Instance Search and Content-based Copy Detection Experiments for TRECVID 2011

Masami Shishibori, Masashi Ohnishi, Yuuki Tanioka and Kenji Kita

Dept. of Information Science and Intelligent Systems, Faculty of Engineering, The University of Tokushima,
2-1 Minami-Josanjima-cho, Tokushima-shi, Tokushima, 770-8506, JAPAN

0. STRUCTURED ABSTRACT

1. *Briefly, what approach or combination of approaches did you test in each of your submitted runs?*

Instance Search (pilot) Task: This system extracts the facial image from each frame image using the Haar-like operator, and then eliminates noise images (non-facial image) using SVM. Next, SIFT features are detected from the true facial image, and noise SIFT features including in the background of the facial image are eliminated. By using only true SIFT features in the face, the similarity of the facial image is calculated.

Content-based Copy Detection Task: This system uses the Time-Series Active Search as the search method of audio signals. The chroma vector is used as the audio features. This system uses only audio data not video data.

2. *What if any significant differences (in terms of what measures) did you find among the runs?*

none.

3. *Based on the results, can you estimate the relative contribution of each component of your system/approach to its effectiveness?*

Estimation is the same as above.

4. *Overall, what did you learn about runs/approaches and the research question(s) that motivated them?*

SIFT-based approach seems to be promising, but the cost of the retrieval time becomes high. Thus, the efficient retrieval algorithm must be considered.

1. INTRODUCTION

This is the second TRECVID participation for Tokushima University. This year, we have participated in the instance search (INS) pilot and the content-based copy detection (CCD). For the INS task, our main focus was to apply SIFT-based image retrieval method with facial images. For the CCD task, the Time-Series Active Search (TSAS) algorithm is applied for the audio signals, which is represented by the chroma vector as the audio features.

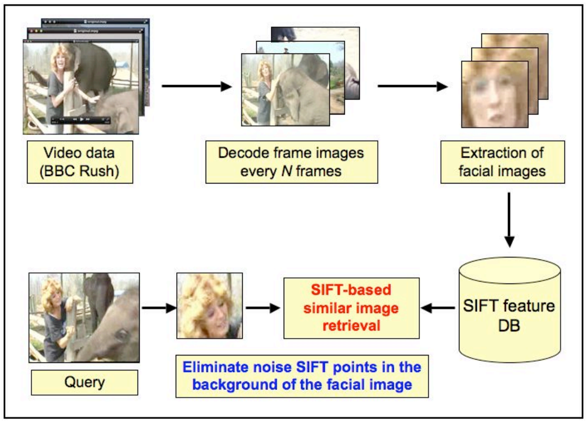

Video data (BBC Rush)
Decode frame images every *N* frames
Extraction of facial images
SIFT feature DB
SIFT-based similar image retrieval
Query
Eliminate noise SIFT points in the background of the facial image

Figure 1: Outline of the instance search system.

2. INSTANCE SEARCH SYSTEM

2.1 Outline of this system

The overview of our retrieval system is shown in Figure 1. At the registration phase, cut scene frames are detected from the video data, and then facial images are extracted from each cut frame using the Haar-like operator [1]. Next, noise images (non-facial image) in the extracted facial images are eliminated using the SVM (Support Vector Machine) [2]. Finally, SIFT (Scale-Invariant Feature Transform) features [3] are detected from true facial images, and these features are registered in the facial database. At the retrieval phase, the user specifies the query image. The person whom the user wants to retrieve is reflected in this image. First, the facial area of the query image is extracted using the same operator as the registration phase. Next, SIFT features are detected from the query facial image. After that, the similarity between query facial image and the facial database is calculated based on 128 dimensional SIFT features.

The different point between TRECVID2010 and this system (TRECVID2011) is that this system eliminates noise SIFT points in the background of the facial image extracted from the query image.

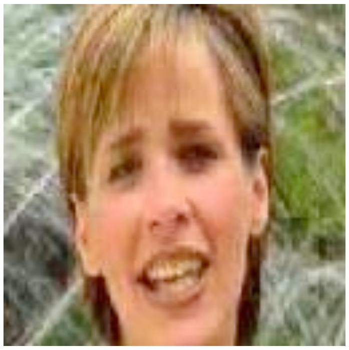


Figure 2: Example of extracted facial images using the Haar-like operator.

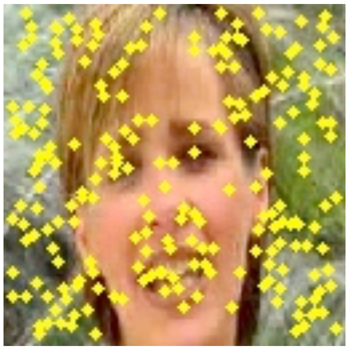


Figure 3: SIFT points detected from the facial image of Figure 2.

2.2 Removal of noise SIFT points

On this system, the face extraction tool in the OpenCV [4] is utilized. This tool is very useful, however the extracted facial image includes some background area. Figure 2 shows the example of extracted facial images using the Haar-like operator. If SIFT points are detected from this facial image, some noise SIFT points in the background area have a bad influence on search accuracy. Figure 3 shows SIFT points detected from the facial image of Figure 2. It is found that many SIFT points are detected in the background area.

In order to solve this problem, this system eliminates noise SIFT points in the background using the color feature. We notice that the color feature between the human face and the background area is different. And, the human face appears in the center of the extracted facial image. First, this system makes the color segmentation image of the extracted facial image. Figure 4 shows the color segmentation image produced from the facial image of Figure 2. Next, as for the color, this system leaves SIFT points in the segmentation area near the flesh color. As for the position, it leaves SIFT points in the segmentation area near the center. Figure 5 shows SIFT points eliminated the noise points in the background from Figure 3.

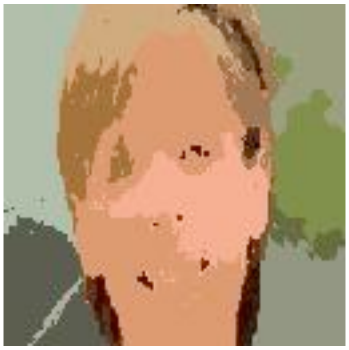


Figure 4: The color segmentation image produced from the facial image of Figure 2.

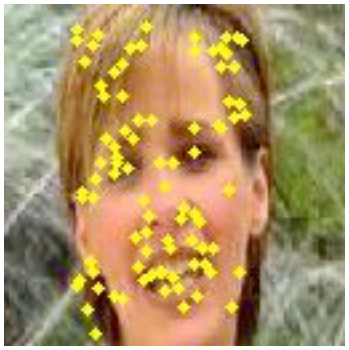


Figure 5: SIFT points eliminated the noise points in the background from Figure 3.

2.3 SIFT-based facial image retrieval

We suppose that L facial images are extracted from the video data and M SIFT features are detected from a facial image, $L*M$ SIFT features are registered in the facial database. At the retrieval phase, the similarity between the query facial image and the facial database is calculated based on the SIFT features. If N SIFT features are detected from the query facial image, the k-nearest neighbor search, which top k similarities can be obtained, is executed N times.

As a result, N retrieval results can be obtained as shown in Figure 6, where one retrieval result has top k similarity facial images. These N retrieval results must be merged into a final result. The final retrieval result is integrated by the vote algorithm. The vote algorithm is executed as following: First, the point according to the order of the similarity is given to each facial image in the first retrieval result. If the retrieval result has k facial images, the top similarity (most similarity) facial image is set to k point. The second similarity image is k-1 point, and the i-th similarity image is k-i+1 point. Next, the same procedure is repeated for other retrieval results. Finally, the final order of the similarity is calculated by the sum of all the points.

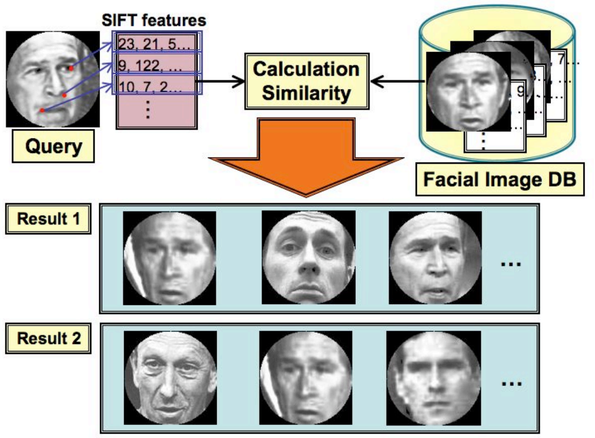


Figure 6: Illustration of the similarity calculation based on SIFT features.

3. COPY DETECTION SYSTEM

3.1 Outline of this system

The overview of our retrieval system is shown in Figure 7. This system uses only audio data not video data. At the pre-processing phase, the audio features are detected from the music data. The chroma vector is used as the audio features, and it is represented by 12 dimensional vector data. Normally, all dimensional data are used as the audio features, however this system uses only first 4 big values and the remaining values are sets to 0 in order to heighten the discernment capability between the features. And then, the vector data is changes to the binary data by using the threshold value. Finally, the unsigned number is calculated by regarding this vector as a binary number. This process is shown in Figure 8. At the retrieval phase, this system uses the Time-Series Active Search [5] as the search method of audio signals.

3.1 Time-Series Active Search

The Time-Series Active Search is the famous method to be able to retrieve the similar scenes quickly for the movie data. This method can be applied not only movie data but also music data. The outline of the Time-Series Active Search Algorithm is shown in Figure 9. This method is executed by the following steps. We suppose that the length of the query is *N frame*, the similarity between the query and the music data is calculated every *N* frames. First, *N* unsigned integer values are acquired from *N* frames as shown in Figure 8. Next, the histogram which represents the distribution of *N* integer values is created. Finally, the similarity can be obtained by calculating the ratio of the overlap between histograms.

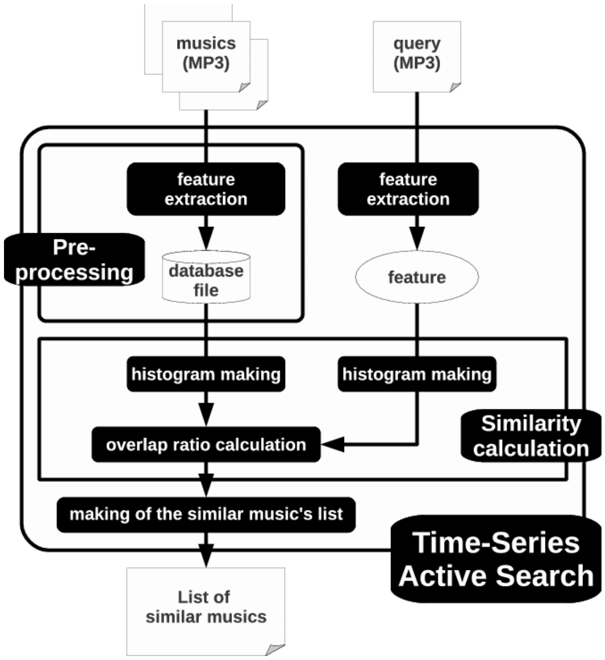


Figure 7: Outline of the copy detection system.

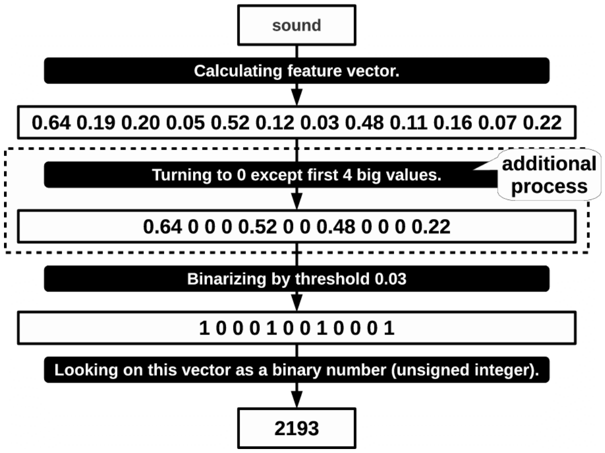
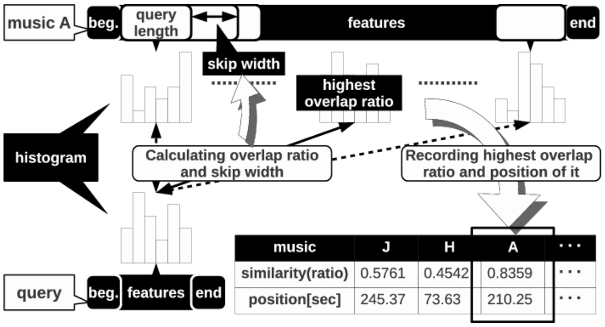

sound
Calculating feature vector.
0.64 0.19 0.20 0.05 0.52 0.12 0.03 0.48 0.11 0.16 0.07 0.22
Turning to 0 except first 4 big values.
additional process
0.64 0 0 0 0.52 0 0 0.48 0 0 0 0.22
Binarizing by threshold 0.03
1 0 0 0 1 0 0 1 0 0 0 1
Looking on this vector as a binary number (unsigned integer).
2193

Figure 8: Example of the audio feature vector.



music	J	H	A	･･･
similarity(ratio)	0.5761	0.4542	0.8359	･･･
position[sec]	245.37	73.63	210.25	･･･

Figure 9: Illustration of the Time-Series Active Search for music data.

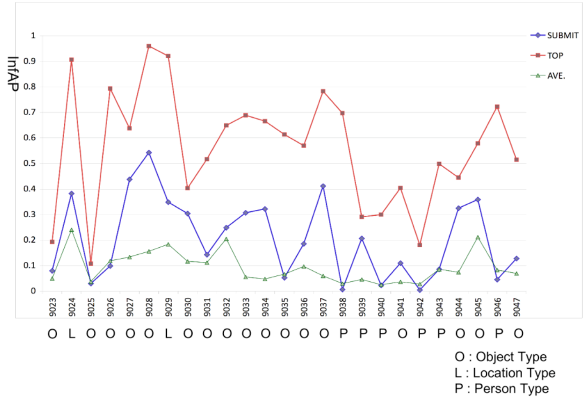


Figure 10: Experimental result of the Instance Search Task.

4. EVALUATIONS

4.1 Evaluation of the Instance Search Task

The video data of TRECVID2011 INS task consists of 20,982 files, which lengths are from 10 seconds to 30 seconds. The video data is decoded every ten frames. Figure 10 shows the experimental result of the INS task. The vertical axis indicates the inferred average precision (InfAP) and the horizontal axis indicates the query number. In this graph, the blue line (submit) is this system accuracy, the red line (top) is the top accuracy and the green line (ave) is the average accuracy. As for the symbols under the query, the symbol “P” shows the person type query, “O” is the object type and “L” is the location type. From the experimental result, the good performance can be obtains for the query number “9039”, because the face image of the query turns to the front. On the other hand, the query number “9038” is bad, because the face image doesn't turn to the front. It is found that this system can search only the same sideways face as the query and can’t search other images of the same person.

3.2 Evaluation of the Copy Detection Task

The video data of TRECVID2011 CCD task consists of 3,200 files, and about 118,115 cut scenes were detected from the video data. Figure 11 shows the experimental result of the CCD task. As this system uses only audio data not video data, this system can’t obtain the good performance. As for the future works, the improvement by introducing the copy detection system using the video data should be considered.

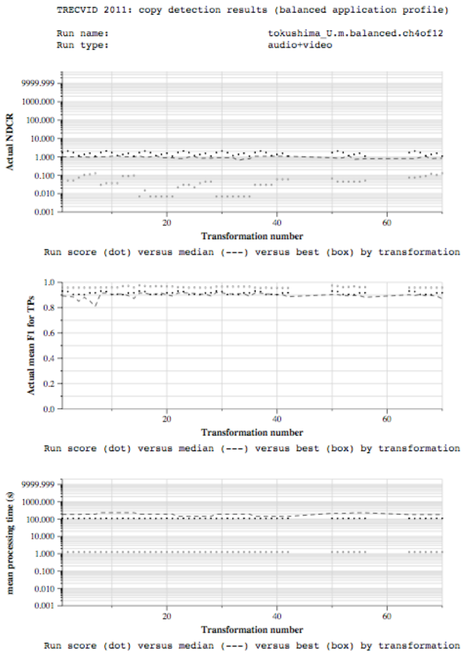


Figure 11: Experimental result of the Copy Detection Task.

REFERENCES

[1] Rainer Lienhart and Jochen Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection”, *IEEE ICIP 2002*, Vol.1, pp. 900-903, Sep. 2002.

[2] V.Vapnik, “The Nature of Statistical Learning Theory”, Springer, 1995.

[3] D.G.Lowe, “Object Recognition from Local Scale-Invariant Features”, *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pp. 1150-1157, 1999.

[4] http://opencv.willowgarage.com

[5] K. Kashino, G. Smith and H. Murase, “A Quick Search Algorithm for Acoustic Signals Using Histogram Features”, *The Transactions of the Institute of Electronics, Information and Communication Engineers*, D-II, pp.1365-1373, 1999.