



Light-Weight Global Feature for Mobile Clothing Search

Guangshan Wen¹, Jing Wu^{1(✉)}, Chengnian Long^{1(✉)}, and Yi-Bing Lin²

¹ Department of Automation, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
{jingwu,longcn}@sjtu.edu.cn

² Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
liny@cs.nctu.edu.tw

Abstract. Mobile clothing search on smartphones is extraordinary challenging on both embedded computer vision and networking search system design due to the flexible object, constrained computing, networking resource, and quick response requirement. In this paper, we propose a light-weight global feature LGF – threshold-LBP and color histogram, for mobile clothing search, which is effective for feature extracting and networking communication on smartphone. In addition, an incremental feature is proposed in rerank module to promote the rerank computation and achieve the quick response requirement. Our approach is evaluated through extensive experiments on the smartphones. It reveals that mobile clothing search based on our global feature is more effective than other clothing retrieval system using prevalent local descriptors, such as SIFT, RGB-SIFT.

Keywords: Smartphone · Computer vision · Flexible object
Global feature · Threshold-LBP · Color histogram · Incremental feature

1 Background

With the rapid development of e-commerce, many commodities are displayed by the way of image on Internet. Besides, as the mobile Internet industry becomes prevalent in these years, an efficient mobile clothing search system is expected for people to retrieve similar clothes from massive amounts of on-line commodities with the image they captured by mobile camera.

Comparing with traditional Content-Based-image-retrieval systems, clothing-item retrieval system is extremely challenging. As a kind of flexible object, it is impossible to extract shape feature from cloth item. In addition, the texture feature caused by clothing folds is hardly distinguished from real clothing

This work was supported by National Nature Science Foundation under Grant 61673275 and 61473184.

texture. Thus, the prevalent local feature such as SIFT, SURF, is not appropriate for cloth item retrieval. Besides, in the view of limited computation ability on mobile device, the extraction of local feature, such as SIFT, SURF, is quite expensive. In addition, the high-dimensional local feature will also aggravate network traffic overhead.

There has been a great deal of work on clothing image retrieval in the past few years [1–8, 11], which mostly based on traditional local feature, such as SIFT, SURF, and the recently developed deep feature. Besides, kind of global feature is also proposed in [1], which is only focus on color information, and the computation of Color Moment in their work is quite expensive. The recently developed deep feature is a kind of high-level feature, which can describe clothing image with multiple levels of abstraction [10], and be robust to clothing variation. However, as we can see from [9], the CNN used in [7] for clothing feature representation has millions of parameters and 500,000 neurons, which causes expensive computation.

The main contributions of this paper can be summarized as follows: (1) we propose a kind of global texture feature – threshold-LBP to collaborate with color histogram, which is more effective than local feature (2) we design sorted color histogram feature to speed-up the computation in re-rank model.

The remainder of this paper is organized as follows. We will deep into the detail of the proposed light-weight global feature in Sect. 2, and the evaluation is in Sect. 3. Finally, we conclude in Sect. 4.

2 Feature Design

2.1 Color Histogram

Color is the most important feature in clothing images, which contains the most information consumers care about.

The normal display system with 8-bit color can display up to 2^{24} colors in RGB space, which means the color histogram extracting from rgb-image will have high dimension of ten million. Apparently, high dimension will lead to expensive computation in the post-processing. Besides, due to the consideration of each kind of color, the color histogram will be sensitive to color variation caused by illumination, reflection and other external environment.

We utilize color quantization to archive dimensionality reduction for color histogram. Color quantization involves dividing the RGB color space cube into a number of small boxes, and then mapping colors fall within box at the center of the box. Our implementation based on uniform quantization, which quantizes each channel uniformly. In our experiment, we partition R, G and B channel into 8 bins separately, while the total number of colors in color histogram will be fixed to $8 \times 8 \times 8$. Suppose the RGB value of pixel x is (r_x, g_x, b_x) , then the quantization color of x is $(\lceil (r_x/32) \rceil \times 16, \lceil (g_x/32) \rceil \times 16, \lceil (b_x/32) \rceil \times 16)$, where $\lceil (x) \rceil$ is the least integer greater than or equal to x .

2.2 Texture Feature

Besides color distribution, the pattern design on clothes is also an important information for clothing image. Texture is an essential feature to describe the clothing pattern, which can extract the spatial arrangement of intensities in an image. As a result, our work makes use of texture feature to implement more accurate retrieval. There is a large number of texture feature can be chosen, such as HOG, GLCM, HAARS. In consideration to the limited computation ability of smartphone, we apply LBP (Local binary patterns), which is a powerful feature for texture classification. In traditional LBP feature extracting, each pixel is compared to its 8 neighbors, and when the pixel's value is greater than the neighbor's value, then the corresponding bit of LBP feature is 1, otherwise is 0.

$$f(x, y) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \quad (1)$$

where

$$s(x) = \begin{cases} 1 & x \geq \text{threshold}, \\ 0 & x < \text{threshold}. \end{cases}$$

However, in the practical scenarios, the high degree of flexibility of clothing object causes the fold and shadow, which raises a large amount of noise for texture detection. In order to reduce the sensitivity of texture feature, the threshold between the comparison of pixel and its neighbors is increased, As shown in Eq. 1, where g_c is centroid of threshold-LBP feature at pixel (x, y) , and the radius of threshold-LBP P is set to 8, and $g_p(p = 0, 1, 2, \dots, P-1)$ is the feature region with the center of g_c . In our experiment, only when the value of pixel is at least 16 greater than its neighbors, can the corresponding bit set 1, which will lead to better result, as shown in the evaluation.

2.3 Sorted Color Histogram

The re-rank module will rerank the candidate result to refine the final returned list.

Given two color histograms $ch_1 = (x_1, x_2, \dots, x_n)$ and $ch_2 = (y_1, y_2, \dots, y_n)$, the traditional algorithm to measure the similarity is Euler distance $d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$, while the complexity of which is liner to the dimension of color histogram, $O(n)$. Based on the characteristics of clothing that the more percentage of the color account for in a clothing, the more important the color is for clothing image retrieval, we proposes an algorithm to compute the similarity between any two color histograms by summing up each color's distance, where the color is distinguished by their importance and the computation sequence is in descending order of color's importance. The detail procedure is shown in Algorithm 1.

Algorithm 1 utilizes two extra parameters to speed up the similarity computation: $dis_{threshold}$ and $per_{threshold}$. $dis_{threshold}$ can stop algorithm when the

Algorithm 1. Efficient algorithm to compute the similarities between two color histograms

Input : Color Histogram ch_1 and ch_2
 Threshold: $dis_{threshold}$, $per_{threshold}$
Output: Distance $distance$

```

1  $pert_1, color_1 = sort(ch_1)$ ;
2  $pert_2, color_2 = sort(ch_2)$ ;
3  $distance = 0$ ;
4  $percentage = 0$ ;
5 for  $(p_{1i}, c_{1i})$  in  $(pert_1, color_1)$  and  $(p_{2i}, c_{2i})$  in  $(pert_2, color_2)$  do
6   if  $dis_{threshold} < distance$  then
7      $distance = +\infty$ ;
8     break;
9   end
10  if  $per_{threshold} < percentage$  then
11    break;
12  end
13   $s = (p_{1i} + p_{2i})/2$ ;
14  if  $c_{1i} == c_{2i}$  then
15     $distance += s \times abs(p_{1i} - p_{2i})$ ;
16  else
17     $distance += max(p_{1i}, p_{2i})$ ;
18  end
19   $percentage += min(pert_1, pert_2)$ ;
20 end
21 return  $distance/percentage$ ;

```

current distance is greater than pre-setted threshold, which means there is small probability that the two color histograms are similar. Besides, the function of $per_{threshold}$ is to control the degree of colors that have been involved in the calculation. A sentinel $percentage$ is used to record the current percentage of colors have been involved. When $percentage > per_{threshold}$, it can be regarded as the dominant colors have been considered, and the remain computation will have little effect on final result. So the algorithm can be stop.

Algorithm 1 makes best use of the fact that most clothes contain several dominant colors, and the proportion of other colors can be ignored. As a result, the computation focus on the few kinds of dominant colors, which will have less side-effect on searching result and be more efficient in time.

3 Evaluation

In this section, we design extensive experiments to evaluate the proposed feature in mobile clothing search. Besides, the comparison with other works will also be presented. Our experiment is based on a publicly available dataset in [7]. The images in the dataset have a variety of variations, such as illumination, occlusion, pose, which can better evaluate the robustness and retrieval accuracy.

3.1 Feature Parameter Analysis

Our work uses threshold-LBP feature and kind of incremental image feature in rerank module to efficiently refine the retrieval candidate result, which is also called Sorted Color Histogram (SCH). In this section, we will analysis the parameters *threshold* of threshold-LBP feature and *per_{threshold}* of SCH.

The result shown in Fig. 1 demonstrates the retrieval accuracy of different threshold for LBP feature. We evaluate the threshold with 0, 6, 10, 12, 16, 20, 24 respectively. As we can see from Fig. 1, threshold with 16 results in the best performance. And the increasing of threshold larger than 16 will significantly reduce the accuracy, which is a result of high threshold that causes texture fuzzy, and leads to detail loss.

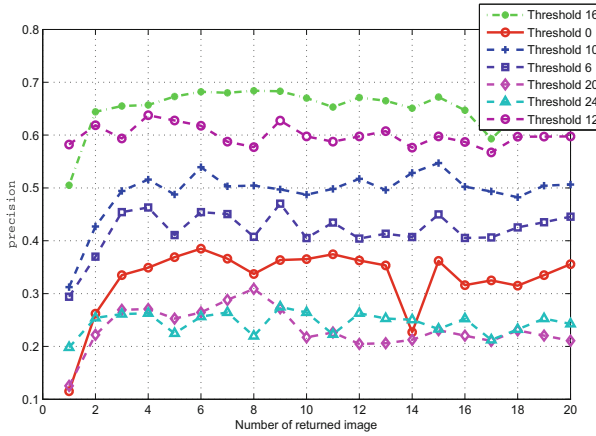


Fig. 1. Experiment of LBP with different threshold

In our experiment, high threshold even has worse performance than raw LBP feature. As clothing object is easily deformed, the folds on clothing surface may be misrecognized as texture, the evaluation result shows that appropriate threshold can rectify the fake texture. As shown in Fig. 1, the result with threshold from 6 to 16 surpass raw LBP. The following evaluation will be based on threshold of 16.

For SCH, as shown in Fig. 2, for our testing image dataset, using 85% of colors can reach nearly the same result with 100%, which means the 85% colors contain almost all the color information in clothing image. In addition, we can see from Fig. 3 that the computing time used in similarity measurement for the most 85% important colors is less than half of the computing time for 100%, which is based on the fact that less than 50% of bins in color histogram includes more than 85% color information in clothing image. In the following evaluation, the *per_{threshold}* of SCH is set to 85%.

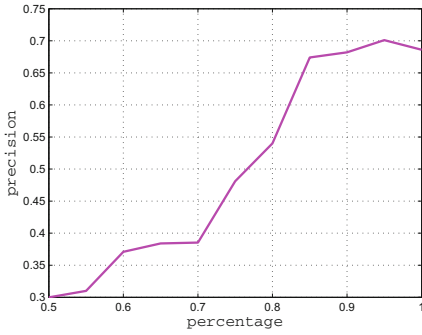


Fig. 2. Percentage vs precision

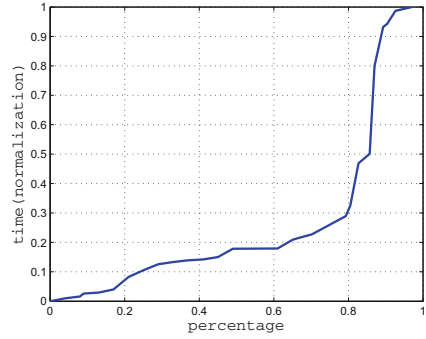


Fig. 3. Percentage vs time

3.2 Comparison and Result

We implement our light-weight global feature based clothing search system on a mobile phone to evaluate the practicability of our cloth retrieval system. The practical experiment is based a SONY Xperia with Android 5.1.1 OS and a remote server with the configuration of core i7. we estimate the retrieval speed by retrieval time. The feature packet for network communication is shown in Fig. 4. In *requestline*, the *m* indicates the number of returned images, *n* reveals the kind of feature for extraction, while the substance is in *requestbody*.

Request line	GET images\m Feature\n Version 1.0\
Request body	<pre> <feature name = color, length = size1, type = int> </feature> \r\n <feature name = texture, length = size2, type = int> </feature> \r\n \r\n\r\n </pre>

Fig. 4. Feature packet

To demonstrate the better performance on retrieval accuracy of our clothing image retrieval system, we compare with some other baseline Content-based Image retrieval methods using prevalent local descriptor, such as SIFT, RGB-SIFT, and Color Moment.

The result in Fig. 5 shows that our light-weight global feature – LGF based retrieval system has a gain of nearly 12.1% in average precision compared with SIFT, which is the result of that traditional SIFT descriptor has little attention in color information. Besides, a improvement about 5.6% and 6.1% respectively compared with Cascaded Color Moment and RGB-SIFT descriptor.

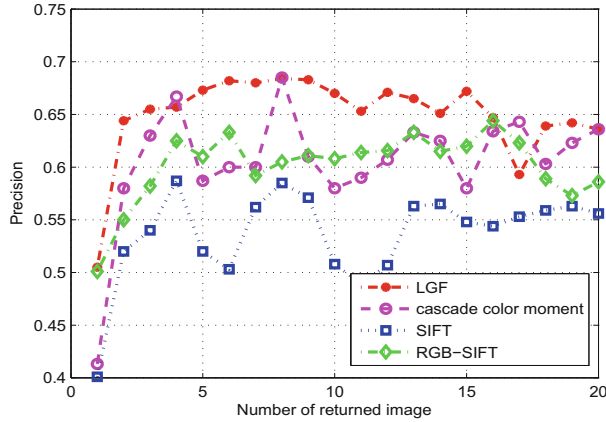


Fig. 5. Precision comparison

Table 1. Efficiency –LGF/SIFT/RGB – SIFT/CascadeColorMoment

Experiment case	Accuracy (10 images)	Retrieval time(s)
1	6/5/6/3	3.312/7.345/8.762/13.081
2	7/4/5/4	4.423/6.534/7.412/15.073
3	7/6/6/5	4.153/9.834/7.215/7.101
4	6/3/5/5	5.002/8.987/9.351/11.205
5	5/6/4/5	4.689/8.023/11.365/13.503
6	6/4/7/5	3.563/7.056/10.279/9.347
7	7/5/6/4	6.761/9.341/7.126/7.149
8	9/6/8/7	4.852/9.371/9.321/10.713
9	5/5/7/3	6.162/7.528/8.721/9.631
10	6/4/5/6	5.239/11.091/8.274/8.905
11	7/7/6/8	3.342/8.801/9.481/10.484
12	9/5/7/6	3.293/5.231/7.258/12.427
13	7/3/8/6	4.303/7.546/9.326/8.261
14	6/5/5/4	6.192/5.628/6.261/9.731
15	8/5/7/5	8.579/7.971/7.259/11.174
Average	6.733/4.867/6.133/5.067	4.924/8.02/8.49/10.52

The experiment result in Table 1 shows that our light-weight global feature based system has better performance on both retrieval accuracy and search efficiency. The light-weight global feature can speed searching up at least two times faster than complicated feature, such as SIFT/RGB-SIFT/Cascade Color Moment, without side-effect on retrieval accuracy.

4 Conclusion

In this paper, we presented a mobile clothing image retrieval system based on the proposed threshold-LBP feature and color histogram. Threshold-LBP feature can better overcome the noisy texture caused by high degree flexibility of clothing object. To refine the candidate result, Sorted Color Histogram is presented in re-rank module. The system evaluation demonstrates the effectiveness of the proposed global feature over other clothing retrieval systems based on prevalent local descriptors.

References

1. Weng, T., Yuan, Y., Shen, L., Zhao, Y.: Clothing image retrieval using color moment. In: Proceedings of the IEEE Conference on Computer Science and Network Technology, pp. 1016–1020 (2013)
2. Wang, X., Zhang, T., Tretter, D.R., Lin, Q.: Personal clothing retrieval on photo collections by color and attributes. *IEEE Trans. Multimed.* **15**(8), 2035–2045 (2013)
3. Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-shop: cross-scenario clothing retrieval via parts alignment and auxiliary set. In: Proceedings of the IEEE Conference on Multimedia, pp. 1335–1336 (2012)
4. Di, W., Wah, C., Bhardwaj, A., Piramuthu, R., Sundaresan, N.: Style finder: fine-grained clothing style detection and retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 8–13 (2013)
5. Cushen, G.A., Nixon, M.S.: Mobile visual clothing search. In: Proceedings of the IEEE Conference on Multimedia and Expo Workshops, pp. 1–6 (2013)
6. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Retrieving similar styles to parse clothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(5), 1028–1040 (2015)
7. Hadi Kiapour, M., Han, X., Lazebnik, S., Berg, A.C., Berg, T.L.: Where to buy it: matching street clothing photos in online shops. In: Proceedings of the IEEE Conference on Computer Vision, pp. 3343–3351 (2015)
8. Fu, J., Wang, J., Li, Z., Xu, M., Lu, H.: Efficient clothing retrieval with semantic-preserving visual phrases. In: Proceedings of the IEEE Asian Conference on Computer Vision, pp. 420–431 (2012)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: ACM International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
11. Liu, Z., Luo, P., Qiu, S., et al.: DeepFashion: powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1096–1104 (2016)