JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 11, No. 1, March 2017, pp. 24-31

A Comparative Study of Local Features in Face-based Video Retrieval

Juan Zhou and Lan Huang*
Department of Computer Science, Yangtze University, Jingzhou, China
zhoujuan_yangtze@163.com, lanhuang@yangtzeu.edu.cn

Abstract
Face-based video retrieval has become an active and important branch of intelligent video analysis. Face profiling and matching is a fundamental step and is crucial to the effectiveness of video retrieval. Although many algorithms have been developed for processing static face images, their effectiveness in face-based video retrieval is still unknown, simply because videos have different resolutions, faces vary in scale, and different lighting conditions and angles are used. In this paper, we combined content-based and semantic-based image analysis techniques, and systematically evaluated four mainstream local features to represent face images in the video retrieval task: Harris operators, SIFT and SURF descriptors, and eigenfaces. Results of ten independent runs of 10-fold cross-validation on datasets consisting of TED (Technology Entertainment Design) talk videos showed the effectiveness of our approach, where the SIFT descriptors achieved an average F-score of 0.725 in video retrieval and thus were the most effective, while the SURF descriptors were computed in 0.3 seconds per image on average and were the most efficient in most cases.

Category: Smart and intelligent computing

Keywords: Video retrieval; Face matching; Harris operators; SIFT; SURF; Eigenfaces

I. INTRODUCTION

Given a static image or a video fragment, face-based video retrieval finds videos that have the same faces as those in the input video. Nowadays, surveillance is ubiquitous, protecting our safety whenever needed. In urgent cases, face-based video retrieval on collected surveillance videos allows us to promptly search and track a target person. Meanwhile, the number of person-centered videos such as talks and lectures on the Internet has rapidly increased. These videos have become a new interest for end users and a new resource for mining video information. For example, using a target person as a keyword to search for related videos might increase a video's click rates. Therefore, face-based video retrieval has become a prominent issue in research as well as in application.

An automatic face-based video retrieval system usually consists of four components: representative frame extraction (i.e., images containing clear faces with a reasonable size), face detection and extraction, face matching (i.e., relevance calculation), and ranking. Among these components, face matching is the most critical and thus the focus of this paper. Further information on the other components can be found in previous works [1-3]. Among the different stages of face matching, feature point detection and the corresponding point-based matching are the most important stages. Although several fea-

Open Access http://dx.doi.org/10.5626/JCSE.2017.11.1.24 http://jcse.kiise.org

Received 13 April 2016; **Revised** 08 March 2017; **Accepted** 09 March 2017
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

ture algorithms have been proposed, their effectiveness in video retrieval is still unknown. Videos have different resolutions, faces vary in scale, and various lighting conditions and angles are used, presenting a greater challenge than static image retrieval. In this paper, we evaluate four representative feature algorithms with respect to the face-based video retrieval task. Given an input query video, the task involves finding all videos in the backend database that contain the same face as that in the query video. Our experimental results provide valuable insight into choosing the appropriate algorithm under different circumstances.

II. RELATED WORK

Video retrieval methods can be roughly categorized into three types: text-based, content-based, and semantic-based [4, 5]. Text-based methods rely on the textual annotation of video content, which is usually added manually, and thus are subjective and tedious to generate. Content-based methods [1] overcome this manual burden by automatically recognizing colors and textures, and have become an active arena of multimedia research [6]. Semantic-based methods [7] aim to derive the meaning behind the scenes and perform matching based on meaning rather than physical traits, i.e., crossing the semantic gap in video retrieval [4]. In this paper, we combine the content-based and semantic-based methods; that is, we extract key faces using content analysis and derive face profiles of a video fragment as its semantic representation.

Image matching is the basis of our video retrieval task, which can be performed based on either templates or features [8]. Image matching based on templates uses mathematical models to generate an overall description of the content of an image. Although rigid, templates usually lack adaptability to noise and template changes. Image matching based on features finds locally unique and invariant features to represent an image, instead of trying to fit the entire image into one template. Therefore, they usually can reduce calculation and adapt to variations (such as rotation, scale, and illumination changes), achieving greater robustness. Harris and Stephens [9] proposed one of the first pixel-based descriptors as image features, later known as the Harris operators. Lowe [10, 11] proposed Scale Invariant Feature Transform (SIFT), which searches for extreme value points in an image. Subsequently, Bay et al. [12] proposed the Speeded Up Robust Features (SURF) to optimize the efficiency of SIFT and its ability to handle light condition variations. For feature-based methods, face matching consists of two steps: extracting features from an image and performing matching by calculating the distance (or similarity) between two sets of features. Due to their many advantages, we adopt the feature-based methods. We extract features from representative face images and construct the face profile of a video fragment, upon which the relevance between video fragments is computed.

Various studies have been carried out related to this area of research [13-17]. Ke and Sukthankar [13] compared SIFT with PCA-SIFT under Gaussian noise and varying viewpoints. Their results showed that PCA-SIFT, which combined Principal Components Analysis (PCA) with SIFT in order to improve efficiency, successfully increased the accuracy and efficiency of an image retrieval task. Mikolajczyk and Schmid [14] studied ten descriptors in image matching, including SIFT, PCA-SIFT, sharp context and gradient location and orientation histogram (GLOH). Their results found SIFT and PCA-SIFT to be the best features for the task. Juan and Gwun [15] tested SIFT, PCA-SIFT, and SURF descriptors, also used in image matching. Le et al. [16] compared Harris operators and SIFT descriptors to find correspondence points in video frame images. They extracted consecutive frames from a video as test images with different lighting conditions and view angles. Their results showed that, while Harris operators showed rapid computing, SIFT descriptors had higher accuracy and robustness. Later, Miksik and Mikolajczyk [17] compared more recent extractors including binary robust independent elementary features (BRIEF) and binary robust invariant scalable keypoints (BRISK) with SIFT and SURF and obtained comparable performances among them. In general, SIFT was found to perform well under scale transformation and rotation, yet suffered from high computation costs. Although extensive research has been performed on comparing these local features, most research has focused on matching static images. Their comparative effectiveness and efficiency in face-based video retrieval remains unknown, which is the key research problem our study attempts to answer. Considering the current usage of the descriptors, we chose four basic schemes (i.e., Harris, SIFT, SURF, and Eigenfaces) in this study, which are probably also the most widely applied schemes.

III. CONSTRUCTING FACE PROFILES USING LOCAL FEATURES

Given a video fragment, our system indexes the video in three steps: identifying key frames, extracting face images, and constructing a face profile of the video. The face profile thus serves as the index entry, such as in traditional information retrieval systems: relevance between videos will be computed based on their corresponding profiles. Obviously, methods for constructing the profiles become the cornerstone of the system. Considering the scope of application, we focus on four mainstream local features: Harris operators, SIFT descriptors, SURF descriptors, and eigenfaces. This section describes these features. Details about the first two steps are presented in Section IV.

A. Harris Operators

Motivated by Morevec's algorithm [18], Harris and Stephens [9] proposed one of the most commonly used corner descriptors: the Harris operators. Given a gray-scale image I and a patch over the area (u, v) by shifting (x, y), a Harris operator is characterized by a large variation of the weighted sum of squared differences $S(x, y)$ in all directions:

$$S(x, y) = \sum_{u,v} w(u, v)(I(u+x, v+y) - I(u, v))^2. \quad (1)$$

Operators from two images are then matched based on their pairwise distances, e.g., in terms of the Euclidean distance. Only pairs with a distance lower than a pre-specified threshold are considered as matching features.

Fig. 1(a) shows the face matching results of the Harris operators. Dots denote the identified operators and lines denote the matching between operators in the different images. When two images are exactly the same, matching is usually perfect, as shown by the horizontal lines in the left pair of images in Fig. 1(a)–(c). In contrast, mismatches occur when the face is slightly rotated, as shown in the right pair of images in Fig. 1(a)–(c). This simple example intuitively demonstrates the challenges in applying static face image matching for face-based video retrieval.

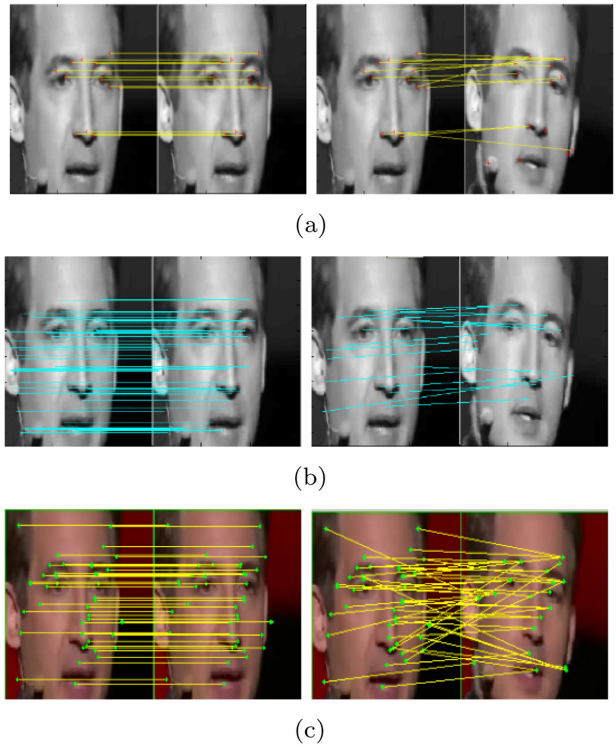

Fig. 1. Features extracted from face images in the Technology Entertainment Design (TED) dataset and their matching results. (a) Harris operators, (b) SIFT descriptors, and (c) SURF descriptors.

B. SIFT Descriptors

Lowe [10] proposed the SIFT descriptors, the extreme points in the Difference of Gaussian (DoG) scale space. Given an image $I(x, y)$, its DoG space can be computed by

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \otimes I(x, y), \quad (2)$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$, $\otimes$ denotes the convolution operation and σ is the scale factor. Usually, $k = 2^{\frac{1}{s}}$, where s is the number of levels in the scale space. Interesting points generate from the local optima in the DoG space. Considering the false positive optima, SIFT removes any low contrast points and any points on the edges. It then plots the neighboring gradient map of the interest points and calculates the primary and the secondary directions of the gradient.

Descriptors from one image are then matched to those from another image. Given a descriptor a in image A, its matching descriptor a' in image B is determined by the following two criteria. First, (a, a') should be the closest to each other among all possible pairs (a, b), where b in B. For example, the Euclidean distance between a and a' should be the lowest and also below a pre-specified threshold. Second, the distance between the descriptors in (a, a') is significantly less than the second least distance, e.g., more than 20% less [11].

C. SURF Descriptors

Similar to SIFT, SURF descriptors also explore the scale space representation of images. However, SURF introduces several changes to reduce computational costs [12]. SURF first uses the determinant of Hessian (DoH) to extract interest points. Given a point $p(x, y)$ in image I, the DoH at p and scale σ is:

$$H(p, \sigma) = L_{xx}(p, \sigma)L_{yy}(p, \sigma) - L_{xy}^2(p, \sigma). \quad (3)$$

$L_{xx}(p, \sigma)$ is the convolution of the Gaussian second-order derivative with image I in point p:

$$L_{xx}(p, \sigma) = \frac{\partial^2}{\partial x^2} G(I, \sigma) \otimes I, \quad (4)$$

where σ, $\otimes$ and $G(I, \sigma)$ are the same as those in the SIFT descriptors. $L_{yy}(p, \sigma)$ and $L_{xy}(p, \sigma)$ can be calculated similarly. SURF further uses integral image and box filters to approximate DoH interest points. Let D_{xx}, D_{xy} and D_{yy} be the convolution outcome of the filters and the image to which the Hessian matrix is simplified:

$$\Delta H = D_{xx}D_{yy} - (0.9D_{xy})^2. \quad (5)$$

Then, SURF uses Haar-wavelet responses to represent

the local neighborhood of an interest point, resulting in a 64-dimensional vector descriptor, whereas SIFT descriptors normally have 128 dimensions.

To match SURF descriptors, the Laplacian of two given descriptors is computed, with a positive outcome, meaning that the two descriptors share the same type of contrast. For such pairs, if their similarity (usually calculated in terms of their Euclidean distance) exceeds a pre-specified threshold, they are considered as matching features and are retained.

D. Eigenfaces

The approach of using eigenfaces for face recognition was developed by Sirovich and Kirby [19] in 1987 and was then later used by Turk and Pentland [20], who used PCA to find eigenfaces for recognition. Applying PCA to face matching usually involves three steps. First, an 'average' face image is computed from a set of training images and then subtracted from each training image. For example, Fig. 2(a) lists ten training images of the same person shown in Fig. 1, and Fig. 2(b) shows the resulting average face. Second, PCA calculates the eigenfaces from the covariance matrix between each training image and the average face image. Fig. 2(c) shows the nine eigenfaces generated from the ten training images. Finally, given a new face image, the matching algorithm compares the input against the training images, and outputs similar images to those with a similarity (e.g., based on the Euclidean distance) value exceeding a pre-specified threshold.

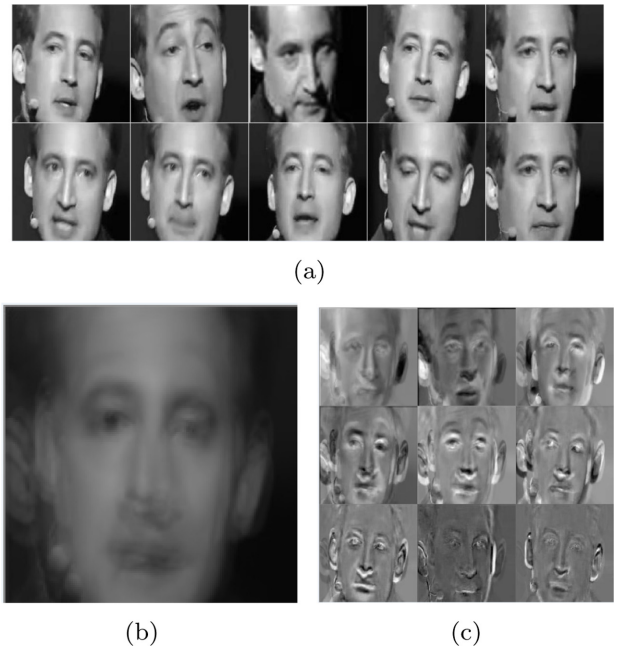

Fig. 2. Training images (a) showing the average face image (b) and the eigenfaces (c).

IV. EXPERIMENTAL SETUP

To evaluate the different face profiling and matching methods, we used video fragments collected from the Technology Entertainment Design (TED) talks. Our experimental dataset consists of 1,197 short video fragments from 73 TED talks. This dataset was created in the face-based video retrieval task of the second big data contest held by the China Computer Federation. Every fragment includes a key speaker. Fig. 3 shows the distribution of the number of fragments per talk. In order to prevent long talks dominating the evaluation, we randomly selected ten talks from the middle of the distribution, i.e., within the [10, 20] range. Table 1 lists the number of fragments of each talk and their total lengths. The length of a single fragment varies from 0.6 to 11.2 minutes, with an average of 4.3 minutes. Fig. 4 shows the speakers of these talks, i.e., the faces for matching. Clearly, the characteristics of the images and those of the ten speakers vary considerably: different gender, outlook, age, posture, background, lighting, etc.

Given a video fragment, our task involves retrieving fragments that have the same speaker as the query fragment. The results show the average of ten independent

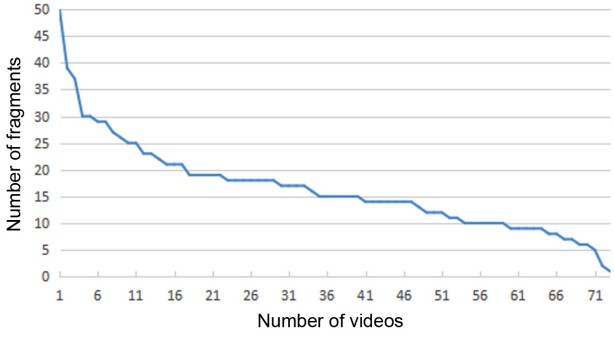

Fig. 3. Distribution of the number of fragments per talk.

Table 1. Ten talks randomly selected from the original dataset

	Talk										Total
	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	
Number of fragments	18	18	17	17	17	16	15	14	14	14	160
Length (min)	76.9	67.1	93.5	65.0	55.3	94.3	40.2	58.4	77.0	55.5	683.2

Fig. 4. Speakers of the selected talks.

runs of a 10-fold cross-validation. In each run, the 160 video fragments were divided into ten different groups: nine for training and one for testing. For each testing fragment, we then rapidly generated its local feature representations, and calculated its relevance to every fragment in the training set. Then, all training fragments were ranked in descending order of relevance and returned to the end user. The standard information retrieval performance measures of *Precision*, *Recall*, and *F-score* were used. *Precision* and *Recall* are calculated as:

$$Precision = \frac{\#\text{correctly retrieved videos}}{\#\text{retrieved videos}},$$

$$Recall = \frac{\#\text{correctly retrieved videos}}{\#\text{relevant videos}}.$$

F-score averages out *Precision* and *Recall*:

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

As described previously, before a video fragment can be indexed and searched, it needs to undergo two preprocessing steps: identifying key frames and extracting the face images therein. Videos in the TED dataset have a frame rate of 15 frames/s; thus, if the current frame contains a face, the next frame is also likely to contain a face. Therefore, we used two sliding window sizes to search for key frames that contain a clear face image of the speaker: forward one frame (i.e., 0.07 second) or three frames (i.e., 0.2 second) respectively, depending on whether or not the current frame is a key frame. Assume $s = \{f_1, f_2, f_3, ..., f_n\}$ is a sequence of frames; we first extract the first frame f_1 and check whether or not f_1 contains a face. If it does, then set f_1 as a key frame and slide one frame forward to extract f_2. Otherwise, abandon f_1 and slide three frames forward to extract f_4, which is more likely than f_2 to contain a face. Then, check whether the extracted frame (f_2 or f_4) contains a face. In practice, we also found it necessary to first split video fragments into shorter clips. This shortened each search and, more importantly, it enabled a parallel search, such as in a distributed computing environment. After initial experimentation and validation, we segmented each video fragment into eight clips, and performed a search within each clip. Each search started from the beginning of a clip until five key frames were found or the end of the clip was reached.

Each key frame was subjected to a face recognition classifier, which combined a frontal face classifier and a mouth classifier to co-locate faces. Successfully identified face images (of size 150×150 pixel) were then extracted from the frame and were saved into the face profile of the video fragment. In order to ensure high consistency of the profile, we applied a filtering process to discard low quality images, such as those with blurry faces. Pairwise Bhattacharyya distance was calculated for every pair of images from the same profile, based on their histograms:

$$d_B(H_1, H_2) = \sqrt{1 - \sum_i \frac{\sqrt{H_1(i) \cdot H_2(i)}}{\sqrt{\sum_i H_1(i) \cdot \sum_i H_2(i)}}},$$

where $H_1(i)$ and $H_2(i)$ are the number of pixels in the ith gray level of the two images. Because the images become more similar as the distance decreases, we recorded the number of times each face image matched another image in the profile, with a distance value lower than 0.3. The images were then sorted in decreasing order of their corresponding counts, and only the top ten were kept, thus forming a coherent profile.

All experiments were performed on a 64-bit Intel Core i3 machine with Windows 8.1. We used FFmpeg [21] for video segmentation and frame extraction, and OpenCV [22] for face recognition. Thresholds for matching features were set to 0.2, 0.3 in Euclidean distance for Harris operators and SURF descriptors, and 0.6 in nearest neighbor distance ratio for SIFT descriptors.

V. EXPERIMENTAL RESULTS AND DISCUSSION

Table 2 lists the best performance of the four local features in face-based video retrieval. Following the standard evaluation procedure in information retrieval, we compared the performances of the four features at different top k result lists. We also implemented a naïve baseline that simply assigned any query video fragment to the longest talk in the dataset, i.e., talk V1 with 18 video fragments (see Table 1). Clearly, SIFT descriptors were the most informative for our task, closely followed by SURF descriptors. Harris operators and eigenfaces retrieved less relevant videos. The overall result is consistent with those reported in the related literature, where these features showed comparable performances in different tasks, i.e., static images matching and face-based video retrieval.

We also examined several factors that might affect performance. As described previously, each video fragment was represented by its profile: a set of face images. The first factor was finding a way to convert the similarity between two sets of images into the relevance between

Table 2. F-score of face-based video retrieval using different local features

	Top5	Top15	Top25	Top35	Top45	Top55	Top65
Baseline	0.029	0.060	0.102	0.129	0.143	0.157	0.167
Harris	0.380	0.486	0.446	0.405	0.365	0.334	0.308
SIFT	0.461	0.725	0.594	0.492	0.420	0.368	0.328
SURF	0.420	0.671	0.598	0.511	0.438	0.385	0.343
Eigenfaces	0.286	0.344	0.331	0.316	0.298	0.282	0.269

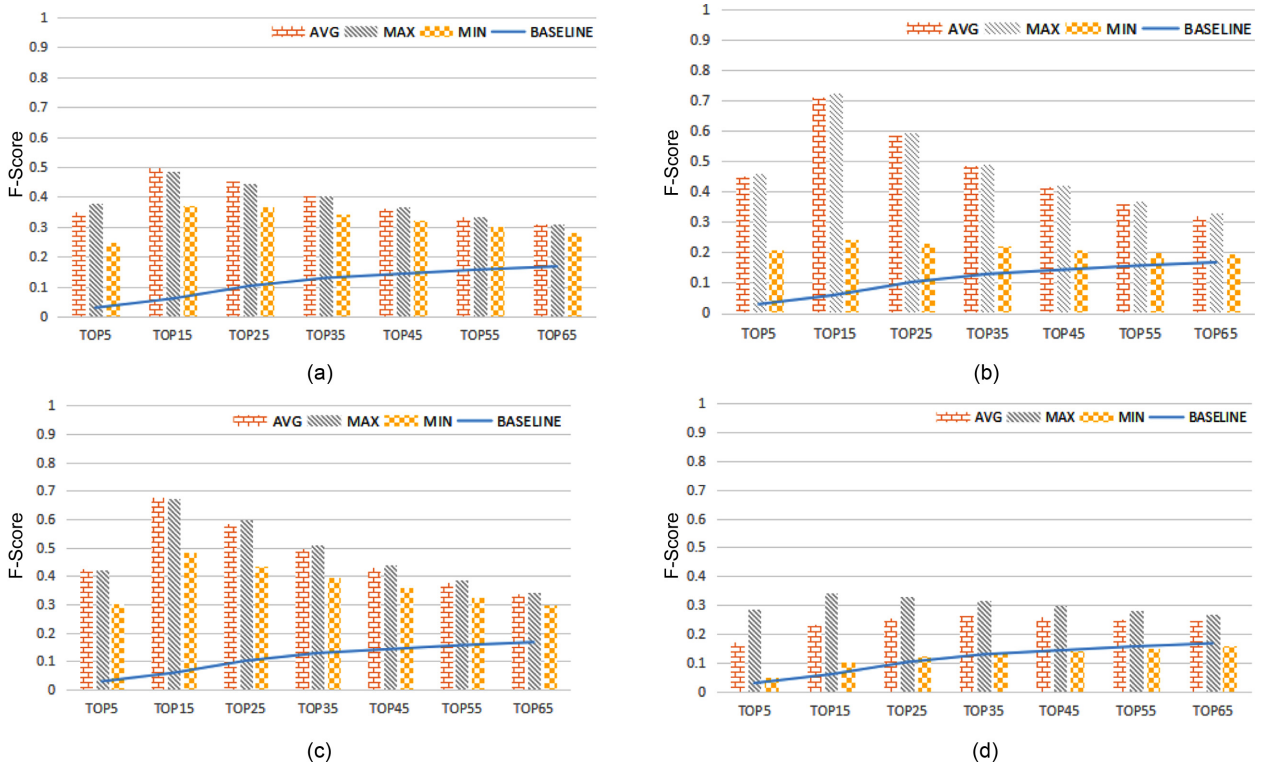

Fig. 5. Performances of different local features in face-based video retrieval: (a) Harris, (b) SIFT, (c) SURF, and (d) eigenfaces.

video fragments. We could take either the maximum (MAX), the minimum (MIN) or the average (AVG) similarity based on all possible pairs. Fig. 5 compares the performances of the different features and with different k values. MAX significantly outperformed AVG and MIN for the eigenfaces, and performed similarly to AVG for the Harris, SIFT, and SURF descriptors, which made it a better choice in general. Therefore, MAX was used to obtain the results reported in Table 2. Comparing across different k values shows that the relevant video fragments were usually ranked within the top 15 results.

The second important factor we considered was efficiency. Video retrieval involves both online and offline computation. Indexing existing videos can be carried out offline, whereas retrieving relevant videos for the query video needs to be performed online. Fig. 6 shows the average time needed for matching the query video against

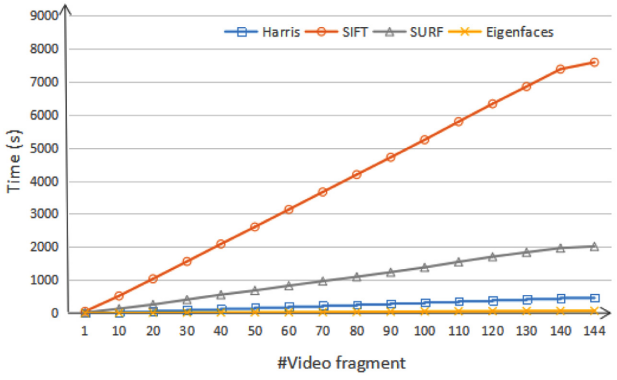

Fig. 6. Time needed for online retrieval.

the training set, which consists of 144 indexed video fragments. In general, eigenfaces were the fastest to

generate, taking almost negligible time compared to the other features. This is partly due to the reduced number of data dimensions. In contrast, for the SIFT and SURF descriptors, the search time grew linearly with the number of video fragments being matched. As shown in Fig. 1, SIFT and SURF both identified more interest points than the Harris operators, and the increased number of features also slowed down the matching process.

Combining the results shown in Table 2 and Fig. 6, we conclude that the SURF descriptors are a better feature for face-based video retrieval. They yielded good retrieval performance, yet their computation was reasonably fast. In contrast, although the SIFT descriptors showed better retrieval performance, they were computationally too expensive, especially for online systems. In the case of a strong requirement for rapid processing time, the Harris operators and eigenfaces can also be considered.

VI. CONCLUSION

Face-based video retrieval is an important application of intelligent video analysis. Constructing coherent and representative face profiles for videos has a crucial impact on the retrieval effectiveness. In this study, we analyzed four mainstream local features for describing face images: Harris corner operators, SIFT descriptors, SURF descriptors, and eigenfaces. Experimental results showed that static face image matching is still an effective approach for video retrieval, and that SURF was the best option in general cases. Our results provide valuable insight into the design and implementation of face-based video retrieval systems. In our future work, we plan to first integrate more recent features into our face-based video retrieval system, and then deploy the system in a distributed computing environment, such as on a Hadoop platform, since all reported work in this paper was completed on a standalone machine.

ACKNOWLEDGMENTS

This research is funded by Yangtze University (Grant No. JY2014032).

REFERENCES

1. P. Geetha and V. Narayanan, "A survey of content-based video retrieval," *Journal of Computer Science*, vol. 4, no. 6, pp. 474-486, 2008.
2. R. Belaroussi and M. Milgram, "A comparative study on face detection and tracking algorithms," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7158-7164, 2012.
3. Y. Chen, X. Li, A. Dick, and R. Hill, "Ranking consistency for image matching and object retrieval," *Pattern Recognition*, vol. 47, no. 3, pp. 1349-1360, 2014.
4. F. Hopfgartner, "Personalised video retrieval: application of implicit feedback and semantic user profiles," Ph.D. dissertation, University of Glasgow, UK, 2010.
5. S. Yu, L. Jiang, Z. Xu, Y. Yang, and A. G. Hauptmann, "Content-based video search over 1million videos with 1 core in 1 second," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR)*, Shanghai, China, 2015, pp. 419-426.
6. A. Mittal, "An overview of multimedia content-based retrieval strategies," *Informatica*, vol. 30, no. 3, pp. 347-356, 2006.
7. S. Memar, L. S. Affendey, N. Mustapha, S. C. Doraisamy, and M. Ektefa, "An integrated semantic-based approach in concept based video retrieval," *Multimedia Tools and Applications*, vol. 64, no. 1, pp. 77-95, 2013.
8. J. Joglekar and S. S. Gedam, "Area based image matching methods: a survey," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 1, pp. 130-136, 2012.
9. C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of 4th Alvey Vision Conference (AVC)*, Manchester, UK, 1988, pp. 147-151.
10. D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the 7th International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 1150-1157.
11. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
12. H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: speeded up robust features," in *Computer Vision: ECCV 2006*. Heidelberg: Springer, 2006, pp. 404-417.
13. Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004, pp. 506-513.
14. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.
15. L. Juan and O. Gwun, "A comparison of sift, PCA-SIFT and SURF," *International Journal of Image Processing*, vol. 3, no. 4, pp. 143-152, 2009.
16. M. H. Le, B. S. Woo, and K. H. Jo, "A comparison of sift and Harris Conner features for correspondence points matching," in *Proceedings of 2011 17th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, Ulsan, Korea, 2011, pp. 1-4.
17. O. Miksik and K. Mikolajczyk, "Evaluation of local detectors and descriptors for fast feature matching," in *Proceedings of 2012 21st International Conference on Pattern Recognition*, Tsukuba, Japan, 2012, pp. 2681-2684.
18. H. P. Morevec, "Towards automatic visual obstacle avoidance," in *Proceedings of International Joint of Conference on Artificial Intelligence*, Cambridge, MA, 1977, pp. 584-584.
19. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical*

Society of America A Optics and Image Science, vol. 4, no. 3, pp. 519-524, 1987.
20. M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.
21. A. Saudagar and H. Mohammed, "A comparative study of video splitting techniques," in *Progress in Systems Engineering*. Cham: Springer International Publishing, 2015, pp. 783-788.
22. P. Timse, P. Aggarwal, P. Sinha, and N. Vora, "Face recognition based door lock system using OpenCV and C# with remote access and security features," *International Journal of Engineering Research and Applications*, vol. 4, no. 4, pp. 52-57, 2014.

Juan Zhou

Juan Zhou received her B.S. degree from the Department of Computer Science and Technology of Yangtze University, China, in 2015. She is currently pursuing an M.S. degree at Yangtze University. Her research interests include computer vision, natural language processing, and data mining.

Lan Huang

Lan Huang received her Ph.D. degree from the Computer Science Department of Waikato University, New Zealand, in 2011. She is working as a lecturer at Yangtze University, in the Department of Computer Science, China. Her research interests include machine learning and text mining.