Clustering-based Object Detection for Low-resolution Video Streaming

Luca Superiori, Olivia Nemethova, Markus Rupp
Institute of Communications and Radio-Frequency Engineering,
Vienna University of Technology, Austria
Gusshausstrasse 25/389, A-1040 Vienna, Austria
Email: {lsuper, onemeth, mrupp}@nt.tuwien.ac.at

Abstract

This paper presents a novel strategy for the detection and tracking of objects in low resolution video sequences. The processing is performed in run-time, considering only few buffered frames. Our approach consists of three main steps: (i) spatial segmentation by means of clustering, (ii) candidate set reduction based on feature extraction (iii) choice of the best candidates. The searching region is chosen adaptively, considering the position and features of the object in previous frames. We evaluate the proposed method on a set of diverse soccer sequences. The results indicate excellent performance of our method in comparison to other approaches, especially in terms of robustness against false error detections.

Keywords: *Mobile TV, Video coding and processing.*

INTRODUCTION

Detection and tracking of small objects in low resolution video sequences is a rather challenging task. Several ideas have been published recently, focusing the detection of ball and/or players in soccer videos [1–4]. In Fig. 1, examples of ball appearance are shown. Due to downsampling (and possibly compression), the ball results to be an object having varying shape and colour reduced to several pixels only, difficult to describe by means of a template.

Figure 1. Typical ball in CIF resolution videos

Thus, object contour tracking or template matching algorithms alone does not perform well. Even after combining with trajectory tracking, these detection methods [3, 5] do not provide sufficient robustness against false object detections due to occlusions, scene cuts or similarity with other objects. Furthermore, trajectory tracking methods typically assume knowledge of the ball positions over future frames, which is not acceptable for run-time processing (important for example for live broadcasting/streaming preprocessing).

To overcome such problems, in this paper we propose a novel and robust approach for detecting small objects in low resolution videos and demonstrate its performance on ball detection in soccer sequences.

The block scheme of the proposed method is depicted in Fig. 2. It consists of three basic components: clustering performed in HSV colour space, description of candidate clusters by an enhanced set of low-complexity features, choice of the best candidate based on the features tracking (instead of position tracking like in [3]) over few past frames. The latter provides robustness against false detections even in the case of occlusions.

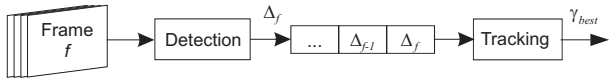


Figure 2. Block scheme of the object detection method

The basic novelty of our approach lies on the tracking procedure. The decision over time depends not only on the *trajectories*, i.e. the spatial positions assumed in time by the candidates, but rather on a *path* analysis. Each path consists of the succession in time of several properties of the candidates belonging to the considered path. The robustness of the tracking is enhanced by the higher number of comparisons needed to append an item to a path.

After presenting related works, the detection and tracking mechanisms is introduced. Results and summary concludes the paper.

RELATED WORKS

Several concept have been published in the last years regarding object detection mechanism. These techniques find application in various fields, such as medical imaging, unsupervised robotic, traffic analysis etc. In this work we deal with the detection of object whose shape is just roughly defined a priori and, therefore, the decision depends on the behavior of candidates in time. In such case, template matching techniques based on simple metrics as then Mean Square Error comparison cannot be used.

An example of such application is the ball tracking. In the pixel domain the ball shape reflects only approximately a circle. Detection mechanisms based on circle

detection, as proposed in [1], or independent component analysis [2] are therefore unsuitable for low resolution videos.
Other approaches are based on trajectory detection. In this case the detection is performed after combining both the information obtained from a single frame and information gathered from the observation of samples over a collection of frames. Such approaches are used in [4,6], but under the assumption of the availability of future frames, allowing the algorithm to base the choice on the knowledge of future ball positions.
A good example compatible with our requirement of live video streaming is represented by RoboSoccer applications. In this case the scenes are recorded by a camera and analyzed real-time by a DSP (Digital Signal Processing) both placed on the robot player. Also in this applications neither threshold nor edge detection methods appear to be efficient. The method proposed in [7] consists of a clustering algorithm able to segment the recorded frame and detect the orange ball used in RoboSoccer. We choose a modified clustering process, considering the fundamental difference between the two environments, as basis for our implementation and built our own tracking mechanism.

DETECTION FRAMEWORK

A block diagram of the proposed detection methods is depicted in Fig. 3.

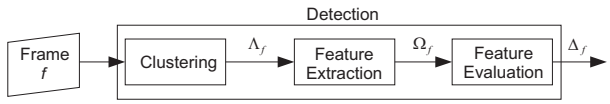


Figure 3. The detection framework

For each frame, a spatial segmentation of each picture is performed by means of clustering. The resulting clusters are evaluated using a feature extraction algorithm as described in the following.

Clustering

Fig. 4 shows the block diagram of the clustering mechanism.

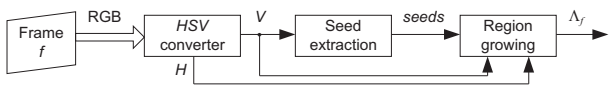


Figure 4. Functional diagram of clustering method

In order to reduce the intrinsic redundancy of the RGB space, we decided to perform the clustering in the HSV colour space. The three components represent the *hue* (the colour pure tone), the *saturation* (the intensity of gray) and the *value* (the colour lightness), respectively. The clustering process is then performed by means of a hybrid thresholding and region growing algorithm [7,8].
The *value* component (V) is used to extract the cluster *seed*. Assuming that the pixels associated to the ball are light, then a set of pixels whose lightness exceeds a threshold, depending on the average frame *value* , are selected.
The *seeds*, ranked for increasing lightness values, are then starting points used to produce clusters by means of a region growing. For each seed, the tone and lightness component of its neighbouring pixels are compared to those of the seed. In case of compatibility, the two are melted in a *cluster* or *region*. Each *cluster*, whose components are the average of the one characterizing the pixel belonging to it, is then further compared with its surrounding pixels. For the specific application of ball detection, the *saturation* component is not taken into account. In the performed simulations, considering the saturation does not improve noticeably the performance of the clustering algorithm, despite a slight increase of the computational complexity.
The result of the clustering process over a frame f is a set of clusters $\Lambda_f = \{\lambda_{f,i}\}$.

Feature Extraction and Evaluation

For each frame, the clusters Λ_f are processed in order to extract a set of properties. The result of the feature extraction is a set of *candidates* $\Omega_f = \{\omega_{f,i}\}$. Each candidate consists of a binary map of pixels (Fig. 5(a)). The white region represents the pixels belonging to the cluster (*items*). The rectangular map is then created around the items by surrounding them with *holes* (black pixels). Additionally, the following properties are included in the description of the candidate.

Position : It represents the cluster center position,

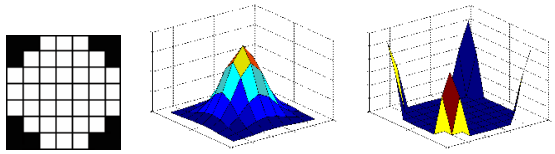
Size : It indicates the candidate dimensions (represented by its vertical and horizontal size).

The following two features are directly related to the characteristics of objects to be detected (in our application the ball). These extracted properties are therefore processed by means of proper feature evaluation.

Shape :

- *Axis Ratio* [9]: Indicates the proportion between the two axis. For a ball it should be near to one.
- *Density* [9]: Is the ratio between the number of *items* and the total candidate rectangular area. It should be similar to the ratio of a circle area to the area of the square it is inscribed into ($\pi/4$).
- *Roundness*: Represents the similarity of the shape to a circle. Since direct shape comparison is not effective, we assume that the desired candidate has to be characterized by *items* in the center and *holes* on the angle (as shown in Fig. 5(a)). Deviations from this requirement are penalized by means of the two-dimensional function in Fig. 5(b) and Fig. 5(c).

Our experiments prove that the influence of the deviation from the desired shape values has to be considered



(a) Circle (b) Fill points (c) Empty points

Figure 5. Ball roundness

with respect to the candidate size: the bigger the candidate size is, the more its characteristics should reflect the one of a real ball. As shown in Fig. 1, wide angle shots lead to extremely small balls, the characteristics of which reflect only roughly the above defined features. In order to gauge this factor, each property is weighted by means of an exponential function

$$f(x) = e^{-(x/\mu)^k}, \tag{1}$$

where λ and μ are parameters depending on the candidate size.

A set $\bar{\Delta}_f = \{\bar{\delta}_{f,i}\}$ is then obtained by discarding the candidates $\omega_{f,i}$ that do not suit the required shape properties. Each $\bar{\delta}_{f,i}$ is then judged in term of its colour.

Colour : Represents the average cluster hue and value component. Due to downsampling, the resulting ball colour can not be white as expected but rather light green. Independently of the tone, for increasing values of lightness the colour becomes indistinguishable from white. We discard the $\bar{\delta}_{f,i}$ with colour values lying outside an interval centered on the background green. The amplitude of the range is a function of the lightness.

The resulting clusters $\delta_{f,i} \in \Delta_f$ are then ranked by means of a *ball score* parameter, obtained by summing up its shape and colour results.

TRACKING FRAMEWORK

Once a sequence of M frames has been processed (with M depending on the frame rate) the algorithm builds a temporal succession of the candidates $\delta_{f,i}$ called *paths*. Each candidate $\delta_{1,i}$ belonging to the first frame is used as root of a path. Each path is initialized by inheriting the spatial and colour properties of the cluster it is starting with.

Once the set of initial paths $\Gamma = \{\gamma_j\}$ has been created, for each of the following frames we evaluate the similarity between the properties of each candidate $\delta_{f,i}$ and each existing path γ_j. We first process the physical distance, that has to be below a certain threshold depending on the frame rate. In case of a single candidate path, the comparison is performed by means of Euclidean distance; otherwise a presumed position is predicted using the average x and y speed and then compared with the real one. In both cases we assume that the considered frame rate forces the ball to move predictably within two frames.

Subsequently, the colour, size, and shape characteristics of the candidates are evaluated with respect to the corresponding properties of each of the already built paths. In order to suit the variability caused by zooming and changing light condition, the path characteristics are averaged over the buffered properties of the last N frames, where N depends on the frame rate. Each cluster is then appended to the path that results to be the most compatible with its characteristics. If a path γ_j in frame f is associated to more than one candidate, then it is duplicated and each replica is connected to a unique candidate. If no path appears to be appropriate for a cluster, a new path is initialized.

Each of the previous comparisons returns a certain quality index — *path score*. The best path is chosen considering the maximum sum between *path score* and the average *ball score* of the path items.

Finally, the path corresponding to the maximum sum of path score and ball score, averaged over all clusters in path, is chosen.

RESULTS

The simulations were performed over a set of soccer video clips in CIF (352 × 288 pixels) resolution. The sequences were recorded at 15 frames per second, the sequence length was chosen to simulate the real camera shot time. Results are shown in Table 1.

Table 1. Detection and tracking performances

Seq.	#frames	#balls	*det*	*false*	*miss*	*acc*
fb1	141	130	124	1	5	95.4 %
fb2	141	137	132	1	4	96.4 %
fb3	141	135	128	1	6	94.8 %
fb4	141	136	131	2	3	96.3 %
fb5	191	126	117	11	7	92.8 %

For each of the investigated sequences we counted manually the number of frames where the ball was visible and not occluded or out of the shot (column 3). The following columns show the number of correct detections (*det*), the number of false detections (*false*), and the number of missed detections (*miss*). As an overall quality metric, the accuracy (*acc*) is used, indicating the number of correct detections compared to the number of frames where the ball was present.

The first four rows of Table 1 show the results for common wide angle shots, both with artificial or natural light, as shown in Fig. 6. The average ball size lies over 4 × 4 pixels and the uniforms of the two teams are easily distinguishable from the ball size and color.

The results are excellent, the number of false detection remain in all cases extremely low and the average accuracy is over 95%.

(a) Natural light (b) Artificial light

Figure 6. Common low resolution sequences

We tested also a worst case, where the shots were characterized by extreme wide angles as shown in Fig. 7.

(a) Wide sngle dhot (b) False detection

Figure 7. Extra-wide angle camera shot

The wide angle results in a ball of 2×2 pixels. Therefore, due to downsampling, the ball colour also appears to be green just lighter than the field. This effect can be observed in Fig. 7(a) where the position of the ball is indicated by an arrow. Moreover, during fast movements, the ball tends to fade to the background green and, in steady frames, it is not easily detectable by a human viewer either.

Another problem arises if one of the teams has uniform colors that can confuse the algorithm. Green shirts with white insert (arms, writing on the front and on the back) and white shorts as well as white socks surrounded by green fields can easily be detected as a false ball. In case of occlusion the thresholds are weakened and objects similar to the ball can cause false detection. Figure 7(b) shows an advertisement writing detected as ball during an occlusion.

However, the impact of those effects is strongly limited by the path analysis. Objects looking similar to the ball vary in their size, shape and colour in time depending on their exposure to the camera. Additionally, the algorithm works in a very conservative way.

Objects found after an occlusion (or, generically, after a certain number of frames where no ball was found) are not added to the path and, therefore, are not used to update the object properties, until the resulting path shows compatibility with the previous one.

In the considered sequence, the ball was occluded or not visible in over 34% of the frames. Despite a higher number of false detections, the algorithm was able to follow the real ball as soon as it was visible again.

CONCLUSION

This paper describes our current approach to detect small object (in the study case, a ball) in low resolution (CIF) video streaming. After detecting clusters by means of a hybrid seed region growing algorithm, we select the candidates whose colour and shape (roundness, axis ratio and density) are compatible with the characteristics of the searched object.

The final decision is then made comparing the candidate properties with the *ball path*, consisting of the evolution in time of the spatial trajectory, shape, size and colour of the tracked ball.

The higher number of comparisons limits the number of false detections even in case of occlusions. The algorithm works in a conservative way, updating the path properties only if the resulting path appears to be consistent. This reduces the effects of false detections.

The method shows promising results in case of standard sequences, it can result in higher number of false detections in some special conditions such as team uniform colors similar to the ball.

Acknowledgements

The authors thank mobilkom austria AG for technical and financial support of this work. The views expressed in this paper are those of the authors and do not necessarily reflect the views within mobilkom austria AG.

REFERENCES

[1] T. D'Orazio, N. Ancona, G. Cicirelli, M. Nitti: "A Ball Detection Algorithm for Real Soccer Image Sequences," Proc. of the 16th Int. Conf. on Pattern Recognition, (ICPR), Quebec City, Quebec, CA, Aug. 2002.

[2] M. Leo, T. D'Orazio, A. Distante: "Independent component analysis for ball recognition in soccer images," Proc. of the Int. Conf. on Intelligent Systems and Control (ISC2003), Salzburg, Austria, Jun. 2003.

[3] S. Jiang, Q. Ye, W. Gao, T. Huang: "A New Method to Segment Playfield and Its Applications in Match Analysis in Sport Video," Proc. of the 12th annual ACM Int. Conf. on Multimedia, New York, NY, Nov. 2004.

[4] X. Yu, Q. Tian, K.W. Wan: "A Novel Ball Detection Framework for Real Soccer Video," Proc. of Int. Conf. on Multimedia and Expo (ICME), Baltimore, US, Jul. 2003.

[5] O. Nemethova, M. Zahumensky, M. Rupp: "Preprocessing of a Ball Game Video Sequences for Robust Transmission over Mobile Network," Proc. of the 9th CDMA Int. Conf. (CIC), Seoul, Korea, Oct. 2004.

[6] X. Yu, C. Xu, H.W. Leong, Q. Tian, Q. Tang, K.W. Wan: "Trajectory-Based Ball Detection and Tracking with Application to Semantic Analysis of Broadcast Soccer Video," Proc. of Int. Conf. on Multimedia (MM'03), Berkeley, CA, USA, Nov. 2003.

[7] Z. Wasik, A. Saffiotti: "Robust Colour Segmentation for the RoboCup Domain," Proc. of 16th Int. Conf. on Pattern Recognition (ICPR), Quebec City, Quebec, CA, Aug. 2002.

[8] R. Adams and L.Bischof: "Seeded Region Growing," In IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pages 641-647, 1994.

[9] J.-R. Ohm: "Multimedia Communication Technology," Springer, 2004.