Detection of Panoramic Takes in Soccer Videos Using Phase Correlation and Boosting

Luiz G. L. B. M. de Vasconcelos
Research & Development Department
Globo TV Network
Rio de Janeiro, Brazil
Email: luiz.vasconcelos@tvglobo.com.br

Sergio L. Netto
PEE/COPPE/POLI/DEL
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
Email: sergioln@lps.ufrj.br

Eduardo A. B. da Silva
PEE/COPPE/POLI/DEL
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
Email: eduardo@lps.ufrj.br

***Abstract*—We propose an automatic engine for panoramic-take detection which relies on an algorithm based on phase correlation and boosting. The motion between two sequential video frames is first estimated through a phase correlation. Then, we are able to extract motion parameters and apply post-processing operations on these parameters in order to feed an AdaBoost-based classifier. The proposed algorithm has been validated over 5 segments of videos of 10000 frames each. Panoramic frame detection achieved around 85% recall and 76% accuracy on a validation set of videos not belonging to the training set.**

I. Introduction

The amount of sports-related multimedia has increased substantially over the years, due to the fact that advances in technology have made it easier to capture, store and retrieve videos. The audience's interest in sports-related content, especially soccer, has grown in a similar manner. Together, these trends point to the necessity of the development of efficient and effective tools to reduce viewers' efforts in searching for what interests them. This led to an increase in the interest in the video summarization and retrieval research area has received more and more investments.

This paper presents an algorithm that is able to detect when a panoramic image occurs during a soccer match. This kind of detection is useful as during a soccer match TV production, teams employ several types of camera takes. For example, at a moment of potentially decisive action the camera take is usually panoramic, but after the moment passes the camera switches to a non-panoramic mode, such as a close-up or an audience take. Figure 1 presents some examples of panoramic and non-panoramic takes.

Fig. 1: Snapshots of panoramic, close up and audience takes.

According to [1], soccer is classified as an MVS (Multiple View Semantics) sport since a single camera position is not able to capture the entire action, on the other hand DSV (Dominant Semantic View) sports, such as tennis, only need one position to do the task.

Among several methods of estimating the camera motion, [2] presents a technique that assumes that the camera motion can be defined by a 2D affine model. However, it is based on an adaptive IRLS (Iterative Reweighted Least Squares) algorithm, which is known to be a computationally expensive algorithm.

An alternative is to estimate the motion through phase correlation described by [3]. It uses only FFTs and frequency-domain multiplications operations, which are much simpler and more efficient than those proposed by [2]. The outputs of the motion are post-processed an fed to an AdaBoost classifier.

This paper is organized as follows: This section outlines the proposed system as well as the video database used during system calibration. Section II discusses how to extract camera motion features and post-process them to obtain useful data. Section III presents the boosting training stage to combine all features extracted as well as to give them weights in order to optimize the error rate. Section IV shows the experimental results taken on a different set of videos from those used in training. Finally Section V draws conclusions and discusses future work.

A. System Overview

The system inputs the soccer match video and outputs labels for each video frame indicating whether it is a panoramic take or not. Figure 2 shows the flowchart of the proposed system which can be separated into two stages: data preparation and classification.

B. Database

Table I shows the video segments that have been used during development, training and validation. They are 2009 Confederations Cup matches held in South Africa. All of them are NTSC standard videos, which implies in a frame rate of 29.97 frames per second and a dimension of 720 columns per 486 lines. Video 1 was used during the technique development in order to do signal and post-processing analysis. Segments 2, 4, 6, 8 and 10 were used during the training stage and 3, 5, 7, 9 and 11 during validation. Notice that although training and validation segments are from the same matches, they are

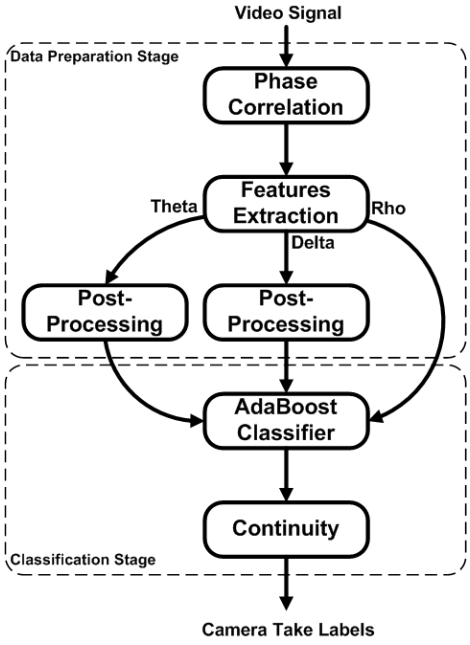

Fig. 2: Conceptual block diagram of the proposed panoramic detection.

from different parts of the video, and therefore able to provide a reliable validation.

TABLE I: Set of videos used during the technique development, training and validation.

Label	Match	Duration
Segment 1	Brazil x United States	1000 frames
Segment 2	Brazil x Egypt	10000 frames
Segment 3	Brazil x Egypt	10000 frames
Segment 4	Brazil x United States	10000 frames
Segment 5	Brazil x United States	10000 frames
Segment 6	Brazil x Italy	10000 frames
Segment 7	Brazil x Italy	10000 frames
Segment 8	Brazil x South Africa	10000 frames
Segment 9	Brazil x South Africa	10000 frames
Segment 10	Spain x United States	10000 frames
Segment 11	Spain x United States	10000 frames

II. Camera Motion Estimation

Two sequential panoramic frames tend to have few differences once all objects displayed on screen are small. However, in scenes of close-up and audience, the objects are large, tending to present quite noticeable movements. This points to the possibility of detecting a panoramic frame based on motion.

A. Phase Correlation

According to [3] is possible to analyze the motion between two sequential frames through Equation (1) where $C(x, y)$ is the 3D correlation map that shows the dominant motion. It means that the dominant motion will appear as a peak on a map position, where x and y represent the horizontal and vertical displacements, respectively.

$$C(x, y) = \mathcal{F}^{-1}\left[\frac{\mathcal{F}_1\mathcal{F}_2^*}{|\mathcal{F}_1\mathcal{F}_2^*|}\right] \quad (1)$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are the Fourier transforms of the adjacent frames and $\mathcal{F}^{-1}$ is the inverse Fourier transform [4].

The map origin is on position (0, 0) which means that a right- and bottom-direction motion causes a peak close to the origin. However, using DFTs, a left- and top-direction motion causes a peak close to the edges of the map due to spectrum repetitions. For an easy understanding and handling of map data, adjustments on 3D correlation map have been done which consisted on inversion of quadrants of the map to thereby place the origin of the map (0, 0) always at the center as shown in Figure 3.

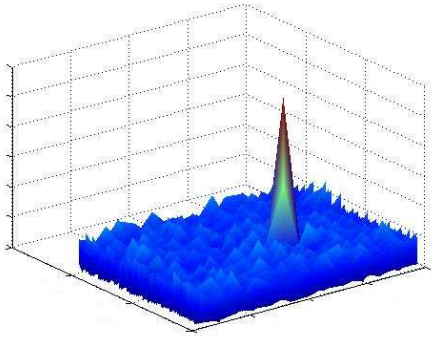

Fig. 3: 3D map derived from phase correlation.

B. Motion Features Extraction

Once the 3D map is adjusted, the next step is to extract information to measure the motion between two frames. At first, we conjectured that using the horizontal and vertical distance from the peak to the center of the map would provide the best performance.

However, by performing a rectangular to polar transformation, as indicated by Equations (2) and (3) below, we will get more meaningful information.

$$\Delta = \sqrt{x^2 + y^2} \quad (2)$$

$$\theta = \arctan\left(\frac{y}{x}\right) \quad (3)$$

This is so because the magnitude of the vector drawn from the origin to the peak, named as Δ, will be described as the size of the motion, while the angle of this vector related to a

reference, named as θ, will be described as the direction of the motion. Moreover, the magnitude of the peak of the 3D correlation map, named as ρ, can determine how well-defined that motion is.

So, in order to analyze the behaviour of Δ, θ, ρ and any other features derived from them by post-processing operations, we have employed Segment 1 described in Section I-B. Figure 4 shows how Δ, θ and ρ evolve with time. There, it is possible to notice the segment divided into four parts. Parts marked as 1 and 3 are composed of non-panoramic frames and the ones marked as 2 and 4 are composed of panoramic frames.

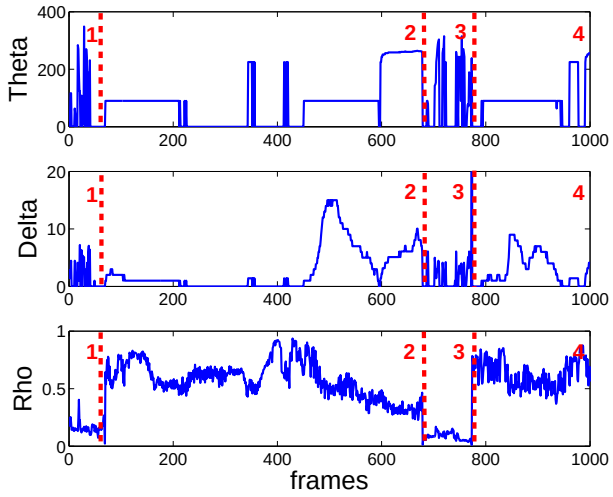

Fig. 4: Δ, θ and ρ timelines.

In Figure 4 it is noticeable that on non-panoramic parts Δ and θ signals vary with a frequency considerably higher than in panoramic parts. This can be explained by the fact that in closes-up and audiences scenes, for example, the objects are larger and move in larger displacements and more directions than in a panoramic take which implies in a large chance of the previous frame to be very different from the present.

Moreover, is also noticeable that on non-panoramic cases ρ presents smaller values than in panoramic cases. The reason is that non-panoramic parts tend to have more and larger movements, and it is not possible to determine a well-defined motion peak in the 3D correlation map.

C. Post-Processing Features

The previous section showed features containing interesting information to the detection of panoramic images. However, a close analysis shows that the ρ signal is the only one that can be used as it is. The other two (Δ and θ) need post-processing before can be input to a classifier.

In the analysis performed above, it was possible to notice that panoramic parts present stable Δ and θ signals. One way to explore such stability is to apply a variance-based operation. However, to do that, we should define a window where the statistics of the signals will be calculated. So, we use a rectangular window of length N moving sample by sample through the signal calculating the variance inside each window. The length N of the window is determined experimentally. For an NTSC video standard (29.97 frames per second) we have adopted the value $N = 15$ because it is unlikely to find out a transition between two different camera takes inside this period. Even if a transition occurs, post-processed signals will not be affected considerably once the past samples will only be used for the variance calculation during half a second. In addition, N can not be much smaller because this would allow, for example, a large difference in variance values for close frames inside the same panoramic view.

Even after post-processing operations, there are parts of the signal where there may occur confusions determining if it is panoramic or not. In order to alleviate this problem, we resorted to an AdaBoost classifier, described in the next section.

III. AdaBoost Classifier

In spite of the fact that in Section II we managed to extract useful features for the panoramic detection task, considered individually these features are not sufficient to reliably classify a take as panoramic or not. Then, they should be considered jointly in order to provide good classification performance.

Among several classification methods we opted to employ Boosting, especially Adaptive Boosting [5], which is widely used. The main idea is that is possible to build a strong classifier from a set of weak classifiers, as described in [6].

There are several AdaBoost-type classifiers, such as Real [7], Gentle [8] and Modest [9] AdaBoost. All of them have been investigated in our work. We used the implementation in the GML AdaBoost Matlab Toolbox available at [10].

During the training stage and analysis, Segments 2, 4, 6, 8 and 10 were used for training while Segment 1 was used for testing during development.

A. Input Data & Training Stage

The first idea is to feed the AdaBoost classifier with the data extracted in Section II, that is ρ, variance of Δ and variance of θ. Figure 5 shows an error rate around 27% for the simple AdaBoost features configuration. The AdaBoost classifiers in the GML toolbox have two main settings, they are the tree depth, that will be set to 3 for this technique, and number of iterations, which means the number of times that AdaBoost learners and weights will be adjusted.

However, AdaBoost classifier itself has no memory in its structure. Therefore, since the evolution of the classification across frames also matters, this implies that we should create a mechanism to also input to the AdaBoost classifier a temporal neighborhood of a frame as well.

Since AdaBoost allows as many features as desired, we solve this problem by also inputing to it features from neighboring frames. Figure 6 shows the error rate for features drawn from a window of 0 (no memory at all) up to 100 neighboring frames. The Gentle and Real AdaBoost algorithms outperform the Modest type in all cases. As the number of iterations

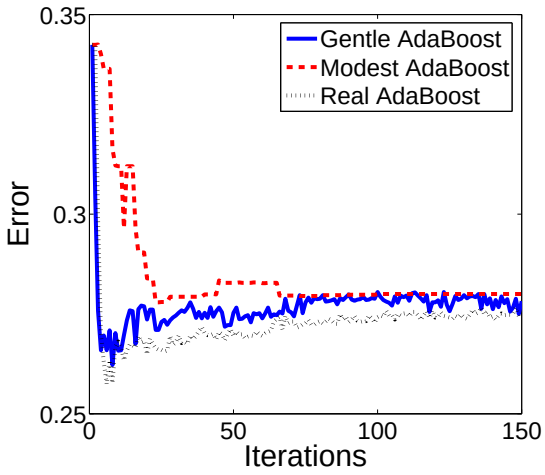

Fig. 5: Performance of the AdaBoost classifiers against the number of iterations for the initial configuration proposed.

increase, the Modest AdaBoost's performance gets closer to the one of the Gentle and Real, but remains inferior. The Gentle and Real AdaBoost algorithms have similar performances, reducing the error rate to 18.5% with 28 past and future samples.

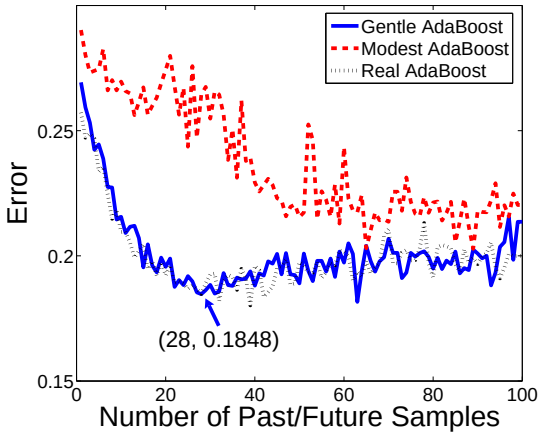

Fig. 6: Performance of the AdaBoost classifiers against the number of neighboring frames providing features for the classifiers.

After optimizing the number of past and future samples, we have to find the optimal number of iterations. From Figure 7 one can see that after around 40 iterations, the error rate does not vary significantly with the minimum obtained at 42 iterations. Once again, the Gentle and Real AdaBoost algorithms performed similarly, yielding an error rate of 17.8%.

B. Continuity

After analyzing the classification output signals, we noticed that significant number of classification errors occur in areas where there is a great deal of variation in the classifier output. The middle graph of Figure 8 shows such a behavior.

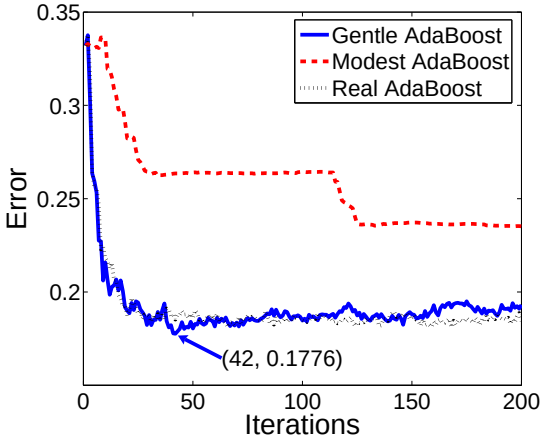

Fig. 7: Performance of the AdaBoost classifiers against the number of iterations for the optimum number of past and future frames.

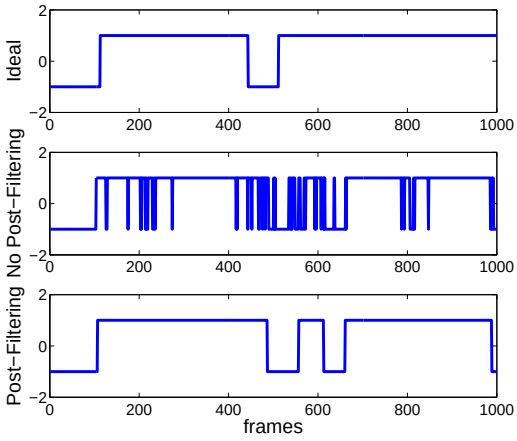

Fig. 8: Results for panoramic classification: ideal, without post-filtering and with post-filtering.

An easy way to overcome this problem is to apply a median filter on the classification. In other words, we classify a frame as panoramic or not by a majority vote among the classifications of M neighboring frames. The effectiveness of the post-processing by the median filter can be assessed in Figure 8. This figure suggests that the median filter is quite effective, in providing a decrease in classification error.

Once verified that median filter succeeds in reducing classification errors, we should define its length M. So, in order to find out the best value for the minimum error rate, Segment 1 has been classified for many window sizes from 1 up to 60 as shown in Figure 9. The window size $M = 29$ provides the best error rate of around 14%.

Although most results show that Gentle and Real AdaBoost perform similarly, Gentle AdaBoost has shown a slightly better error rate during filter size determination. Therefore, we opted to use only the Gentle AdaBoost for the rest of the

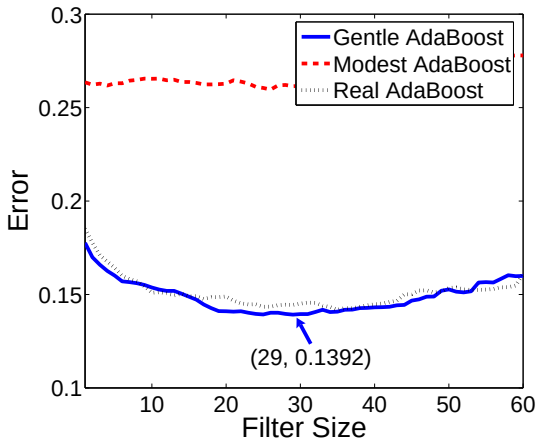

Fig. 9: Performance of AdaBoost classifiers after post-processing by a temporal median filter, against the median filter size.

experimental validation.

IV. Validation

In this section we validate the technique developed in the previous sections, including the several parameters that were determined experimentally. In order to do so, we used a different set of video segments, that had not been used during the development stages. Table II assesses the proposed techniques using two measurements: accuracy rate, which quantifies how many samples have been correctly classified, and recall rate, which indicates the number of panoramic frames that have been correctly classified. In other words, accuracy rate means the overall precision of the technique while recall rate means the technique precision for panoramic takes.

TABLE II: Accuracy and Recall rate for validation videos.

	Without Median Filter		**With Median Filter**	
	Accuracy	**Recall**	**Accuracy**	**Recall**
Segment 3	82.01%	84.13%	84.68%	88.43%
Segment 5	72.61%	89.27%	72.99%	91.72%
Segment 7	68.31%	79.93%	67.19%	80.59%
Segment 9	77.52%	76.54%	79.58%	79.29%
Segment 11	73.69%	84.11%	76.40%	87.51%
Mean	74.82%	82.80%	76.16%	85.51%
Std Deviation	3.95%	3.64%	4.86%	4.45%

Table II assesses two versions of the proposed technique: with and without a median filter at the output of the AdaBoost classifier. For most cases the median filter improves accuracy rate as well as recall rate, suggesting that its use tends to improve the performance of the proposed classifier.

Validation results have shown around 76% of mean accuracy rate using median filter with mean recall rate of around 85%. It is important to note the stability of the proposed techniques since the standard deviations in Table II are only around 3 to 4%.

V. Conclusion

This paper proposed an automatic panoramic take detection algorithm based on motion estimation between two sequential frames feeding a machine learning algorithm.

For that, we have performed motion estimation via phase correlation, providing motion information that has been post-processed and then input to an AdaBoost classifier. After parameter optimization, we have verified that the use of features from neighboring frames is beneficial. Moreover, we have found that a median filter applied to the AdaBoost classifier output improves the classification performance.

Finally, once the technique and its parameters have been defined, validation experiments have been performed. Results showed that technique achieved around 76% accuracy rate and 85% recall rate. Considering that only motion features have been employed, this is a reasonably good result.

One should also bear in mind that the panoramic frame detection is not an end in itself. It is intended to be used as a building block in the development of a video summarization and retrieval framework. For example, other features, e.g., audio features [11] can be included in a complete system, which will tend to improve the classification performance. In this context, the obtained results are quite encouraging.

Acknowledgments

The authors would like to thank Globo TV Network for providing videos used in this research.

References

[1] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros, and I. Sezan, "Browsing sports video: trends in sports-related indexing and retrieval work," *Signal Processing Magazine, IEEE*, vol. 23, no. 2, pp. 47–58, March 2006.

[2] F. Coldefy and P. Bouthemy, "Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2004, pp. 268–271.

[3] D. Pearson, *Image Processing*, D. Pearson, Ed. Mcgraw-Hill, 1991.

[4] P. Diniz, S. Netto, and E. D. Silva, *Digital Signal Processing: System Analysis and Design*. New York, NY, USA: Cambridge University Press, 2002.

[5] R. E. Schapire and Y. Freund, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[6] Y. Freund and R. E. Schapire, "A short introduction to boosting," in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1999, pp. 1401–1406.

[7] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," 1999.

[8] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of Statistics*, vol. 28, p. 2000, 1998.

[9] A. Vezhnevets and V. Vezhnevets, "Modest adaboost-teaching adaboost to generalize better," in *Graphicon*, 2005.

[10] L. GRAPHICS, "Gml adaboost matlab toolbox." [Online]. Available: http://graphics.cs.msu.ru/en/science/research/machinelearning/adaboosttoolbox

[11] L. Vasconcelos, S. Netto, L. Biscainho, and C. Prado, "Marcao automtica de eventos usando sinal de udio em transmisses esportivas de TV," in *Anais do 6o. Congresso de Engenharia de udio AES-Brasil*, vol. 1, 2008, pp. 58–64.