p$ of p, we define a template $T(i, j)$ centered on it and do variable template matching by taking account of rotations and scale changes.

We manually place the template $T(i, j)$ at a pixel (a, b) close to the corresponding point in the other image $I(i, j)$ (such an approximate point can be automatically found by a coarse-to-fine strategy, but this is irrelevant in our context, so we omit this stage). Then, we adjust the translation, the rotation, and the scale by template matching.

Suppose the template $T(i, j)$ matches the image $I(i, j)$ if it is translated by (x, y), rotated by angle θ, and scaled s times from this position. The values (x, y), θ, and s are determined by minimizing

$$J(x, y, \theta, s) = \sum_{(i,j)\in\mathcal{N}_p} w_{ij}\Big(T(i, j) - \mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i, j)\Big)^2 \quad (15)$$

where we define the similarity mapping $\mathcal{T}^{(a,b)}_{(x,y,\theta,s)}$ by

$$\mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i, j) = I\big(a+x+s(\cos\theta - j\sin\theta),$$
$$b+y+s(i\sin\theta + j\cos\theta)\big) \quad (16)$$

We numerically search the discretized parameter space of x, y, θ, and s for the minimum by recursively subdividing the search region (we omit the details). If the illumination and exposure conditions change between the two images, we use instead of Eq. (15)

$$J(x, y, \theta, s) = \sum_{(i,j)\in\mathcal{N}_p} w_{ij}\mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i, j)^2$$

$$-C\left(\sum_{(i,j)\in\mathcal{N}_p} w_{ij}T(i,j)\mathcal{T}^{(a,b)}_{(x,y,\theta,s)}I(i,j)\right)^2 \qquad (17)$$

where

$$C = \frac{1}{\sum_{(i,j)\in\mathcal{N}_p} w_{ij}T(i,j)^2} \qquad (18)$$

which is a constant independent of x, y, s, and θ (Appendix 4).

4.2. Real image experiments

Figure 4(a) shows two images of an outdoor scene. We arbitrarily placed a 3×3 grid in the left image. In the right image, we manually defined, using a mouse, a grid (in solid lines) that approximately corresponds to the grid in the left image. Then, we corrected each vertex position by similarity template matching. The corrected grid is shown in dashed lines.

If the covariance matrix reflects the accuracy of locating the feature position, the deviation $(\Delta x, \Delta y)$ of the corrected position from its true position should have a positive correlation with the magnitude of the covariance matrix. Thus, we repeated the experiment described above many times in different feature locations and in different images. Figure 4(b) plots, on logarithmic scales, the absolute deviation $\sqrt{\Delta x^2 + \Delta y^2}$ on the vertical axis versus the square root of the trace of the covariance matrix computed from the first image on the horizontal axis. The latter should be proportional to the root-mean-square error $\sqrt{E[\Delta x^2 + \Delta y^2]}$, as seen from Eq. (1).

In this experiment, the true positions were determined as follows. Noting that the two images of a far scene should be related by a homography, we carefully chose by hand a large number (53 in this case) of corresponding points in the two images and optimally computed the homography from them (the details will be given later). Then, we mapped the vertices in the left image of Fig. 4(a) to the right image via the computed homography and regarded the resulting positions as the true corresponding points.

We can see from Fig. 4(b) that although there are large variations, the plots cluster roughly along a line with slope 1. This implies that the accuracy of template matching has more or less a positive correlation with the covariance matrix computed from the gray levels.

5. Covariance-Based Optimization

The remaining question is: Does the covariance matrix computed from the gray levels really improve the accuracy of the optimization using it? We now investigate this by computing the homography and the fundamental matrix from two images.

Suppose we are given two sets of corresponding points $\{(x_\alpha, y_\alpha)\}$ and $\{(x'_\alpha, y'_\alpha)\}$ along with their covariance matrices $\{\Sigma^0_\alpha\}$ and $\{\Sigma^{0\prime}_\alpha\}$, $\alpha = 1, \ldots, N$. We represent these points by vectors

$$\boldsymbol{x}_\alpha = \begin{pmatrix} x_\alpha/f_0 \\ y_\alpha/f_0 \\ 1 \end{pmatrix}, \quad \boldsymbol{x}'_\alpha = \begin{pmatrix} x'_\alpha/f_0 \\ y'_\alpha/f_0 \\ 1 \end{pmatrix} \qquad (19)$$

where f_0 is an appropriate scale constant, such as the image size, taken so that x_α/f_0, y_α/f_0, x'_α/f_0, and y'_α/f_0 have the order of 1. Since the third components are 1, their normalized covariance matrices are singular with rank 2 in the form

(a) (b)

Fig. 4. (a) Two images of an outdoor scene. The left grid is mapped to the right image and corrected with subpixel accuracy. (b) Correlation between the covariance matrix and the template matching accuracy: the residual-based approach (×); the derivative-based approach (+).

(a) (b)

Fig. 5. (a) Corresponding points and their covariance matrices. (b) Errors in the homography computation using the covariance matrices evaluated from the gray levels (solid lines) and their default values (dashed lines). The numbers on the abscissa indicate individual instances.

$$V_0[\boldsymbol{x}_\alpha] = \begin{pmatrix} \Sigma_\alpha^0 & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix}, \quad V_0[\boldsymbol{x}'_\alpha] = \begin{pmatrix} \Sigma_\alpha^{0\prime} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{pmatrix} \tag{20}$$

Theoretically, the derivative-based approach may be more consistent than the residual-based approach (Appendixes 2 and 3). In the actual computation, however, differentiation needs to be approximated by a difference filter, which is susceptible to noise and greatly influenced by the associated smoothing operation (Appendix 1). The residual-based approach, on the other hand, relies on integration (approximated as summation), which is more robust than differentiation. Since in most cases no marked difference is found in the final results, we adopt the residual-based approach in the subsequent experiments.

5.1. Optimal homography computation

If we take two images of a planar object or a far scene, the two images are related by a mapping called *homography*. Namely, there exists a nonsingular matrix $\boldsymbol{H}$ such that $\{\boldsymbol{x}_\alpha\}$ and $\{\boldsymbol{x}'_\alpha\}$ are related in the following form [3, 6]:

$$\boldsymbol{x}'_\alpha = Z[\boldsymbol{H}\boldsymbol{x}_\alpha] \tag{21}$$

Here, $Z[\cdot]$ designates scale normalization to make the third component 1. The matrix $\boldsymbol{H}$ is also called the *homography* whenever no confusion can arise. Since it is defined only up to a scale factor, we normalize it to $\|\boldsymbol{H}\| = 1$ [the norm of a matrix $\boldsymbol{A} = (A_{ij})$ is defined by $\|\boldsymbol{A}\| = \sqrt{\Sigma_{i,j=1}^3 A_{ij}^2}$].

Computing the homography from images is a necessary step in many vision applications such as image mosaicking and camera calibration [6, 8]. In the presence of image noise, a statistically optimal estimate of $\boldsymbol{H}$ is obtained by minimizing the following function [3]:

$$J(\boldsymbol{H}) = \frac{1}{N}\sum_{\alpha=1}^{N} (\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{x}_\alpha, \boldsymbol{W}_\alpha(\boldsymbol{x}'_\alpha \times \boldsymbol{H}\boldsymbol{x}_\alpha)) \tag{22}$$

$$\boldsymbol{W}_\alpha = \Big(\boldsymbol{x}'_\alpha \times \boldsymbol{H}V_0[\boldsymbol{x}_\alpha]\boldsymbol{H}^\top \times \boldsymbol{x}'_\alpha + (\boldsymbol{H}\boldsymbol{x}_\alpha) \times V_0[\boldsymbol{x}'_\alpha] \times (\boldsymbol{H}\boldsymbol{x}_\alpha)\Big)_2^{\frac{1}{N}} \tag{23}$$

Here, $(\boldsymbol{a}, \boldsymbol{b})$ denotes the inner product of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$; $\boldsymbol{a} \times \boldsymbol{A}$ is the matrix whose columns are the vector products of $\boldsymbol{a}$ and the columns of $\boldsymbol{A}$; $\boldsymbol{A} \times \boldsymbol{a}$ is the matrix whose rows are the vector products of $\boldsymbol{a}$ and the rows of $\boldsymbol{A}$ [3]. The operation $(\cdot)_r^-$ denotes the (Moore–Penrose) generalized inverse computed after replacing the smallest $n - r$ eigenvalues by zeros [3]. The program code for computing the minimizing solution of Eq. (22) by a technique called *renormalization* is publicly available.*

Figure 5(a) shows two images of a far scene. We arbitrarily placed a grid in the left image and mapped it to a roughly corresponding position in the right image by hand, using a mouse. Then, we corrected the vertex positions by similarity template matching. The covariance matrices computed from the gray levels are displayed as ellipses at the vertices. Using these covariance matrices, we optimally computed the homography $\boldsymbol{H}$ from the corrected point matches.

For comparison, we also computed the homography $\boldsymbol{H}_0$ using the default values in Eq. (2) and the "true" homography $\overline{\boldsymbol{H}}$, which was estimated by using many (53 in this case) points carefully chosen by hand. The discrepancies of $\boldsymbol{H}$ and $\boldsymbol{H}_0$ from $\overline{\boldsymbol{H}}$ measured in norm are

$$\|\boldsymbol{H} - \overline{\boldsymbol{H}}\| = 0.007190, \quad \|\boldsymbol{H}_0 - \overline{\boldsymbol{H}}\| = 0.008358$$

Figure 5(b) shows 24 instances of such errors for different images and different grid positions. From this, we conclude that the accuracy is more or less improved by optimization using the covariance matrix computed from the images.

5.2. Optimal fundamental matrix computation

As is well known, corresponding points in two images of the same scene satisfy the *epipolar equation* [3, 4]

*http://www.ail.cs.gunma-u.ac.jp/Labo/programs-e.html

(a) (b)

Fig. 6. (a) Covariance matrices and corresponding points. (b) Errors in the fundamental matrix computation using the covariance matrices evaluated from the gray levels (solid lines) and their default values (dashed lines). The numbers on the abscissa indicate individual instances.

$$(\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha) = 0 \tag{24}$$

where $\boldsymbol{F}$ is a singular matrix of rank 2, called the *fundamental matrix*. Since its absolute scale is indeterminate, we normalize it to $\|\boldsymbol{F}\| = 1$.

Computing the fundamental matrix is a first step of 3D reconstruction from images [4]. In the presence of image noise, a statistically optimal estimate of $\boldsymbol{F}$ is obtained by minimizing the following function [3]:

$$J(\boldsymbol{F}) = \frac{1}{N}\sum_{\alpha=1}^{N} W_\alpha (\boldsymbol{x}_\alpha, \boldsymbol{F}\boldsymbol{x}'_\alpha)^2 \tag{25}$$

$$W_\alpha = \frac{1}{(\boldsymbol{x}'_\alpha, \boldsymbol{F}^\top V_0[\boldsymbol{x}_\alpha]\boldsymbol{F}\boldsymbol{x}'_\alpha) + (\boldsymbol{x}_\alpha, \boldsymbol{F}V_0[\boldsymbol{x}'_\alpha]\boldsymbol{F}^\top \boldsymbol{x}_\alpha)} \tag{26}$$

The program code for computing the solution by renormalization is also available publicly.*

Figure 6(a) shows two images of an indoor scene. We arbitrarily placed a 3×3 grid in the left image and mapped it to a roughly corresponding position in the right image by hand, using a mouse. Then, we corrected the vertices by similarity template matching. The covariance matrices computed from the gray levels are displayed as ellipses at the vertices. Using these covariance matrices, we optimally computed the fundamental matrix $\boldsymbol{F}$ from the corrected point matches.

For comparison, we also computed the fundamental matrix $\boldsymbol{F}_0$ using the default values in Eq. (2) and the "true" fundamental matrix $\bar{\boldsymbol{F}}$, which was estimated by using many (68 in this case) points carefully chosen by hand. The discrepancies of $\boldsymbol{F}$ and $\boldsymbol{F}_0$ from $\bar{\boldsymbol{F}}$ measured in norm are

$$\|\boldsymbol{F} - \bar{\boldsymbol{F}}\| = 0.009806, \quad \|\boldsymbol{F}_0 - \bar{\boldsymbol{F}}\| = 0.015141$$

*http://www.ail.cs.gunma-u.ac.jp/Labo/programs-e.html

Figure 6(b) shows 24 instances of such errors for different images and different grid positions. From this, we conclude that the accuracy is more or less improved by optimization using the covariance matrix computed from the images.

6. Conclusions

Our conclusion is as follows:

- It suffices to assume the default value for the covariance matrix if feature points are chosen by hand or by a feature detector.
- If feature points are specified randomly or independently of the image content, then:
  - the covariance matrix computed from the gray levels of the image has a positive correlation with the accuracy of template matching;
  - the accuracy of optimization is improved, though only slightly, by the use of the covariance matrix computed from the gray levels of the image.

An example of the latter situation is the following semiautomatic image matching system. First, the computer defines a regular grid in the first image. Then, a human operator roughly specifies its corresponding position in the second image, using a mouse, or alternatively adjusts the grid copied by the computer in the second image, dragging the mouse. Finally, the computer corrects each grid position by template matching with subpixel accuracy.

The real image experiments in this paper were done using such a system, which we experimentally constructed. If we do image mosaicking or 3D reconstruction from feature points obtained in this way, the accuracy is expected to increase by doing optimization using the covariance matrix computed from the gray levels of the image.

The conclusions in this paper were all derived from relatively few experimental results. It would be ideal if decisive conclusions could be obtained by simulations. However, this is theoretically impossible, because simulations only confirm the effectiveness of a method by simulating the assumptions or models *on which the method is built*. In contrast, the purpose of this paper is to examine to what extent a method is effective for real data, for which the assumptions or models may not necessarily be valid. In this sense, we have illuminated both the usefulness and the limitations of the statistical approach based on the covariance of feature points.

REFERENCES

1. Förstner F. Reliability analysis of parameter estimation in linear models with applications to mensuration problems in computer vision. Comput Vision Graphics Image Process 1987;40:273–310.
2. Harris C, Stephens M. A combined corner and edge detector. Proc 4th Alvey Vision Conf, 1988, Manchester, UK, p 147–151.
3. Kanatani K. Statistical optimization for geometric computation: Theory and practice. Elsevier Science; 1996.
4. Kanatani K, Mishima H. 3-D reconstruction from two uncalibrated views and its reliability evaluation. IPSJ J: CVIM 2001;42:1–8. (in Japanese)
5. Kanatani K, Ohta N, Kanazawa Y. Optimal homography computation with a reliability measure. Trans IEICE 2000;E83-D:1369–1374.
6. Kanazawa Y, Kanatani K. Stabilizing image mosaicking by the geometric AIC. Trans IEICE 2000;J83-A:686–693. (in Japanese)
7. Li SZ. Markov random field modeling in computer vision. Springer; 1995.
8. Matusnaga C, Kanatani K. Calibration of a moving camera using a planar pattern: Optimal computation, reliability evaluation, and stabilization by the geometric AIC. Trans IEICE 2000;J83-A:694–701. (in Japanese) Electron Commun Japan Part 3 2001;84:12–21. (English translation)
9. Mishima H, Kanatani K. Optimal computation of the fundamental matrix and its reliability evaluation. IPSJ SIG Notes, 99-CVIM-118-10, p 67–74, 1999. (in Japanese)
10. Morris DD, Kanade K. A unified factorization algorithm for points, line segments and planes with uncertainty models. Proc Int Conf Comput Vision, 1998, Bombay, India, p 696–702.
11. Ohta N. Image movement detection with reliability indices. IEICE Trans 1991;E74:3379–3388.
12. Singh A. An estimation-theoretic framework for image-flow computation. Proc 3rd Int Conf Comput Vision, 1990, Osaka, Japan, p 168–177.
13. Shi J, Tomasi C. Good features to track. Proc IEEE Conf Comput Vision Pattern Recog, 1994, Seattle, WA, p 593–600.
14. Smith SM, Brady JM. SUSAN—A new approach to low level image processing. Int J Comput Vision 1997;23:45–78.
15. Tomasi C, Kanade T. Detection and tracking of point features. CMU Tech Rep CMU-CS-91-132, April 1991; http://vision.stanford.edu/~birch/klt/.

APPENDIX

1. Smooth Differentiation Filter

The smoothing filter with weight $w(x, y)$ for a continuous image $I(x, y)$ has the form

$$\tilde{I}(x,y) = C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w(t-x, s-y)I(t,s)dtds \qquad (27)$$

where C is the normalization constant. Differentiating this with respect to x, we obtain

$$\begin{aligned}\tilde{I}_x(x,y) &= -C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_x(t-x, s-y)I(t,s)dtds \\ &= C\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} w_x(t,s)I(t+x, s+y)dtds \qquad (28)\end{aligned}$$

In discrete approximation, we have

$$\tilde{I}_x(i,j) = -C\sum_{(k,l)\in\mathcal{N}_p} w_x(k,l)I(k+i, l+j) \qquad (29)$$

Determining the constant C so that $\tilde{I}_x(i, j) = 1$ identically for $I(i, j) = i$, we obtain the following differentiation filter in the x direction:

$$D_x(i,j) = \frac{w_x(i,j)}{\sum_{(k,l)\in\mathcal{N}_p} w_x(k,l)k} \qquad (30)$$

Here, the weight $w(x, y)$ is assumed to be an even function of x and y. In our experiments, we used the Gaussian weight $w(x, y) = e^{-(x^2+y^2)/2\sigma^2}$ and adjusted the standard deviation σ according to the size of the neighborhood $\mathcal{N}_p$. The differentiation filter in the y direction can be derived similarly.

2. Cramer–Rao Lower Bound

Given a two-variable function $I(u, v)$ and data values I_{ij}, $(i, j) \in \mathcal{N}_p$, where $\mathcal{N}_p$ represents the set of grid points in the neighborhood of point p, consider the problem of deter-

mining (x, y) such that $I_{ij} \approx I(i+x, j+y)$, $(i,j) \in \mathcal{N}_p$. If we introduce the model

$$I_{ij} = I(i+x, j+y) + \varepsilon_{ij} \tag{31}$$

this problem can be regarded as statistical estimation. If the noise ε_{ij} is an independent Gaussian random variable of mean 0 and standard deviation σ_{ij}, the likelihood of the data I_{ij}, $(i,j) \in \mathcal{N}_p$ is written as

$$p = \prod_{(i,j)\in\mathcal{N}_p} \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-(I_{ij}-I(i+x,j+y))^2/2\sigma_{ij}^2} \tag{32}$$

Suppose the true value of (x, y) is $(0, 0)$. If we put $w_{ij} = \sigma^2/\sigma_{ij}^2$, the *score* $l = (\partial \log p/\partial x, \partial \log p/\partial y)^\top$ is written in the form

$$l = \frac{1}{\sigma^2} \begin{pmatrix} \sum_{(i,j)\in\mathcal{N}_p} w_{ij}\varepsilon_{ij} I_i \\ \sum_{(i,j)\in\mathcal{N}_p} w_{ij}\varepsilon_{ij} I_j \end{pmatrix} \tag{33}$$

where I_i and I_j represent the derivatives of $I(u, v)$ with respect to u and v, respectively, evaluated at $(u, v) = (i, j)$. From Eq. (33), the *Fisher information matrix* has the form

$$\begin{aligned} E[ll^\top] &= \frac{1}{\sigma^2} \begin{pmatrix} \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_i^2 & \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_i I_j \\ \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_j I_i & \sum_{(i,j)\in\mathcal{N}_p} w_{ij} I_j^2 \end{pmatrix} \\ &= \frac{1}{\sigma^2} H \end{aligned} \tag{34}$$

where H is the matrix defined in Eq. (11). In the above derivation, we have used the fact that $E[\varepsilon_{ij}\varepsilon_{kl}]$ is σ_{ij}^2 for $i = k$ and $j = l$ and is 0 otherwise. From Eq. (34), we obtain the following *Cramer–Rao inequality* for the covariance matrix $\Sigma(= \sigma^2\Sigma^0)$ of an estimate $(\hat{x}, \hat{y})$ of (x, y) [3]:

$$\Sigma^0 \succ H^{-1} \tag{35}$$

Here, $\succ$ means that the difference between the left-hand side and the right-hand side is a positive semidefinite symmetrical matrix. The right-hand side is known as the *Cramer–Rao lower bound*.

3. Covariance Matrix of Optical Flow

If x and y are small, we have the approximation $I(i+x, j+y) \approx I + I_i x + I_j y$. Then, the maximum likelihood solution that maximizes Eq. (32) is obtained by minimizing

$$J = \frac{1}{2} \sum_{(i,j)\in\mathcal{N}_p} w_{ij}(I_{ij} - I - I_i x - I_j y)^2 \tag{36}$$

This can be interpreted as computing the *optical flow* (x, y) by solving the *gradient constraint* [3, 11]

$$I_i x + I_j y = I_{ij} - I \tag{37}$$

by weighted least squares. Differentiating Eq. (36) with respect to x and y and equating the result to 0, we obtain the normal equation

$$H \begin{pmatrix} x \\ y \end{pmatrix} = l \tag{38}$$

where H and l are defined by Eqs. (11) and (33), respectively. The covariance matrix of the solution is given by

$$\begin{aligned} E[\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}^\top] &= H^{-1} E[ll^\top] H^{-1} \\ &= H^{-1} H H^{-1} = H^{-1} \end{aligned} \tag{39}$$

This has the same form as the Cramer–Rao lower bound given in Eq. (35).

4. Illumination-Invariant Template Matching

All we need is to introduce a coefficient c that compensates the change of illumination and minimize

$$J(x, y, \theta, s) = \sum_{(i,j)\in\mathcal{N}_p} w_{ij} \left(cT(i,j) - \mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i,j) \right)^2 \tag{40}$$

Differentiating this with respect to c and equating the result to 0, we see that c is given by

$$c = \frac{\sum_{(i,j)\in\mathcal{N}_p} w_{ij} T(i,j) \mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i,j)}{\sum_{(i,j)\in\mathcal{N}_p} w_{ij} T(i,j)} \tag{41}$$

Substitution of this into Eq. (40) yields Eq. (17). If the third term $\Sigma_{(i,j)\in\mathcal{N}_p} w_{ij} (\mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i,j))^2$ in the expansion of the right-hand side of Eq. (40) does not depend on x, y, θ, or s, minimizing Eq. (40) and minimizing Eq. (17) are both equivalent to maximizing the following *correlation function*:

$$\sum_{(i,j)\in\mathcal{N}_p} w_{ij} T(i,j) \mathcal{T}^{(a,b)}_{(x,y,\theta,s)} I(i,j) \tag{42}$$

AUTHORS (from left to right)

Yasushi Kanazawa received his M.S. degree in information engineering from Toyohashi University of Technology in 1987 and Ph.D. degree in information and computer science from Osaka University in 1997. After engaging in research and development of image processing systems at Fuji Electric Co., and serving as a lecturer of information and computer engineering at Gunma College of Technology, he is currently an associate professor in the Department of Knowledge-Based Information Engineering at Toyohashi University of Technology. His research interests include image processing and computer vision.

Kenichi Kanatani received his M.S. and Ph.D. degrees in applied mathematics from the University of Tokyo in 1974 and 1979. After serving as a professor of computer science at Gunma University, he is currently a professor of information technology at Okayama University. He is the author of *Group-Theoretical Methods in Image Understanding* (Springer, 1990), *Geometric Computation for Machine Vision* (Oxford, 1993), and *Statistical Optimization for Geometric Computation: Theory and Practice* (Elsevier, 1996). He is an IEEE Fellow.