ISSN 1064-2269, Journal of Communications Technology and Electronics, 2014, Vol. 59, No. 12, pp. 1446–1450. © Pleiades Publishing, Inc., 2014.
Original Russian Text © D. Miramontes-Jaramillo, V.I. Kober, V.H. Díaz-Ramírez, V.N. Karnaukhov, 2014, published in Informatsionnye Protsessy, 2014, Vol. 14, No. 1, pp. 56–63.

A Novel Image Matching Algorithm Based on Sliding Histograms of Oriented Gradients

D. Miramontes-Jaramillo[a], V. I. Kober[a, c], V. H. Díaz-Ramírez[b], and V. N. Karnaukhov[c]

[a] *Department of Computer Science, CICESE, Carretera Ensenada-Tijuana 3918,
Zona Playitas, Ensenada B.C. 22860, Mexico*

[b] *Instituto Politécnico Nacional – CITEDI, Ave. del Parque 1310, Mesa de Otay, Tijuana B.C.22510, Mexico*

[c] *Institute for Information Transmission Problems, Russian Academy of Sciences,
Bol'shoi Karetnyi per. 19, str. 1, Moscow, 127994 Russia*

e-mail: vnk@iitp.ru

Received February 17, 2014

Abstract—A novel algorithm for image matching based on recursive calculation of histograms of oriented gradients over several circular sliding windows and pyramidal image decomposition is presented. The algorithm gives good results for geometrically distorted and scaled scene images. The results of computer simulation obtained with the proposed algorithm are compared to those of available algorithms in terms of matching accuracy and processing time.

Keywords: image matching, fast algorithm, histogram of oriented gradients, circular window

DOI: 10.1134/S1064226914120146

INTRODUCTION

Recently, numerous algorithms for image matching employing various features and keypoints have been proposed. Among them, much attention is attracted by two algorithms: the scale invariant feature transform (SIFT) [1] and the speeded-up robust features (SURF) [2]. These algorithms and their variants [3–5] may be considered as benchmarks for the comparison with new proposed methods for image matching. Although the feature-based methods are very popular, template-based algorithms offer a very attractive alternative for applications operating in real time. The template-based methods have a very good formal justification. In addition, such methods can be implemented with a high processing speed in hybrid opticodigital systems [6], in high-performance hardware such as graphic processing units (GPUs) [7] and field-programmable gate arrays (FPGAs) [8]. Another approach to this problem may be a combination of feature- and template-based matching algorithms. An example of such algorithms is the scale invariant compressed histogram transform (SICHT) [9] in which histograms of oriented gradients (HoG) [10] calculated in a sliding window are features.

The main advantages of the proposed algorithm are the following:

(i) The algorithm matches histograms of oriented gradients in several circular windows sliding over the input image. The final decision is made using the results of combined comparison for all windows.

(ii) For speed-up of the processing, the algorithm utilizes decimation. Local threshold filtering of histograms, which is followed by decimation of the resulting image, is used instead of the classical pyramidal approach of low-pass filtering.

(iii) A trade-off between the complexity of the algorithm and the accuracy of matching, which is controlled by specified false alarm and missing error probabilities, on the one hand, and the processing time, on the other hand, is possible.

The performance of the proposed algorithm in a image test database was compared with the performance of the SIRF, SURF, and newer, oriented FAST and rotated BRIEF, algorithms [3] in terms of the accuracy of matching and the processing time.

1. FAST IMAGE MATCHING ALGORITHMS

Let us define a set of circular windows $\{W_i, i = 1, ..., M\}$ on a reference image as a set of the following closed disks:

$$W_i = \{(x, y) \in \mathbb{R}^2 : (x - x_i)^2 + (y - y_i)^2 \le r_i\}, \quad (1)$$

where (x_i, y_i) are the coordinates of the window center and r_i the radius of the ith disk. The disks form a geometrical structure with relative distances and angles between the window centers. Histograms of oriented gradients are calculated in the circular regions and used for the matching. It should be noted that, in any position of the structure, each disk contains a rotation-invariat region. Therefore, the histograms of oriented gradients calculated in the circular regions are rotation-invariat too. The use of only one window gives poor results of matching. Therefore, it is recommended to choose such a minimum number of disks as to fill the maximum possible interior of the reference object. Histograms of oriented gradients are good features for matching [10], because they have good discrimination ability and are robust to small deformations such as rotation and scaling.

For each position of the ith circular window, we calculate in the scene image the gradients inside the window by using the Sobel operator [11]. Then, using the magnitude values of gradients $\{Mag_i(x, y) : (x, y) \in W_i\}$ and the direction angles quantized into Q levels $\{\varphi_i(x, y) : (x, y) \in W_i\}$, we calculate the histograms of oriented gradients as follows:

$$HoG_i(\alpha) = \begin{cases} \sum\limits_{(x, y) \in W_i} \delta(\alpha - \varphi_i(x, y)), & \text{if } Mag_i(x, y) \geq Med, \\ 0, & \text{else}, \end{cases} \quad (2)$$

where $\alpha = \{0, ..., Q - 1\}$ is a discrete value (bin) of the histogram, Med is the median of the magnitude values of the gradients of circular windows, and $\delta(z) = \begin{cases} 1, & \text{if } z = 0, \\ 0, & \text{else} \end{cases}$ is the Kronecker delta. It should be noted that the calculations in Eq. (2) require approximately $[\pi r_i^2]$ addition operations, where [.] is the integer part operator. In order to reduce the computational complexity, the histogram can be calculated recursively at any position of a sliding window. This approach leads to a computational complexity on the order of $[2\pi r_i]$ addition operations. Figure 1 shows recursive update of the histogram along columns for transition from the kth to the $(k + 1)$th iteration. To provide the rotation invariance, we use a normalized correlation operator for comparison of the histograms of the reference image and the scene. Let us calculate the centered and normalized histogram of oriented gradients of the reference image in the following form:

$$\overline{HoG_i^R}(\alpha) = \frac{HoG_i^R(\alpha) - Mean^R}{\sqrt{Var^R}}, \quad (3)$$

where $Mean^R$ and Var^R are the sample mean and the variance of the histogram, respectively.

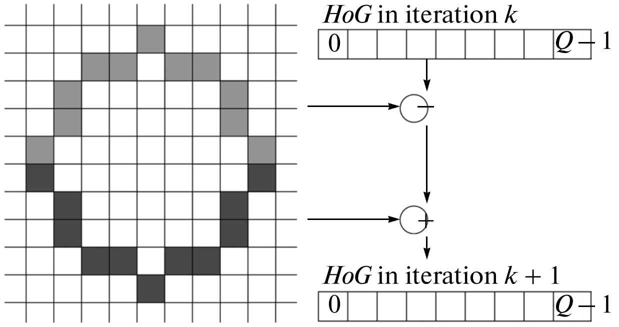

Fig. 1. Recursive update of the histogram along columns in transition from the kth to the $(k + 1)$st iteration.

The correlation function for each ith window in the kth position can be calculated with the use of the inverse Fourier transform [11] as

$$C_i^k(\alpha) = IFT\left[\frac{HS_i^k(\omega)HR_i^*(\omega)}{Q\sum\limits_{q=0}^{Q-1}(HoG_i^k(q))^2 - (HS_i^k(0))^2}\right], \quad (4)$$

where $HS_i^k(\omega)$ is the Fourier transform of the histogram of oriented gradients inside the ith window of the input scene, $HR_i(\omega)$ is the Fourier transform of $\overline{HoG_i^R}(\alpha)$, and asterisk denotes the complex conjugate. The correlation peak is a measure of similarity between two histograms; it can be obtained as

$$P_i^k = \max_{\alpha}\{C_i^k(\alpha)\}. \quad (5)$$

The heights of these correlation peaks are in the range [–1, 1]. It should be emphasized that normalized correlation peaks (5) possess two important properties. The first property is the rotation invariance, because a cyclic shift of a histogram corresponds to a cyclic shift of the peak values. The second property is that the normalization in Eqs. (3) and (4) makes it possible to take into account a small scale difference between the reference image and the scene image. Calculation of centered and normalized histograms for all circular windows from the reference image as well as the Fourier transforms can be performed as preprocessing of the reference data.

In order to accelerate the matching process, we do not use the classical pyramidal approach, which is based on low-pass filtering followed by the sampling of the resulting image for obtaining low-resolution images of smaller size [12]. Simple decimation of the input image along rows and columns [13] seems to be more appropriate for speed-up of matching of histograms of oriented gradients.

Matching is performed between the first reduced scene image and the reduced reference image. It should be noted that decomposition of the reference image into a set of reduced images can be performed as preprocessing of the reference data. Another important parameter that can accelerate the process of matching is number Q of discrete levels of the histograms.

The proposed matching procedure can be briefly represented in the following form.

(i) Independent fast matching is performed for circular windows with a reduced number, e.g., Q_R, of discrete levels of histograms. The number of points to be considered in the next matching iteration is reduced keeping a low value of the probability of miss errors.

(ii) Only the remaining selected points are considered for the second matching iteration with standard number Q of discrete levels of histograms for each window.

(iii) Further lowering of the probability of miss errors as well as the probability of false alarms is reached by means of the combined analysis of features in all considered circular regions of the reference image.

A detailed description of the algorithm steps and their explanations can be summarized as follows:

1. Define a set of circular windows $\{W_i, i = 1, ..., M\}$ of constant radius r on the reference image using Eq. (1). Calculate the distance between the window centers

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad i = 1, ..., M;$$
$$j = i + 1, ..., M$$

and the angles between each three adjacent centers of the whole set of windows, i.e.,

$$\gamma_i = \cos^{-1}\left[\frac{D^2_{i,j+1} + D^2_{i,j+2} - D^2_{i,j+3}}{2D_{i,j+1}D_{i,j+2}}\right],$$
$$i = 1, ..., M - 2.$$

Specify a threshold value Th_{QR} that provides the specified probability value of miss errors at a reduced number Q_R of discrete levels of histograms. In a similar manner, specify a threshold value Th_Q that provides a trade-off between the miss error and false alarm probabilities for Q discrete levels of histograms.

2. Perform decimation of the reference image with step L along columns and rows and obtain L^2 reduced images. For each circular window of the reference image, calculate the centered and normalized histograms of oriented gradients $\{\overline{HoG_i^R}(\alpha),\ i = 1, ..., M\}$ and calculate the Fourier transforms $\{HR_i(\omega),\ i = 1, ..., M\}$ using Eqs. (2) and (3).

3. Perform the compression of $\{\overline{HoG_i^R}(\alpha),\ i = 1, ..., M\}$ with Q discrete levels of histograms and calculate Fourier transforms $\{CHR_i(\omega), i = 1, ..., M\}$.

4. Perform decimation of the scene image with step L along rows and columns and obtain L^2 reduced images. Start the matching process using all circular windows of the scene image for $k = 1$. Since all windows have the same size, only one sliding window is used. Index i of the histogram of the scene image may be omitted. Calculate the histogram of oriented gradients of the first reduced scene image $HoG^1(\alpha)$ and its Fourier transform $HS^1(\omega)$. Perform compression of $HoG^1(\alpha)$ with Q discrete levels of histograms in order to obtain $CHoG^1(\alpha)$ with Q_R discrete levels of histograms and calculate Fourier transform $CHS^1(\omega)$.

5. Calculate correlation peaks between $CHoG^1(\alpha)$ and $\{\overline{CHoG_i^R}(\alpha),\ i = 1, ..., M\}$ using Eqs. (4) and (5).

6. If any of the correlation peaks calculated in step 5 is higher than Th_{QR}, then calculate correlation peaks $\{P_i^1,\ i = 1, ..., M\}$ between $HoG^1(\alpha)$ and $\{\overline{HoG_i^R}(\alpha),\ i = 1, ..., M\}$, using Eqs. (4) and (5). If any of the correlation peaks $\{P_i^1,\ i = 1, ..., M\}$ is higher than Th_Q, then store this point for further analysis.

7. Increase k, update the histograms of oriented gradients of the scene image along rows and columns, and repeat steps 5 and 6.

8. Make final decision on the presence of the reference image at the kth position. During the analysis, we consider possible deformation of the initial structure caused by the scaling and out-of-plane rotation. The analysis begins at the position of the highest peak calculated for the first window. Then a circular region with the outer and inner radii $D_{1,2} \pm \epsilon_D$, respectively, is determined for the search for the peak inside the second window. The search for successive correlation peaks can be performed rapidly if angles between the window centers are taken into account. For example, for the third window, we calculate the intersection between the annular region with the outer and inner radii $D_{2,3} \pm \epsilon_D$ and an open circular sector with the origin at the position of the peak of the second window and bounded inner and outer radii with slopes $\gamma_1 \pm \epsilon_D$.

2. RESULTS OF EXPERIMENTS

In this section, we present the results of experiments using the ALOI image data base [14]. We used ten scene images with the size of 1280 × 1024 pixels and ten reference images with the size of 144 × 144 pixels with various reference objects. Each reference object was randomly placed in the scenes at 100 different positions. The performance of the proposed algorithm was compared with the that of the SIFT and SURF algorithms as well as the newer ORB algorithm [3]. Parameters of the first two algorithms were taken from [1, 2]. Parameters of the ORB algorithm were chosen empirically to obtain the results comparable in terms of the quality and the processing time. The algorithms were tested in various conditions such as in-plane/out-of-plane rotations and small scaling. The performance was estimated in terms of the number of correct matches and the processing time. The proposed algorithm, referred to as the circular window-histograms of oriented gradients (CW-HOG) algorithm, uses two circular windows of radius r (depending on the object size). In order to obtain better matching, we use $Q_R = 16$ and $Q = 64$ discrete levels of histograms instead of 9 levels proposed in [10]. The parameters of the algorithm in the experiments are as follows:

(i) $M = 2$, $Q = 64$, $Q_R = 16$, $\epsilon_D = 0.2r$, and $L = 4$.

(ii) The decision thresholds are specified as follows: for $Q_R = 16$ (the first run), the threshold $Th_{QR} = 0.8$ provides the probability of miss errors $P_{ME} = 0.05$ on the test data base and selects approximately 46 points for their further analysis in the second run with $Q = 64$;

(iii) For $Q = 64$ (the second run), the threshold $Th_Q = 0.7$ provides the probability of miss errors $P_{ME} = 0.05$ and the probability of false alarms $P_{FA} = 0.21$;

(iv) The final decision reduces the probability of false alarms to $P_{FA} = 0.05$.

The accuracy of image matching with the use of the tested algorithms as a function of the image rotation angle for in-plane rotations is shown in Fig. 2. It is readily seen that the CW-HOG algorithm yields the matching performance comparable with that of the SIFT algorithm in terms of the in-plane rotation invariance. The accuracy of image matching with the use of the chosen algorithms as a function of the image rotation angle for out-of-plane rotations is shown in Fig. 3. It should be noted that the CW-HOG algorithm gives more stable matching performance than other algorithms for out-of-plane rotations. Figure 4 illustrates the robustness of the proposed algorithm to scaling of the scene in the range [0.8, 1.2]. One can observe that the accuracy of image matching by the CW-HOG algorithm is rather good in this case too.

Comparison of the chosen algorithms in terms of the processing time has shown that the processing time of the CW-HOG and ORB algorithms is about 1.7 s and the processing time of the SIFT and SURF algorithms is 9.82 s and 0.95 s, respectively. The algorithms were implemented on a standard personal computer with Intel Core i7 processor (3.2 GHz, 8 GB RAM). Implementation of the SIFT, SURF, and ORB algorithms was taken from the OpenCV library with the Intel TBB multithreading library. The proposed CW-HOG algorithm was also implemented using the OpenCV library with the OpenMP multithreading library.

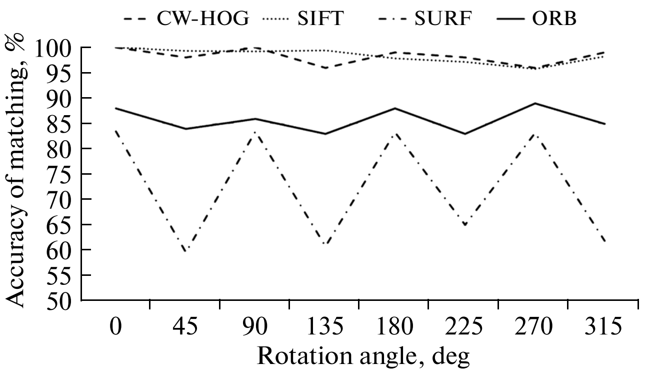

Fig. 2. Accuracy of image matching by the tested algorithms for in-plane rotations.

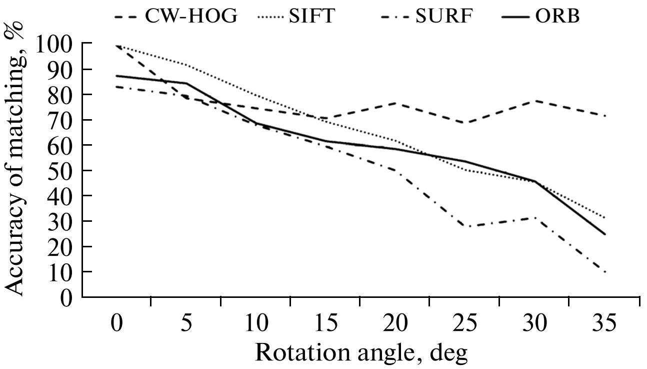

Fig. 3. Accuracy of image matching by the tested algorithms for out-of-plane rotations.

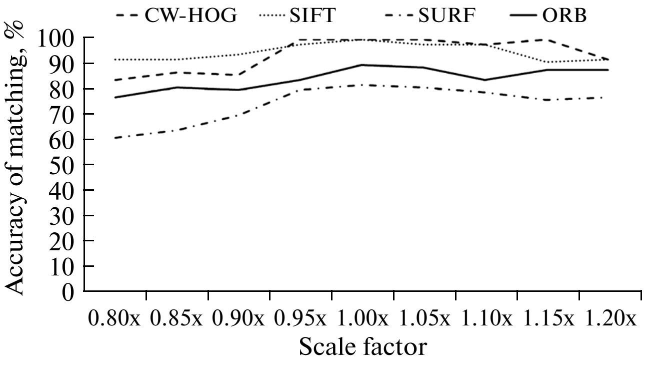

Fig. 4. Accuracy of image matching by the tested algorithms for small scaling of the scene image.

Object tracking algorithms usually employ prediction methods [15] for evaluation of the object position and trajectory in a sequence of frames and for reduction of the search region. The characteristics of the proposed algorithm in terms of the processing time are the following:

for fragments of 640 × 512 pixels, the processing time is 0.47 s;

for fragments of 426 × 341 pixels, the processing time is 0.22 s;

for fragments of 320 × 256 pixels, the processing time is 0.14 s.

Obviously, the processing time rapidly decreases with decrease in the fragment size. Therefore, a real-time tracking based on the proposed algorithm becomes possible.

CONCLUSIONS

In this paper, we have proposed a fast algorithm for image matching based on recursive calculation of histograms of oriented gradients over several circular sliding windows. Multicore processors with inherent parallel architecture can be used for implementation of the proposed algorithm for high-speed matching of large images. The results of experiments conducted in this work show that the proposed algorithm outruns known algorithms for in-plane rotations, yields a similar performance to that of the SIFT for out-of-plane rotations, and has a processing speed close to the SURF algorithm. The proposed algorithm shows promises for real-time applications such as object tracking, when rotation invariance matching with a slight scaling is needed.

ACKNOWLEDGMENTS

This work was supported by the CONACYT (grant 157211) and the Ministry of Education and Science (grant 2.1766.2014K).

REFERENCES

1. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th Int. Conf. on Computer Vision, Crete, 1999* (IEEE, New York, 1999), Vol. 2, pp. 1150–1157.
2. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," Comput. Vis. Image Underst. **110**, 346–359 (2008).
3. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. on Computer Vision, Barcelona, 2011* (IEEE, New York, 2011), pp. 2564–2571.
4. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proc. 11th Eur. Conf. on Computer Vision. (ECCV'10), Hersonissos Heraklion Crete, Greece, 2010* (Springer-Verlag, 2010), pp. 778–792.
5. R. Ortiz, "FREAK: Fast Retina Keypoint," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'12, Providence, RI, June, 2012* (IEEE, New York, 2012), pp. 510–517.
6. V. H. Díaz-Ramírez and V. Kober, "Adaptive phase-input joint transform correlator," Appl. Opt. **46**, 6543–6551 (2007).
7. Y. Ouerhani, M. Jridi, and A. Alfalou, and C. Brosseau, "Optimized preprocessing input plane GPU implementation of an optical face recognition technique using a segmented phase only composite filter," Opt. Commun. **289**, 33–44 (2013).
8. K. L. Rice, T. M. Taha, A. M. Chowdhury, A. A. S. Awwal, and D. L. Woodard, "Design and acceleration of phase-only filter-based optical pattern recognition for fingerprint identification," Opt. Eng. **48** (11), 117–206 (2009).
9. B. A. Zalesky and P. V. Lukashevich, "Scale invariant algorithm to match regions on aero or satellite images," Proc. Pattern Recogn. Inf. Process. **11**, 25–30 (2011).
10. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Comput. Vis. Pattern Recogn. **1**, 886–893 (2005).
11. W. K. Pratt, *Digital Image Processing* (Wiley, New York, 2007).
12. T. Lindeberg, "Scale-space theory: A basic tool for analysis structures at different scales," J. Appl. Statis. **21**, 225–270 (1994).
13. B. Liu and A. Zaccarin, "New fast algorithms for the estimation of block motion vectors," IEEE Trans. Circuits Syst. Video Technol. **3**, 148–157 (1993).
14. J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, The Amsterdam library of object images, International Journal of Computer Vision, 2005, vol. 61, no. 1, pp. 103–112, http://staff.science.uva.nl/ aloi/
15. Li X. Rong and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models," IEEE Trans. Aerosp. Electron. Syst. **39**, 1333–1364 (2003).

Translated by E. Chernokozhin