Effective recognition of the video pattern in a recorded video stream

Atanas Nikolov, Dimo Dimov, Vassil Kolev,
Miroslav Ivanov, Krassimira Ivanova, Ognian Kounchev,
Maroussia Bojkova, Plamen Mateev

1. Introduction

The advertisements' detection and processing, as part of the media content analysis (MCA), has a number of uses and offers significant benefits to companies, organizations, different agencies and – particularly those that receive wide media coverage. MCA [1] is increasingly used commercially because of the key roles of the mass media. Fig. 1 provides an overview of the four roles and uses of MCA mainly within the two areas – strategic planning and evaluation. Since the advertisement TV block is media part of video stream and also a key in MCA, advertisements' detection and recognition is very important.

The media analysis of companies shows how the customer's company is represented in the media – which most represents the company, which experts comment company or the product, and also the same analysis for the competitive institution in order to compare their publicity parameters with those of the customer's com-

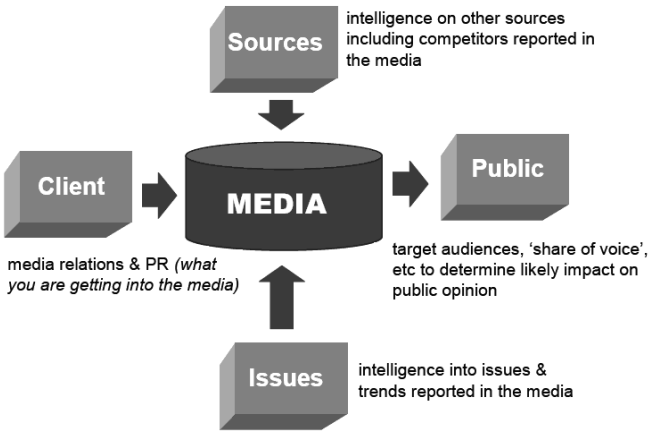


Figure 1: The four roles of media content analysis [1]

pany. The considerable part of the TV announcements of the customer's company and/or competition companies is the broadcast of theirs advertisements. The advertisements are broadcasted on television channels every day in advertisements' blocks, spread within other program elements (Fig.2). Companies pay a lot of money to place their advertisements on certain channels and in certain time slots. This way companies can assure that their advertisements were broadcasted.

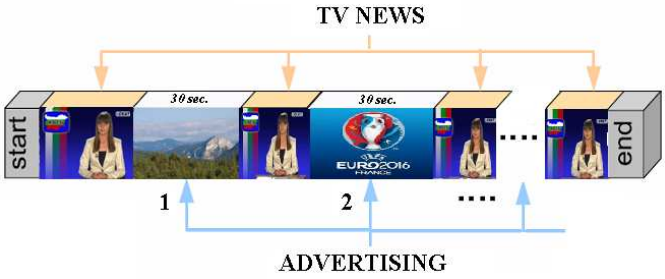


Figure 2: Example of a video stream with advertisements blocks

As it is mentioned in [2] there are some specifics of advertisements. Some of them are directly measurable, such as:

- repeated video sequence;
- restricted temporal length (generally between 10–60 sec);
- higher hard cut rate (scene changes);
- absence of correlation between consecutive scenes (due to camera and viewpoint change).

Other features are indirectly measurable, such as:

- high action rates (high motion & normalized difference energy);
- short shot lengths;
- drastically change of the visual style (like dominant colours and light);
- removal of the network-logo during advertisement blocks;
- turning up the volume of the audio signal during advertisements (in spite of the fact that some laws try to forbid this manner).

Another feature, mentioned in [2], connected to separating the consecutive advertisements by 5–10 blank frames, is no longer observed maybe because of better use of time for advertisements.

It should be noted that, the broadcasting of the advertisement can be transformed (usually by removing parts of it) depending on the viewer's preferences or

the TV time relevant. In [3] it is shown an example of detection and recognition of advertising trademarks from TV video stream of sport media.

2. Problem formulation

The problem can be formulated as follows:
Input:

1. A set of duration 30min video in MPEG4 format (25 fps; resolution 448x336 pixels), where the broadcast of 24 hours (or only prime time) of TV program is saved;
2. A set of advertisement video templates (about 30sec) in the same format.

Constrains:

The algorithms need to be appropriate with respect to time processing and hardware.

Output:

The time locations of a given advertisement template in the recorded TV stream, if this advertisement was broadcasted in the recorded day.

3. The team propositions

3.1. First scenario – Direct frame-by-frame pixel comparison

The simplest, but the slowest method for checking if a frame (image) of a given template video is contained in another video stream can be done by direct frame-by-frame pixel comparison of the two video sets. Thus, the maximum number of comparison will be $N_0 \left(\sum_{j=1}^{J} N_j \right)$, where N_0 is the number of frames in the video

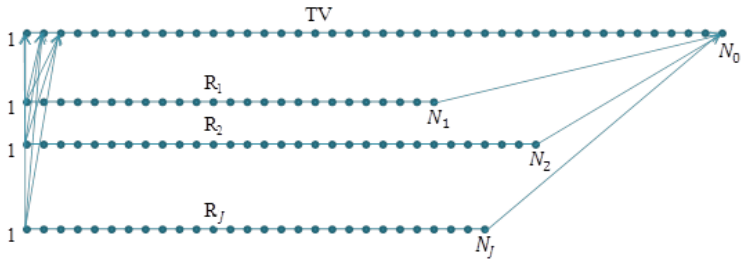


Figure 3: The direct frame-by-frame pixels comparison

stream; N_j is the number of frames in the j-th advertisement; J is the number of the observed advertisements. Fig. 3 shows the complexity of the algorithm. According to time processing this approach is not appropriate for us.

3.2. Second scenario – Speed-up, knowing JPEG's DCT coefficients in advance

Using DCT (Discrete Cosine Transform) coefficients of 8x8 region(s) from the JPEG frames, some acceleration in the frame comparison function could be achieved. This is true, because many of the DCT coefficients in 8x8 regions are zeros, and we will compare less numbers than pixels in the same region. In spite of this, the maximum number of comparison remains as in previous point. Furthermore, using DCT coefficients would have sense if we could extract them directly, but not computing them again.

3.3. Third scenario – Localizing the advertisement block and applying other scenarios in that block

The localization of the advertisements' blocks is very important in order that it solves more complex task – to focus the attention only on the frames of the advertisements blocks. This way, applying some more complicated algorithms only on this restricted area (searching for some objects as company's logo, slogan, some specific objects that are typical for the company or observed branch, etc.) one can find not only known, but also the new advertisements of the observed company.

In this case there can be analysed the place where the network logo is shown, taking into account the fact that during the advertisements blocks this logo is removed.

3.4. Fourth scenario – Matching the scenes duration of the advertisement and the TV stream

We stop our attention on the scenario, which decreases the complexity of the algorithm by replacing the frame-to-frame comparison with a scene-to-scene matching. This algorithm gives the advantages in case when the cardinality of the set of observed advertisements is bigger.

We represent the videos as 1D chain of numbers, which characterize the duration (in number of frames) of separate scenes of the videos. We consider a scene as a portion in a video where there is no sharp change between two consecutive frames. Thus, we compare the scenes duration belonging to the frames, and only for the suspicious scenes (which are able to match) we perform one-to-many frames comparison. This strategy has the big benefit of a much smaller number in-frame comparisons than previous cases.

4. Description of the proposed algorithm for the fourth scenario

Here we propose one simple and fast approach for scene detection analysing the difference between each two consecutive frames. This approach ensures detection of whole advertisements and/or arbitrary their pieces in a TV video stream.

Our algorithm consists of four steps:

1. Script the video streams as number vectors, representing the difference between neighbour frames (differential videos).
2. Split the stream to sequences of scenes and represent as number vector containing the durations (in frames) of each scene.
3. Comparison of number series (instead of frame series) between TV stream and potential candidates of scenes from the advertisement set.
4. In case of matching or inclusion of time interval from TV stream with time interval of some advertisement: frame comparison of representative frame from TV video with the frames in scene-candidate from the advertisement.

4.1. Forming differential videos (for TV stream and for advertisements)

The differential video can be estimated as:

$$V_{diff}(n) = \frac{1}{X \times Y} \sum_{\forall(x,y)} f\left(C_n(x,y), C_{n+1}(x,y)\right),$$

where

$n = 0, 1, \ldots, N-1$ are frame numbers in an observed video stream (TV record or advertisement);

$C_n(x,y)$ is the examined characteristic of a pixel (x,y) from the n-th frame. The examined characteristics can be luminosity, RGB colour, etc.;

$X \times Y$ is the frame size in pixels;

f is a function of the difference between $C_n(x,y)$ and $C_{n+1}(x,y)$. The function f can be defined in different ways.

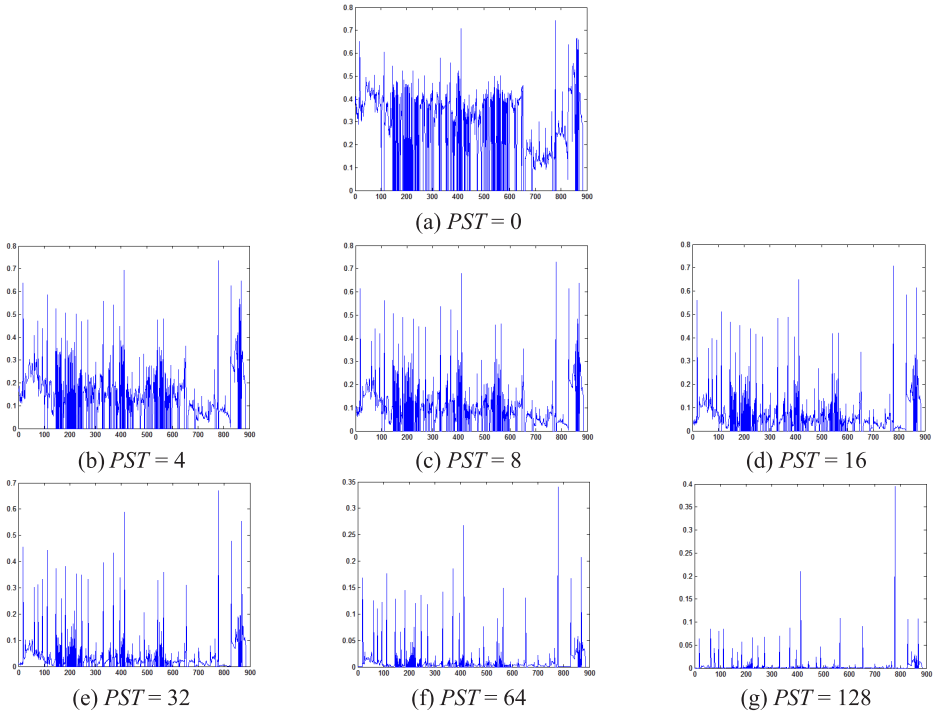
The simplest one is to calculate the absolute difference of the colour characteristics values:

$$f = \left|C_{n+1}(x,y) - C_n(x,y)\right|.$$

In order to get better true scenes separability from the in-scenes noise, a Pixel Similarity Threshold (PST) can be applied [6] [7]. This way the function f can be defined as:

$$f = \begin{cases} 1 & \text{if } \left|C_{n+1}(x,y) - C_n(x,y)\right| > PST, \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 4 shows the alteration of the resulting sequence for different values of *PST*.



(a) $PST = 0$ (b) $PST = 4$ (c) $PST = 8$ (d) $PST = 16$ (e) $PST = 32$ (f) $PST = 64$ (g) $PST = 128$

Figure 4: The results from applying different values of the threshold PST

As we can see, when the threshold *PST* is too low, the considerable noise from the background is leaked into the foreground. On the other hand, when the threshold is too high, information from the foreground can be lost, since the system understands it as background. The objective is then to find such a threshold *PST* where the most information from the foreground pixels remains while the level noise is reduced.

Applying this algorithm on the TV video stream and on the advertisements, we receive:

- for the TV video stream: $(\mathrm{I}_{diff}(1), \ldots, \mathrm{I}_{diff}(n-1))$;
- for the j-th advertisement $(j = 1, \ldots, \mathrm{J})$, respectively: $(\mathrm{R}^{j}_{diff}(1), \ldots, \mathrm{R}^{j}_{diff}(n-1))$.

This algorithm for scene split guarantees the split in equal manner for the TV stream and for the advertisements.

4.2. Scenes detection and representation of videos as integer chains

The scene separation is connected with the process of finding the outliers in the integer chain obtained in the previous step (Fig. 5). The simplest approach to mark outliers is to choose a threshold thr_1, which is the same for the TV video and for the advertisement.

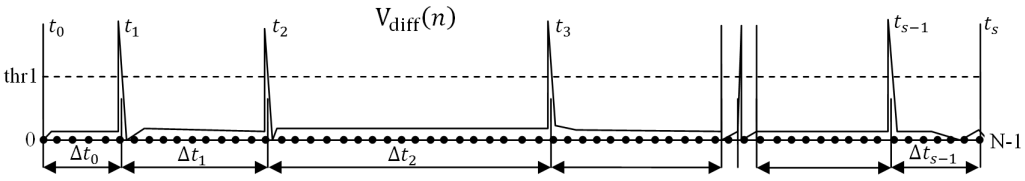


Figure 5: Visualisation of the process of scene detection

If $V_{diff}(n) > thr_1$ then the new scene starts $t_i \leftrightarrow n$; $\Delta t_i = t_{i+1} - t_i - 1$ is the length (in frame numbers) of i-th detected scene; $i = (0, \ldots, s-1)$; s is the number of scenes.

This way, the i-th scene is the chain of frames in the interval $[t_i; t_{i+1}] = [t_i; t_i + \Delta t_i]$.

The value of thr_1 can be obtained using different approaches. One possible variant is the statistical one based on the interquartile range ($thr_1 = Q_3 + 1.5(Q_3 - Q_1)$ [6]) of the learning sample. There are different datasets that can be used for this purpose:

1. to give as a learning set the advertisement stream. This would be good in case if we search only for one advertisement, because the calculation of thr_1 will be fast (this stream is no more than a minute) and also will take into account the exact specifics of the observed advertisement. For searching a set of advertisements this will lead to recalculating of the TV-chain for each advertisement, which will extremely slow up the process;
2. to give as a learning set the concrete observed video stream. This approach is also not good, because it is a slow process from one side, and all the advertisements also have to be recalculated each time from the other side;
3. to determine once on a training sample containing streams from various times of day and different TV channels. Because of the specifics of the advertisements, already mentioned in [2], this approach also is not so good;

4. to determine once using as a training sample a set of advertisements that are broadcasted in a certain time by different TV channels. This way the threshold will be obtained at once and will take into account more precisely the specifics of the advertisements.

After applying this algorithm on the TV video and on the advertisements we receive a set of chains:

- the TV chain: $I_{chain} = (\Delta t_0, \Delta t_1, \ldots, \Delta t_{s-1})$,
- the advertisements chains: $R^j_{chain} = (\Delta \tau^j_0, \Delta \tau^j_1, \ldots, \Delta \tau^j_{s_j-1})$,

where Δt_i, $i = 1, \ldots, s-1$ are the durations (in number of frames) of respective consecutive scenes in the TV stream and $\Delta \tau^j_k$, $k = 1, \ldots, s_j - 1$ is the similar but for the j-th advertisement stream, $j = 1, \ldots, J$.

Below we show the results of one experiment of scene detection and the creation of time-interval chain for 20 sec advertisement and for 15 min TV stream.

The function that determines the absolute difference of pixels (x, y) between n-th and $n+1$-th frame is:

$$f = |R_{n+1}(x,y) - R_n(x,y)| + |G_{n+1}(x,y) - G_n(x,y)| + |B_{n+1}(x,y) - B_n(x,y)|,$$

where R, G, B are resp. Red, Green, Blue values of the corresponding pixels.

The threshold thr_1 is calculated as IQR-outlier boundary using as a learning set the advertisement.

Figures 6 and 7 show the results of scene detection and creation of time-interval chain for 20 sec advertisement.

Note, that the breaks of the first frames into separate scenes is due to the fact that in the beginning of the advertisement the frames include moving of objects and background simultaneously, which leads to increasing the difference between neighbour frames.

Figures 8 and 9 show the result of scene detection and time-interval chain for 15 min TV stream of TV talk show. It is widely seen the increasing of the intensity of scene changes in the advertisement block.

The work of obtaining optimal value of thr_1 (constant or variable) has to continue in order to overcome splitting into too low scenes. But the algorithm must keep the property to split the scenes in equal manner in TV video and in the advertisements. Only this way we can reduce the task to the algebraic one.

4.3. Detection of potential matching or inclusion of scenes from TV in advertisements

In order to escape the damages of the length of the first and the last time intervals (because of the noising from neighbour broadcasts) they are excluded

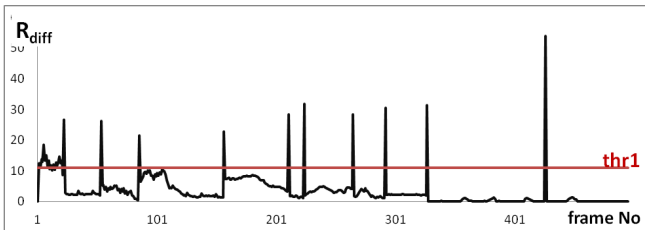


Figure 6: Scene detection of 20 sec advertisement

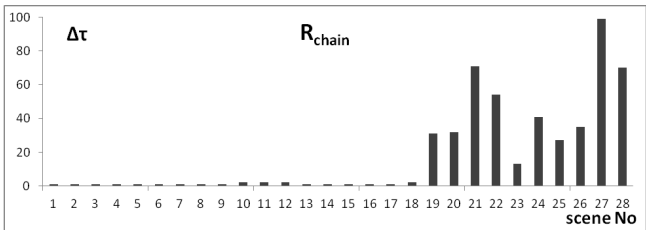


Figure 7: The time-interval chain that represents scenes duration for this advertisement

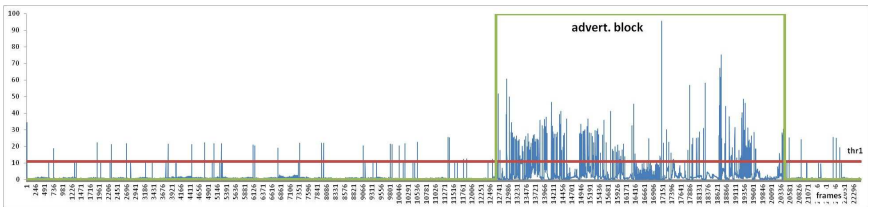


Figure 8: Scene detection of 15 min TV stream with one advertisement block

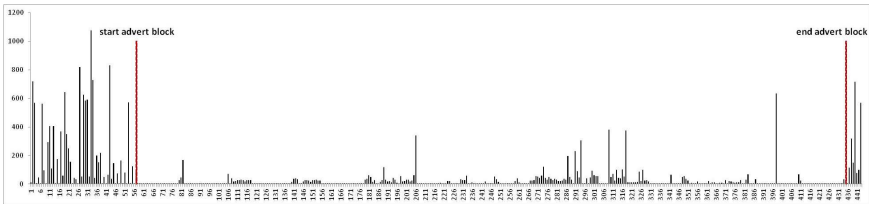


Figure 9: The time-interval chain that represents scenes duration for this TV stream

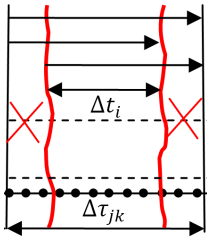


Figure 10: Cases of matching/inclusion of a scene from TV stream to a scene from the advertisement

from the observation. As it was mentioned in [2], one of the specific features of the advertisements is higher hard cut rate, which assures the bigger probability R^j_{chain} to be unique [7].

Main rule: One scene from a TV video can be part of a scene in a given advertisement (i.e. to match or to be shorter), while the inversed situation is not always valid.

This determines the searching direction for scenes comparison from a video stream to the set of advertisements.

The necessity of this rule arises because of the cases when we are given only the full advertisement template, but not the shorter variants, which consist of part(s) of the whole scenes $\Delta\tau^j_i$ from the full advertisement.

Usage of shorter variants is a regular practice of including the advertisements in the TV stream due to duration limitations, prices, thematic limitations, number of already made broadcasts of this advertisement, etc.

Thus, we can define four possible cases of inclusion of the scene Δt_k from video broadcast:

1. The Δt_k scene exactly matches with the $\Delta\tau^j_i$ scene from j-th advertisement;
2. Only the beginning of the Δt_i scene matches with the beginning of $\Delta\tau^j_k$ scene;
3. Only the end of the Δt_i scene matches with the end of $\Delta\tau^j_k$ scene;
4. The beginning and the end of Δt_i scene are internal for the $\Delta\tau^j_k$ scene. The Algorithm is applicable also for the cases when more than one scene from video is internal for a given scene from the advertisement.

Shortly the algorithm can be explained as:

Scan the scenes from TV stream consecutively ($\Delta t_i, i = 1, \ldots, s-1$)
If a previous scene was recognized as a scene from the j-th advertisement
Scan from the current scene $\Delta\tau_k^j$ for possible inclusion (comparing scenes duration)
If yes: continue with frame-to-frame comparison for adopting/rejecting hypothesis for finding a scene from the j-th advertisement (algorithm is explained in the next step)
If no: start searching for other advertisement comparing from the beginning of the scenes of each advertisement from the set
If a previous scene was not recognized as a scene from the j-th advertisement
Comparing Δt_i with scenes $\Delta\tau_k^j$ traversing all $j = 1, \ldots, J$ and $k = 1, \ldots, s_j - 1$ until find some scene as possible candidate or exhaust the scenes from the advertisements set.
If there was scene-candidate: continue with frame-to-frame comparison for adopting/rejecting hypothesis for finding a scene from the j-th advertisement.

4.4. Comparison of a frame from observed TV video scene with the frames in scene-candidate from the advertisement (if it exists)

This part is applicable when in the previous step the algorithm extracts the scene from the advertisement to be potentially matching or covering the observed scene from the TV stream.

Let:

- i be the index of observed scene from TV stream and k_j is k-th scene of the j-th advertisement that was nominated as candidate;
- f_1 and $f_{\Delta t_i}$ be respectively first and last frame from the i-th scene of TV stream;
- φ_x, $x = 1, \ldots, \Delta\tau_k^j$ be a frame from the scene-candidate (k) from the j-th advertisement.

In order to cover easily all possible variants of scene inclusion we make the frame comparison in the following manner (Fig. 11):

1. start with comparison of $f_1 = \varphi_1$
2. if yes – it covers first variant or second variant of inclusion
3. if no – start with comparison of $f_{\Delta t_i} = \varphi_{\Delta\tau_k^j}$
4. if yes – it covers the third variant of inclusion (the first one have to be already detected)
5. if no – consecutively continue comparison of: a) f_1 with the next frame from the beginning of the advertisement scene (inc(x):1,...,y) and b) $f_{\Delta t_i}$ with the previous scene from the end of the advertisement scene (dec(y):$\Delta\tau_k^j$,...x) until one of two situations arises:
   a. the equal frames are found – i.e. the scene of TV video belongs to the advertisement;
   b. two counters x and y meet each other – i.e. there was no frame in the advertisement scene equal to the first or last frame of the TV video.

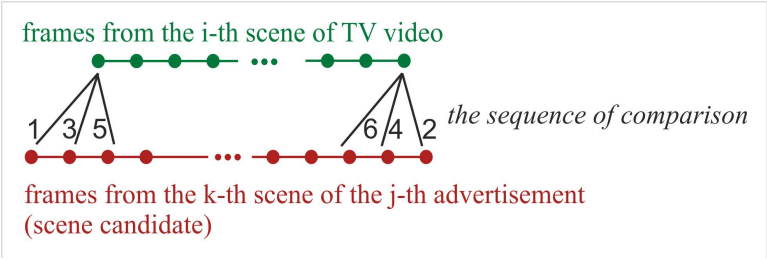


Figure 11: The sequence of frames comparison between scene from TV stream and scene-candidate from the advertisement

Conclusions

The observed approach ensures detection of the advertisements in a TV video stream for appropriate time processing and storage requirements.

MATLAB proved to be an important tool when developing prototypes due to its built-in video processing and mathematical tools. For real time implementation the use of lower level languages is required.

The first experiments of the program, realized on C#, showed the promised results.

References

[1] Macnamara J. Media content analysis: its uses, benefits and best practice methodology. In: Asia Pacific Public Relations Journal, vol. 6, no. 1, pp. 1–34, 2005.

[2] Tanwer, A. and P. S. Reel. Effects of threshold of hard cut based technique for advertisement detection in TV video streams. In: Proc. of the 2010 IEEE Students Technology Symposium, 03–04 April 2010, Kharagpur, India. DOI: 10.1109/TECHSYM.2010.5469157

[3] Lamberto B., B. Marco, and J. Arjun. A system for automatic detection and recognition of advertising trademarks in sports videos. In: Proc. of the 16th ACM Int. Conf. on Multimedia, pp. 991–992, 2008.

[4] Marcenaro, L., G. Vernazza, C. S. Regazzoni. Image stabilization algorithms for video-surveillance applications, IEEE Int. Conf. on Image Processing, 2001, Vol. 1, pp. 349–352.

[5] Dimov, D., A. Nikolov. Real time video stabilization for handheld devices, In: Rachev, B., A. Smrikarov (Eds.) Proceedings of CompSysTech'14, June 27, 2014, Ruse, Bulgaria, also in 2014 – ACM International Conference Proceeding Series, (to appear)

[6] Han, J. and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufman Publisher, Elsevier, 2006.

[7] Cover, T., J. Thomas. Elements of Information Theory, 2nd ed., John Wiley and Sons, 2006.