PAPER

Extending Interrupted Feature Point Tracking for 3-D Affine Reconstruction

Yasuyuki SUGAYA[†a)] *and* **Kenichi KANATANI**[†], *Members*

SUMMARY Feature point tracking over a video sequence fails when the points go out of the field of view or behind other objects. In this paper, we extend such interrupted tracking by imposing the constraint that under the affine camera model all feature trajectories should be in an affine space. Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space, coupled with an outlier removal process. Using real video images, we demonstrate that our method can restore a sufficient number of trajectories for detailed 3-D reconstruction.

key words: *feature tracking, missing data estimation, outlier removal, factorization, 3-D reconstruction*

1. Introduction

The factorization method of Tomasi and Kanade [15] can reconstruct the 3-D shape of a scene from feature point trajectories tracked over a video sequence. The computation is very efficient, requiring only linear operations. The solution is sufficiently accurate for many practical purposes and can be used as an initial value for iterations of a more sophisticated reconstruction procedure [3].

However, the feature point tracking fails when the points go out of the field of view or behind other objects. In order to obtain a sufficient number of feature trajectories for detailed 3-D reconstruction, we need to extend such interrupted tracking to the final frame. There have been several such attempts in the past.

Tomasi and Kanade [15] reconstructed the 3-D positions of partly visible feature points from their visible image positions and reprojected them onto the frames in which they are invisible. The camera positions were estimated from other visible feature points.

Saito and Kamijima [12] projectively reconstructed tentative 3-D positions of the missing points by sampling two frames in which they are visible and then reprojected them onto the frames in which they are invisible. The camera positions were computed up to projectivity.

Using the knowledge that all trajectories of feature points should be in a 4-dimensional subspace of the data space, Jacobs [5] randomly sampled four trajectories, constructed a high-dimensional subspace by letting the missing data have free values, and computed its orthogonal complement. He repeated this many times and computed by least squares a 4-dimensional subspace approximately orthogonal to the resulting orthogonal complements[*]. Partial trajectories were extended so that they were compatible with the estimated subspace. A similar method was also used by Kahl and Heyden [6].

Brandt [1] reconstructed tentative 3-D positions of the missing points using a tentative camera model and reprojected them onto all frames. From the visible and reprojected feature points, he estimated the camera model. Iterating these, he optimized both the camera model and the feature positions.

For all these methods, we should note the following:

- We need not reconstruct a tentative 3-D shape. 3-D reconstruction is made possible by some geometric constraints over multiple frames. One can directly map 2-D point positions to other frames if such constraints[**] are used.
- If a minimum number of frames are sampled for tentative 3-D reconstruction, the accuracy of computation depends on the sampled frames. Rather, one should make full use of all information contained in all frames.
- The observed trajectories are not necessarily correct, but existing methods treat outlier removal and trajectory extension separately.

In this paper, we present a new scheme for extending partial trajectories based on the constraint that under the affine camera model all trajectories should be in a 3-dimensional affine space, which we call the "affine space constraint". Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space.

If the motion were pure rotation, one could do exact maximal likelihood estimation, e.g., by using the method of Shum et al. [13], but it cannot be applied to translational motions. Here, we simplify the optimization procedure by introducing to each partial trajectory a weight that reflects its length.

Also, we incorporate outlier removal and trajectory extension into a single process, testing in every step of the optimization if each trajectory, extended or not, is reliable and removing unreliable ones as outliers.

Thus, the contribution of this paper is as follows:

Manuscript received July 14, 2003.
Manuscript revised November 18, 2003.

[†] The authors are with the Depertment of Information Technology, Okayama University, Okayama-shi, 700-8530 Japan.

a) E-mail: sugaya@suri.it.okayama-u.ac.jp

[*] In actual computation, he interchanged the roles of points and frames: he sampled two frames, i.e., two lists of x coordinates and two lists of y coordinates. The mathematical structure is the same.

[**] The projective reconstruction of Saito and Kamijima [12] is equivalent to the use of the *trilinear constraint* [3].

1. We present a succinct mathematical formulation for extending interrupted trajectories based on the affine space constraint without referring to any particular camera model such as orthography. Our constraint is stronger than that used by Jacobs [5]. No reprojection of tentative 3-D reconstruction is necessary.
2. We present a procedure that integrates reliability evaluation of perfect and imperfect trajectories, outlier removal, and optimization of the affine space into a single process.

Section 2 summarizes our affine space constraint. Section 3 describes our initial outlier removal procedure. Section 4 describes how we extend partial trajectories and test their reliability. In Sec. 5, we show real video examples and demonstrate that our method can restore a sufficient number of trajectories for detailed 3-D reconstruction. Section 6 presents our conclusion.

In Appendix, the factorization algorithm for 3-D reconstruction based on our affine space constraint is concisely described. This description is slightly different from that in the literature: no matrix factorization by SVD (singular value decomposition) is involved.

2. Affine Space Constraint

We first summarize the geometric constraints on which our method is based. The same constraints have already been used in our previous studies [7]–[9], [14]. We reiterate them here, because they play a fundamental role in our trajectory extension method.

2.1 Trajectory of Feature Points

Suppose we track N feature points over M frames. Let $(x_{\kappa\alpha}, y_{\kappa\alpha})$ be the coordinates of the αth point in the κth frame. We stack all the coordinates vertically and represent the entire trajectory by the following $2M$-dimensional *trajectory vector*:

$$\boldsymbol{p}_\alpha = \begin{pmatrix} x_{1\alpha} & y_{1\alpha} & x_{2\alpha} & y_{2\alpha} & \cdots & x_{M\alpha} & y_{M\alpha} \end{pmatrix}^\top. \quad (1)$$

For convenience, we identify the frame number κ with "time" and refer to the κth frame as "time κ".

We regard the XYZ camera coordinate system as the world frame, relative to which the scene is moving. Consider a 3-D coordinate system fixed to the scene, and let $\boldsymbol{t}_\kappa$ and $\{\boldsymbol{i}_\kappa, \boldsymbol{j}_\kappa, \boldsymbol{k}_\kappa\}$ be, respectively, its origin and basis vectors at time κ. If the αth point has coordinates $(a_\alpha, b_\alpha, c_\alpha)$ with respect to this coordinate system, the position with respect to the world frame at time κ is

$$\boldsymbol{r}_{\kappa\alpha} = \boldsymbol{t}_\kappa + a_\alpha \boldsymbol{i}_\kappa + b_\alpha \boldsymbol{j}_\kappa + c_\alpha \boldsymbol{k}_\kappa. \quad (2)$$

2.2 Affine Camera Model

If an affine camera model (generalizing orthographic, weak perspective, and paraperspective projections [10]) is assumed, the image position of $\boldsymbol{r}_{\kappa\alpha}$ is

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \boldsymbol{A}_\kappa \boldsymbol{r}_{\kappa\alpha} + \boldsymbol{b}_\kappa, \quad (3)$$

where $\boldsymbol{A}_\kappa$ and $\boldsymbol{b}_\kappa$ are, respectively, a 2 × 3 matrix and a 2-dimensional vector determined by the position and orientation of the camera and its internal parameters at time κ. Substituting Eq. (2), we have

$$\begin{pmatrix} x_{\kappa\alpha} \\ y_{\kappa\alpha} \end{pmatrix} = \tilde{\boldsymbol{m}}_{0\kappa} + a_\alpha \tilde{\boldsymbol{m}}_{1\kappa} + b_\alpha \tilde{\boldsymbol{m}}_{2\kappa} + c_\alpha \tilde{\boldsymbol{m}}_{3\kappa}, \quad (4)$$

where $\tilde{\boldsymbol{m}}_{0\kappa}, \tilde{\boldsymbol{m}}_{1\kappa}, \tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ are 2-dimensional vectors determined by the position and orientation of the camera and its internal parameters at time κ. From Eq. (4), the trajectory vector $\boldsymbol{p}_\alpha$ in Eq. (1) can be written in the form

$$\boldsymbol{p}_\alpha = \boldsymbol{m}_0 + a_\alpha \boldsymbol{m}_1 + b_\alpha \boldsymbol{m}_2 + c_\alpha \boldsymbol{m}_3, \quad (5)$$

where $\boldsymbol{m}_0$, $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ are the $2M$-dimensional vectors obtained by stacking $\tilde{\boldsymbol{m}}_{0\kappa}, \tilde{\boldsymbol{m}}_{1\kappa}, \tilde{\boldsymbol{m}}_{2\kappa}$, and $\tilde{\boldsymbol{m}}_{3\kappa}$ vertically over the M frames, respectively.

2.3 Affine Space Constraint

Equation (5) implies that all the trajectories are constrained to be in the 4-dimensional subspace spanned by $\{\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3\}$ in $\mathcal{R}^{2M}$. This is called the *subspace constraint* [7], [8], on which the method of Jacobs [5] is based.

In addition, the coefficient of $\boldsymbol{m}_0$ in Eq. (5) is identically 1 for all α. This means that the trajectories are in the 3-dimensional affine space within that 4-dimensional subspace. This is called the *affine space constraint* [9].

If all the feature points are tracked to the final frame, we can define the coordinate origin at the centroid of their trajectory vectors $\{\boldsymbol{p}_\alpha\}$, thereby regarding them as defining a 3-dimensional subspace in $\mathcal{R}^{2M}$. The Tomasi-Kanade factorization [15] is based on this representation, and Brandt [1] tried to find this representation by iterations. In this paper, we directly use the affine space constraint without searching for the centroid.

Unlike existing studies, we describe our trajectory extension scheme without referring to any particular camera model, such as orthographic, weak perspective, or paraperspective projection, except that it is affine. Of course, existing methods described with respect to a particular camera model can automatically be generalized to all affine cameras, but our formulation makes this fact more explicit.

3. Outlier Removal

Before extending partial trajectories, we must remove incorrectly tracked trajectories, or "outliers", from among observed complete trajectories.

This problem was studied by Huynh and Heyden [4], who fitted a 4-dimensional subspace to the observed trajectories by LMedS [11], removing those trajectories sufficiently apart from it. However, their distance measure was introduced merely for mathematical convenience without giving much consideration to the statistical behavior of image noise.

Sugaya and Kanatani [14] fitted a 4-dimensional subspace to the observed trajectories by RANSAC [2], [3] and removed outliers using a χ^2 criterion derived from the error behavior of actual video tracking. Here, we modify their method specifically for the affine space constraint. Our method is a direct consequence of the principle given in [14], but we describe it here, because it plays a crucial role for our optimization procedure we introduce later.

3.1 Procedure

Let $n = 2M$, where M is the number of frames, and let $\{\boldsymbol{p}_\alpha\}$, $\alpha = 1, ..., N$, be the observed complete trajectory vectors. Our outlier removal procedure is as follows:

1. Randomly choose four vectors $\boldsymbol{q}_1$, $\boldsymbol{q}_2$, $\boldsymbol{q}_3$, and $\boldsymbol{q}_4$ from among $\{\boldsymbol{p}_\alpha\}$.
2. Compute the $n \times n$ moment matrix
$$\boldsymbol{M}_3 = \sum_{i=1}^{4}(\boldsymbol{q}_i - \boldsymbol{q}_C)(\boldsymbol{q}_i - \boldsymbol{q}_C)^\top, \qquad (6)$$
where $\boldsymbol{q}_C$ is the centroid of $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3, \boldsymbol{q}_4\}$.
3. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the three eigenvalues of the matrix $\boldsymbol{M}_3$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the orthonormal system of corresponding eigenvectors.
4. Compute the $n \times n$ projection matrix
$$\boldsymbol{P}_{n-3} = \boldsymbol{I} - \sum_{i=1}^{3}\boldsymbol{u}_i\boldsymbol{u}_i^\top. \qquad (7)$$
5. Let S be the number of points $\boldsymbol{p}_\alpha$ that satisfy
$$\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2 < (n-3)\sigma^2, \qquad (8)$$
where σ is an estimate of the noise standard deviation.
6. Repeat the above procedure a sufficient number of times[†], and determine the projection matrix $\boldsymbol{P}_{n-3}$ that maximizes S.
7. Remove those $\boldsymbol{p}_\alpha$ that satisfy
$$\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2 \geq \sigma^2\chi^2_{n-3;99}, \qquad (9)$$
where $\chi^2_{r;a}$ is the ath percentile of the χ^2 distribution with r degrees of freedom.

The term $\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2$, which we call the *residual*, is the squared distance of point $\boldsymbol{p}_\alpha$ from the fitted 3-dimensional affine space. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ, the residual $\|\boldsymbol{P}_{n-3}(\boldsymbol{p}_\alpha - \boldsymbol{q}_C)\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $n-3$ degrees of freedom. Hence, its expectation is $(n-3)\sigma^2$. The above procedure effectively fits a 3-dimensional affine space that maximizes the number of the trajectories whose residuals are smaller than $(n-3)\sigma^2$. After fitting such an affine space, we remove those trajectories which cannot be regarded as inliers with significance level 1% (Fig. 1). We have confirmed that the value $\sigma = 0.5$ can work well for all image sequences we tested [14].

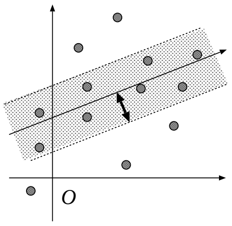

Fig. 1 Removing outliers by fitting a 3-dimensional affine space.

3.2 Final Affine Space Fitting

After removing outlier trajectories, we optimally fit a 3-dimensional affine space to the resulting inlier trajectories. Let $\{\boldsymbol{p}_\alpha\}$, $\alpha = 1, ..., N$, be their trajectory vectors. We first compute their centroid
$$\boldsymbol{p}_C = \frac{1}{N}\sum_{\alpha=1}^{N}\boldsymbol{p}_\alpha. \qquad (10)$$
Then, we compute the $n \times n$ moment matrix
$$\boldsymbol{M} = \sum_{\alpha=1}^{N}(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)^\top. \qquad (11)$$
Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ be the largest three eigenvalues of the matrix $\boldsymbol{M}$, and $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ the orthonormal system of corresponding eigenvectors. The optimally fitted 3-dimensional affine space is spanned by the three vectors of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ starting from $\boldsymbol{p}_C$.

Mathematically, this affine space fitting is equivalent to the factorization operation using SVD (singular value decomposition) [15]. It follows that no SVD is necessary for 3-D reconstruction once an affine space is fitted[††] (see Appendix).

4. Trajectory Extension

We now describe our trajectory extension scheme.

4.1 Reliability Test

If the αth feature point can be tracked only over κ of the M frames, its trajectory vector $\boldsymbol{p}_\alpha$ has $n-k$ unknown components (as before, we put $n = 2M$ and $k = 2\kappa$). We

[†] In our experiment, we stopped if S did not increase for 200 consecutive iterations.

[††] The statement that the method of Tomasi and Kanade [15] is based on matrix factorization using SVD is not correct. It simply means 3-D affine reconstruction based on the affine camera model. The SVD is merely one of many equivalent computational tools for it.

partition the vector $\boldsymbol{p}_\alpha$ into the k-dimensional part $\boldsymbol{p}_\alpha^{(0)}$ consisting of the k known components and the $(n-k)$-dimensional part $\boldsymbol{p}_\alpha^{(1)}$ consisting of the remaining $n-k$ unknown components. Similarly, we partition[‡†] the centroid $\boldsymbol{p}_C$ and the basis vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ into the k-dimensional parts $\boldsymbol{p}_C^{(0)}$ and $\{\boldsymbol{u}_1^{(0)}, \boldsymbol{u}_2^{(0)}, \boldsymbol{u}_3^{(0)}\}$ and the $(n-k)$-dimensional parts $\boldsymbol{p}_C^{(1)}$ and $\{\boldsymbol{u}_1^{(1)}, \boldsymbol{u}_2^{(1)}, \boldsymbol{u}_3^{(1)}\}$ in accordance with the division of $\boldsymbol{p}_\alpha$.

We test if each of the partial trajectories is sufficiently reliable. Let $\boldsymbol{p}_\alpha$ be a partial trajectory vector. If image noise does not exist, the deviation of $\boldsymbol{p}_\alpha$ from the centroid $\boldsymbol{p}_C$ should be expressed as a linear combination of $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$. Hence, there should be constants c_1, c_2, and c_3 such that

$$\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} = c_1\boldsymbol{u}_1^{(0)} + c_2\boldsymbol{u}_2^{(0)} + c_3\boldsymbol{u}^{(0)} \tag{12}$$

for the known part. In the presence of image noise, this equality does not hold. If we let $\boldsymbol{U}^{(0)}$ be the $k\times 3$ matrix consisting of $\boldsymbol{u}_1^{(0)}$, $\boldsymbol{u}_2^{(0)}$, and $\boldsymbol{u}_3^{(0)}$ as its columns, Eq. (12) is replaced by

$$\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} \approx \boldsymbol{U}^{(0)}\boldsymbol{c}, \tag{13}$$

where $\boldsymbol{c}$ is the 3-dimensional vector consisting of c_1, c_2, and c_3. Assuming that $k \geq 3$, we estimate the vector $\boldsymbol{c}$ by least squares in the form

$$\hat{\boldsymbol{c}} = \boldsymbol{U}^{(0)-}(\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)}), \tag{14}$$

where $\boldsymbol{U}^{(0)-}$ is the generalized inverse of $\boldsymbol{U}^{(0)}$. It is computed by

$$\boldsymbol{U}^{(0)-} = (\boldsymbol{U}^{(0)\top}\boldsymbol{U}^{(0)})^{-1}\boldsymbol{U}^{(0)\top}. \tag{15}$$

The residual, i.e., the squared distance of point $\boldsymbol{p}_\alpha^{(0)}$ from the 3-dimensional affine space spanned by $\{\boldsymbol{u}_1^{(0)}, \boldsymbol{u}_2^{(0)}, \boldsymbol{u}_3^{(0)}\}$ is $\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2$. If the noise in the coordinates of the feature points is an independent Gaussian random variable of mean 0 and standard deviation σ, the residual $\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2$ divided by σ^2 should be subject to a χ^2 distribution with $k-3$ degrees of freedom. Hence, we regard those trajectories that satisfy

$$\|\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)} - \boldsymbol{U}^{(0)}\hat{\boldsymbol{c}}\|^2 \geq \sigma^2\chi^2_{k-3;99} \tag{16}$$

as outliers with significance level 1%.

4.2 Extension and Optimization of Trajectories

The unknown part $\boldsymbol{p}_\alpha^{(1)}$ is estimated from the constraint implied by Eq. (12), namely

$$\boldsymbol{p}_\alpha^{(1)} - \boldsymbol{p}_C^{(1)} = c_1\boldsymbol{u}_1^{(1)} + c_2\boldsymbol{u}_2^{(1)} + c_3\boldsymbol{u}^{(1)} = \boldsymbol{U}^{(1)}\boldsymbol{c}, \tag{17}$$

where $\boldsymbol{U}^{(1)}$ is the $(n-k)\times 3$ matrix consisting of $\boldsymbol{u}_1^{(1)}$, $\boldsymbol{u}_2^{(1)}$, and $\boldsymbol{u}_3^{(1)}$ as its columns. Substituting Eq. (14) for $\boldsymbol{c}$, we obtain

$$\hat{\boldsymbol{p}}_\alpha^{(1)} = \boldsymbol{p}_C^{(1)} + \boldsymbol{U}^{(1)}\boldsymbol{U}^{(0)-}(\boldsymbol{p}_\alpha^{(0)} - \boldsymbol{p}_C^{(0)}). \tag{18}$$

Evidently, this is an optimal estimate in the presence of Gaussian noise. However, the underlying affine space is computed only from a small number of complete trajectories; no information contained in the partial trajectories is used, irrespective of how long they are. So, we incorporate partial trajectories by iterations.

Note that if three components of $\boldsymbol{p}_\alpha$ are specified, one can place it, in general, in any 3-dimensional affine space by appropriately adjusting the remaining $n-3$ components. In view of this, we introduce the "weight" of the trajectory vector $\boldsymbol{p}_\alpha$ with k known components in the form

$$W_\alpha = \frac{k-3}{n-3}. \tag{19}$$

Let N be the number of all trajectories, complete or partial, inliers or outliers. The optimization goes as follows:

1. Set the weights W_α of those trajectories, complete or partial, that are so far judged to be outliers to 0. All other weights are set to the value in Eq. (19).
2. Fit a 3-dimensional affine space to all the trajectories. The procedure is the same as described in Sec. 3.2 except that Eq. (10) is replaced by the *weighted* centroid
$$\boldsymbol{p}_C = \frac{\sum_{\alpha=1}^N W_\alpha \boldsymbol{p}_\alpha}{\sum_{\alpha=1}^N W_\alpha}, \tag{20}$$
and Eq. (11) is replaced by the *weighted* moment matrix
$$\boldsymbol{M} = \sum_{\alpha=1}^N W_\alpha(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)(\boldsymbol{p}_\alpha - \boldsymbol{p}_C)^\top. \tag{21}$$
3. Test each trajectory if it is an outlier, using Eq. (16).
4. Estimate the unknown parts of the inlier partial trajectory vectors, using Eq. (18).

These four steps are iterated until the fitted affine space converges. Eq. (18) implies that the estimated components do not contribute to the residual of the extended vector $\boldsymbol{p}_\alpha$ from the affine space, so the reliability of extended trajectories is tested only from their known components using Eq. (16). In the course of this optimization, trajectories once regarded as outliers may be judged to be inliers later, and vice versa. In the end, inlier partial trajectories are optimally extended with respect to the affine space that is optimally fitted to all the complete and partial inlier trajectories.

However, the resulting solution is not guaranteed to be globally optimal; its accuracy largely depends on the quality of the initial guess. The outlier removal procedure of Sec. 3 is incorporated for obtaining as accurate an initial guess as possible, even though all trajectories are reexamined later.

[†††] This is merely for the convenience of description. In real computation, we treat all data as n-dimensional vectors after multiplying them by an appropriate diagonal matrix consisting of 1s and 0s.

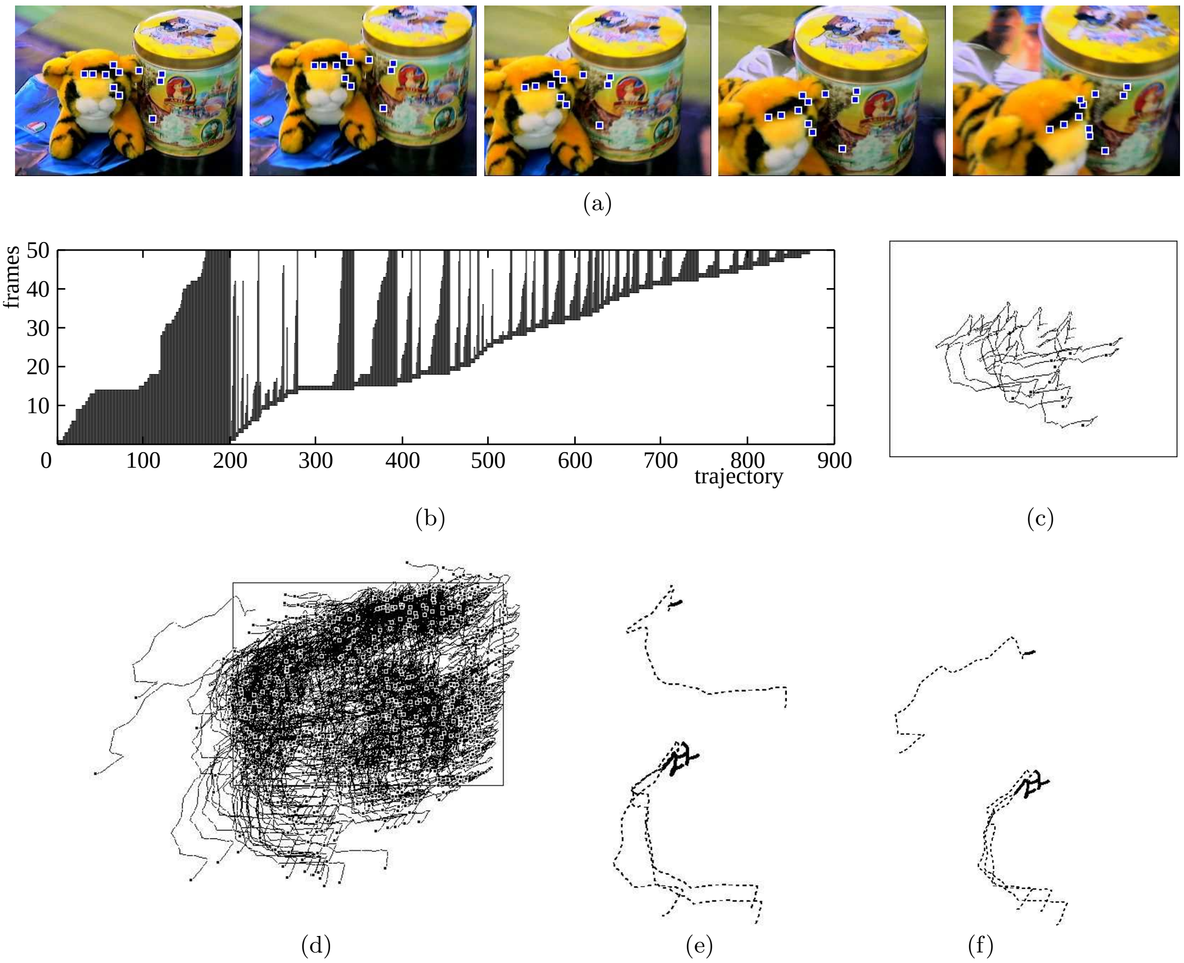

Fig. 2 (a) Five decimated frames from a 50 frame sequence and 11 points correctly tracked throughout the sequence. (b) The life spans of the detected 871 trajectories. (c) The 11 complete inlier trajectories. (d) The 560 optimal extensions of the trajectories. (e) Four trajectories before optimization. The real lines show the original data, and the dotted lines show the estimated parts. (f) The corresponding optimized trajectories.

The iterations may not converge if the initial guess is very poor or a large proportion of the trajectories are incorrect. In that case, we must conclude that the original feature tracking does not provide meaningful information. However, this did not happen in any of our experiments using real video sequences.

We need at least three complete trajectories for guessing the initial affine space. If no such trajectories are given, we may use the method of Jacobs [5] for an initial guess. However, it is much more practical to segment the sequence into overlapping blocks, extending partial trajectories over each block separately and connecting all the blocks to find complete trajectories.

5. Experiments

We tested our method using real video sequences. Figure 2(a) shows five decimated frames from a 50 frame sequence (320 × 240 pixels) of a static scene taken by a moving camera. We detected 200 feature points and tracked them using the Kanade-Lucas-Tomasi algorithm [16]. When tracking failed at some frame, we restarted the tracking after adding a new feature point in that frame. Figure 2(b) shows the life space of the 871 trajectories thus obtained: they are enumerated on the horizontal axis in the order of disappearance and new appearance; the white part corresponds to missing data.

Among them, 29 are complete trajectories, of which 11 are regarded as inliers by the procedure described in Sec. 3. The marks □ in Fig. 2(a) indicate their positions; Figure 2(c) shows their trajectories.

Using the affine space they define, we extended the partial trajectories and optimized the affine space and the extended trajectories. The optimization converged after 11 iterations, resulting in the 560 inlier trajectories shown in Fig. 2(d). The computation time for this optimization was 134 seconds. We used Pentium 4 2.4B GHz for the CPU with 1 GB main memory and Linux for the OS.

Figure 2(e),(f) shows four enlarged trajectories that underwent significant corrections by the optimization: the trajectories in Fig. 2(e), which appeared to scatter inconsistently, were corrected into those in Fig. 2(f), which are more consistent with the global motion. The

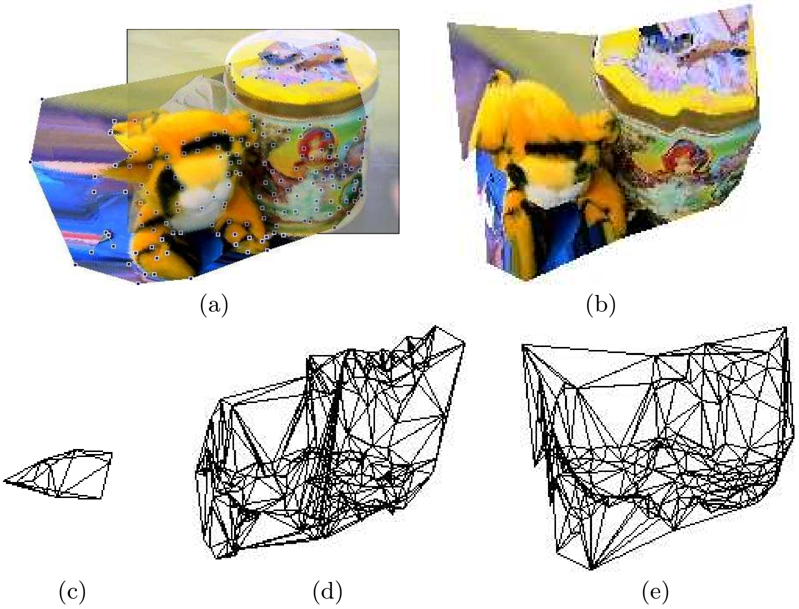

Fig. 3 (a) The extrapolated texture-mapped image of the 33th frame. (b) The reconstructed 3-D shape. (c) The patches reconstructed from the 11 initial complete (d) The patches reconstructed from all extended trajectories without optimization. (e) The corresponding result with optimization.

solid lines indicate the original data; the dashed lines indicate the estimated parts.

Figure 3(a) is the extrapolated image of the 33th frame after missing feature positions are restored: using the 180 feature points visible in the first frame, we defined triangular patches, to which the texture in the first frame is mapped.

We reconstructed the 3-D shape by factorization based on weak perspective projection (see Appendix). Figure 3(b) is the top view of the texture-mapped shape. Figure 3(c) shows the patches reconstructed from the 11 initial trajectories in Fig. 2(c). Evidently, a meaningful 3-D shape cannot be reconstructed from such a small number of feature points. Figure 3(d) shows the patches reconstructed from extended trajectories without optimization; Figure 3(e) is the corresponding shape after optimization.

From these results, we can see that a sufficient number of trajectories can be restored for detailed 3-D reconstruction by extending the partial trajectories and that incorrect trajectories are removed or corrected by the optimization process. According to visual inspection, the reconstructed 3-D shape appears to be better after the optimization, but the difference is small. This is probably because the effects of trajectory errors are suppressed by our factorization algorithm (Appendix), which optimizes the solution using all the data in all the frames.

6. Concluding Remarks

We have presented a new method for extending interrupted feature point tracking for 3-D affine reconstruction. Our method consists of iterations for optimally extending the trajectories and for optimally estimating the affine space. In every step, the reliability of the extended trajectories is tested, and those judged to be outliers are removed. Using real video images, we have demonstrated that a sufficient number of trajectories can be restored for detailed 3-D reconstruction.

Acknowledgments

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan, under a Grant in Aid for Scientific Research C(2) (No. 15500113), the Support Center for Advanced Telecommunications Technology Research, and Kayamori Foundation of Informational Science Advancement.

References

[1] S. Brandt, “Closed-form solutions for affine reconstruction under missing data,” Proc. Statistical Methods in Video Processing Workshop, pp.109–114, Copenhagen, Denmark, June 2002.

[2] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Commun. ACM, vol.24, no.6, pp.381–395, June 1981.

[3] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge, U.K., 2000.

[4] D. Q. Huynh and A. Heyden, “Outlier detection in video sequences under affine projection,” Proc. IEEE Conf. Comput. Vis. Pattern Recog., vol.2, pp.695–701, Kauai, HI, U.S.A., Dec. 2001.

[5] D. W. Jacobs, “Linear fitting with missing data for structure-from-motion” Comput. Vis. Image Unders., vol.82, no.1, pp.57–81, April 2001.

[6] F. Kahl and A. Heyden, “Affine structure and motion from points, lines and conics,” Int. J. Comput. Vis., vol.33, no.3, pp.163–180, Sept. 1999.

[7] K. Kanatani, "Motion segmentation by subspace separation and model selection," Proc. 8th Int. Conf. Comput. Vis., vol.2, pp.301-306, Vancouver, Canada, July 2001.

[8] K. Kanatani, "Motion segmentation by subspace separation: Model selection and reliability evaluation," Int. J. Image Graphics, vol.2, no.2, pp.179–197, April 2002.

[9] K. Kanatani, "Evaluation and selection of models for motion segmentation," Proc. 7th Euro. Conf. Comput. Vis., pp.335–349, Copenhagen, Denmark, June 2002.

[10] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.3, pp.206–218, March 1997.

[11] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection, Wiley, New York, 1987.

[12] H. Saito and S. Kamijima, "Factorization method using interpolated feature tracking via projective geometry," Proc. 14th British Machine Vision Conf., vol.2, pp.449–458, Norwich, UK, Sept. 2003.

[13] H.-Y. Shum, K. Ikeuchi and R. Reddy, "Principal component analysis with missing data and its application to polyhedral object modeling," IEEE Trans. Pattern Anal. Mach. Intell., vol.17, no.3, pp.854–867, Sept. 1995.

[14] Y. Sugaya and K. Kanatani, "Outlier removal for motion tracking by subspace separation," IEICE Trans. Inf. & Syst., vol.E86-D, no.6, pp.1095–1102, June 2003.

[15] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography—A factorization method," Int. J. Comput. Vis., vol.9, no.2, pp.137–154, Nov. 1992.

[16] C. Tomasi and T. Kanade, "Detection and tracking of point features," CMU Tech. Rep. CMU-CS-91-132, April 1991; http://vision.stanford.edu/˜birch/klt/

Appendix : 3-D Reconstruction Algorithm

Input

- $2M$-dimensional trajectory vectors $\boldsymbol{p}_\alpha$, $\alpha = 1, ..., N$.
- Focal length f_κ for the κth frame, $\kappa = 1, ..., M$.
- Average depth Z_c.

Output

- 3-D positions $\{\hat{\boldsymbol{r}}_\alpha\}$ and $\{\hat{\boldsymbol{r}}'_\alpha\}$ (mutually mirror images) of the N feature points.

Algorithm

1. Let $\boldsymbol{p}_C$ be the centroid of $\{\boldsymbol{p}_\alpha\}$, and $\boldsymbol{U}$ the $2M \times 3$ matrix consisting of the orthonomal basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\}$ of the fitted 3-D affine space (cf. Sec. 3.2).
2. Let $\boldsymbol{u}^\dagger_{\kappa(a)}$ ($\kappa = 1, ..., M$, $a = 1, 2$) be the $(2(\kappa - 1) + a)$th column of $\boldsymbol{U}^\top$.
3. (*Metric condition*) Compute the 3×3 matrix $\boldsymbol{T}$ that minimizes the following function (the procedure is described later):

$$K = \sum_{\kappa=1}^{M} \Big[\Big((\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(1)}) - (\boldsymbol{u}^\dagger_{\kappa(2)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)}) \Big)^2 + (\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)})^2 \Big]. \qquad (\text{A}\cdot 1)$$

4. Compute the Z coordinate of the translation $\boldsymbol{t}_\kappa$ by

$$t_{z\kappa} = f_\kappa \sqrt{\frac{2}{(\boldsymbol{u}^\dagger_{\kappa(1)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(1)}) + (\boldsymbol{u}^\dagger_{\kappa(2)}, \boldsymbol{T}\boldsymbol{u}^\dagger_{\kappa(2)})}}. \qquad (\text{A}\cdot 2)$$

5. Let $\tilde{t}_{x\kappa}$ and $\tilde{t}_{y\kappa}$ be, respectively, the $(2(\kappa-1)+1)$th and $(2(\kappa-1)+2)$th components of the centroid $\boldsymbol{p}_C$.
6. Compute the X and Y coordinates of the translation $\boldsymbol{t}_\kappa$ by

$$t_{x\kappa} = \frac{t_{z\kappa}}{f_\kappa}\tilde{t}_{x\kappa}, \qquad t_{y\kappa} = \frac{t_{z\kappa}}{f_\kappa}\tilde{t}_{y\kappa}. \qquad (\text{A}\cdot 3)$$

7. Let λ_1, λ_2, and λ_3 be the eigenvalues of the matrix $\boldsymbol{T}$, and $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ the orthonormal system of corresponding eigenvectors.
8. Compute the $2M$-dimensional vector

$$\boldsymbol{m}_i = \sqrt{\lambda_i} \begin{pmatrix} (\boldsymbol{u}^\dagger_{1(1)}, \boldsymbol{v}_i) \\ (\boldsymbol{u}^\dagger_{1(2)}, \boldsymbol{v}_i) \\ (\boldsymbol{u}^\dagger_{2(1)}, \boldsymbol{v}_i) \\ \vdots \\ (\boldsymbol{u}^\dagger_{M(2)}, \boldsymbol{v}_i) \end{pmatrix} \qquad (\text{A}\cdot 4)$$

for $i = 1, 2, 3$.

9. Let $\boldsymbol{M}$ be the $2M \times 3$ matrix consisting of $\boldsymbol{m}_1$, $\boldsymbol{m}_2$, and $\boldsymbol{m}_3$ as its columns.
10. Let $\boldsymbol{m}^\dagger_{\kappa(a)}$ ($\kappa = 1, ..., M$, $a = 1, 2$) be the $(2(\kappa - 1) + a)$th column of $\boldsymbol{M}^\top$.
11. Compute the following SVD:

$$\frac{t_{z\kappa}}{f_\kappa} \begin{pmatrix} \boldsymbol{m}^\dagger_{\kappa(1)} & \boldsymbol{m}^\dagger_{\kappa(2)} & \boldsymbol{0} \end{pmatrix} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{U}^\top. \qquad (\text{A}\cdot 5)$$

12. Compute the rotation matrices $\{\boldsymbol{R}_\kappa\}$ as follows:

$$\boldsymbol{R}_\kappa = \boldsymbol{U}\mathrm{diag}(1, 1, \det(\boldsymbol{V}\boldsymbol{U}^\top))\boldsymbol{V}^\top. \qquad (\text{A}\cdot 6)$$

13. Recompute the matrix $\boldsymbol{M}$ by

$$\boldsymbol{M} = \sum_{\kappa=1}^{M} \Pi_\kappa^\top \boldsymbol{R}_\kappa, \qquad (\text{A}\cdot 7)$$

where $\Pi_\kappa = (\Pi_{\kappa(ij)})$ is a $3\times2M$ matrix with element $\Pi_{\kappa(ij)} = f_\kappa/t_{z\kappa}$ for $(i, j) = (2, 2\kappa - 1)$, $(2, 2\kappa)$ and 0 otherwise.

14. Compute the 3-D shape vectors

$$\boldsymbol{s}_\alpha = (\boldsymbol{M}^\top\boldsymbol{M})^{-1}\boldsymbol{M}^\top(\boldsymbol{p}_\alpha - \boldsymbol{p}_C). \qquad (\text{A}\cdot 8)$$

15. Compute $\{\boldsymbol{s}'_\alpha\}$ and $\boldsymbol{R}'_1$ as follows:

$$\boldsymbol{s}'_\alpha = -\boldsymbol{s}_\alpha, \ \boldsymbol{R}'_1 = \mathrm{diag}(-1, -1, 1)\boldsymbol{R}_1. \qquad (\text{A}\cdot 9)$$

16. Compute $\{\hat{\boldsymbol{r}}_\alpha\}$ and $\{\hat{\boldsymbol{r}}'_\alpha\}$ as follows:

$$\hat{\boldsymbol{r}}_\alpha = \frac{Z_c}{t_{z1}}(\boldsymbol{R}_1\boldsymbol{s}_\alpha + \boldsymbol{t}_1), \qquad \hat{\boldsymbol{r}}'_\alpha = \frac{Z_c}{t_{z1}}(\boldsymbol{R}'_1\boldsymbol{s}'_\alpha + \boldsymbol{t}_1). \qquad (\text{A}\cdot 10)$$

Metric condition (Step 3 in the above algorithm)

1. Define the $3 \times 3 \times 3 \times 3$ tensor $\mathcal{A} = (A_{ijkl})$ by

$$A_{ijkl} = \sum_{\kappa=1}^{M} \Big[(\boldsymbol{u}^\dagger_{\kappa(1)})_i (\boldsymbol{u}^\dagger_{\kappa(1)})_j (\boldsymbol{u}^\dagger_{\kappa(1)})_k (\boldsymbol{u}^\dagger_{\kappa(1)})_l$$

$$
\begin{aligned}
&-(\boldsymbol{u}^{\dagger}_{\kappa(1)})_i(\boldsymbol{u}^{\dagger}_{\kappa(1)})_j(\boldsymbol{u}^{\dagger}_{\kappa(2)})_k(\boldsymbol{u}^{\dagger}_{\kappa(2)})_l\\
&-(\boldsymbol{u}^{\dagger}_{\kappa(2)})_i(\boldsymbol{u}^{\dagger}_{\kappa(2)})_j(\boldsymbol{u}^{\dagger}_{\kappa(1)})_k(\boldsymbol{u}^{\dagger}_{\kappa(1)})_l\\
&+(\boldsymbol{u}^{\dagger}_{\kappa(2)})_i(\boldsymbol{u}^{\dagger}_{\kappa(2)})_j(\boldsymbol{u}^{\dagger}_{\kappa(2)})_k(\boldsymbol{u}^{\dagger}_{\kappa(2)})_l\\
&+\frac{1}{4}\Big((\boldsymbol{u}^{\dagger}_{\kappa(1)})_i(\boldsymbol{u}^{\dagger}_{\kappa(2)})_j(\boldsymbol{u}^{\dagger}_{\kappa(1)})_k(\boldsymbol{u}^{\dagger}_{\kappa(2)})_l\\
&+(\boldsymbol{u}^{\dagger}_{\kappa(2)})_i(\boldsymbol{u}^{\dagger}_{\kappa(1)})_j(\boldsymbol{u}^{\dagger}_{\kappa(1)})_k(\boldsymbol{u}^{\dagger}_{\kappa(2)})_l\\
&+(\boldsymbol{u}^{\dagger}_{\kappa(1)})_i(\boldsymbol{u}^{\dagger}_{\kappa(2)})_j(\boldsymbol{u}^{\dagger}_{\kappa(2)})_k(\boldsymbol{u}^{\dagger}_{\kappa(1)})_l\\
&+(\boldsymbol{u}^{\dagger}_{\kappa(2)})_i(\boldsymbol{u}^{\dagger}_{\kappa(1)})_j(\boldsymbol{u}^{\dagger}_{\kappa(2)})_k(\boldsymbol{u}^{\dagger}_{\kappa(1)})_l\Big)\Big],
\end{aligned}
\qquad (\mathrm{A}\cdot 11)
$$

where $(\boldsymbol{u}^{\dagger}_{\kappa(a)})_i$ is the ith component of $\boldsymbol{u}^{\dagger}_{\kappa(a)}$.

2. Define the 6 × 6 matrix

$$
\boldsymbol{A} = \begin{pmatrix}
A_{1111} & A_{1122} & A_{1133} & \sqrt{2}A_{1123} & \sqrt{2}A_{1131} & \sqrt{2}A_{1112}\\
A_{2211} & A_{2222} & A_{2233} & \sqrt{2}A_{2223} & \sqrt{2}A_{2231} & \sqrt{2}A_{2212}\\
A_{3311} & A_{3322} & A_{3333} & \sqrt{2}A_{3323} & \sqrt{2}A_{3331} & \sqrt{2}A_{3312}\\
\sqrt{2}A_{2311} & \sqrt{2}A_{2322} & \sqrt{2}A_{2333} & 2A_{2323} & 2A_{2331} & 2A_{2312}\\
\sqrt{2}A_{3111} & \sqrt{2}A_{3122} & \sqrt{2}A_{3133} & 2A_{3123} & 2A_{3131} & 2A_{3112}\\
\sqrt{2}A_{1211} & \sqrt{2}A_{1222} & \sqrt{2}A_{1233} & 2A_{1223} & 2A_{1231} & 2A_{1212}
\end{pmatrix}. \qquad (\mathrm{A}\cdot 12)
$$

3. Compute the 6-dimensional eigenvector $\boldsymbol{\tau}$ for the smallest eigenvalue of the matrix $\boldsymbol{A}$.
4. Let $\boldsymbol{T}$ be the 3 × 3 matrix

$$
\boldsymbol{T} = \begin{pmatrix}
\tau_1 & \tau_6/\sqrt{2} & \tau_5/\sqrt{2}\\
\tau_6/\sqrt{2} & \tau_2 & \tau_4/\sqrt{2}\\
\tau_5/\sqrt{2} & \tau_4/\sqrt{2} & \tau_3
\end{pmatrix}. \qquad (\mathrm{A}\cdot 13)
$$

5. If $\det\boldsymbol{T} < 0$, then let $\boldsymbol{T} \leftarrow -\boldsymbol{T}$.

Kenichi Kanatani received his M.S. and Ph.D. in applied mathematics from the University of Tokyo in 1974 and 1979, respectively. After serving as Professor of computer science at Gunma University, Japan, he is currently Professor of information technology at Okayama University, Japan. He authored many books on computer vision. He is an IEEE Fellow.

Yasuyuki Sugaya received his M.S. and Ph.D. from the University of Tsukuba, Japan, in 1998 and 2001, respectively. He is currently Assistant Professor of information technology at Okayama University, Japan. His research interests include image processing and computer vision.