Near-Duplicate Video Matching with Transformation Recognition

Zhipeng Wu
Graduate University of
Chinese Academy of Sciences
Beijing, 100049, China
zpwu@jdl.ac.cn

Shuqiang Jiang
Key Lab of Intell. Info. Process., Inst.
of Comput. Tech., Chinese Academy
of Sciences, Beijing, 100190, China
sqjiang@jdl.ac.cn

Qingming Huang
Graduate University of
Chinese Academy of Sciences
Beijing, 100049, China
qmhuang@jdl.ac.cn

ABSTRACT

Nowadays, the issue of near-duplicate video matching has been extensively studied. However, transformation, which is one of the major causes of near-duplicates, has been little discussed. In this paper, we focus on the fact that a certain kind of feature may perform excellently to deal with one type of transformation while not be that good on another. We present a self-similarity matrix based near-duplicate video matching scheme with an additional transformation recognition module. By detecting the type of transformations, the near-duplicates can be treated with the 'best' feature which is decided experimentally. Thus, we obtain an enhanced matching result by employing the selected feature. Our work includes seven features and ten transformations respectively, and experimental results show the effectiveness of transformation recognition and the promotion it brings to boost the near-duplicate matching scheme.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Performance, Design, Experimentation

Keywords

Multi-features, Transformation Recognition, Near-Duplicate, Copy Detection, Self-Similarity, SSM

1. INTRODUCTION

Nowadays, with the exponential growth of digital video resources, the near-duplicate issue, which appears to be important in many areas such as copyright infringement and media tracking, has aroused researchers' great attention. According to TRECVID benchmark [11], the near-duplicates or copies, are segments of videos derived from other videos, usually by means of various transformations such as addition, deletion, modification, etc.

Tracing back to the cause of near-duplicates, transformation, which dominates the final formation of the duplicate videos, is important in research and applications.

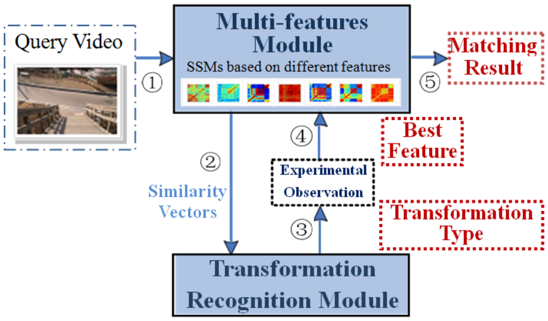


Figure 1. Flowchart of our framework

It helps us to better analyze the original video content. Besides, we can classify the duplicates according to the transformation and mine the relationship between them. Moreover, if we know the exact type of transformations, the task of near-duplicate matching could be much easier. That is, we analogize the duplicate video as a patient who wants to be prescribed from the doctor. Only if the kind of disease (transformation) is detected can we use the proper medicine (feature) to treat the illness (matching the original source).

In review of the study of near-duplicate issue, various kinds of features are adopted as 'all-round medicines' to treat with all kinds of transformations (diseases) but without taking a further step to link the transformation types to the features. Researchers extract low-level global features such as color moment and color histogram in frames [10, 14]. However, the global features suffer from serious problems like lighting changes and fail in more complex tasks. Alternatively, some approaches resort to Local Interest Points (LIPs), which appear to be prominent in retrieval tasks [3, 13, 14, 15, 17]. Although these approaches have shown their effectiveness in near-duplicate issue, the regardless of different transformation types and their 'best' features still leave a big constraint to the bottleneck.

In this paper, we present a Self-Similarity Matrix (SSM) based near-duplicate video matching scheme with a transformation recognition module. The Self-Similarity Matrix (SSM) description for video is succinct and discriminative [16]. We implement it with various features as different SSM detectors to detect transformation types as well as match near-duplicates. To a certain query video, firstly, it enters the multi-features module. It is treated with SSMs based on different features to calculate the similarities to the source files. In that case, several similarity vectors are obtained by different features respectively. Then, all the extracted similarity vectors are input to a pre-trained SVM based

MM'09, October 19–24, 2009, Beijing, China.
Copyright 2009 ACM 978-1-60558-608-3/09/10...$10.00.

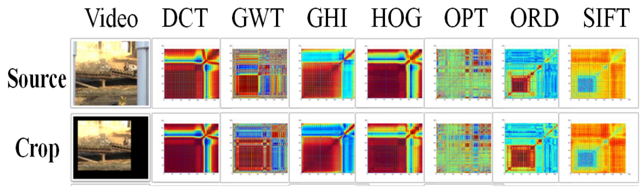


Figure 2. SSMs based on different features

transformation recognition module with an output of transformation type. Finally, the best feature is selected according to experimental observation, and the matching result of the SSM detector which based on the best feature is returned to the users. Figure 1 illustrates the flowchart of our framework.

The rest of the paper is organized as follows. In section 2, we provide a simple review of the SSM based near-duplicate similarity measurement. Then the transformation recognition and near-duplicate matching scheme are fully discussed in section 3. The experimental results are provided in section 4. In section 5, we conclude the paper.

2. SSM BASED NEAR-DUPLICATE SIMILARITY MEASUREMENT

2.1 SSM Definition

Given a certain video clip $V = \{F_1, F_2, ..., F_n\}$ of n frames, SSM is built on exhaustively calculating the distance between every of the two feature vectors extracted from a single frame [16]:

$$SSM_{n\times n} = [d_{ij}]_{i,j=1,2,...,n} = \begin{bmatrix} 0 & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & 0 & d_{23} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \cdots & 0 \end{bmatrix} \qquad (2.1)$$

where d_{uv} is the distance between frame u and v. The SSM is a symmetric positive semi-define matrix with zero value along the main diagonal.

2.2 SSMs Based on Different Features

In the previous work of SSM based near-duplicate video matching, optical flow is extracted in sequential frames to build SSMs [16]. In order to make a relatively comprehensive study of the transformations and the features, we here explore seven commonly used features in computer vision and implement seven different SSMs based on them. Our feature pool contains texture (Gabor Wavelet Transform, GWT [6]), intensity (Grayscale Histogram Intersection, GHI; Ordinal intensity, ORD [2]), motion (Optical Flow, OPT [4]), gradient (Histograms of Oriented Gradients, HOG [1]), frequency domain (Discrete Cosine Transform, DCT), and LIPs (histogram of SIFT visual words, SIFT [7]). Figure 2 describes a set of different SSMs built by various kinds of features. The first row is an original video from MPEG corpus [9] following its cropped version in the second row. Noticing to a certain transformation (here refers to crop), different SSM detectors may perform differently. We can find out that OPT and SIFT show relatively robust matching power while GWT could not handle this case at all. According to the analogy mentioned above, OPT and SIFT are the right 'medicine' for the 'disease' crop. Further discussions about the features and transformations could be found in section 3.

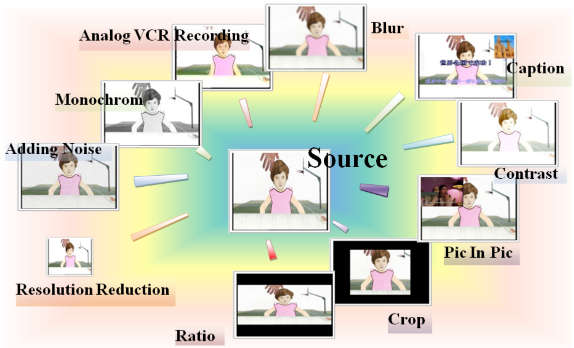


Figure 3. Ten types of transformations included in our dataset

2.3 SSM Based Similarity Measurement

We employ Visual Character-String (VCS) descriptor to describe and measure the similarity between SSMs [16]. The VCS is a string-like vector with float values as characters. The characters are computed by summing up the elements in square regions along the main diagonal of SSM and then formed in turn into one string. (More details are included in [16])

We employ edit distance [5] to calculate the similarity between two visual strings. The measurement [5], typically considers the minimum number of operations on characters to transform one string to another. And the operations here include inserting, substituting, and deleting one character. We quantify the float characters VC_i and VC_j in threshold ∂ :

$$\begin{cases} \textit{same characters} & |VC_i - VC_j| \le \partial \\ \textit{different characters} & |VC_i - VC_j| > \partial \end{cases} \qquad (2.2)$$

While we measure the similarities between query video and source videos with SSM description and edit distance metric, the duplicate-origin video pairs can be matched.

3. SSM BASED TRANSFORMATION RECOGNITION AND NEAR-DUPLICATE MATCHING

3.1 Transformations and Their Best Features

Typically, near-duplicate video is referred as a result by the combination of several kinds of transformations from the original source [11, 8]. However, to conduct a basic exploration, we build type-tagged video pieces on single transformation and aim to find the characteristics of them respectively. Our dataset contains 1000 source video clips and 4000 duplicate queries. It is formed by 10-second clips randomly selected and intercepted from CIVR07 Copy Detection Corpus [8], TRECVID08 benchmark [11], and MPEG Evaluation Experiments for Video Signature [9]. The 4000 queries are separated into ten groups (10*400) and transformed upon one of the ten kinds of transformations: Analog VCR Recording (AVC), Blur, Caption Insertion, Contrast Changing, Crop, Monochrome, Adding Noise, Picture in Picture, Ratio, and Resolution Reduction as shown in figure 3.

Noticing there are no duplicates in the source dataset, we aim to match the query to its only-one origin in the source. We use the evaluation metric in CIVR07 Copy Detection Showcase [8] as the criterion of our scheme:

$$Quality = \frac{NCorrect_{\textit{Num of Correct Matchings}}}{NQueries_{\textit{Num of Total Queries}}} \qquad (3.1)$$

We employ the seven features mentioned in section 2.1 by introducing 2000 queries (10 transformations * 200 pieces each) to match their origins in the source. Table 1 illustrates the average matching qualities.

Table 1. Average matching qualities of different features

	DCT	GWT	GHI	HOG	OPT	ORD	SIFT
Quality	75.5%	44.3%	75.0%	61.0%	90.8%	80.2%	77.2%

It can be observed from experimentation that optical flow and ordinal intensity are the two features which are comparatively robust for SSM based matching. However, are they robust enough to handle every transformation? Take ORD as an example, the qualities received on ten transformations are shown in table 2:

Table 2. Matching qualities of ORD on ten transformations

	AVC	Blur	Caption	Contrast	Crop
Quality	74.5%	100.0%	90.5%	99.5.0%	24%
	Mono	**Noise**	**PicInPic**	**Ratio**	**Resolution**
Quality	99.5.0%	100.0%	44.5%	69.5%	100.0%

ORD performs well on Blur, Contrast, Mono, Noise, and Resolution, but it fails when it comes to Crop, Picture in Picture. Fortunately, in our experiments, we have OPT (90.0% on Crop), SIFT (86.5% on Crop), GHI (91.0% on PicInPic), and they are the right features (like medicine for certain disease) that perform better than ORD. According to this observation, different features should be employed to handle different transformations in order to obtain a better matching quality. Table 3 shows the transformation types and their best features observed in our experiments.

Table 3. Transformations and best features

	AVC	Blur	Caption	Contrast	Crop
Features	DCT	ORD	OPT	GWT	OPT
	Mono	**Noise**	**PicInPic**	**Ratio**	**Resolution**
Features	OPT	ORD	GHI	OPT	ORD

3.2 Transformation Recognition

We detect the type of transformations via two modules: multi-features module and transformation recognition module as shown in figure 1. The transformation recognition module is based on a pre-trained SVM model. Our training data includes 2000 (10 transformations * 200 each) type-tagged video clips randomly selected in the dataset mentioned above.

3.2.1 Multi-features Module

In the multi-feature module, each query is submitted to different SSMs based on different features. Then the similarities between the query and the source videos under every feature are output as middle result for transformation recognition.

First of all, a transformed query Q is treated with seven SSM detectors (based on different features) mentioned above. Then, the seven SSMs are represented as VCSs. To a single feature, by extensively calculating the edit distances between the query and all the source files, we obtain the similarity vector:

$$SV^{K}_{Similarity\ Vector} = \{dis^{K}_{1}, dis^{K}_{2}, dis^{K}_{3}, ..., dis^{K}_{m}\} \tag{3.2}$$

where K denotes the seven different features (K=1~7), m is the total video number in the source, and dis^{K}_{i} is the distance between query Q and video i in the source.

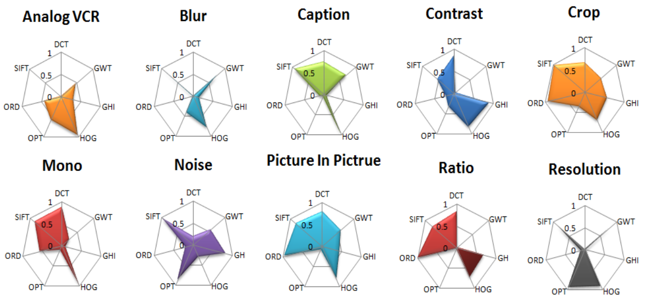


Figure 4. Average responses of the training data

3.2.2 Transformation Recognition Module

With the obtained similarity vector SV^{K}, we extract response R^{K} from it as training/testing data of SVM. Generally, R^{K} should effectively reflect the ability feature K owns to cope with certain type of transformations. We define R^{K} as the minimum value in SV^{K}, which potentially denotes Q's origin in the source. Intuitively, if R^{K} is quite small, it means the feature K is robust to the current transformation type. Otherwise, the feature may be unsuitable to it. Finally, we normalize the responses R^{K} (K=1~7) and form into a vector R as the SVM input. Figure 4 illustrates the average responses of the training data on multi-features. Take the Analog VCR and Blur as examples, we can easily distinguish the two types according to the ORD and SIFT response.

3.2.3 SVM Training

After we extract the seven responses R^{K} (K=1~7) of every training data, we combine them with their transformation type-tagged labels to train an SVM based on linear kernel. The SVM takes the response vector $\{R^{1}, R^{2}, ..., R^{7}\}$ as input and outputs the recognized transformation type.

3.3 Near-Duplicate Matching on Best Feature

After we detect the transformation types of an input query, following the rules in table 3, we can select the best feature to cope with the query. Then, the matching result of the SSM detector based on chosen feature in multi-features module is returned to the user. In such case, we obtain an enhanced result by using the right 'medicine' rather than only relying on one constant feature.

4. EXPERIMENTS

4.1 Transformation Recognition

We evaluate our approach on ten transformations and seven features mentioned in section 2 and 3. Details of our dataset could be found in section 3.1.

Following the scheme mentioned above, we prepare 2000 (10 transformation * 200 each) type-tagged queries as the training data and another 1000(10*100) for verification. The training and testing data are randomly selected without intersection. The recognition quality is defined as:

$$Quality = \frac{NCorrect_{Num\ of\ Correct\ Recognized}}{NTotal_{Num\ of\ Total\ Queries}} \tag{4.1}$$

Table 4 shows the result on the test data. Recognition qualities of the ten transformations are illustrated respectively:

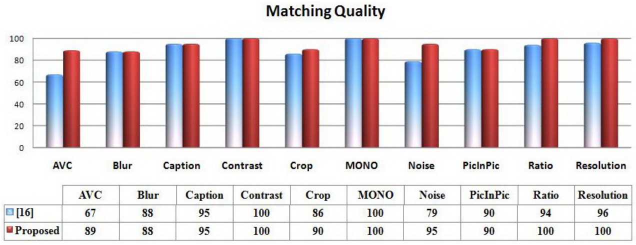
Table 4. Transformation Recognition Qualities

	AVC	Blur	Caption	Contrast	Crop
Quality	90.0%	77.0%	97.0%	75.0%	60.0%
	Mono	Noise	PicInPic	Ratio	Resolution
Quality	63.0%	76.0%	83.0%	73.0%	93.0%

The average recognition quality receives at 78.7%. This might be restricted by the limited number of training data. However, such quality still shows its effectiveness in the matching task. Taking the "Crop" (60.0%) as an example, most of the false recognized queries are decided as "Caption" while the chosen feature for both "Crop" and "Caption" is OPT. In that case, the processing error will not be accumulated to the matching step.

4.2 Near-Duplicate Video Matching

To verify the effectiveness of our framework, we compare it with the original single-feature version presented in [16]. We use the 1000 queries in section 4.1, and figure 5 illustrates the performance promotion of the proposed approach. The average quality rises from 89.5% to 94.7%.



	AVC	Blur	Caption	Contrast	Crop	MONO	Noise	PicInPic	Ratio	Resolution
[16]	67	88	95	100	86	100	79	90	94	96
Proposed	89	88	95	100	90	100	95	90	100	100

Figure 5. Comparison of the matching qualities

4.3 Test on Multi-Transformation Data

Generally speaking, near-duplicate videos are combinations of two or more transformations. Under such complicated circumstances, the proposed method fails in pointing out the transformation types, but it can still select a comparative feature and enhance the matching quality. To evaluate the approach under real situations (multi-transformation with some kinds which are not included in the ten types mentioned above, e.g. Flip, Shift, Cam Cording…), we bring another 200 10-second clip pairs from TRECVID08 [11] and CIVR07 [8] CBCD corpus into our dataset to test the performance of the proposed framework. The queries are all combinations of two or more transformations. And the average matching quality receives at 90.5%, which is higher than using the single-feature version in [16] (88.0%).

5. CONCLUSION

In this paper, we present a near-duplicate video matching scheme as well as finding the exact transformation type. Using the pre-trained SVM based transformation recognition approach, the transformation type is recognized and the best feature is selected. Experimental results show the effectiveness of transformation recognition and the promotion it brings to boost the near-duplicate matching scheme. Future work will be deployed on large training dataset, which would promote the recognition performance. Also, the multi-transformation conditions will be studied.

6. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China: 60773136 and 60833006, in part by National Hi-Tech Development Program (863 Program) of China: 2006AA01Z117, in part by National Basic Research Program of China (973 Program): 2009CB320906, and in part by Beijing Natural Science Foundation: 4092042.

7. REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp.886-893, 2005.

[2] A. Hampapur, K. Hyun, and R. Bolle., "Comparison of Sequence Matching Techniques for Video Copy Detection," *In SPIE. Storage and Retrieval for Media Databases*, vol.4676, pp. 194-201, San Jose, CA, USA, Jan. 2002.

[3] Y. Ke, R. Sukthankar, and L. Huston, "Efficient near-duplicate detection and sub-image retrieval system. *In ACM MULTIMEDIA '04*, pp. 869–876. ACM, 2004.

[4] B. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision," *In: Image Understanding Workshop*. (1981) 121-130.

[5] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics* Doklady 10 (1966):707–710.

[6] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8), pp. 837-842, 1996.

[7] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer, "Learning object representations for visual object class recognition," *The PASCAL VOC'07 Challenge Workshop*.

[8] MUSCLE-VCD-2007, http://www-rocq.inria.fr/imedia/civr-bench/index.html.

[9] MPEG Call for Proposals for video signature tools, http://www.chiariglione.org/mpeg/hot_news.htm.

[10] A. Qamra, Y. Meng, and E. Y. Chang, "Enhanced perceptual distance functions and indexing for image replica recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):379–391, 2005.

[11] TRECVID, http://www-nlpir.nist.gov/projects/trecvid.

[12] Wikipedia. http://en.wikipedia.org/wiki.

[13] X. M. Wu, M. Takimoto, S. Satoh, and J. Adachi, "Scene duplicate detection based on the pattern of discontinuities in feature point trajectories," *In ACM MULTIMEDIA '08*, pp. 51–60. ACM, 2008.

[14] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," *In ACM MULTIMEDIA '07*, pp. 218-227. ACM, 2007.

[15] X. Wu, W.L. Zhao, C.W. Ngo, "Near-duplicate keyframe retrieval with visual keywords and semantic context," *in Proc. Of ACM Int. Conf. on Image and video retrieval,* pp.162-169, 2007.

[16] Z.P. Wu, Q.M. Huang, and S.Q. Jiang, "Robust copy detection by mining temporal self-similarities," *IEEE Int. Conf. on Multimedia and Expo,* 2009. Accepted.

[17] D. Xu, T.J. Cham, S. Yan and S.-F. Chang, "Near duplicate image identification with spatially aligned pyramid matching," *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pp.1-7, 2008.