Novel averaging window filter for SIFT in infrared face recognition

Junfeng Bai (柏俊峰)[1], Yong Ma (马 泳)[2*], Jing Li (黎 静)[2],
Fan Fan (樊 凡)[2], and Hongyuan Wang (王宏远)[1]

[1]*Department of Electronic and Information Engineering,*
Huazhong University of Science and Technology, Wuhan 430074, China
[2]*Department of Electronic and Information Engineering, Wuhan National Laboratory for Opto-Electronics,*
Huazhong University of Science and Technology, Wuhan 430074, China
[*]*Corresponding author: mayong@hust.edu.cn*

Received December 21, 2010; accepted March 24, 2011; posted online June 16, 2011

The extraction of stable local features directly affects the performance of infrared face recognition algorithms. Recent studies on the application of scale invariant feature transform (SIFT) to infrared face recognition show that star-styled window filter (SWF) can filter out errors incorrectly introduced by SIFT. The current letter proposes an improved filter pattern called Y-styled window filter (YWF) to further eliminate the wrong matches. Compared with SWF, YWF patterns are sparser and do not maintain rotation invariance; thus, they are more suitable to infrared face recognition. Our experimental results demonstrate that a YWF-based averaging window outperforms an SWF-based one in reducing wrong matches, therefore improving the reliability of infrared face recognition systems.

OCIS codes: 100.2000, 100.3008, 100.2960.
doi: 10.3788/COL201109.081002.

Infrared human face recognition has become an area of growing interest in literature[1]. Most representative methods include elemental shape matching, eigenface, metrics matching, template matching, symmetry waveforms, and face codes[2−4] are introduced from the visible domain. Among these methods, symmetry waveforms and face codes utilize the anatomical structure by analyzing the infrared vascular pattern, while the others extract and match thermal contours[1]. Compared with recognition in visible-spectrum imagery, face recognition in the thermal infrared domain has received relatively little attention in literature.

For infrared face recognition, there are several successful candidate visual approaches based on invariant feature extraction[5]. Mikolajczyk *et al.* made the first effort in this area and achieved rotation invariance[6]. Lowe extended this approach and achieved scale invariance[7,8]. Many researchers have reported achievements in affine transformation invariance and rotation, including scale invariant feature transform (SIFT), independent component analysis, improved Harris corner detector, and fractal and genetic algorithms[9−14]. Among them, the methods based on scale–space feature extraction, i.e., SIFT[10] and improved Harris corner[14], are most applicable to infrared human face recognition. Between them, features extracted by SIFT are more dispersed in spatial distribution, more stable for occlusion, and are relatively large in quantity[15]. Therefore, the former is more suitable for infrared features, and SIFT is taken as our candidate method for investigation.

By introducing SIFT into the infrared domain, several problems in infrared human face recognition, such as wearing glasses and facial rotation, can be solved directly. However, SIFT has an intrinsic defect, that is, it generates mismatches for points with similar textures around them. Experiments by Tan *et al.* showed that most of these mismatches differ in mean brightness[16]. By applying a star-styled averaging window, mismatches can be removed effectively. Tan's study examined only one pattern out of many possible averaging window filters applicable in this scenario. In this letter, two other candidate filter patterns, namely, cross-styled window filter (CWF) and Y-styled window filter (YWF), are proposed, which are proved to be more effective in yielding better filtering results than star-styled window filter (SWF).

The elegant design of the four SIFT stages enables it to extract distinctive invariant features from an image better than other algorithms. However, the construction of the SIFT features is completely performed in the scale space and the features of the original image space are not used. Patches with similar local textures will therefore result in similar keypoint descriptors, leading to incorrect features when used for object recognition. The proof is given below.

Let A and B be similar patches in $I(x,\ y)$. Let a and b be points in the same relative physical location of A and B, respectively, i.e., $a \in A$, $b \in B$ and $a \approx kb$, where k is the gray level ratio. In addition, let a_0 and b_0 be local extrema in A and B, respectively. Our derivation follows the stages similar to those in SIFT.

In the first stage, the two patches are subjected to the difference of Gaussian (DoG) transformation. Since the DoG operator is linear, we have

$$\mathrm{DoG}(a) \approx \mathrm{DoG}(k \cdot b) = k \cdot \mathrm{DoG}(b). \tag{1}$$

Meanwhile, the relative physical locations of a_0 and b_0 are the same because $a \approx kb$.

In the second stage, the previous two criteria are checked on a_0 and b_0. Since $a_0 \approx kb_0$, it is impossible that one extrema is along an edge and the other is not. Assume that a_0 and b_0 are stable. Thus, both of them will pass the test of the second stage.

In the third stage, the magnitude assignment formula is linear as well. Since $\mathrm{DoG}(a) \approx k{\cdot}\mathrm{DoG}(b)$, we have

$$m[\mathrm{DoG}(a)] \approx m[k \cdot \mathrm{DoG}(b)] = k \cdot m[\mathrm{DoG}(b)]. \quad (2)$$

For the orientation assignment formula, the division in it eliminates the effect of the coefficient k. Thus, we have

$$\theta[\mathrm{DoG}(a)] \approx \theta[k \cdot \mathrm{DoG}(b)] = \theta[\mathrm{DoG}(b)]. \quad (3)$$

The normalization procedure in the fourth stage guarantees that the normalized version of the magnitude m' and the orientation θ' of a and b are approximately equal, i.e.,

$$m'[\mathrm{DoG}(a)] \approx m'[\mathrm{DoG}(b)], \quad (4)$$

and

$$\theta'[\mathrm{DoG}(a)] \approx \theta'[\mathrm{DoG}(b)]. \quad (5)$$

After that, the following weighting, projection, and summing procedure to obtain the SIFT descriptor $\mathrm{SIFT}[\mathrm{DoG}(x_i, y_i, \sigma_i\ )]$ are all linear. We therefore have

$$\mathrm{SIFT}[\mathrm{DoG}(a_0)] \approx \mathrm{SIFT}[\mathrm{DoG}(b_0)]. \quad (6)$$

The consequence of Eq. (6) is that a matching procedure may consider the match between two different extremas generated by SIFT as a better match than the true match between the keypoints with the same physical location. Figure 1 shows the SIFT wrong matches between two scenes, both of which are adapted from Yan Ke's work[17]. The local texture similarity can be observed clearly.

In our experiments, two characteristics of most mismatches were found. Firstly, the mismatches differ in the brightness of their local texture. Secondly, rotation along the axis perpendicular to the page is necessary to provide spatial correspondence for patches around mismatched keypoints. In other words, patches around mismatched keypoints cannot set up spatial correspondence without rotation. Based on the two observations, a straightforward solution to filtering mismatched keypoints is to use the gray level information in an averaging window with a pattern that is sparser and does not preserve rotation invariance.

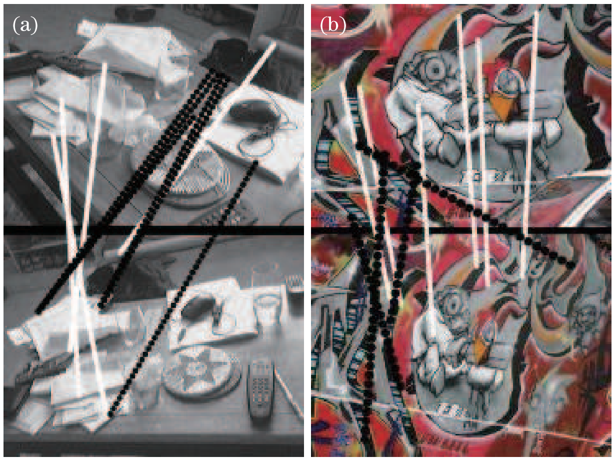

Fig. 1. Illustration of mismatched points resulting from local texture similarity. (a) A photo of a cluttered coffee table; (b) a mural painting. The solid lines represent the correct matches while the dotted lines represent the incorrect ones.

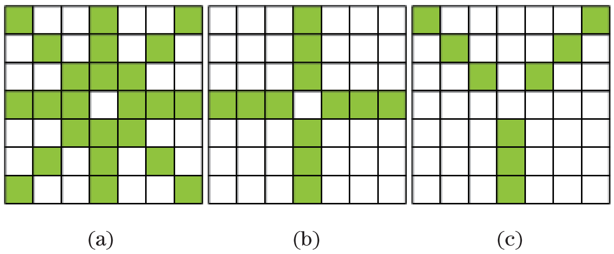

Fig. 2. General pattern of averaging window. The keypoint is the center pixel and the deep colored locations are pixels selected for the window. (a) The SWF proposed in Ref. [16]; (b) the CWF; (c) the YWF.

Two averaging window patterns (Fig. 2) are proposed to test this hypothesis. CWF is designed as a cross-styled pattern to keep the rotation invariance features of SIFT. Compared with SWF, CWF is sparser and should therefore outperform SWF. In the present letter, to further avoid the rotation invariance property, YWF is proposed as the candidate averaging window pattern and is expected to have the best performance.

Our approach keeps the original SIFT stages and adds additional stages to evaluate the tradeoffs between eliminating mismatched points and preserving correctly matched points. To ensure the accuracy of our results easy comparison to the results of the previous work, we use the original SIFT source code provided by Vedaldi[18] and the SWF algorithm presented in Ref. [16].

Our algorithm adds two additional stages to SIFT: averaging information extraction and averaging information thresholding. The former stage generates a proper value to represent the local texture around the keypoint. The latter stage eliminates mismatches by choosing an optimized threshold.

Averaging the brightness information of the local patch is a direct and effective method for eliminating wrong. The topology of point locations selected from the local texture is called filter window. The choice of shape and size affects the filtering performance. Figure 2 illustrates the general style of averaging window. Figure 2(a) is the pattern for SWF. Figures 1(b) and (c) give the general patterns for the proposed cross-style and Y-style averaging window, respectively. The point at the center of the patch is the keypoint; the pixels with deep color are locations selected for the averaging window. The average gray-scale value at the selected pixels is named averaging information (AI). In general, the averaging window size could be $(2N{+}1){\times}(2N{+}1)$, $N{=}2, 3, 4\ldots$.

Let $I(x,\ y)$ be the center of the averaging patch, i.e., the keypoint. The $\mathrm{AI}[I(x,\ y)]$ is evaluated as

$$\mathrm{AI}[I(x,y)] = \mathrm{mean}\left[\sum_{i,j} I(i,j)\right], \quad (7)$$

where $I(i,j)$ refers to the brightness of the pixel located in (i,j), and i, j are selected such that the pixel is located in the averaging window.

We take the same method to threshold AI as used in

SIFT and SWF, i.e., minimum Euclidian distance.

$$\mathrm{AI}[I_1(x_i, y_i)] - \mathrm{AI}[I_2(x_j, y_j)] = \begin{cases} \leqslant I_\mathrm{T}, & \text{Accept} \\ > I_\mathrm{T}, & \text{Reject} \end{cases}, \quad (8)$$

where $\mathrm{AI}[I_1(x_i, y_i)]$ and $\mathrm{AI}[I_2(x_j, y_j)]$ are AI for the compared matched pair $I_1(x_i, y_i)$ and $I_2(x_j, y_j)$, respectively.

The Terravic Research's Infrared Human Faces Database contains infrared picture frames of the faces of 20 people taken in a variety of cases: wearing glasses, wearing a hat, or with face rotation. We choose 3 to 6 frames of every person's infrared face picture sequences, resulting in 102 frames in total. The frames are chosen based on the rule that each frame represents a case stated in the previous paragraph. More specifically, the frames chosen for the face rotation case are taken at 15°. from the front side direction. This angle choice is typical because the cameras are located 30°. away in three-dimensional photography.

Recall-Precision is chosen as our evaluation metric. Recall and 1-Precision are defined as[1]

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}), \quad (9)$$

$$1 - \text{Precision} = \text{FP}/(\text{TP} + \text{FP}), \quad (10)$$

where TP (true-positive) is a match generated by the algorithm where the two points correspond to the same physical location; FP (false-positive) is a match generated by the algorithm where the two points correspond to different physical locations; and FN (false-negative) is a match corresponding to the same physical location but not identified by the algorithm. In our experiment, the values of FP and (TP+FP) differ little. Therefore, the ratio Recall/(1-Precision) is used to represent the performance of the algorithm instead.

The cases in the experiment, especially the cases of wearing hat and wearing glasses, cannot be easily modeled. Thus, generating TP and FN automatically would be difficult. Fortunately, despite the large number of keypoints generated by SIFT in high resolution images, the keypoints are far more less in the case of infrared human picture. The matches are mostly around 100. We therefore identify the parameter TP+FN manually to ensure correctness.

We compare the Recall-Precision performance among SWF, CWF, and YWF. The SIFT result is given as the baseline. Three cases of infrared human faces matching are tested: (1) rotation of 15°. including both right and left rotation; (2) wearing glasses, where the part of the face behind the glasses is completely shielded; and (3) wearing a hat. There is no separate case for brightness variation because the pictures tested for the previous three cases were already subjected to brightness variation.

Figure 3 presents typical results for the wearing glasses case in our experiments. The bold lines represent the incorrect matches resulting form similar local textures. Figures 3(b)–(d) show the matching results of SWF, CWF, and YWF applied to the same pair of images used in Fig. 3(a), respectively. CWF and YWF clearly dominate SWF in eliminating incorrect matches. There is only one incorrect match in both Figs. 3(c) and (d), whereas Fig. 3(b) has two errors. When comparing CWF and YWF, the number of total matches in Fig. 3(c) is slightly smaller than that in Fig. 3(d). Furthermore, YWF generates more correct matches.

The size of the averaging window is determined by the picture's resolution and the noise level, which are characterized by the database tested. The same infrared database in Ref. [16] is used; hence, the same group of N, i.e., N=2, 3, 4, is tested. The result is shown in Table 1. The TP+FP AVG column gives the average of total matches. The FP AVG column gives the average of false matches. PR ratio represents the ratio of Recall/(1-Precision). The Recall/(1-Precision) ratio of N=3 is about 8% higher than that of N=2, and 11% higher than that of N=4. Therefore N=3 is the best choice.

Table 2 compares different thresholds of YWF. As shown in the table, along with threshold increase, both total matches and false matches increase. From I_T =0.15 to I_T =0.17, PR ratio increases, while from I_T =0.17 to I_T =0.19, it decreases. Therefore, I_T =0.17 makes the best performance. A small value of I_T would, therefore, degrade the performance because the gain from decreased false matches is less than the loss from decreased total matches. Meanwhile, a large I_T value would also degrade the performance because the gain from increased total matches is less than the loss from increased false matches. I_T =0.17 is the value which balances both the gain and the loss. Therefore, 0.17 is chosen in our comparison experiment.

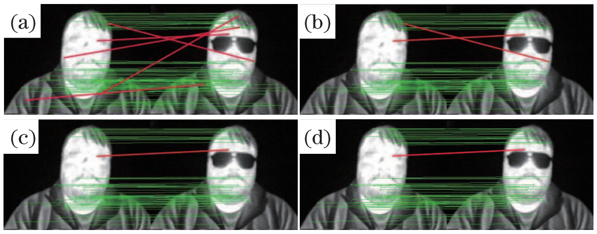

Fig. 3. Comparison of filtering effectiveness of SIFT versus three different kinds of averaging window filters. An infrared face with and without glasses is chosen as the candidate picture. (a) Match result for SIFT; (b)–(d) match results for SWF, CWF, and YWF averaging window, respectively. The bold lines denote the wrong matches while the thin ones denote the correct matches.

Table 1. Recall/(1-Precision) Performance of YWF with Different Window Sizes

N	TP+FP AVG	FP AVG	PR ratio
2	32.35135	1.324324	4.240739
3	32.36486	1.216216	4.63773
4	32.37838	1.216216	4.11642

Table 2. Performance of YWF with Different Threshholds

Threshold	TP+FP AVG	FP AVG	PR Ratio
0.15	30.82432	1.148649	4.455639
0.16	31.61282	1.175676	4.585721
0.17	32.36486	1.216216	4.63773
0.18	32.83784	1.333333	4.341221
0.19	33.45946	1.527027	3.914781

Table 3. Recall/(1-Precision) Performance of SIFT, SWF, and the Proposed Methods

Algorithm	TP+FP AVG	FP AVG	PR Ratio
SIFT	53.79487	6.589744	2.156079
SWF	35.51282	1.794872	3.732640
CWF	32.22973	1.256757	4.444179
YWF	32.36486	1.216216	4.637730

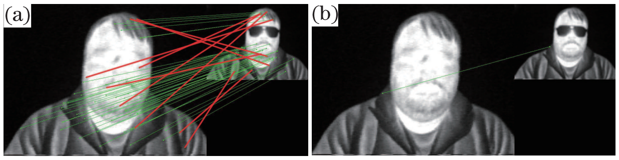

Fig. 4. (a) SIFT and (b) YWF averaging window's performance in scale variation. The tested picture frames are down sampled to half their original size. The bold lines denote the wrong matches while the thin ones denote the correct matches.

Table 3 compares the SIFT and SWF results, and the proposed CWF and YWF averaging window filters. The number of false matches decreases significantly and the number of total matches also decreases using YWF compared to those using SWF. Based on these results, YWF discards false matches at the cost of filtering some correct matches, i.e., there is a tradeoff between false matches and total matches. As for the ratio of Recall/(1-Precision), YWF clearly dominates SWF in the experiment. The Recall/(1-Precision) ratio of YWF is about 16% better than that of SIFT.

YWF eliminates some of the correct matches while reducing the incorrect matches. Hence, a natural question to ask is what kind of correct matches it can pick out. Our experiments show that the correct matches mistakenly filtered by YWF are subjected to strong noise. In this situation, the differences of average brightness exceed the threshold and are judged as mismatches. Strong noise cannot be handled by simply applying an averaging window method. This is beyond the reach of the algorithms considered in this letter. Major improvements of our algorithm or alternative algorithms may be needed to minimize or eliminate the adverse effect of strong noise.

YWF inherits the four basic stages of SIFT; therefore, it is not robust to scale variation either. The facial infrared pictures used in our experiment do not provide scale variation samples. Therefore, a simple experiment is performed by down sampling the picture frames to half their original size. They are then tested with both SIFT and YWF-SIFT against the original ones. Both algorithms incorrectly eliminated most of the matches.

Figure 4 illustrates the typical result of SIFT and YWF-SIFT in scale variation. Compared to Fig. 3, performance of both algorithms obviously decreases. More false matches are generated by SIFT and few total matches by YWF-SIFT. This phenomenon is attributed to the intrinsic defect of SIFT, whose performance degrades for infrared images with low definition. Meanwhile, scale variation feature is not considered in the design of the averaging window.

In conclusion, we propose a novel averaging window filter YWF for applying SIFT to infrared human face recognition. Compared with SWF recently proposed by Tan *et al.*[16], YWF patterns are sparser and are designed to avoid rotation invariance. Experimental results show that YWF is more suitable for eliminating wrong matches generated by SIFT. YWF could also be a viable method for filtering false matches of color images because it only utilizes the local texture information around a keypoint. We are currently exploring this idea in other infrared application scenarios and color images.

This work was supported by the Natural Science Foundation of Hubei Province under Grant No. 2009CDB320.

References

1. S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, Comput. Vis. Image Und. **97,** 103 (2005).
2. D. Socolinsky, L. Wolff, J. Neuheisel, and C. Eveland, in *Proceeding of IEEE Workshop Computer Vision and Pattern Recognition* **1,** I-527 (2001).
3. A. Selinger, *Appearance-Based Facial Recognition Using Visible and Thermal Imagery: a Comparative Study* (Equinox Corp, New York, 2006).
4. F. Prokoski, in *Proceeding of IEEE Workshop Computer Vision Beyond Visible Spectrum: Methods and Applications* **3,** 1688 (2000).
5. K. Mikolajczyk and C. Schmid, in *Proceedings of Computer Vision and Pattern Recognition* **2,** II-257 (2003).
6. C. Schmid and R. Mohr, IEEE Trans. Pattern Anal. **19,** 530 (1997).
7. D. G. Lowe, in *Proceedings of International Conference on Computer Vision* 1150 (1999).
8. D. G. Lowe, Int. J. Comput. Vision. **60,** 91 (2004).
9. A. Baumberg, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition* 774 (2000).
10. T. Tuytelaars and L. Van Gool, in *Proceedings of the 11th British Machine Vision Conference* 412 (2000).
11. M. Brown and D. Lowe, in *Proceedings of the 13th British Machine Vision Conference* 253 (2002).
12. P. L. Ding, J. F. Mei, L. M. Zhang, J. Infrared Millim. Wav. **20,** 5 (2001).
13. G. Chen and F. H. Qi, J. Infrared and Millim. Wav. **19,** 5 (2000).
14. Q. Zeng, L. Liu, and J. Li, Chin. Opt. Lett. **8,** 573 (2010).
15. M. Brown, R. Szeliski and S. Winder, in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **1,** 510 (2005).
16. C. Tan, H. Wang, and D. Pei, Tsinghua Sci. Technol. **15,** 357 (2010).
17. Y. Ke and R. Sukthankar, in *Proceedings of the* 2004 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2,** 506 (2004).
18. Andrea Vedaldi, "SIFT for Matlab", http://www.vlfeat.org/∼vedaldi/code/sift.html (October, 2005).