IEICE TRANS. INF. & SYST., VOL.E94–D, NO.4 APRIL 2011

LETTER

# Automatic Adjustment of the Distance Ratio Threshold in Nearest Neighbor Distance Ratio Matching for Robust Camera Tracking

**Hanhoon PARK**[†a)], **Hideki MITSUMINE**[†b)], *Nonmembers*, *and* **Mahito FUJII**[†c)], *Member*

**SUMMARY** In nearest neighbor distance ratio (NNDR) matching the fixed distance ratio threshold sometimes results in an insufficient number of inliers or a huge number of outliers, which is not good for robust tracking. In this letter, we propose adjusting the distance ratio threshold based on maximizing the number of inliers while maintaining the ratio of the number of outliers to that of inliers. By applying the proposed method to a model-based camera tracking system, its effectiveness is verified.

***key words:*** *distance ratio threshold adjustment, NNDR matching, robust camera tracking*

## 1. Introduction

A variety of local invariant feature detectors, descriptors, and their combinations have been proposed [1]–[5]. Along with them a number of criteria for feature matching have been devised and evaluated. In previous research [4], three common criteria, fixed threshold, nearest neighbor, and nearest neighbor distance ratio (NNDR), were compared. The results showed that NNDR outperformed the others, so NNDR matching has been mostly used. However, in NNDR matching, the number of inliers (= correct matches) and outliers (= incorrect matches) varies greatly depending on the distance ratio threshold [3]. In particular, distance ratio thresholds that are too large (close to 1) cause loose matching (including too many incorrect matches) and distance ratio thresholds that are too small (smaller than 0.3) cause tight matching (including few correct matches). Therefore, based on a heuristic, Lowe [3] set the distance ratio threshold to 0.8 for his object recognition task. Likewise, most researchers have set the distance ratio threshold to a fixed value, which was an effective method for static environments. However, in dynamic environments where light and viewpoint can vary greatly, which is common in camera tracking, the fixed distance ratio threshold sometimes suffers from a lack of inliers or an excess of outliers and thus cannot always ensure robust camera tracking. This problem becomes more serious in scenes that include a small number of features.

For robust camera tracking, the distance ratio threshold must therefore be adjusted adaptively to environmental changes. In this letter, we propose an adjustment method which counts the number of inliers and outliers and maximizes the number of inliers (a small number of inliers causes poor results in camera tracking [6]) while maintaining the ratio of the number of outliers to that of inliers. The adjustment method can provide optimized distance ratio thresholds in static and dynamic environments without a prior or heuristic knowledge.

Manuscript received July 1, 2010.
Manuscript revised November 25, 2010.

[†]The authors are with NHK (Japan Broadcasting Corporation) Science & Technology Research Laboratories, Tokyo, 157–8510 Japan.
a) E-mail: hanhoon.park@strlstaff.strl.nhk.or.jp
b) E-mail: mitsumine.h-gk@nhk.or.jp
c) E-mail: fujii.m-ii@nhk.or.jp

DOI: 10.1587/transinf.E94.D.938

## 2. NNDR Matching

For making this letter more self-contained, we briefly explain NNDR matching [3] in this section.

The candidate neighbors of a feature are found by computing the Euclidean or Mahalanobis distances between the descriptor of the feature and those of other target features and thresholding the distances with a threshold. The candidate neighbors are ranked by the magnitude of their distances. Then, letting *feature B* be the nearest neighbor of *feature A* and *feature C* be the second nearest neighbor of *feature A*, the NNDR is defined as

$$nndr = \frac{d_1}{d_2} = \frac{D_A - D_B}{D_A - D_C}, \tag{1}$$

where $d_1$ and $d_2$ are the nearest and second nearest neighbor distances, and $D_A$, $D_B$, and $D_C$ are the descriptors of features. Finally, if *nndr* is smaller than a threshold ($r_{th}$), called *distance ratio threshold* in this letter, *feature B* is determined to match *feature A*.

## 3. Adjustment of the Distance Ratio Threshold for Robust Camera Tracking

The distance ratio threshold is adjusted as follows.

$$r_{th} = \begin{cases} r_{th} + \Delta r_{th}, & \text{if } N_{inlier} < \tau, \\ r_{th} - \Delta r_{th}, & \text{else if } e(\mathbf{E}) > \epsilon, \\ r_{th} + \Delta r_{th}, & \text{else if } \frac{N_{outlier}}{N_{inlier}} \le n_{th}, \\ r_{th} - \Delta r_{th}, & \text{otherwise,} \end{cases} \tag{2}$$

where $\Delta r_{th}$ is an increment of $r_{th}$, $N_{inlier}$ and $N_{outlier}$ are the number of inliers and outliers, $e(\mathbf{E})$ is the mean reprojection error of inliers caused by the estimated camera pose $\mathbf{E}$, and $\tau$, $n_{th}$, and $\epsilon$ are the predefined thresholds. In Eq. 2 $r_{th}$ increases if the ratio of the number of outliers to that of the inliers is smaller than $n_{th}$ and thus the number of inliers increases. That is, the number of inliers is maximized given

$n_{th}$, which is helpful for robust camera tracking [6]. In contrast, $r_{th}$ decreases if the ratio of the number of outliers to that of inliers is larger than $n_{th}$ and thus the number of outliers decreases. That is, the ratio of the number of outliers to that of inliers is maintained smaller than $n_{th}$. In Eq. 2 the conditions, $N_{inlier} < \tau$ and $e(\mathbf{E}) > \epsilon$, are required for preventing camera tracking from being unstable (diverging).

$\Delta r_{th}$ can be a small constant or adjusted. Both are evaluated in this letter with the adjustment done as follows. If $r_{th}$ is successively incremented or successively decremented in Eq. 2, $\Delta r_{th}$ is added by 0.01. Otherwise, $\Delta r_{th}$ is subtracted by 0.01. Finally, $\Delta r_{th}$ is clipped to 0 and 0.1.

The inliers and outliers are simply separated as follows. First, features extracted in a frame are matched with those extracted in the next frame using the NNDR matching in Sect. 2. Next, the Euclidean distances between the features and their matches are computed. Then, the median distance is computed. Finally, a feature with a distance larger than the doubled median distance is considered an outlier. Otherwise, the feature is considered to be an inlier.

## 4. Experimental Results and Discussion

A model-based camera tracking system[†] [6], which uses SURF [2] for detecting and matching robust features in a high speed, was used for evaluating the performance of the proposed method. The camera pose (3 translation and 3 rotation parameters[††]) was estimated from the 640 × 480 images that a virtual scene (600(w) × 600(h) × 300(d)) in Fig. 1 was captured by freely moving a virtual camera (vertical view angle = 64.62 degrees and aspect ratio = 4 : 3). Its ground truth is shown in Fig. 2. The total number of reference features, which were found from the scene using SURF by fixing the distance ratio threshold to 0.65 in advance in an offline step [6], was 480.

The performance of the proposed method depends on $\Delta r_{th}$ and $n_{th}$. Therefore, we first analyzed the variation of the distance ratio threshold, the number of inliers, and the camera pose due to different constant $\Delta r_{th}$ (ranging from 0.01 to 0.1) and $n_{th}$ (ranging from 0.1 to 0.5[†††]). The initial distance ratio threshold could be given randomly but was set to 0.65 for fair evaluation, and $\tau$ and $\epsilon$ were set to 10 and 5, respectively. As shown in Table 1, the proposed method returned the optimized distance ratio threshold and the maximized number of inliers for each $n_{th}$ regardless of the values of $\Delta r_{th}$ and $n_{th}$. However, when $\Delta r_{th}$ was set to large values, it was difficult to maintain the ratio between the number of inliers and that of outliers within the given threshold and the camera pose error was larger. When $n_{th}$ was set to a value that was too small (making $r_{th}$ small and causing tight matching) or too large (making $r_{th}$ large and causing loose matching), the camera pose error was larger. Therefore, care should be taken in determining $n_{th}$.

Then, we analyzed the effect of the adjustment of $\Delta r_{th}$. As explained in Sect. 3, $\Delta r_{th}$ was adjusted between 0 and 0.1. As shown in Table 2, $\Delta r_{th}$ and $r_{th}$ were optimized consistently regardless of the initial value of $\Delta r_{th}$. Due to the optimized $\Delta r_{th}$ and $r_{th}$, the proposed method could maximize the number of inliers and minimize the camera pose error

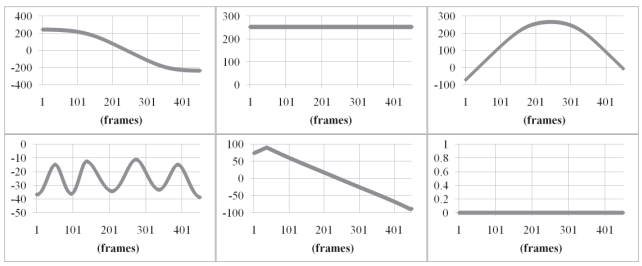


**Fig. 2** Ground truth camera pose. Upper row: translations, lower row: rotations.

**Table 1** The mean of the distance ratio threshold, the number of inliers, the ratio between the number of inliers and that of outliers, and the camera pose error resulted from fixed $\Delta r_{th}$ and $n_{th}$.

| $\Delta r_{th}$ | $n_{th}$ | $r_{th}$ | $N_{inlier}$ | $\frac{N_{outlier}}{N_{inlier}}$ | pose error (trans. & rot.) |
|---|---|---|---|---|---|
| 0.01 | 0.1 | 0.76 | 72.15 | 0.10 | 1.76 & 0.20 |
| | 0.2 | 0.82 | 83.61 | 0.21 | 1.56 & 0.18 |
| | 0.3 | 0.85 | 89.19 | 0.30 | 1.46 & 0.17 |
| | 0.4 | 0.87 | 92.66 | 0.40 | 1.52 & 0.18 |
| | 0.5 | 0.88 | 95.31 | 0.50 | 1.55 & 0.19 |
| 0.05 | 0.1 | 0.76 | 71.59 | 0.12 | 1.80 & 0.20 |
| | 0.2 | 0.82 | 83.18 | 0.23 | 1.57 & 0.18 |
| | 0.3 | 0.85 | 89.52 | 0.35 | 1.48 & 0.18 |
| | 0.4 | 0.87 | 93.46 | 0.44 | 1.62 & 0.19 |
| | 0.5 | 0.88 | 97.91 | 0.50 | 2.08 & 0.25 |
| 0.10 | 0.1 | 0.74 | 67.61 | 0.16 | 1.87 & 0.22 |
| | 0.2 | 0.83 | 85.08 | 0.34 | 1.55 & 0.19 |
| | 0.3 | 0.84 | 87.47 | 0.37 | 1.66 & 0.20 |
| | 0.4 | 0.85 | 94.84 | 0.39 | 1.62 & 0.21 |
| | 0.5 | 0.87 | 124.10 | 0.41 | 1.67 & 0.24 |

**Fig. 1** Wired 3D model and camera images of a virtual scene used in our experiments.

[†]Although it was devised to use two types of features, i.e. edges and points, cooperatively, there was no problem in working with only either one. In this letter, only point features were used.

[††]The translation parameters have no unit and the unit of rotation parameters is *degrees*.

[†††]When $n_{th}$ was larger than 0.5, the camera tracking system became severely unstable. This will be because the least-squared-based camera tracking system [6] is sensitive to the number of outliers.

**Table 2** The mean of the distance ratio threshold, the number of inliers, the ratio between the number of inliers and that of outliers, and the camera pose error resulted from *varying* $\Delta r_{th}$ and fixed $n_{th}$.

| initial $\Delta r_{th}$ | $n_{th}$ | $r_{th}$ | $\Delta r_{th}$ | $N_{inlier}$ | $\frac{N_{outlier}}{N_{inlier}}$ | pose error (trans. & rot.) |
|---|---|---|---|---|---|---|
| 0.01 | 0.1 | 0.76 | 0.016 | 72.61 | 0.11 | 1.70 & 0.20 |
| | 0.2 | 0.82 | 0.015 | 83.90 | 0.21 | 1.62 & 0.18 |
| | 0.3 | 0.85 | 0.015 | 89.34 | 0.31 | 1.61 & 0.19 |
| | 0.4 | 0.87 | 0.011 | 92.92 | 0.41 | 1.54 & 0.18 |
| | 0.5 | 0.89 | 0.012 | 96.46 | 0.53 | 1.46 & 0.18 |
| 0.05 | 0.1 | 0.77 | 0.016 | 73.18 | 0.11 | 1.68 & 0.19 |
| | 0.2 | 0.82 | 0.015 | 83.81 | 0.21 | 1.62 & 0.18 |
| | 0.3 | 0.85 | 0.015 | 89.38 | 0.32 | 1.60 & 0.19 |
| | 0.4 | 0.87 | 0.011 | 92.92 | 0.42 | 1.53 & 0.18 |
| | 0.5 | 0.89 | 0.012 | 96.37 | 0.53 | 1.46 & 0.18 |
| 0.10 | 0.1 | 0.76 | 0.017 | 72.72 | 0.11 | 1.70 & 0.19 |
| | 0.2 | 0.83 | 0.016 | 84.17 | 0.22 | 1.61 & 0.18 |
| | 0.3 | 0.85 | 0.012 | 88.85 | 0.30 | 1.60 & 0.18 |
| | 0.4 | 0.87 | 0.011 | 92.89 | 0.41 | 1.51 & 0.18 |
| | 0.5 | 0.89 | 0.013 | 95.85 | 0.51 | 1.46 & 0.18 |

**Table 3** The mean of the number of inliers, the ratio between the number of inliers and that of outliers, and the camera pose error when fixing the distance ratio threshold.

| $r_{th}$ | $N_{inlier}$ | $\frac{N_{outlier}}{N_{inlier}}$ | pose error (trans. & rot.) |
|---|---|---|---|
| 0.55 (fixed) | 29.08 | 0.01 | 2.29 & 0.26 |
| 0.65 (fixed) | 47.72 | 0.03 | 1.91 & 0.21 |
| 0.75 (fixed) | 68.72 | 0.08 | 1.66 & 0.18 |
| 0.85 (fixed) | 86.35 | 0.25 | 1.71 & 0.19 |

When $r_{th}$ was fixed larger than 0.85 (too many outliers) or smaller than 0.5 (too few inliers), the camera tracking system was unstable.

with little care in determining $n_{th}$. One noticeable thing is that it seemed better to set $n_{th}$ to as large a value as possible. However, if a setting of $n_{th}$ is too large it would be risky. In practice, the camera pose error increased rapidly with $n_{th}$ larger than 0.5 and the camera tracking system was unstable with $n_{th}$ larger than 0.6 (the data was omitted here).

In order to show the adverseness of using a fixed distance ratio threshold, $r_{th}$ was fixed to 0.55, 0.65, 0.75, and 0.85 and their results were compared with those by the proposed method. As shown in Table 3, the camera pose error of fixing $r_{th}$ was larger than that of the proposed method even when fixing to 0.85 (close to the optimized ones resulted by the proposed method). Furthermore, the camera tracking system was unstable (often failed to estimate the camera pose due to insufficient inliers or excessive outliers) when $r_{th}$ was fixed larger than 0.85 or smaller than 0.5. However, the proposed method worked stably even when $r_{th}$ was initialized larger than 0.85 or smaller than 0.5. Figure 3 shows the variation of $r_{th}$ that resulted from the proposed method when $r_{th}$ was initialized to different values. Regardless of its initial value, it was optimized consistently and the resulting mean camera pose error was not different (although the camera pose data was omitted here). The speed that $r_{th}$ converges to the optimal value depends on the initial

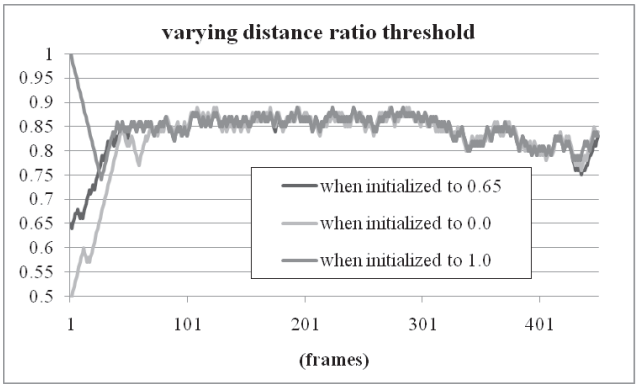


**Fig. 3** Variation of the distance ratio threshold when it was initialized to different values ($\Delta r_{th}$ and $n_{th}$ were fixed to 0.01 and 0.3). The mean was the same (0.85) for different initial values. These results did not greatly change when $\Delta r_{th}$ was adjusted.

value of $r_{th}$, $\Delta r_{th}$, $n_{th}$, scene conditions, etc. In our experiments, the convergence took about 30 to 60 frames.

## 5. Conclusion

In this letter, we proposed to adjust the distance ratio threshold in NNDR matching to be adaptive to dynamic environments for robust camera tracking. Through the experimental results with synthetic camera images, we verified that the proposed method computed the optimized distance ratio for maximizing the number of inliers while maintaining the ratio of the number of outliers to that of inliers every frame. This approach permitted more accurate and stable camera tracking than a fixing the distance ratio threshold.

In our experiments where $r_{th}$ was neither largely nor abruptly changed, the effectiveness of the proposed method against fixing $r_{th}$ could not be as fully evaluated as expected. Currently, we are trying to analyze the performance of the proposed method in more detail by applying it to synthetic or real scenes in different various environments.

### References

[1] M. Agrawal, K. Konolige, and M.R. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," Proc. Euro. Conf. on Computer Vision, pp.102–115, 2008.
[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Proc. Euro. Conf. on Computer Vision, pp.404–417, 2006.
[3] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.92–110, 2004.
[4] K. Mikolajczyk and C. Schmid, "A performance evaluation of lcoal descriptors," IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.10, pp.1615–1630, 2005.
[5] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," Int. J. Comput. Vis., vol.65, no.1/2, pp.43–72, 2005.
[6] H. Park, H. Mitsumine, M. Fujii, and J.-I. Park, "Analytic fusion of visual cues in model-based camera tracking," Proc. Int. Conf. on Virtual-Reality Continuum and Its Applications in Industry, 2009.