I-013

頑健なモデルベースカメラトラッキングのための適応的エッジ検出およびトラッキング

Adaptive Edge Detection and Tracking for Robust Model-Based Camera Tracking

朴 漢薫† 三ッ峰 秀樹† 藤井 真人†
Hanhoon Park Hideki Mitsumine Mahito Fujii

1. Introduction

In model-based camera tracking, since edge does not have its descriptor, it is usually tracked locally with a weak prior knowledge and thus likely to be tracked incorrectly. Incorrectly tracked edges result in inaccurate and unstable camera tracking. One can say that this problem could be alleviated by using so-called probabilistic methods [3, 7]. However, making edges tracked correctly will be more effective. In this regard, methods of using robust estimators [2, 8] or cooperatively tracking multiple visual cues (e.g. edges and feature points [5]) have been proposed. However, they are those that do something on a given (or unadjustable) edge map (extracted explicitly or implicitly) and without any information about the true correspondences (= hypotheses). Therefore, their performance is limited to some extent. If having the information about the true correspondences and enhancing the edge map (by suppressing the false hypotheses), it will be much easier to make edges tracked correctly. This paper presents such a method. The method trains the gradients of the true correspondences in the previous frames and improves a conventional edge detector to selectively detect edges that have similar gradients to the true correspondences in the current frame.

A similar work was done by Wuest et al. [9]. However, they trained the pixel values (i.e. colors) of edges, in contrast to our approach of training the gradients of edges. In addition, they used the trained information to find the true hypotheses from the implicitly detected multiple hypotheses, in contrast to our approach of adjusting the thresholds of a conventional edge detector using the trained information to explicitly detect only the true hypotheses.

2. Testbed System

As a testbed, we use a linear and iterative model-based camera tracking system [5]. In the system, edges of the 3D scene model are tracked (= matched with their correspondences) as follows. First, the edges are tested if they are visible in the current camera view. Then, the visible edges are sampled with an equi-distance and the sampled edge points are projected onto the camera image plane. Next, strong image edges are detected from the current frame using the Canny operator in [4]. Then, the projected edge points are matched with the detected image edges closer to them. The camera pose is estimated by minimizing the weighted Euclidean distance from the projected edge points to their matches.

Consequently, the accuracy and stability of the testbed system greatly depend on the edge detection results by the Canny operator. Therefore, to increase the accuracy and stability, a method for enhancing the edge detection results is explained in the next section.

3. Proposed Method

3.1 Modeling and Updating the Visual Properties of Edge

An edge's visual property can be modeled as a mixture of N Gaussian distributions [6]. That is, the probability of observing the edge's visual property x at time t is

$$P(x_t) = \sum_{i=1}^{N} w_{i,t}\Phi(x_t, \mu_{i,t}, \sigma_{i,t}) \quad (1)$$

where Φ is the Gaussian probability density function. The weight w_i represents a measure of what portion of data is taken into account by the i-th Gaussian.

Given a new property value x_t at time t, the mixture model is updated as follows. First, if there is a distribution k that is matched with x_t, i.e. $|x_t - \mu_{k,t-1}| < 2.5\sigma_{k,t-1}$, the parameters of the distribution are updated as follows [6].

$$w_{k,t} = (1-\lambda)w_{k,t-1} + \lambda, \quad (2)$$

$$\mu_{k,t} = (1-\rho)\mu_{k,t-1} + \rho x_t, \quad (3)$$

$$\sigma_{k,t} = \sqrt{(1-\rho)\sigma_{k,t-1}^2 + \rho(x_t - \mu_{k,t})^2}. \quad (4)$$

where λ is the learning rate and $\rho = \lambda\Phi(x_t, \mu_{k,t-1}, \sigma_{k,t-1})$. Second, if there is no such a matched distribution, the mean of the least probable distribution j is replaced by x_t, its standard deviation is initialized with a high value (σ_0), and its weight is initialized with a low value (w_0). Finally, the weight of the other distributions that are neither matched with x_t nor the least probable distribution is updated as follows.

$$w_{i,t} = (1-\lambda)w_{i,t-1},\ 1 \le i \le N, i \ne k, i \ne j. \quad (5)$$

Their means and standard deviations remain the same.

3.2 Adaptive Edge Detection

A gray-scaled camera image is divided into $B_w \times B_h$ blocks[1]. In each block, the gradients ($g = |g_x| + |g_y|$ where g_x and g_y are computed by applying the horizontal and vertical Sobel operators to the gray-scaled camera image, respectively.) of the correspondences (= true hypotheses) of correctly tracked edges[2]

† NHK Science & Technology Research Laboratories

[1] Since the visual properties of adjacent edges are similar, the process can be done blockwise.

[2] In this paper, it is determined that edges were correctly tracked

are modeled and updated by a mixture of N Gaussian distributions as explained in Sect. 3.1. The parameter λ in Eqs. (2) and (5) is controlled by the accuracy of the pose estimation, i.e. $\lambda = \lambda_0/(err_e+1)$ where err_e is the reprojection error of edges. Then, among the N distributions, the mean (μ) and standard deviation (σ) of a distribution that has w larger than a constant W_{min} and also has a largest w/σ are used to determine the thresholds of the Canny operator[1] as follows. If $(\mu - 2.5\sigma) > \mu/2$,

$$h_{th} = \mu - 2.5\sigma,\ l_{th} = \mu/2. \quad (6)$$

else,

$$h_{th} = \mu/2,\ l_{th} = \mu - 2.5\sigma. \quad (7)$$

And the edges, of which g is larger than $\mu + 2.5\sigma$, are removed. This indicates to adjust the thresholds so as to detect only the edges that have similar gradients to the true hypotheses and suppress the others (i.e. false hypotheses having the weaker or stronger gradients) in the next frames.

If a block does not include correctly tracked edges, its Gaussian mixture model is reset. If there is no distribution having w larger than W_{min}, the thresholds are initialized to pre-defined values.

Note that, by more precisely dividing the image using a sophisticated segmentation method, by separately computing the gradients in each color channel, by training the gradients of all pixels (true hypotheses, false hypotheses, and pixels that are not edges), or by using the multiple distributions having large w/σ, it would be possible to adjust the thresholds more precisely. However, they cost lots of time (in practice, they did in our preliminary experiments). The method presented in this section is a simple and efficient one among the possible ones.

4. Experiments and Discussion

For experiments, video sequences (640 × 480 pixels) were obtained by freely moving a calibrated camera (Logitech Qcam Pro 9000) around a real scene (Fig. 1) several times. From the sequences, the camera poses were estimated using the testbed camera tracking system (without or with the adaptive edge detection). Then, pose errors were computed by differencing the poses from those that were initialized by the ARToolKit [1] and optimized by applying sparse bundle adjustment [10] to the correspondences of feature points stacked over L frames. Jitters were computed from the temporal differentiation of the pose errors. The errors and jitters were averaged over the video sequences. In the adaptive edge detection, N, σ_0, w_0, B_w, B_h, λ_0, and W_{min} was heuristically set to 3, 100, 0.1, 64, 48, 0.3, and 0.7.

As guessed in Sect. 3, the proposed method assumes that the gradients of the true hypotheses and those of the false hypotheses can be separated (specifically, their difference must be larger than at least 2.5σ). To show that the assumption is satisfied, we also looked into the gradients of the false hypotheses and compared them with those of the true hypotheses. In the upper figure of Fig. 2, the solid line represents the difference between the mean of the gradients of the true hypotheses and that of the false hypotheses in a block and the dashed line represents the value of 2.5σ. In the lower figure, '1' indicates that there exists a distribution that has w larger than a constant W_{min} and also has a largest w/σ among the N distributions for the true hypotheses, i.e. the adaptive thresholds computed by Eqs. (6) and (7) are used. In contrast, '0' indicates that the predefined thresholds are used. When the adaptive thresholds were used, the difference between the mean of the gradients of the true hypotheses and that of the false hypotheses was always larger than 2.5σ. Consequently, we can know that the assumption is satisfied in practice.

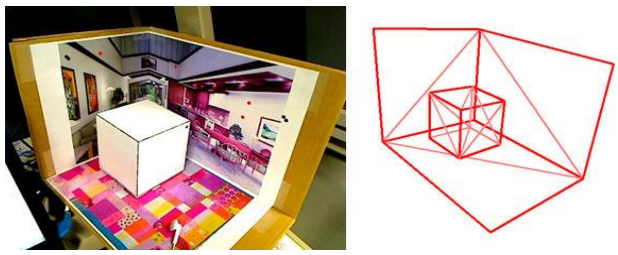


Fig. 1. A real scene (215 × 310 × 215 mm) and its 3D wired model used in our experiments.

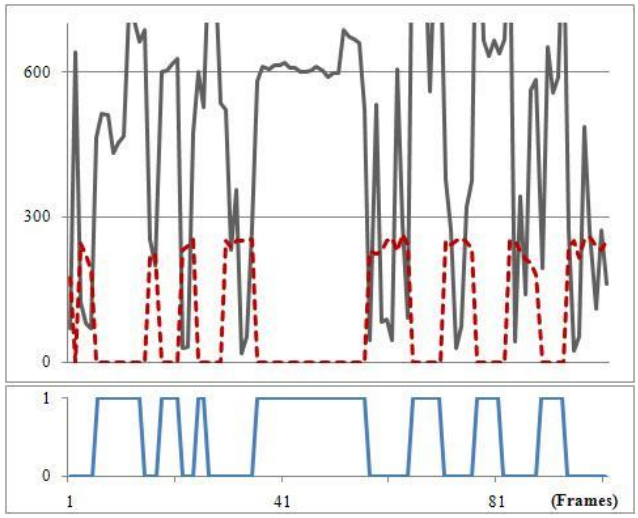


Fig. 2. Upper: The solid line represents the difference between the mean of the gradients of the true hypotheses and that of the false hypotheses in a block and the dashed line represents the value of 2.5σ, lower: '1' and '0' indicate that the adaptive edge detection is available or not.

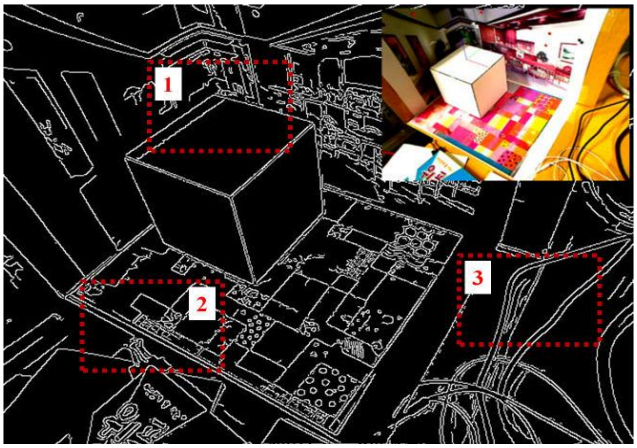
Without the adaptive edge detection, the projected edges had about 1.898 hypotheses on average and were relatively likely to be matched with the false hypotheses as shown in Fig. 3. The camera pose errors in translations and rotations were about 4.439 mm and 0.541 degree, respectively (Fig. 4 and Tab. 1). In addition, the jitters in translations and rotations were about 1.383 mm/frame and 0.177 degree/frame, respectively (Fig. 4 and Tab. 1).

However, by using the adaptive edge detection, the number of

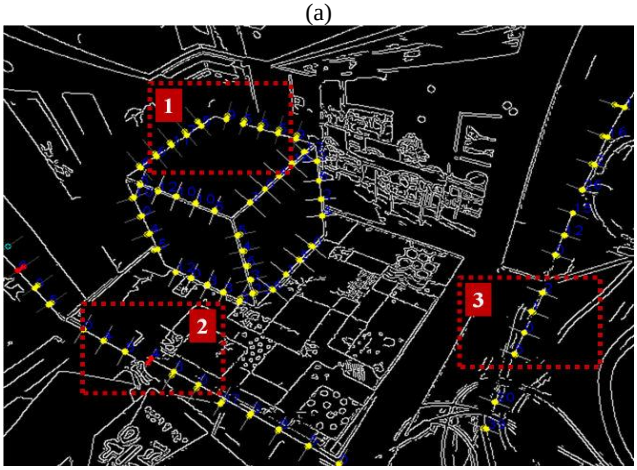
when their reprojection errors were lower than 2.

[1] Since, in our testbed system, the Canny operator was used for detecting edges, we described a method of adjusting the thresholds (high and low ones for hysteresis thresholding) of the Canny operator in this paper. However, the scheme can be used for other edge detectors.

hypotheses was reduced to 1.793 (the false hypotheses were less detected) and only the hypotheses that have similar gradients to the true hypotheses remained. The projected edges were likely to be matched with the true hypotheses as shown in Fig. 3.



(a)



(b)

Fig. 3. Edge tracking. (a) Edges detected by the conventional Canny operator, (b) edges detected by the adaptive edge detection method and their tracking. By reducing the false hypotheses, the probability of matching with the false hypotheses decreased. In the images, small empty and filled circles indicate the projections of the sampled edge points on the 3D scene model and their correspondences, respectively, and short straight lines passing through the circles indicate the range for searching the correspondences. In this paper, the searching range was set to 30.

By using the adaptive edge detection, the camera pose errors were reduced to 3.967 mm in translations and 0.493 degree in rotations and the jitters were reduced to 1.305 mm/frame in translations and 0.166 degree/frame in rotations as shown in Fig. 4 and Tab. 2. As mentioned before, the proposed method is based on the assumption that the gradients of the true hypotheses are separated from those of the false hypotheses. Therefore, in the regions where the gradients of the true and false hypotheses are similar, it is possible that the variation of the gradients caused by illumination or viewpoint changes is larger than the difference in gradients between the true and false hypotheses. In that case, the true hypotheses may be rejected by the wrong Gaussian distributions. Thus, the rejection was forced to be conservative to avoid false positives and the overall improvement was not impressive as expected in our experiments where the scene conditions were not manipulated for satisfying the assumption.

Table 1. Camera pose error and jitter without the adaptive edge detection

	t_1	t_2	t_3	r_1	r_2	r_3
Error	4.336	4.867	4.112	0.688	0.453	0.484
Jitter	1.326	1.528	1.295	0.214	0.164	0.152

Table 2. Camera pose error and jitter with the adaptive edge detection

	t_1	t_2	t_3	r_1	r_2	r_3
Error	4.260	4.291	3.351	0.558	0.448	0.472
Jitter	1.378	1.337	1.199	0.211	0.140	0.146

The adaptive edge detection increased the processing time of the testbed system by about 4 ms and slightly dropped the frame rate from 25 Hz to around 23 Hz on a laptop PC (2.8 GHz dual-core CPU). Therefore, we would like to say that the proposed method focuses on increasing the accuracy and stability at the small expense of speed.

5. Conclusion

In this paper, we proposed an adaptive edge detection and tracking method that models the edge gradients as a mixture of Gaussian distributions and adjusts the thresholds of the Canny operator. Based on the method, we could reduce the number of false hypotheses by 12% and improve the accuracy and stability of the testbed model-based camera tracking system by 10% and 6%, respectively.

In this paper, the lighting environment was fixed. However, the proposed method can be used for the purpose of robustly tracking edges under a time-varying lighting environment. Its verification remains as a future work.

References

[1] ARToolKit, http://www.hitl.washington.edu/artoolkit/
[2] T. Drummond and R. Cipolla, “Real-time visual tracking of complex structures,” IEEE T. on PAMI, vol. 24, no. 7, pp. 932-946, 2002.
[3] E. Eade and T. Drummond, “Edge landmarks in monocular SLAM,” Ima. and Vis. Comp., vol. 27, no. 5, pp. 588-596, 2009.
[4] OpenCV, http://sourceforge.net/projects/opencvlibrary/
[5] H. Park, J. Oh, B.-K. Seo and J.-I. Park, “Automatic confidence adjustment of visual cues in model-based camera tracking,” Comp. Anim. and Virt. Worlds, vol. 21, no. 2, pp. 69-79, 2010.
[6] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in CVPR, pp. 2246-2252, 1999.
[7] C. Teuliere, E. Marchand, and L. Eck, “Using multiple hypothesis in model-based tracking,” in ICRA, pp. 4559-

4565, 2010.
[8] L. Vacchetti, V. Lepetit, and P. Fua, “Combining edge and texture information for real-time accurate 3D camera tracking,” in ISMAR, pp. 62-69, 2005.
[9] H. Wuest, F. Vial, and D. Stricker, “Adaptive line tracking with multiple hypotheses for augmented reality,” in ISMAR, pp. 62-69, 2005.
[10] M. A. Lourakis and A. Argyros, “SBA: A software package for generic sparse bundle adjustment,” ACM T. on Math. Soft., vol. 36, no. 1, pp. 1-30, 2009.

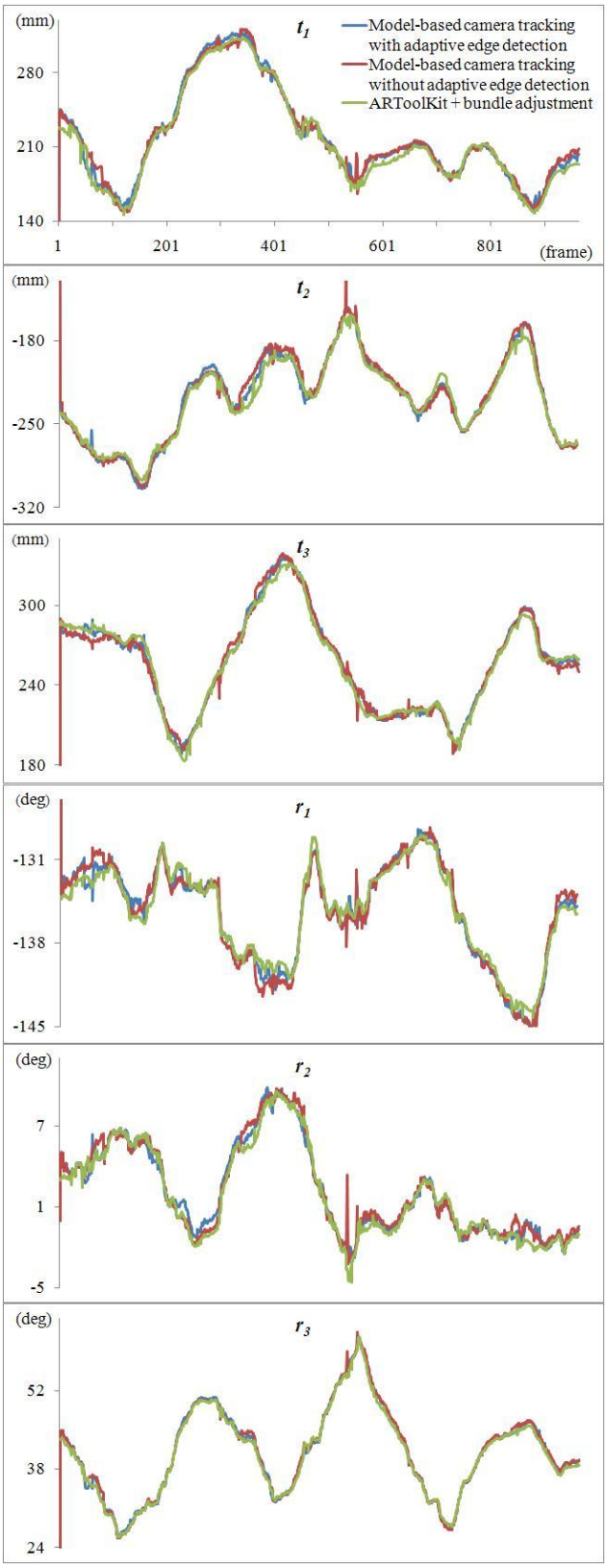


Fig. 4. Camera pose results of a video sequence.