Block Operator Context Scanning for Commercial Tracking

Ioannis Giannoukos[1], Vassilis Vrachnakis[1], Christos-Nikolaos Anagnostopoulos[2], Ioannis Anagnostopoulos[3], and Vassili Loumos[1]

[1] Electrical & Computer Engineering School, National Technical University of Athens
[2] Cultural Technology and Communication Dpt., University of the Aegean
[3] Computer Science and Biomedical Informatics Dpt., University of Central Greece
igiann@medialab.ntua.gr, canag@ct.aegean.gr, janag@ucg.gr, loumos@cs.ntua.gr

Abstract. The industry that designs and promotes advertising products in television channels is constantly growing. For effective market analysis and contract validation, various commercial tracker systems are employed. However, these systems mostly rely on heuristics and, since commercial broadcasting varies significantly, are often inaccurate. This paper proposes a commercial tracker system based on the Block Operator Context Scanning (Block - OCS) algorithm, which is both accurate and fast. The proposed method, similar to coarse-to-fine strategies, skips a large portion of the image sequences by focusing only on Regions of Interest. In this paper, a video matching algorithm is also proposed, which compares image sequences using time sliding windows of frames. Experimental results showed 100% accuracy and 50% speed increase compared to traditional block-based processing methods.

Keywords: Block-Operator Context Scanning, Commercial Tracking, Sliding Windows, Video Matching.

1 Introduction

The advertising industry functions as an intermediate between the manufacturers and the customers, creating products of substantial cost. In this industry, broadcast monitoring for commercial detection and tracking are crucial tools for contract validation and market analysis. These applications monitor TV channels to detect how many times a commercial clip is broadcasted and in which time zone. This information is extremely useful for copyright owners to collect royalties and for advertisers to efficiently manage the commercial broadcasting for the validation of their contracts with the television networks. This process, however, introduces a high computational cost, making it inappropriate for real-world applications.

This paper presents an accurate and fast commercial tracking system which implements a modified version of the Operator Context Scanning (OCS) algorithm, proposed in our previous works [1-3]. This algorithm acts as an alternative to the traditional exhaustive block-based processing. It uses operators in the form of sliding windows and skips a large portion of the input image pixel space, focusing only on

I. Maglogiannis, V. Plagianakos, and I. Vlahavas (Eds.): SETN 2012, LNAI 7297, pp. 369–374, 2012.

Regions of Interest. The Regions of Interest in an image are defined by the sliding window operator and present the features that the operator is searching for. To further improve the speed and accuracy of the system, a novel video matching method is proposed that uses temporally sliding windows, which group consecutive adjacent frames of the stored video clips.

2 Block Operator Context Scanning

The OCS algorithm uses pixel operators in the form of sliding windows, which associate a pixel neighborhood to the possibility of belonging to a Region of Interest (RoI). It focuses the processing of an image on the regions that have a large density of the desired features and uses a low sampling rate for the rest of the image regions.

Specifically, the Block-OCS algorithm can be formulated as follows. Let I be an image, which can be divided into K sub-regions, arranged in M columns and N rows. Therefore, each image sub-region size is $\frac{W}{M} x \frac{H}{N}$, where W and H is the width and the height of image I respectively. Additionally, let $V=[v_1, v_2,\ldots, v_n]$ be a feature vector that a Sliding Windows OPerator (SWOP) of size $\frac{W}{M} x \frac{H}{N}$ is extracting. The SWOP operator evaluates a fitness function f in the region, based on the feature vector V, and when f exceeds a predefined threshold, R is marked as that of a Region of Interest (SWOP=1); otherwise the SWOP output is '0'. Therefore, the Block-OCS algorithm produces a binary mask, the positive pixels of which indicate the Regions of Interest.

The main advantage of the algorithm is the introduction of the factor of velocity during image scanning. The algorithm velocity is manipulated using an Additive Increase Multiplicative Decrease (AIMD) window-based control scheme, which is commonly used to prevent data traffic congestion on computer networks [4]. Specifically, if the currently processed region centroid is (x, y), the next region centroid to be scanned in the horizontal direction will be (x,y'), assuming for simplicity that y'<H. The new vertical coordinate y' is calculated as follows (equation 1):

$$y' = y + v_H(p) \tag{1}$$

where v_H is, the "velocity" parameter of the Block-OCS algorithm defined in the horizontal axis, $v_{H,max} \geq v_H \geq W/M$, $v_H \in N$, and $v_{H,max}$ the maximum value of the velocity parameter in the horizontal axis. The parameter *p* in the aforementioned equation refers to the pth region that the Block-OCS examines in the image I and $1 \leq p \leq K$, $p \in N$. Compared to the original OCS, the velocity in the Block-OCS cannot be assigned values in the range [0, W/M), since the factor of the minimum vertical velocity $v_{min,H}=W/M$ prevents the re-calculation of feature vector V multiple times in the same region. The values of parameter v_H follow the pattern of equation 2.

$$v_H(p+1) = \begin{cases} v_{\min,H} + a + v_H(p),\ if\ SWOP(x,y) = 0, \\ (additive\ increase),\ a > 0 \\ \\ v_{\min,H} + d \cdot v_H(p),\ if\ SWOP(x,y) = 1, \\ (multiplicative\ decrease),\ 1 > d > 0 \end{cases} \tag{2}$$

where α is the "acceleration" parameter and d refers to the "deceleration". In accordance with the AIMD scheme [4], the Block-OCS algorithm considers the high density SWOP zero outputs as "data" to be "transmitted" as fast as possible. Thus, it searches for "traffic congestion", since this would indicate a Region of Interest. Similar to the AIMD scheme, the Block-OCS velocity parameter value is directly correlated to the appropriate window size that is required to achieve the optimal transmission rate. Consequently, a large portion of the image is skipped, reducing in this way the computational cost of the image segmentation method.

The aforementioned process, described for the horizontal axis, is also applicable to the vertical axis of the image. If the SWOP operator detects a small number of candidate pixels after scanning an image row, then a number of image rows are skipped according to the vertical velocity parameter. Additionally, a minimum vertical velocity factor $v_{min,V}$=H/N is also considered in the vertical axis of the image.

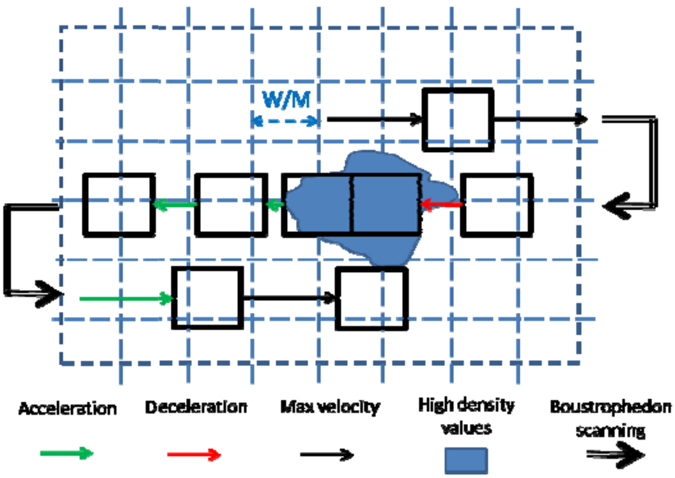

Fig. 1. Block Operator Context Scanning algorithm

The Block-OCS algorithm scans the image following a "boustrophedon" scheme, which was introduced in the original OCS (as seen in Figure 1). In this scheme, the scanning window slides in alternating opposite directions in a descending series of horizontal lines, like the course of plough in successive furrows. This way, the velocity does not reset after scanning an image row, since the next region to be processed is always near the previous pixel neighborhood.

3 Video Matching Algorithm

The video matching algorithm compares a video stream to a database of stored image sequences. The algorithm reduces search space using the temporal information of the stored image sequences. In the following, the proposed algorithm is described in detail.

Let H_{Block} be the block histogram of image I that contains K bins, corresponding to K sub-images of the image, arranged in M columns and N rows. A block in coordinates (i,j) corresponds to the [(i-1) * M +j] bin of the histogram. Each bin contains an activation value, which describes the area coverage of the feature that is searched. This value is derived from the Block-OCS algorithm, that is, the number of pixels that present the desired features in the region defined by the operator. When the operator finds a Region of Interest, it distributes the number of candidate pixels found to the respective histogram bins, according to the percentage coverage of the operator region surface to the MxN sub-images of image I.

To compare two frames, the Mean Square Error (MSE) of the respective histograms is used. Specifically, let H_{in} be the histogram of an input frame of the TV program and H_c (where $0 \leq c \leq C$) the histogram of the c^{th} stored commercial in the database. Each histogram contains K bins. Then, a frame in the database is compared to the current frame of the input sequence as follows:

$$MSE\left(H_{in,}H_c\right) = \frac{1}{K}\sum_{j=1}^{K}(H_{in}^{j} - H_c^{j})^2 \tag{3}$$

However, the frame that presents the minimum MSE does not always correspond to the correct commercial. So, instead of finding the frame with the minimum MSE value, a threshold MSE_{thres} is defined, where every commercial with frames below this threshold is considered a candidate. In addition, time sliding windows are applied in order to exploit temporal information during video matching. More specifically, this technique groups adjacent frames to define a window with the length of 2m+1 frames ($m \in \mathbb{N}$), as follows:

$$W_t = [H_{t-m}, H_{t-m-1,} \ldots H_t, H_{t+1}, \ldots, H_{t+m}] \tag{4}$$

This time sliding window is applied to both input stream (TV broadcast) and the stored image sequences. The proposed algorithm compares the frames of the input stream only to the stored frames defined by the time sliding windows. A number of candidate commercials may be indicated by this process. Then, the windows of both the input stream and the stored commercials slide, by t_{step} seconds, to include the next group of frames (Fig. 2).

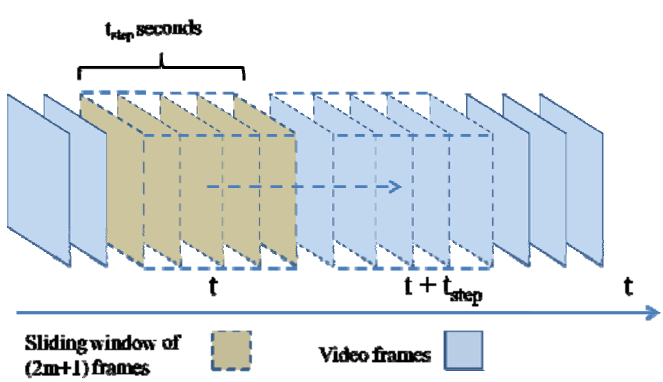

Fig. 2. Time sliding window technique

To match an entire commercial to the input stream, a novel algorithm is used, which utilizes the temporal information of the stored image sequences. This algorithm categorizes the stored commercials into three sets, namely, true candidates (corresponding to set T), false candidates (set F) and rejected (set R). The goal of the algorithm is to filter the stored sequences and find the candidate which best matches the input stream.

Initially, all stored image sequences are included in set F and the time sliding windows contain their first 2m+1 frames. The filtering process initiates when, compared to the input stream, a number of commercials present a MSE below the threshold and are transferred to set T. In the next steps, the following actions may be taken: a commercial in set T which exceeds the threshold MSE_{thres} is moved to set F, a commercial in set F that presents a MSE below the threshold is transferred to set T and a commercial that remains in set F for a time period $t>t_{thres}$ is rejected (sent to set R). All the commercial clips in set R are not processed further. As far as the time sliding windows are concerned, they slide for 2m+1 frames, corresponding to t_{step} seconds, at each step of the algorithm until they reach the end of the respective image sequences. The procedure ends when no commercials remain in sets T and F, or when the time sliding windows reach the end of the image sequences in the database. Then, the image sequence that has appeared the most in set T is matched to the input stream. When this procedure finishes, the algorithm resets and restarts.

In parallel, the timestamps that correspond to the beginning and to the end of the input stream and the found commercials are compared. If the identified commercial has the same duration in both the input stream and the database, then it is considered a positive match. On the contrary, if the duration is not the same, a warning is issued. This warning could mean that either a shortened or a longer version of the stored commercial was aired.

4 Experimental Results

One hundred forty seven (147) commercial clips were collected from the video sample, including clips that were aired on different channels. Approximately 80% of those (116 commercial clips) were stored in a database. To evaluate the proposed method, two test cases were designed. The first included non-commercial program (a talk show), commercial clips previously stored in the database and other commercials (unknown to the system). Therefore, the first test case is considered to be a typical example of normal TV broadcast. The second test case contained commercials that were deliberately shortened; commercial shortening may occur in TV channel broadcast, in order to save broadcast time. This test case was designed to evaluate the proposed system in terms of estimating the duration of the commercials aired.

In both test cases, all the commercials stored in the database were accurately identified in the input stream, including those that were aired by a different TV channel. Additionally, the non-commercial program and the unknown commercials were correctly classified as unknown video streams. As far as shortened commercials are concerned, the method correctly identified 14 as the shortened versions of known commercials and 10 were classified as unknown. The values of the method parameters were as follows. The Block-OCS max velocity parameter was assigned the value 20, to produce an image histogram of K=99 blocks. The time sliding window had a length of 5 frames (m=2) corresponding to 0.5 seconds. The threshold t_{thres} over which a commercial is rejected was 3, corresponding to 1.5 seconds of broadcast time.

5 Discussion

This paper presents a commercial tracker system that measures how many times a commercial clip stored in a database has been aired and calculates its broadcast duration. The system is based on the Block Operator Context Scanning algorithm which uses pixel operators to process the image sequences in a coarse-to-fine manner. The algorithm uses the popular Canny edge detector to find Regions of Interest that include edges in high density. The main advantage of Block-OCS is that it focuses only on Regions of Interest, by applying a low sampling rate to the rest. Compared to typical block based processing, the method was found to be 50% faster (9 frames per second instead of 6 which is found in the literature). The personal computer used in the experiments has a Core 2 Duo processor at 2GHz and 4GB RAM. The system was developed using the Open Computer Vision [5] library and the MySQL database management system [6].

References

1. Giannoukos, I., Anagnostopoulos, C.-N., Loumos, V., Kayafas, E.: Operator context scanning to support high segmentation rates for real time license plate recognition. Pattern Recognition 43, 3866–3878 (2010)
2. Anagnostopoulos, C.N., Anagnostopoulos, I., Kayafas, E., Loumos, V.: A license plate recognition algorithm for Intelligent Transportation System applications. IEEE Transactions on Intelligent Transportation Systems 7(3), 377–392 (2006)
3. Anagnostopoulos, C., Anagnostopoulos, I., Vergados, D., Kayafas, E., Loumos, V.: Sliding Windows: A software method for real-time inspection in textile surface. Textile Research Journal 74(7), 646–651 (2004)
4. Chiu, D.M., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Computer Networks and ISDN Systems 17, 1–14 (1989)
5. OpenCV. Open Computer Vision Library (February 1, 2012) `http://sourceforge.net/projects/opencv/`
6. MySQL. MySQL relational database management system (February 1, 2012), `http://www.mysql.com/`