Marwa Mohamed Dawoud, Dr. Mohamed El Mahallawy, Prof. Dr. Mohamed Waleed Fakhr, Prof. Dr. Mostafa Abdel Azeem / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 1, Jan-Feb.2012, pp. 339-344

Video Copy Detection Based On Hidden Markov Models and Invariant Moments

Marwa Mohamed Dawoud*, Dr. Mohamed El Mahallawy, Prof. Dr. Mohamed Waleed Fakhr *, Prof. Dr. Mostafa Abdel Azeem***

*(College of Computing & Information Technology, Arab Academy for science and Technology & Maritime Transport
Cairo, Egypt)
** (College of Engineering Technology, Arab Academy for science and Technology & Maritime Transport
Cairo, Egypt)

ABSTRACT

Video has recently been commonly used for transferring information due to the continuous growth of the Internet. Because of this, illegal video copy detection is one of the most needed technologies. Existing video copy detection methods compare the whole video frames, so it may take a long time to detect a copy among many reference videos. In this paper we have used a recognition-based approach with hidden Markov models as the recognition tool. HMM is trained using the original video sequences and its manipulated versions. The HMM is tested with the query videos and the score determines if the query video is a copy or not. Invariant Moments are used as features for the HMM. To evaluate this approach, we used 18 reference videos and 6 video files. We measured the detection rate by comparing these original video copies to their manipulated versions. The experiments show that our framework has a recognition rate of 99.9% accuracy as well as to test it on a large -scale videos database.

Keywords - **Video Copy Detection, Invariant Moments, Hidden Markov Model, recognition-based, dynamic range, video segments.**

I. INTRODUCTION

Video copy detection is essential for many applications, for example, discovering copyright infringement of multimedia content, monitoring commercial air time, querying video by example, etc [1]. Generally there are two complimentary approaches for video copy detection: digital video watermarking and content-based copy detection (CBCD). The first approach refers to the process of embedding irreversible information in the original video stream, where the watermarking can be either visible or invisible. The second approach extracts content-based features directly from the video stream, and uses these features to determine whether one video is a copy of another [2]. Growing broadcasting of digital video content on different media brings the search of copies in large video databases to a new critical issue. Digital videos can be found on TV Channels, Web-TV, Video Blogs and the public Video Web servers. The massive capacity of these sources makes the tracing of video content into a very hard problem for video professionals. At the same time, controlling the copyright of the huge number of videos uploaded everyday is a critical challenge for the owner of the popular video web servers. Content Based Copy Detection (CBCD) presents an alternative to the watermarking approach to identify video sequences and to solve this challenge [2]. For the protection of copyright, watermarking and CBCD are two different approaches: Watermarking inserts non-visible information into the media which can be used to establish ownership [2]. In a CBCD approach, the watermark is the media itself. Generally CBCD consists in extracting a small number of pertinent features (called signatures or fingerprints) from the images or the video stream and matching them with the database according to a dedicated voting function [2]. The goal of video copy detection is to locate segments within a query video that are copied or modified from an archive of reference videos. Usually the copied segments are subject to various transformations such as rotation, reflection and translation. All these transformations make the detection task more challenging [1]

II. RELATED WORK

Several kinds of techniques have been proposed in the literature: in order to find pirate videos on the

Internet, Indyk et al. use temporal fingerprints based on the shot boundaries of a video sequence. This technique can be efficient for finding a full movie, but may not work well for short episodes with a few shot boundaries in [3]. Oostveen et al. in [4] present the concept of video fingerprinting or hash function as a tool for video identification. They have proposed a spatio-temporal fingerprint based on the differential of luminance of partitioned grids in spatial and temporal regions. B. Coskun et al. [5] propose two robust hash algorithms for videos both based on the Discrete Cosine Transform (DCT) for identification of copies. Hampapur and Bolle [6] compare global descriptions of the video based on motion, color and spatio-temporal distribution of intensities. This ordinal measure was originally proposed by Bhat and Nayar [7] for computing image correspondences, and adapted by Mohan, for video purposes [8]. Different studies use this ordinal measure [9, 10] and it has been proved to be robust to different resolutions, illumination shifts and display formats. Other approaches focus on exact copy detection for monitoring commercials; an example being Y. Li et al. [11] who use a compact binary signature involving color histograms. The drawback of the ordinal measure is its lack of robustness as regards logo insertion, shifting or cropping, which are very frequent transformations in TV post-production. The authors of [12] show that using local descriptors is better than ordinal measure for video identification when captions are inserted. When considering post-production transformations of different kinds, signatures based on points of interest have demonstrated their effectiveness for retrieving video sequences in very large video databases, like in the approach proposed in [13].

III. FRAMEWORK FOR VIDEO COPY DETECTION

This section presents the framework for video copy detection used for study in this paper A good video copy detection system should have high precision of rates and should detect all copies in a video stream possibly subjected to videos transformations such as rotation with different degrees, reflection, brightness, dimensions, video quality... etc. In order to design the required framework, we need video collections with different intensities and different number of frames. For implementation, we will use variety of tools: MATLAB, Hidden Markov Model Toolkit (HTK), as well as programming in C language. Our framework for video copy detection system is composed of: feature extraction, training phase and recognition phase.

We use hidden markov model as recognizer; to recognize the identical or non-identical video copies and score the recognition rate, to evaluate the performance of our framework, as illustrated by Figure1. Feature Extraction is a special form of dimensionality reduction. The input data will be transformed into a reduced representation set of features (also named feature vector), the feature extraction works on grayscale level and also the number of features must be enough to get high recognition rate. The feature vector is composed of seven invariant moments (also named hu invariant moments, discussed later in Section *A*. Due to the wide variation in the range of the feature vectors components, the normalization of these feature vectors components is needed to balance, the difference dimensions, of the feature vectors while calculating the distances between points in features space, during the recognition phase.

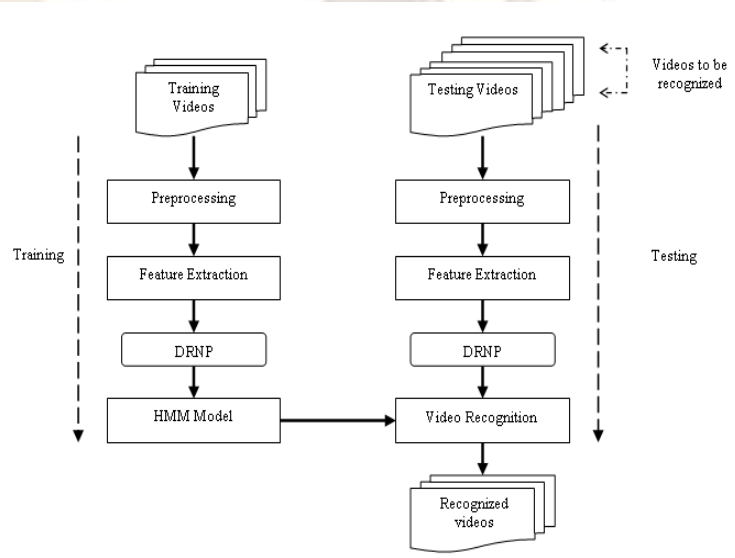

Figure 1 : Video Copy Detection Framework

In addition, we use dynamic range normalization parameter (DRNP); this module is used to detect the effective dynamic range of all the feature vectors, components, to calculate the normalization parameters for each component using the population of the feature vectors in the training data. HMM Model receives the observations sequence, which is the sequence of quantized feature vectors, of each frame along with its sequence.

In testing phase, we use a collection of videos to be recognized with gray scale level; also we use DRNP to calculate the normalization parameters for each component using the population of the feature vectors.

3.1 INVARIANT MOMENTS

Moment invariants are functions of image moments, invariant to certain class of image degradations: Rotation, translation, scaling. Moments are "projections" of the image function into a polynomial basis. The most common moments: central moments, normalized central moments, scale invariant moments, and rotation invariant moments. Central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\bar{x})^p(y-\bar{y})^q f(x,y)dxdy \quad (1)$$

where $\bar{x}=\frac{M_{10}}{M_{00}}$ and $\bar{y}=\frac{M_{01}}{M_{00}}$ are the components of the centroid. If $f(x, y)$ is a digital image, then the previous equation becomes

$$\mu_{pq}=\sum_x\sum_y (x-\bar{x})^p(y-\bar{y})^q f(x,y) \quad (2)$$

Central moments are translational invariant. Scale Invariant Moments, Moments ηi j where i + j ≥ 2 can be constructed to be invariant to both translation and changes in scale by dividing the corresponding central moment by the properly scaled (μ_{00}) th moment, using the following formula.

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}\left(1+\frac{i+j}{2}\right)} \quad (3)$$

Rotation invariant moments, it is possible to calculate moments which are invariant under translation, changes in scale, and also rotation. Most frequently used are the Hu set of invariant moments [15]:

$$\varphi_1 = \eta_{20}+\eta_{02} \quad (4)$$

$$\varphi_2 = (\eta_{20}-\eta_{02})^2+4\eta_{11}^2 \quad (5)$$

$$\varphi_3 = (\eta_{30}-3\eta_{12})^2+(3\eta_{21}-\eta_{03})^2 \quad (6)$$

$$\varphi_4 = (\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2 \quad (7)$$

$$\varphi_5 = (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right] + (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right] \quad (8)$$

$$\varphi_6 = (\eta_{20}-\eta_{02})\left[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right] + 4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03}) \quad (9)$$

$$\varphi_7 = (3\eta_{21}-\eta_{03})(\eta_{30}+\eta_{12})\left[(\eta_{30}+\eta_{12})^2-3(\eta_{21}+\eta_{03})^2\right] + (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})\left[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2\right] \quad (10)$$

The first one φ_1 is analogous to the moment of inertia around the image's centroid, where the pixels' intensities are analogous to physical density. The last one, φ_7, is skew invariant, which enables it to distinguish mirror images of otherwise identical images. Hu's seven moment invariants have the desirable properties of being invariant under image scaling, translation, and rotation. We can see that to compute the higher order of Hu's seven moment invariants is quite complex, and to recover the image from the results is deemed to be difficult as in [15, 16].

3.2 HIDDEN MARKOV MODELS

Hidden Markov Models (HMM) are a powerful tool that has been shown to be of great use in speech and signal processing [17]. Hidden Markov Models (HMMs) are increasingly being used in computer vision for applications such as: gesture analysis, action recognition from video, and illumination modeling. Their use involves an off-line learning step that is used as a basis for on-line decision making (i.e. a stationary assumption on the model parameters). But, real world applications are often non-stationary in nature. This leads to the need for a dynamic mechanism to learn and update the model topology as well as its parameters. These models (along with variants) are increasingly being applied by the vision community to problems such as image segmentation, face recognition, gesture interpretation, event understanding, and background modeling (e.g. [18], [19]). HMMs provide the framework for modeling dynamical or spatial dependencies and correlations between measurements. The dynamical dependencies are modeled implicitly by a Markov chain with a specified number of hidden states and a transition matrix, with observations that are conditionally independent given a state. Major issues in the use of HMMs in real world applications involve two points: real-time computation, and topology modification to address non-stationeries due to dynamically varying conditions. A hidden Markov model is a stochastic finite state machine, specified by a record (S, A, π) where **S** is a discrete set of hidden states with cardinality N, π is the probability distribution for the initial state $\pi(i) = p(s_i) \quad s_i \in S$. **A** is the state transition matrix with probabilities: $a_{ij} = p(s_j \mid s_i) \quad s_i, s_j \in S$ where the state transition coefficients satisfy $\sum_{s_i} \in S\, a_{ij} = \mathbf{1}, \quad s_i \in S$. The states themselves are not observable. The information accessible consists of symbols from the alphabet of observations $O = (O_1, \ldots, O_T)$ where **T** is the number of samples in the observed sequence. For every state an output distribution is given as $b_i(k) = P(O_t = k \mid s_i) \quad k \in O, s_i \in S$. Thus, the set of HMM parameters θ consists of the initial state distribution, the state transition probabilities and the output probabilities. HMMs can be used for classification and pattern recognition by solving the following problems:

3.1.1 The Evaluation Problem: Given the model with parameters θ, calculate the probability for an observation sequence O. Let $O = (O_1, \ldots, O_T)$ denote the observation sequence and $S = (S_1, \ldots, S_T)$ a state sequence. The probability $P(O \mid \theta)$ can be obtained by Forward Algorithm [14].

3.1.2 The Decoding Problem: Find the optimal state sequence for an observation sequence $\mathbf{argmax}_{S \in S^T} P(S \mid O, \theta)$. This can be done by the Viterbi algorithm [14].

3.1.3 The Learning Problem: Given an observation sequence O and the HMM parameters, find the parameters $\hat{\theta}$ which maximize $P(O \mid \theta)$, i.e. $\hat{\theta} = \mathbf{argmax}_{\theta} P(O \mid \theta)$. This question corresponds to training an HMM. The state sequence is not observable. Therefore, the problem can be viewed as a missing-data problem, which can be solved via an EM-type algorithm. In the case of HMM training, this is the Baum-Welch algorithm [14].

3.3 EXPERIMENTS RESULTS

Our framework for detecting video copies is to take original video sequence as training, and its manipulated versions as testing. We use Invariant Moments (7 moments) as features for the HMM, that we used it as a recognizer to determine if the query video is a copy or not. We use six videos were collected from MATLAB and YUV Sequences [20], with different resolutions (320×240, 160 × 120, 352 × 288) and resample at different frame rates (15 ~ 30 fps).

Manipulated versions were produced by applying some modifications to the original videos sequences, such as half size, blur and rotation. In our experiments, we searched for the best match for detecting videos. According to our first experiment, we used the original videos sequences themselves as the training data set(757 frames) and its manipulated versions as the testing data set (757 frames), invariant moments were computed for each frame. Other experiment, we used same original videos sequences for training but with different number of frames (1518 frames) and their manipulated versions for testing with same number of frames (1518 frames), also invariant moments were computed for each frame. The normalization of these feature vectors components is needed to balance, the difference dimensions, of the feature vectors while calculating the distances between points in features space, during the recognition phase. Results are illustrated in table 1. Table 1 shows average recognition rate after normalization using left-right HMMs.

Table 1

Exp. No.	Average Recognition Rate before and after normalization using left-right HMMs			
	Description	***No of frames***	***Average Rate Before Normalization***	***Average Rate After Normalization***
1	Train → Original (Half No of frames) Test → Effects (Half No of frames)	757 frames	94.45%	99.9%
2	Train → Original (Total No of frames) Test → Effects (Total No of frames)	1518 frames	83.33%	99.9%

IV. CONCLUSION

In This paper we have presented a video copy detection framework based on Invariant Moments and Hidden Markov Models. Our framework uses original videos sequences and its manipulated versions by applying some modifications such as half-size, blur and rotation. For training, we use original videos sequences themselves. For testing, we use manipulate versions of original videos sequences using HMM as recognizer. The computation of the Invariant Moments (7 moments) as features of a sequences and the recognition, are both fast. Experiments have showed that our framework has a recognition rate of 99.9% accuracy, as well as to test it on a large-scale videos database.

REFERENCES

Proceedings Papers:

[1] Zhu Liu, Tao Liu, David Gibbon, Behzad Shahraray, Effective and Scalable Video Copy Detection, MIR'10, March 29–31, 2010

[2] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. "Video copy detection: a comparative study". In Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR'07), pages 371-378, 2007.

[3] P. Indyk, G. Iyengar, and N. Shivakumar. Finding pirated video sequences on the internet. Technical report, Stanford University, 1999.

[4] J. Oostveen, T. Kalker, and J. Haitsma. Feature extraction and a database strategy for video fingerprinting. In VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems, London, UK, Springer-Verlag, pages 117–128, 2002.

[5] B. Coskun, B. Sankur, and N. Memon. Spatio-temporal transform-based video hashing. IEEE Transactions on Multimedia, 8(6):1190–1208, 2006.

[6] A. Hampapur and R. Bolle. Comparison of sequence matching techniques for video copy detection. In Conference on Storage and

Retrieval for Media Databases, pages 194–201, 2002.

[7] D. Bhat and S. Nayar. Ordinal measures for image correspondence. ieee Transactions on Pattern Analysis and Machine Intelligence, 20(4):415–423, 1998.

[8] R. Mohan. Video sequence matching. In Int. Conference on Audio, Speech and Signal Processing,1998.

[9] X.-S. Hua, X. Chen, and H.-J. Zhang. Robust video signature based on ordinal measure. In International Conference on Image Processing, 2004.

[10] C. Kim and B. Vasudev. Spatiotemporal sequence matching techniques for video copy detection. IEEE Transactions on Circuits and Systems for Video Technology, 1(15):127–132, Jan. 2005.

[11]] Y. Li, L. Jin, and X. Zhou. Video matching using binary signature. In Int. Symposium on Intelligent Signal Processing and Communication Systems, pages 317–320, 2005.

[12] K. Iwamoto, E. Kasutani, and A. Yamada. Image signature robust to caption superimposition for video sequence identification. In International Conference on Image Processing, 2006.

[13] A. Joly, C. Frelicot, and O. Buisson. Feature statistical retrieval applied to content-based copy identification. In International Conference on Image Processing, 2004.

[15] M. K. Hu, "Visual Pattern Recognition by Moment Invariants", IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962.

[16] A. Khotanzad and Y. H. Hong, “Invariant image recognition by Zemike moments,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vo1.12, NOS, pp.489-497, 1990.

[17] L.R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE , Vol. 77, No. 2, pp. 257-286, 1989.

[18] M. Brand and V. Kettnaker, ”Discovery and Segmentation of Activities in Video,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 844-851, August 2000.

[19] J. Rittscher, J. Kato, S. Joga, A. Blake, A Probabilistic Background Model for Tracking, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2000.

[20] http://trace.eas.asu.edu/yuv/ , YUV Sequences.

Books:

S.Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X.Liu, G.Moore, J. Odell, D. Ollason, D. Povey, V.Valtchev, P. Woodland "*The HTK Book (for HTK Version 3.4)*", March 2009.