PAPER Special Section on Recent Advances in Multimedia Signal Processing Techniques and Applications

Efficiently Finding Individuals from Video Dataset

Pengyi HAO[†a)], *Nonmember and* Sei-ichiro KAMATA[†b)], *Member*

SUMMARY We are interested in retrieving video shots or videos containing particular people from a video dataset. Owing to the large variations in pose, illumination conditions, occlusions, hairstyles and facial expressions, face tracks have recently been researched in the fields of face recognition, face retrieval and name labeling from videos. However, when the number of face tracks is very large, conventional methods, which match all or some pairs of faces in face tracks, will not be effective. Therefore, in this paper, an efficient method for finding a given person from a video dataset is presented. In our study, in according to performing research on face tracks in a single video, we also consider how to organize all the faces in videos in a dataset and how to improve the search quality in the query process. Different videos may include the same person; thus, the management of individuals in different videos will be useful for their retrieval. The proposed method includes the following three points. (i) Face tracks of the same person appearing for a period in each video are first connected on the basis of scene information with a time constriction, then all the people in one video are organized by a proposed hierarchical clustering method. (ii) After obtaining the organizational structure of all the people in one video, the people are organized into an upper layer by affinity propagation. (iii) Finally, in the process of querying, a remeasuring method based on the index structure of videos is performed to improve the retrieval accuracy. We also build a video dataset that contains six types of videos: films, TV shows, educational videos, interviews, press conferences and domestic activities. The formation of face tracks in the six types of videos is first researched, then experiments are performed on this video dataset containing more than 1 million faces and 218,786 face tracks. The results show that the proposed approach has high search quality and a short search time.

key words: *face retrieval, face track, video dataset, scene clustering*

1. Introduction

We are interested in retrieving video shots or videos containing particular people from a video dataset. There are many applications of such a capability for example, given a long video, it would be interesting if the video could jump to the next scene containing the people of interest or if all the shots containing a particular family member could be found from the thousands of short video sequences captured using a typical modern digital camera, or if movies on a website containing an actor of interest could be searched for. The purpose of this retrieval system is as follows: given an imaged person, a face image or a very short video sequence as a query, it should retrieve all the videos and shots of the same person from a dataset. The key issue is the extraction and organization of face information in the video dataset.

Owing to the large variations in pose, illumination conditions, occlusions, hairstyles and facial expressions, robust face matching is a challenging problem. Fortunately, videos of people can take advantage of the abundance of frames that contain multiple examples of each person in a form that can easily and automatically be associated with using straightforward visual tracking [1]. In other words, face tracks are used instead of single faces to track people in video. For example, in Ref. [1], each face track was modeled as a histogram of facial part appearance, and then these histograms were matched within shots to find face sets. In Ref. [2] face tracks associated with names were used to label the appearances of characters in TV or film material, where the minimum distances between the descriptors in each unlabeled face track and in the exemplar tracks were used. In Ref. [3] an interactive feedback set was constructed by searching for tracks with a particular distance from one of the tracks in the query set. These methods exhibit better performances than those using single faces. However, the numbers of individuals and face tracks used in experiments were rather small. Therefore, the conventional approach of computing the minimum distance among all pairs of faces in two face tracks is not applicable.

To handle a large dataset consisting of hundreds of hours of videos, in Ref. [4], the problem was transformed from that of finding the most similar subset of faces to that of finding the densest component in a graph representing the similarities of faces. In Ref. [4], even though no face tracks were extracted and employed, good recall of retrieval for a news video dataset was achieved. Finding the faces associated with the name of the specified person was required initial step. Another approach for dealing with face retrieval from a large video dataset was proposed in Ref. [5], where the mean face from a subset of faces selected from the original face track (this method was called 'k-Faces' in this study) was used to compute the similarity with k faces selected from the face tracks in the dataset. Although good performance was exhibited for news videos in Ref. [5], if a huge number of face tracks are extracted, matching between face tracks will be ineffective. In addition, in the above studies, no results were given for the formation of face tracks from different types of videos.

In this work, we demonstrate that exploring the relationship among individuals from videos can increase the speed of retrieval. Because one person may appear in several scenes in a video, and different videos may contain the same

Manuscript received May 1, 2011.
Manuscript revised August 9, 2011.

[†]The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan
a) E-mail: haopy@toki.waseda.jp
b) E-mail: kam@waseda.jp

DOI: 10.1587/transinf.E95.D.1280

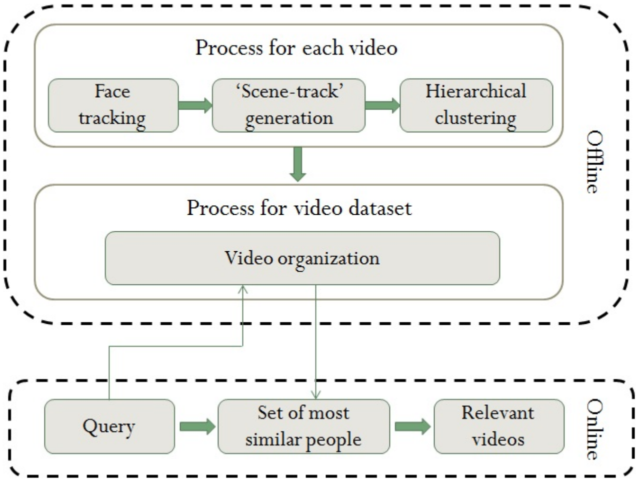


Fig. 1 Flow chart of the proposed approach.

person, the management of individuals in a single video and the organization among different videos will be useful. Therefore, in addition to performing research on face tracks in a single video, we also consider how to organize all the faces in videos in a dataset. Given a query, we describe our method of automatically searching for people appearing in videos from the following three viewpoints: (i) the organization of face tracks in each video; (ii) structuring the faces in all the videos in a dataset; (iii) the search process after submitting a query. Figure 1 outlines the structure of the proposed approach. From the flow chart it can be seen that the organization of videos in the dataset is an offline process, whereas the query process is executed online.

In our approach, scene information with a time constriction in each video is first employed to connect face tracks of the same person located in a given period, all the people in one video are organized by the proposed hierarchical clustering method. After obtaining the organizational structure of all the people in one video, the people are clustered into an upper layer. Here, we call a cluster a 'club'. In our clustering, a club contains at least one person, and a person can belong to more than one club. This process increases the speed of retrieval by shortening the time spent searching videos, and decreases the error of person localization when finding the correct clusters. Finally, in the query process, we first choose clubs that may contain the person we wish to find, then a set of individual candidates for the query is formed from the selected clubs. A remeasuring is also performed in this step to improve the retrieval accuracy.

For evaluation, we build a video dataset that contains six types of videos: films, TV shows, educational videos, interviews, press conferences and domestic activities. Experiments were performed on this video dataset containing 218,786 face tracks and more than 1 million faces. The formation of face tracks from these six types of videos is first researched, then experiments on finding people from these videos are performed. The results show that the proposed approach has high search quality with a short query time.

The rest of the paper is organized as follows. Section 2 describes the organization of face tracks in each video, Sect. 3 shows how to structure all the videos in the dataset so that the retrieval is immediate at run time, and the search process is presented in Sect. 4. Experimental results are reported in Sect. 5, and conclusions and future works are presented in Sect. 6.

2. Organizing People in Each Video

2.1 Face Track Extraction

A face detector is first executed on every frame of a video. Then faces detected in different frames of the same shot are associated into face tracks by tracking, such that each track corresponds to a single character. A typical episode of a TV series (here, "Friends" series 9, episode 1) contains around 23,100 detected faces and can generate a few thousand tracks. Determining the correspondence between faces within each shot reduces the volume of data to be processed and allows us to obtain additional examples of the facial appearance at no extra cost [1]. Consequently, subsequent work is expected to target face tracks rather than individual detection.

There are several approaches that can be used to group faces into face tracks. For example, Sivic et al. tracked facial regions and connected them to form groups [1]. This approach was accurate but had a high computational cost. To reduce the computational cost while maintaining accuracy, in the approach proposed by Everingham et al. [2], tracked points were obtained using the Kanade-Lucas-Tomasi (KLT) tracker. However, the points of interest generated in a certain frame in this method resulted in a few features on faces that were sensitive to changes in illumination, occlusions and false face detection, so that face tracks obtained by this method were sometimes fragmented. To avoid this problem, in Ref. [6] the feature tracker was seeded with features on every detected face, and the tracker was run from the first frame to the last frame of each shot, then symmetrically from the last frame to the first frame.

Here, we also employ tracked points to find evidence for connections between faces detected from an individual in a video. In contrast to Refs. [2] and [6], we recompute the points after every two frames. For a given pair of faces in different frames, the number of points that pass through both faces is measured, and if most of the points extracted in the previous face can be tracked in the current one, the current face is added to the track. If this ratio exceeds 0.6, the faces will be considered as candidates for connection. With no tuning of the parameters the method connects, for example 23,100 face detections into 1980 tracks.

2.2 'Scene-track' Generation

After face detection and tracking, faces are represented by face tracks, where each track is a sequence of faces of the same person, ideally in consecutive frames. Although face

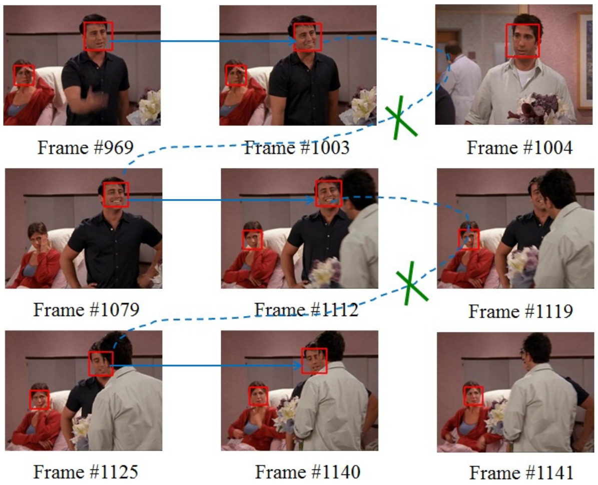


Fig. 2 Example of a disconnected track.

tracks are sufficient for some applications such as naming people in a video [2], they are not efficient for retrieving a particular person from large-scale videos. Faces of the same person may disappear in particular frames and then reappear later in the same scene. This may occur because of occlusion or turning. An example is given in Fig. 2, which shows a video sequence from frame 969 to frame 1140 of "Friends". The character Joey is tracked from frames 969 to 1003 but disappears between frames 1004 and 1078 because of the appearance of Ross. However Joey reappears in frame 1079 but is occluded by Ross from frame 1113. He reappears again in frame 1125 and is tracked until frame 1140. Although there are three different tracks for Joey from frames 969 to 1140, they represent the same person. Thus, a person may have more than one track in one scene.

To represent a person accurately and perform retrieval quickly, small tracks such as the three tracks in Fig. 2 should be connected and considered as a single track. Hence, the next important step is to connect face tracks from shots into scenes on the basis of scene information so that one scene has one or several tracks and each track represents one person. We call the connected tracks 'scene tracks'.

For each face track, we select the middle face of every three faces as the key face, and the corresponding frames are considered as key frames. Note that if a track only has one face or two faces, all of them are key faces. Then we define the distance between tracks T_i and T_j with a time restriction as follows:

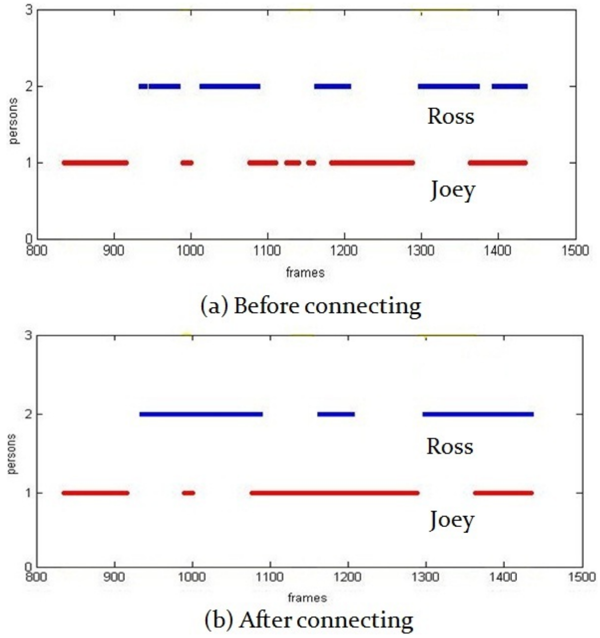
$$D(T_i, T_j) = \begin{cases} 1 - d(T_i, T_j), \text{if } |j - i| < t \\ \infty, \qquad\qquad otherwise. \end{cases} \qquad (1)$$

Here, $d(,)$ is the visual similarity between T_i and T_j, and t is the time threshold, where 2s is used in our experiments. The visual similarity $d(T_i, T_j)$ is defined as

$\frac{1}{m}\Sigma_{u=1}^{m} \min\{d(f_{i,u}, f_{j,1}), d(f_{i,u}, f_{j,2}), \ldots, d(f_{i,u}, f_{j,n})\}$,

where $d(f_{i,u}, f_{j,n})$ is the distance between frames in track T_i and track T_j, and m and n are the numbers of key frames in these two tracks, respectively.

Figure 3 gives an example of connecting tracks. Figure 3 (a) shows the distribution of the face tracks of Ross and Joey extracted from about 700 frames from episode 1 of series 9 of "Friends". Figure 3 (b) shows the result after connecting on the basis of visual similarity with a time restriction.



(a) Before connecting

(b) After connecting

Fig. 3 Example of connecting tracks.

2.3 Organizing People

Given a collection of scene tracks, we now wish to merge the scene tracks that contain the same person. Because one person can appear several times in a video, our objective is that after organizing these scene tracks we know how many people are in this video and which person is represented by which scene tracks. Then all the scene tracks of the searched person can be retrieved as a whole. Agglomerative clustering appears to be suitable for performing this task. It has also been widely used in previous studies [1], [6]–[9]. In Refs. [1], [7] and [6], agglomerative clustering was applied as a complementary method of face tracking to obtain better face groups. In Refs. [8] and [9], agglomerative clustering was applied to group face tracks into as homogenous clusters as possible. However, conventional agglomerative clustering, which merges data until the distance matrix is reduced to a single element, cannot satisfy our requirements.

Actually, although a scene track is superior to a face track for obtaining examples of a person's face, many scene tracks of the same person still exist nearby owing to occlusion, movement of the face away from the camera, changes in lighting, and so forth. Therefore, hierarchical clustering is used here. After processing by hierarchical clustering, we obtain some groups. We call each group a 'person'. Each person has several scene tracks.

Figure 4 illustrates processing by our hierarchical clustering method. In this example, there are four people, de-

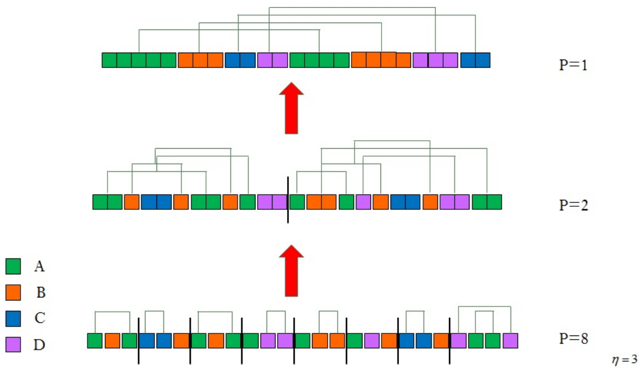


Fig. 4 Example of organizing people in one video.

noted by A, B, C, and D. At the beginning, the scene tracks of the four people are arranged in the bottom layer. The middle layer shows their arrangement after the first phase of clustering. The top layer represents the final clustering. The parameters displayed in this figure will be explained in the following algorithm.

A scene track, denoted as ST, has S face tracks. For each face track, a representative face is computed by averaging out all the feature vectors of its key faces. Then the scene track is represented as a set of these S vectors. Let Ψ denote the set of scene tracks in a video. The hierarchical clustering is performed by the following steps:

1. Ψ is partitioned into p parts.

2. For each part, each ST is initially a single cluster, then an iteration is performed by the following two steps:

(a) For each cluster P_x, the distances to each of the current clusters P_y are calculated. The closest cluster P_z is obtained by

$$D(P_x, P_y) = \min_{ST\in P_x, ST'\in P_y} d(ST, ST'), \tag{2}$$

and the distance between scene track ST in P_x and scene track ST' in P_y is given by

$$d(ST, ST') = \frac{1}{|ST|\cdot|ST'|}\Sigma_i^{|ST|}\Sigma_j^{|ST'|}\frac{(s_{ij}-s'_{ij})^2}{s_{ij}+s'_{ij}}. \tag{3}$$

(b) Clusters P_x and P_z are merged into a new cluster, labeled P^*.

This process is terminated when the number of clusters in this part is equal to ω. Here, ω is the number of scene tracks whose face tracks are larger than a minimum value, which is between the average number of face tracks and the middle number of face tracks in the set of scene tracks.

3. After applying the clustering process to all parts, a new track set is formed. If $p = 1$, the algorithm is stopped, otherwise, the new set is divided into $max\{[\frac{p}{\eta}], 1\}$ parts, p is replaced by $max\{[\frac{p}{\eta}], 1\}$, then steps 1 and 2 are repeated. We set $p = 100$ and $\eta = 5$ in our experiments.

3. Organizing Videos of Dataset

For a large video dataset, it is desirable for retrieval systems to find the videos containing faces of interest effectively. However, the numbers of individuals and videos used in most previous studies were rather small, for example, in Refs. [1], [2], [6], [7], [10]–[12], only one or two films or several TV shows were used. Therefore, scalability was not taken into account. Although in Ref. [5] 370 hours of news videos were used, it was still necessary to perform a linear search on a high-dimensional dataset, thus, the matching time was unacceptable.

Our goal is to minimize the response time to a user's query, which is a major challenge since the exhaustive and redundant computation of similarities is required. Cluster-based indexing approaches, which were originally designed for images [13], provided us with the inspiration to solve this problem. If we can efficiently organize people in each video of the dataset into an upper-layer structure, such as cluster-tree [13], the response time can be reduced. However, the number of types of individuals in the dataset is unknown, meaning that common clustering algorithms are not suitable. Fortunately, there is a clustering method named affinity propagation (AP) [14], which is different from conventional methods, on that it is unnecessary to specify and fix the number of clusters. Moreover, it is not sensitive to the initial center points.

Here, each scene track of each person is a data point. The input of AP is a list of numerical similarities between pairs of scene tracks. By viewing each data point as a node in a network, current nodes exchange messages with all other nodes along the edges of the network until a good set of examples and corresponding clusters emerge. We call each final cluster a 'club', which is defined as a triad Ξ : $(ClubID, Center, Person)$. $ClubID$ is the indexing number of the club; $Center$ is the feature of the example, denoted by CF, and the example denotes the cluster center of this club; $Person$ is a quadruplet of data $(PersonID, VideoID, Tracknumber, STs)$, $PersonID$ and $VideoID$ are the indices of person and video, respectively, $Tracknumber$ is the number of scene tracks for this person, and STs denote the scene tracks containing the person in this video. The club is constructed by connecting each scene track to the example that best represents it.

There are two types of messages exchanged between data points, one is "responsibility" $\gamma(i, k)$, which is sent from data point i to candidate example point k, and reflects the accumulated evidence for how well suited point k is as the example for point i, taking into account other potential exemplars for point i; the other is "availability" $\alpha(i, k)$, which is sent from candidate example point k to point i, and reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its example, taking into account the support from other points for which point k could be an example. After a fixed number of iterations, for person i, the person k that maximizes $\alpha(i, k) + \gamma(i, k)$ is identified as its example.

After the AP process, the videos in the dataset are organized as shown in Fig. 5. From this figure, it can be seen clearly that an upper layer is built for videos that can avoid

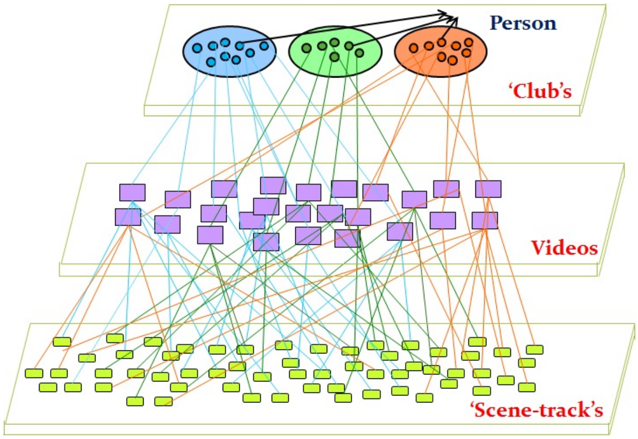


Fig. 5 Organization structure of video dataset.

a linear search of the dataset. Hence, the retrieval speed is increased. In addition, because scene tracks are used to perform clustering, the same person may be clustered into several clubs. That is to say one club can have more than one person, and one person can be a member of several clubs. This feature benefits our queries resulting in an increase in search quality. Section 4 gives details of the querying process, and the search results using the structure shown in Fig. 5 are given in Sect. 5.

4. Querying from Video Dataset

Images of a person's face contain variations induced by changes in pose, expression, illumination and so forth. Therefore, in the clustering described in Sect. 3, a person can be clustered into several clubs. Thus, given a query, several possible clubs that may contain the person are first selected. Then the most similar people from such clubs should be found for the query. Simply measuring the distance between the query and people in clubs is not sufficiently powerful to find the closest match owing to the organization of people by hierarchical clustering. Therefore, we first obtain the candidate people who are similar to the person in the query from the selected clubs. Then the most similar people are obtained by considering the variations among the candidate people. Finally, the videos containing these most similar people are listed to answer the query. Figure 6 shows this process clearly.

To find the most similar people, we remeasure each candidate person based on the average distance to referenced people. A candidate person will be added to the relevant person set when the person is similar to both the person in the query and the people selected in the previous iteration.

Assuming that there are C clubs, CF denotes the features of each club, PF, which is the average of the STs, denotes the feature of the person, cd denotes the candidate person, R is the relevant set and is initialized by PF, $|R|$ is the number of elements in R, and q is a query. The algorithm for obtaining the set of people for the query is as follows:

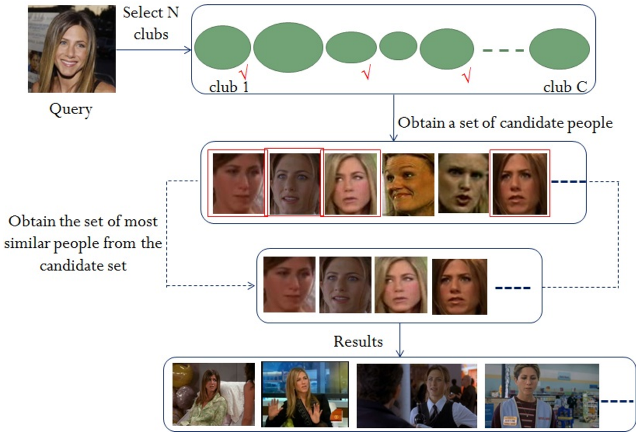

Select N clubs
club 1
√
√
√
club C
Query
Obtain a set of candidate people
Obtain the set of most similar people from the candidate set
Results

Fig. 6 Example of querying from video dataset.

Algorithm 1 Algorithm for obtaining set of people for query

```
for $i = 1$ to $C$ do
    $\alpha_i = d(q, CF_i)$;
end for
Select the top $N$ clubs with smallest $\alpha$;
for $i = 1$ to $N$ do
    $\beta_i = d(q, PF_i)$;
end for
Select the top $M$ 'person's with smallest $\beta$; Here $M$ is not larger than $N$.
$R = \{cd_1\}$;
repeat
    for $i = 1$ to $M$ do
        $\Gamma_i = d(q, PF_i) + \frac{1}{|R|} \sum_{j=1}^{|R|} d(R_j, cd_i)$;
    end for
    if $cd_k, k \in [1, M]$ is $\min\{\Gamma_1, \Gamma_2, \ldots, \Gamma_M\}$ then
        if $|R| = 1$ then
            $R = \{cd_k\}$;
        else
            if $cd_k \in R$ then
                break;
            else
                $R \leftarrow cd_k$;
            end if
        end if
    end if
until There is no change in $R$
```

The most similar people selected by the above approach are not only similar to the person in the query but also similar to each other. Then, the videos containing this person will be returned and the periods containing this person will be displayed on the basis of the records of scene tracks.

5. Experiments

We built a video dataset containing six types of videos: two long films ("Along Came Polly" and "The Good Girl"), 20 TV shows (2 episodes per series of "Friends"), 20 educational videos from NASA, 100 interviews, 100 press conferences and 100 domestic activities downloaded from the YouTube website. 218,786 face tracks with about one million faces were extracted. Although the number of faces is

Table 1 Statistics of the six types of videos.

Video Type	Length (min.)	# Face, FD(#/sec.)	# Face track, FTD(#/sec.)	# Scene track, STR(#ST/#FT)
Film	180	139896, 12.953	17959, 1.662	5760, 0.320
TV Show	440	618548, 23.429	57790, 2.189	21120, 0.365
Education	480	309111, 10.644	20657, 0.711	8831, 0.427
Interview	347	349710, 16.796	29603, 1.421	7807, 0.264
Press Conference	166	70372, 7.065	8311, 0.834	2752, 0.331
Domestic Activity	149	89783, 10.042	9667, 1.081	3939, 0.407

Table 2 Comparison of search quality and query time.

Method			mAP	Query Time (s)
k-Faces [5]			46.39	116.7
Sivic's method [1]			49.87	84.5
Proposed approach	'club'	remeasure		
	no	no	49.25	21.9
	yes	no	51.06	8.4
	yes	yes	53.43	9.1

not larger than that in Ref. [5], the numbers of face tracks is much larger because it is easier to extract consecutive face tracks from the news videos used in Ref. [5] than it is from films, TV shows, interviews and domestic activities. The number of face tracks in our dataset is also larger than those in previous studies [1], [2], [6], [7], [10]–[12]. After scene track formation, these were 50,209 tracks. Then, ten people such as Jennifer Aniston were annotated from 2455 face tracks. Details of the six types of videos are given in Table 1.

In Table 1, FD denotes the face density per second, FTD is the density of face tracks per second, and STR denotes the rate of scene tracks formed from face tracks. From the statistics in Table 1, we can see that TV shows have the highest density of faces and face tracks. This is because many scenes that appeared in "Friends" were taken at a coffee shop, where there are many people unimportant to the plot. It is possible that a few such faces will form face tracks and split the face tracks of the main actors in one shot into several tracks. Although the number of faces extracted from interview videos per second is larger than those of the remaining four types, face tracks obtained from this type of video are fewer owing to the factors such as exposure, flash lamp, occlusion and so forth. Among the six types of videos, educational videos have the smallest density of face tracks.

For the formation rate of scene tracks, we conclude that smaller is better for computation and retrieval. Since the interviewer and interviewee play a very important roles in the interview videos, most of the faces extracted in this type of video are the interviewer and interviewee. Moreover, the backgrounds in most of the frames do not change; thus, face tracks can be more easily grouped into scene tracks. Hence, interviews have the smallest STR. In contrast, for educational videos, it is difficult to obtain scene tracks because detected faces are not only few but also separated by very far.

We use the mean average precision (mAP), commonly used in content-based image retrieval, to evaluate the performance of various video retrieval methods. For each query we obtain the average precision computed as the area under the precision-recall curve. Precision is the number of positive retrieved videos and scene tracks relative to the total number of videos and tracks retrieved. Recall is the number of positive retrieved videos and tracks relative to the total number of positives in the corpus. mAP is thus the mean for a set of queries.

Table 2 gives the mAPs and query times of three methods: k-Faces, proposed in Ref. [5], using k faces selected from the face track to compute the mean vector and then perform matching; the method of Sivic et al. [1], which models each face track as a histogram of facial part appearance, and the proposed approach. Here, to ensure a fair comparison, we use the same parameter value (k = 5) as that in Ref. [5]. In Table 2, we also give the performance of video organization by constructing an upper layer with clubs and remeasuring based on the index structure of the video dataset.

As shown in Table 2, using k-Faces to represent a face track gives the worst result in terms of mAP, although it has better performance for the news videos used in Ref. [5]. The reason for this is that the six types of videos in our dataset have large variations, which causes the method to fail when the k faces of two face tracks have different poses, illumination conditions and so forth. In contrast, using a histogram of facial parts to represent face tracks performs slightly better. For the proposed method, when only scene tracks are used for searching, it has slightly higher precision than k-Faces because multiple examples are considered, but it has lower performance than the histogram representation. The reason for this is that in the method in Ref. [1] a large number of facial parts are used to represent the eyes and mouth, which cover changes in expression naturally. However, after the video organization and remeasuring, the precision of our approach is significantly improved. The individuals detected in the videos are organized, which reduces the error of the search. In addition, remeasuring prevents the selection of videos and tracks far from the center of the current reference videos and finally obtains a set of videos and tracks that are not only similar to the query but also similar to each other.

In terms of the query time, it is clear from Table 2 that the proposed approach is faster than the other approaches because the numbers of tracks is reduced from 218,786 to 50,209. Using the upper layer structure, the speed of the proposed approach is further improved.

Figure 7 shows the query time plotted against the number of face tracks in the dataset. As expected, the number of face tracks increases with the number of videos. In this case, conventional methods, which find desired tracks by match-

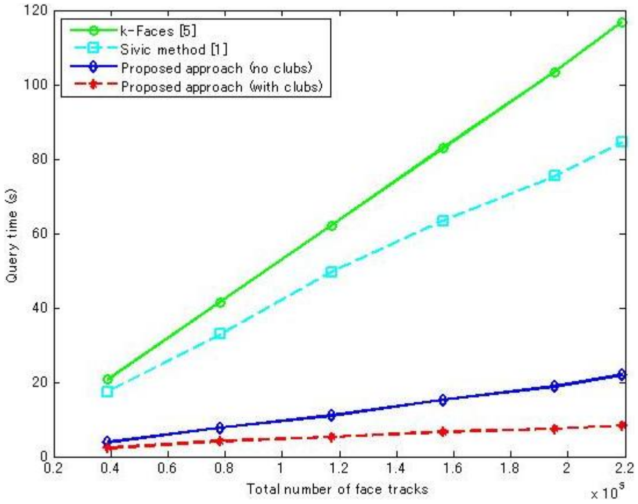


Fig. 7 Comparison of query time for different methods plotted against number of face tracks extracted from video dataset.

ing the distance between faces in two face tracks, will fail. In our experiments, when 195,345 face tracks were obtained from about 25 hours of videos, the query time taken by k-Faces increases almost five fold compared with the time taken to search about 5 hours of videos. When searching from 218,786 face tracks, the method in Ref. [1] requires more than 84 seconds to search through about 25 hours of videos, which is about 4.8 times the time required for 5 hours of videos. Using the proposed method it takes about 8.4 seconds to obtain the relevant tracks from 218,786 face tracks. The time is reduced by about 86.1% and 90% compared with those for k-Faces and the histogram distribution method, respectively.

The offline process takes a long time to perform face detection, face tracking, video feature extraction and video dataset indexing. It took about 2.5 days on a 2.66 GHz CPU with 8 GB memory to process 340 videos. However, it is easy to add new videos. First, person information is obtained using the process for a single video shown in Fig. 1, then the new video is assigned to the most suitable club by calculating the distance from the clubs in the dataset. After that, the selected club is updated to include the person and scene track information of the new video. When deleting videos, by detecting the video ID, scene tracks and the person information in clubs will also be deleted.

6. Conclusions and Future Work

In this study, we proposed an approach for retrieving video shots or videos containing particular people from a video dataset. Since conventional methods, which match all or some pairs of faces in face tracks, will fail when the number of face tracks is very large, we presented an efficient method for finding a given person from a video dataset in this study. In addition to research on face track extraction from videos, we also investigated how to organize all the faces in the videos in a dataset and how to improve the search quality in the query process. First, face tracks of the same person located in a period in each video were connected on the basis of scene information with a time constriction, then people in a single video were organized by the proposed hierarchical clustering method. After obtaining the organizational structure of all people in each video, people were clustered into an upper layer by affinity propagation. Finally, in the process of querying, remeasuring based on the index structure of videos was performed to improve the retrieval accuracy.

We also built a video dataset containing six types of videos: films, TV shows, educational videos, interviews, press conferences and domestic activities. Experiments were performed on this video dataset. 218,786 face tracks were extracted from this dataset, which contained more than 1 million faces. Compared with two other well-known methods, the results showed that the proposed approach markedly improved the search quality and shortened the query time. In addition, we also compared the formation of face tracks in different types of videos. The method of organizing the videos and the remeasuring method were also evaluated.

Although the mean average precision obtained by the proposed approach is higher than that of the conventional approaches, it is still low for practical application owing to large variations in poses, illumination conditions, occlusions, hairstyles and facial expressions. The query time also cannot yet satisfy the requirements of users. Therefore, in future, the proposed approach will be improved and more effective face representation methods will also be investigated.

Acknowledgments

We would like to thank all the people for providing their data for testing and all the observers for contributing to this study.

References

[1] J. Sivic, M. Everingham, and A. Zisserman, "Person spotting: Video shot retrieval for face sets," ACM International Conference on Image and Video Retrieval, pp.226–236, 2005.

[2] M. Everingham, J. Sivic, and A. Zisserman, ""Hello, My name is... Buffy" - Automatic naming of characters in TV video," Proc. British Machine Vision Conference, pp.899–908, 2006.

[3] M. Fischer, H.K. Ekenel, and R. Stiefelhagen, "Interactive person re-identification in TV series," International Workshop on Content-Based Multimedia Indexing, pp.1–6, June 2010.

[4] D. Ozkan and P. Duygulu, "Interesting faces: A graph-based approach for finding people in news," Pattern Recognit., vol.43, no.5, pp.1717–1735, May 2010.

[5] T. Nguyen, T. Ngo, D.-D. Le, S. Satoh, B. Le, and D. Duong, "An efficient method for face retrieval from large video datasets," ACM International Conference on Image and Video Retrieval, July 2010.

[6] J. Sivic, M. Everingham, and A. Zisserman, ""Who are you?" learning person specific classifiers from video," IEEE Conference on Computer Vision and Pattern Recognition, pp.1145–1152, 2009.

[7] D. Ramanan, S. Baker, and S. Kakade, "Leveraging archival video for building face datasets," International Conference on Computer Vision, pp.1–8, Oct. 2007.

[8] M. Zhao, J. Yagnik, H. Adam, and D. Bau, "Large scale learning and recognition of faces in web videos," IEEE International Conference on Automatic Face & Gesture Recognition, pp.1–7, Sept. 2008.

[9] A. Holub, P. Moreels, A. Islam, A. Makhanov, and R. Yang, "Towards unlocking web video: Automatic people tracking and clustering," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp.47–54, 2009.

[10] A. Anjulan and N. Canagarajah, "Object based video retrieval with local region tracking," Signal Processing: Image Commun., vol.22, no.7, pp.607–621, Aug. 2007.

[11] D.-D. Le, S. Satoh, M. Houle, and D. Nguyen, "Finding important people in large news video databases using multimodal and clustering analysis," IEEE International Workshop on Multimedia Databases and Data Management, pp.127–136, 2007.

[12] X.F. Ren, "Finding people in archive films through tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, June 2008.

[13] D.T. Yu and A.D. Zhang, "ClusterTree: Integration of cluster representation and nearest-neighbor search for large data sets with high dimensions," IEEE Trans. Knowl. Data Eng., vol.15, no.5, pp.1316–1337, 2003.

[14] B. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol.315, no.16, pp.972–977, Feb. 2007.

Pengyi Hao is a Ph.D. student of the Graduate School of Information, Production and Systems, Waseda University, Japan. She received her first M.E. degree in computer application and technology from Shanghai University, China, in March 2010, and her second M.E. degree in computer science from Waseda University, Japan, in July 2010. Her current research interests are multimedia retrieval and pattern recognition.

Sei-ichiro Kamata received his M.S. degree in computer science from Kyushu University, Japan, in 1985, and his doctor of engineering degree from the department of Computer Science, Kyushu Institute of Technology, Japan, in 1995. From 1985 to 1988, he was with NEC Ltd., Kawasaki, Japan. In 1988, he joined the faculty at Kyushu Institute of Technology. From 1996 to 2001, he was an associate professor in the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been a professor at Graduate School of Information, Production and Systems, Waseda University. In 1990 and 1994, he was a visiting researcher at the University of Maine, Orono. His research interests are image processing, pattern recognition, image compression, remotely sensed image analysis, space-filling curves and fractals. Prof. Kamata is a member of the IEEE, and the ITE in Japan.