NEWS VIDEO ANALYSIS BASED ON IDENTICAL SHOT DETECTION

Shin'ichi Satoh

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

ABSTRACT

The paper presents a method to detect identical video segments from video footages in broadcasted video archives, and its application to news video analysis. We define identical video segments as distinctively similar video segments. After giving the definition of identical images, efficient algorithm to detect identical shots in videos is shown. Its effectiveness is shown by applying the algorithm to news video analysis. As the first experiment, the algorithm successfully locates identical shots derived from the same video materials shared by semantically related topics. This obviously is useful for topic tracking by a visual cue. Then the second experiment shows that aggregating detected identical shots can tell news video structure, i.e., can locate filler shots such as opening shots, anchor shots, weather forecast, etc. The result is quite helpful for news video parsing. The experiments reveal that, although the method does not use any news-specific a priori knowledge, it could be used as a powerful tool to explore useful knowledge from large-scale news video archives.

1. INTRODUCTION

Technical innovation by high-speed broadcasting and networking, digital video transmission, huge storage devices, etc., raises a demand for large-scale video database and its content-based access. These technologies are expected to enable "mining" knowledge from large-scale video archives that meets users' need. As a basic component of content-based video access, researchers tend to use feature-based similarity search for images and videos. However, several papers have recently been published which state that similarity search is getting more noisy and useless as the image/video archives are getting larger, but instead, searching the "identical" image/video is becoming useful [1–3]. We define identical video segments as distinctively similar video segments.

In this paper, we designed identical video segment detection method for broadcast video archives. After giving the definition of identical images based on image intersection, efficient algorithm to detect identical segments (shots particularly) in videos is shown combining video fragment decomposition, color histogram intersection calculation, and image intersection calculation. We then evaluate our method with news video archives. Two types of news video analysis are conducted. As the first experiment, the algorithm is employed to locate identical shots derived from the same video materials shared by semantically related topics. This obviously is useful for topic tracking by a visual cue. Then as the second experiment, filler shots extraction such as opening shots, anchor shots, weather forecast, etc., is realized by aggregating detected identical shots. The result is quite helpful for news video parsing. Since the identical shot detection does not use any news-specific a priori knowledge, it can be used as a robust and powerful tool for news video analysis.

2. IDENTICAL VIDEO SEGMENT

2.1. Definition of Image Identifiability

Let an image I be composed of $n =$ width × height pixels (352 × 240 in our case), x_i, $i = 1, \dots, n$, ordered in particular order, e.g., raster scan. Here we define the intersection between two images I^1 and I^2 as follows:

$$I_I(I^1, I^2) = \frac{1}{n}||\{(x_i^1, x_i^2)|d_p(x_i^1, x_i^2) < \theta_p\}|| \quad (1)$$

where $d_p(x_i^1, x_i^2)$ is the distance between corresponding pixels of two images, and θ_p is the uppermost allowed distance between identifiable pixels. The image intersection is intended to provide the covered area where corresponding pixels between two images are identical. We regard two images I^1 and I^2 are identical if $I_I(I^1, I^2) > \theta_c$. We use $\theta_c = 60\%$ for our experiments. Let each pixel be represented by color values $x = [r\ g\ b]^t$. The pixel distance d_p can be defined by the Euclidean distance $|x^1 - x^2|$, however, due to the nonlinear nature of human sensation, this may cause unintended results. To compensate for this, we introduce nonlinear conversion function γ in d_p as follows:

$$d_p(x^1, x^2) = \left| \begin{bmatrix} \gamma(r^1) - \gamma(r^2) \\ \gamma(g^1) - \gamma(g^2) \\ \gamma(b^1) - \gamma(b^2) \end{bmatrix} \right|. \quad (2)$$

Several nonlinear functions can be used for the conversion function, such as log, square root, cubic root, etc. We use square root function for our experiments.

2.2. Color Histogram Intersection

To detect identical images in videos by using image intersection, pixel-wise comparison is required for each pair of frames. However, this process is unrealistic for large scale news video archives in terms of the size of the data to be stored and the amount of computation. In order to save the database size as well as computation time, much smaller representation is preferred. We use color histograms for this purpose. The color space (for instance the RGB space) is first decomposed into m regions. Then the histogram of an image I is calculated by counting pixels in the image falling into each region: $H = [h_1\ h_2\ \dots\ h_m]$, where h_i is the number of pixels in the i-th region. To evaluate similarity between histograms H^1 and H^2 of images I^1 and I^2, we use the normalized histogram intersection:

$$I_H(H^1, H^2) = \sum_i \min\left(\frac{h_i^1}{n}, \frac{h_i^2}{n}\right). \quad (3)$$

If corresponding pixels of two images are identifiable, they likely fall into the same region in the decomposed color space. Thus the

0-7803-7304-9/02/$17.00 C2002 IEEE

following inequality holds:

$$
\begin{aligned}
& nI_H(H^1, H^2) \\
& \quad > \quad \|\{(x_j^1, x_j^2) | d_p(x_j^1, x_j^2) < \theta_p\}\| - O(\theta_p). \qquad (4)
\end{aligned}
$$

The term $O(\theta_p)$ is due to the boundary condition between histogram bins. The histogram intersection gives (approximate) upper bounds for image intersection;

$$
\begin{aligned}
I_I(I^1, I^2) &= \frac{1}{n}\|\{(x_j^1, x_j^2) | d_p(x_j^1, x_j^2) < \theta_p\}\| && (5) \\
&< I_H(H^1, H^2) + O(\theta_P). && (6)
\end{aligned}
$$

Therefore $I_H > \theta_c$ can be used as a dirty filter for the identical image detection. This is theoretically true, however, this might be too strict bounds (i.e., $I_H \gg I_I$), mostly due to the fact that the histogram intersection gives the fraction of the image region having identical pixels in terms of quantized color, with disregarding pixel locations. Thus we will use a larger threshold, e.g., $\theta'_c = 70 \sim 80\%$ to test I_H, i.e., $I_H > \theta'_c$. The result can easily be extended to subregion histograms. We use 2×2 subregion histograms where $4 \times 4 \times 4$ division in the RGB space for our experiments. In order to reflect the effect of nonlinear conversion, division in the color space is chosen so that the division is even in the converted domain. We chose this configuration among others, e.g., with different number of division, division in hue dimension in the HSV color space, etc., because our experiments exhibited that it achieved better results in approximating image intersection by histogram intersection, i.e., I_H is much closer to I_I, while retaining reasonable number of dimension. By converting images ($352 \times 240 \times 3$ values) into histograms ($2 \times 2 \times 4 \times 4 \times 4$ values), the data size for each frame as well as intersection computation is reduced to 1/1,000.

2.3. Video Fragment Decomposition

Based on the definition of identical images, we then design algorithm to detect identical video segments composed of consecutive frames. Since successive frames in videos tend to be similar, identical video segment detection can be accelerated by taking advantage of this property. To do this, a video is decomposed into video fragments, each of which includes contiguous frames, so that distances between any pair of frames in a fragment do not exceed a threshold. Assume that contiguous m frames $f_i, i = 1, \cdots, m$ compose a video fragment. In order to decide whether the next frame f_{m+1} can be included in the fragment, we need to test the inequality $\max_{i \in [1,m]} I_I(f_i, f_{m+1}) < \theta_c$, which requires m-times distance evaluation. Instead, to further save computations, we will use the test $I_I(f_1, f_{m+1}) < \theta_c$, which needs only one distance evaluation at the cost of approximation. The algorithm to decompose a video footage including n frames ($f_i, i \in [1, n]$) into fragments is as follows:

```
current = $f_1$;
for ($i$=2; $i \le n$; $i$++) {
    if ($I_I(current, f_i) > \theta_c$)
        continue;
    current = $f_i$;
    output $i$ as the start frame of a video fragment;
}
```

This algorithm requires costly image intersection computations, but only once for each frame at registration.

Given video fragments, we then design an identical fragment detection algorithm. We define that two fragments are identical if any pair of frames of the fragments are identical (i.e., $I_H > \theta_f$). Basic idea is to first dirty-filter hopeless fragments out by comparing color histograms, then to compare histograms frame by frame. Assume that we are given two fragments F^1, F^2 from video footages, whose first frames are f_0^1 and f_0^2 respectively. The fragments are identical if the following two inequalities hold:

$$
\begin{aligned}
& I_H(f_0^1, f_0^2) > 1 - (1 - \theta_f) - 2(1 - \theta'_c) && (7) \\
& \max(I_H(f^1, f^2) | f^1 \in F^1, f^2 \in F^2) > \theta_f. && (8)
\end{aligned}
$$

Eq. (8) requires two fold iteration due to the max operation, but by evaluating Eq. (7) first, unnecessary search will be pruned based on the triangular inequality of the Manhattan distances of histograms. To save computations, only color histogram intersections are evaluated at this stage. We set $\theta_f = \theta'_c$ for our experiments. In the actual implementation, the excessive sum optimization is employed to accelerate evaluation of inequalities.

2.4. Identical Shot Detection

Finally, aforementioned techniques are combined into the integrated identical shot detection method. We define that two shots are identical if any pair of fragments of the shots are identical. Our goal is to locate pairs of identical shots in given two videos. In doing this, we first need to decompose videos into shots by applying scene change detection. Any scene change detection methods can be used, however, in our experiments, accumulation of histogram intersection values for a certain duration (t_s frames) is used, i.e., $\sum_{t=t_0}^{t_0+t_s} I_H(f_t, f_{t+1})$, with hysteresis using two threshold values. When accumulation value goes below the lower threshold, that point is regarded as the start of scene change. On the other hand, the end of scene change is detected when the value goes beyond the upper threshold. By accumulating for the duration, it can also detect gradual scene changes, while at the same time, noisy result is avoided by hysteresis. We use $t_s = 10$ frames for our experiments.

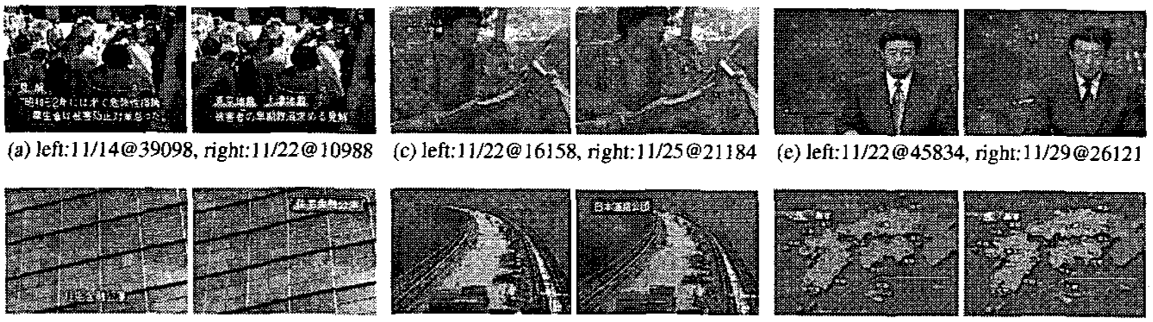
By combining shot decomposition and fragment decomposition, we obtain shot lists SL^1 and SL^2, and each shot $\{S_j^i\}$ of a shot list SL^i is composed of a list of fragments $S_j^i = \{F_{j,k}^i, k = 1, \ldots, n_j^i\}$, from two videos respectively. Given these lists, identical shot detection is realized as follows:

```
foreach $S^1$ in $SL^1$ {
  foreach $S^2$ in $SL^2$ {
    if ($\max(I_H(F^1, F^2) | F^1 \in S^1, F^2 \in S^2) > \theta_f$) {
      $(F^1, F^2) = \mathrm{argmax}(I_S(F^1, F^2) | F^1 \in S^1, F^2 \in S^2)$;
      $(f^1, f^2) = \mathrm{argmax}(I_H(f^1, f^2) | f^1 \in F^1, f^2 \in F^2)$;
(*)   if ($I_I(f^1, f^2) > \theta_c$) {
        output $(S^1, S^2)$ as identical shots;
} } } }
```

where $I_H(F^1, F^2) = \max(I_H(f^1, f^2) | f^1 \in F^1, f^2 \in F^2)$. In evaluating $I_H(f^1, f^2)$, the pruning technique described in the previous section (Eq. (7)) is used. The line (*) finally verifies identifiability between shots by using image intersection, which is computationally costly but accurate.

3. RELATED TOPIC SHOT DETECTION

The proposed identical shot detection method was implemented and tested with actual news videos. Currently we are constructing news video archive system, which collects news videos from



(a) left:11/14@39098, right:11/22@10988 (c) left:11/22@16158, right:11/25@21184 (e) left:11/22@45834, right:11/29@26121

(b) left:11/22@8859, right:11/23@29089 (d) left:11/16@45339, right:11/22@2973 (f) left:11/22@58434, right:11/29@58564

Figure 1: Detected Identical Shots (the number preceded by @ is the frame number)

NHK (the largest broadcast station in Japan) everyday, 30 minutes (or around, depending on the program) per a day. We used 64 news videos broadcasted from November 2001 to early January 2002. The method was implemented in C and runs on Pentium IV 2GHz PC. For each 30 minute program, it takes about 30 minutes to compute color histograms and to decompose it into shots and fragments. This process is required only once for each footage. Then to detect identical shots between two 30 minute videos, the system takes about 40 minutes. This process needs to be repeated for each combination of video pairs. For instance, adding today's news to archives consisting 1000 footages, 1000 repetitions are required. Thus the latter stage should further be accelerated.

In news videos, each footage is composed of several topics as pointed out by many researchers in news video parsing (e.g., [4]). Once large-scale news video archives are realized, topic relation between videos is getting important. For example, especially for "big news," day by day change of the topic is reported everyday as related topics. Detecting such relation will be useful for content-based browsing, automated comprehensive commentary generation of a specific topic, or summarizing one week news by removing redundancy. This task is recognized as topic tracking, and studied by text and speech analysis researchers so far [5]. However, their results still suffer from errors. On the other hand, related topics in news videos sometimes share the same video materials with slightly different digital video effects or video captions, so they share identical shots defined in our work. Once identical shots are detected, originating topics are very likely related topics. As the first experiment to assess usefulness of the work, we apply our method to detect identical shots shared by related topics.

In order to detect identical shots generated from the same video materials, we use tighter thresholds, i.e., $\theta_c = 80\%$ in the experiment. The news video on November 22nd was used to be compared with 23 videos in November, and the system found 286 identical shot pairs. Figure 1 shows successful extraction of identical shots. Figure 1(a)-(d) actually corresponds to shots from related topics broadcasted on the different days. In addition to shots from related topics, identical shots shown in Fig. 1(e)-(f) were also detected, which are not necessarily from related topics, but are "filler" shots, e.g., anchor, opening, weather forecast shots, etc., detailed in the next section. Among detected pairs, 14 pairs were error (5% error rate), 261 were filler (91%), and 11 were from related topics (4%). We can regard the system achieves 95% accuracy in identical shots detection, however, most of them are filler shots. The results motivate us to conduct the next experiment.

4. MINING NEWS VIDEO STRUCTURE

Typical news program includes filler shots as important components, such as opening shots, anchor person shots, weather forecast CG shots, etc., each type of which looks visually similar each other among everyday broadcasts. They are less informative in the sense of topic contents, but provide very important clues of news video structure. Typical news video has very rigid structure which starts with an opening shot followed by highlights, then several topics. Each topic is composed of an anchor shot as the first element, followed by couple of file shots. The program then terminates with weather forecast. Based on this observation, many researchers tried to "parse" news videos by using filler shots (mostly anchor shots) as the most important "tokens." Filler shot detection is the most important part in news video parsing, and taking advantage of visual uniformity for each type of filler shots, researchers tend to use tailored models to detect filler shots, such as almost fixed location for anchor's face, constant background color, etc. However, such approaches suffer from the problem how the models should be built, or stiffness of the models which cannot adapt to slight change in news studio setting. On the other hand, the previous experiment shows that identical shot detection results among a set of news videos provide rich information to locate filler shots, since any type of filler shots are expected to appear almost everyday which are "identical" each other. Thus by properly applying identical shot detection to a certain volume of news video archives, it is possible to realize unsupervised filler shot detection for news video parsing, with no a priori knowledge or models of filler shots.

Assume that we have sufficient volume of news video archives, including N news videos, $V^i, i = 1, \ldots, N$, each of which are processed so that fragments and shots are extracted, and color histograms are calculated. Today's news video V^0 is also processed to obtain fragments, shots, and color histograms. In order to find filler shots in V^0, for each shot S_i^0 in V^0, the system detects the identifiable shot set $IDENT(S_i^0) = \{S_k^j(\in V^j)\}$ in all videos in the archives, where S_i^0 and S_k^j are identical shots. If a shot S is a filler shot, its identifiable shot set $IDENT(S)$ is likely to include some shots from almost all videos. To evaluate this, the system counts the number of videos which include shots identifiable to the shot S:

$$NIDENT(S) = ||\{V^i|\exists jS_j^i \in IDENT(S)\}||. \qquad (9)$$

If S is a filler shot such as an anchor person shot, $NIDENT(S)$ is ideally N, or close enough to N in actual cases, since anchor person shots appear everyday. Thus $P_f(S) = NIDENT(S)/N$ can be used

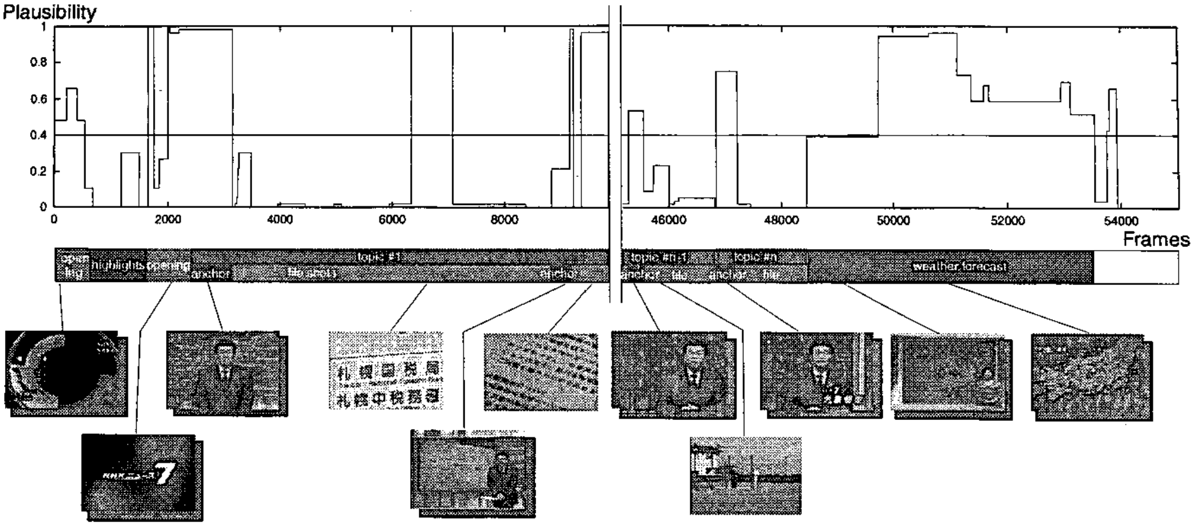


Figure 2: Filler Shot Detection by News Video Mining (thumbnails with shadows are filler shots)

as the plausibility of S being a filler shot. In calculating $P_f(S)$, the system counts the number of news videos including identifiable shots to S. This can be regarded as a kind of aggregation, i.e., data mining operation, for news video database.

We evaluated the plausibility of filler shots of a news video (January 10th) comparing with 64 news footages. The news video was decomposed into shots and the plausibility for each shot was evaluated. Figure 2 shows the results. The plot is associated with thumbnails, where thumbnails with shadows present actual filler shots. The figure shows that most filler shots correspond to higher plausibility values, but there are some false detection such as shots around the 6,500th and 9,500th frames in Fig. 2, where frames has high intensity at almost all pixels. Due to the limitation of the current definition of image identifiability, such frames are regarded as identical each other. This problem should somehow be relieved by, for example, introducing much accurate image identifiability. Even with the current definition, by thresholding the plausibility values at 0.3-0.4, filler shots can be detected. By thresholding at 0.4, 13 filler shots are successfully detected while generating six errors, and two anchor shots are missed. The accuracy is not very high (62%), however, the method is still promising and useful because the method does not require any a priori knowledge or models of filler shots, thus it is self-adaptive to changes in setting of filler shots, or even adaptive to the other types of news shows.

5. CONCLUSIONS

A method to detect identical video segments from video footages in broadcast video archives is presented. A definition of identical images is introduced followed by efficient algorithm of identical shot detection. Then the method is evaluated with two experiments; one is identical shot detection shared by related topics, and another is filler shot extraction which is useful for news video parsing. The successful results are shown to reveal the method's practical effectiveness.

The major contribution of this paper is that we reveal effectiveness of identical shot detection in large-scale video archives. If the size of the archive is small, we find no, or only very few pairs of identical shots. However, if the archive size getting larger, the number of identical shot pairs increases. Moreover, identical shot pairs may provide semantic relations between videos. This can be regarded as a type of knowledge discovery in large-scale video archives. In addition, by applying aggregation operation to the results of identical shot detection such as counting the number of identical shots to a particular shot, new knowledge can be mined from video archives, i.e., unsupervised filler shot detection in our example.

The acceleration treatment described in Sec. 2 reduces the computation in its magnitude but not in its order. For instance, the method runs at $O(N^2)$ time for the size of video archives N. Since this could be very critical especially when the size of archives becomes extremely huge, it should be a challenging task to remedy this, e.g., by incorporating high-dimensional index structure such as [6] to achieve the order of $O(N \log N)$ or lesser time complexity.

6. REFERENCES

[1] S. Aksoy and Robert M. Haralick, "Graph-theoretic clustering for image grouping and retrieval," in *Proc. of CVPR*, 1999.

[2] S. S. Cheung and A. Zakhor, "Video similarity detection with video signature clustering," in *Proc. of ICIP*, 2001.

[3] N. Katayama and S. Satoh, "Distinctiveness-sensitive nearest-neighbor search for efficient similarity retrieval of multimedia information," in *Proc. of ICDE*, 2001.

[4] H.-J. Zhang, S. Tan, S. Smoliar, and Y. Gong, "Automatic parsing and indexing of news video," *Multimedia Systems*, vol. 2, 1995.

[5] "The topic detection and tracking evaluation project," http://www.nist.gov/speech/tests/tdt/tdt2001/index.htm, 2001.

[6] N. Katayama and S. Satoh, "The SR-tree: an index structure for high-dimensional nearest neighbor queries," in *Proc. of SIGMOD*, 1997.