A Spatio-temporal Approach for Multiple Object Detection in Videos Using Graphs and Probability Maps

Henrique Morimitsu[1](✉), Roberto M. Cesar Jr.[1], and Isabelle Bloch[2]

[1] University of São Paulo, São Paulo, Brazil
[2] Institut Mines Télécom, Télécom ParisTech, CNRS LTCI, Paris, France
henriquem87@gmail.com

Abstract. This paper presents a novel framework for object detection in videos that considers both structural and temporal information. Detection is performed by first applying low-level feature extraction techniques in each frame of the video. Then, additional robustness is obtained by considering the temporal stability of videos, using particle filters and probability maps, which encode information about the expected location of each object. Lastly, structural information of the scene is described using graphs, which allows us to further improve the results. As a practical application, we evaluate our approach on table tennis sport videos databases: the UCF101 table tennis shots and an in-house one. The observed results indicate that the proposed approach is robust, showing a high hit rate on the two databases.

Keywords: Object detection · Structural information · Graph · Tracking · Video

1 Introduction

Several works address the detection of each object as an individual task, i.e. do not consider the possible relationships between objects. This approach is appropriate for certain tasks, such as image retrieval [12] or detecting everything that belongs to a given class [5]. It has also been applied for videos, in which case the most common approach is the use of background subtraction methods combined with a blob descriptor for people detection. Almajai *et al.* [1], for example, use such a method to extract player blobs from tennis videos. The blobs are then filtered by classifying 3DHOG descriptors.

However, individual detection is a difficult task, and often produces undesirable results that could be avoided by considering more information, such as the spatial relations. This approach has been explored lately for scene understanding in static images. The work of Choi *et al.* [4] shows that by detecting multiple objects and considering their relationships it is possible to improve the detection of each object as a whole. The idea of considering the global scene

A. Campilho and M. Kamel (Eds.): ICIAR 2014, Part II, LNCS 8815, pp. 421–428, 2014.
DOI: 10.1007/978-3-319-11755-3_47

configuration has been receiving great attention in the field of more complex activity recognition as well [7]. Wang *et al.* [14] encoded structural relations using Markov Random Fields in order to perform player action detection in tennis matches. Even though widely studied for action and activity recognition tasks, this approach has not been well explored for the task of object detection in videos. Perhaps the most similar field is that of multi-object tracking, where the inclusion of structural information has been showing some very interesting results [15].

In this work we present a framework for multiple object detection and tracking in videos using graphs and temporal information obtained from probability maps to improve the tracker performance. The process starts by performing detection in each frame of the video using classic low-level features, such as color and motion data. Videos in general present several challenges, such as arbitrary view-points, possibly moving camera, unstable, low-quality image (e.g., obtained by a cell phone camera). Therefore, the results obtained from this step are usually very noisy. In order to face these challenges, the temporal properties of the video are used to filter undesired detections. This step is accomplished by obtaining information from what is referred to as probability maps and also from trackers implemented using particle filters. A structural approach is also employed, where a graph is built using the detected objects. This allows the scene to be described using higher-level information considering the relationships between the objects.

The contributions of this paper are twofold. First, we present a framework that encodes structural and temporal information about the objects in videos using graphs and trackers. Secondly, we explore the collected information to improve object detection in videos that share a common structure.

In Sec. 2 we show how low-level features are used to obtain detection candidates. In Sec. 3 the use of temporal information is explored to remove inconsistent candidates. In Sec. 4 structural information is used to improve the detection. Then, the results of the methods applied on table tennis videos are presented in Sec. 5.

2 Low-Level Object Detection

The low-level detection is highly dependent on the type of video being analyzed. However, in the proposed framework, the only requirement is that the chosen detector produces several candidate detections, that must include the correct ones. Note that different detectors can be used for the same task in order to produce more candidates, if necessary. Examples of possible detectors include, but are not limited to: background subtraction methods for moving objects, HOG [5] for people, keypoints methods like SIFT [9] or keygraphs [11] for rigid objects, parts-based detector [6] for multiview scenarios.

3 Temporal Consistency

Videos, as opposed to static images, provide a very important information in the form of temporal consistency. In other words, it is expected that two consecutive frames do not present very large differences. This information can be used to improve current results by relying on past information. In this work, temporal consistency is considered by means of two approaches: probability maps and object trackers.

3.1 Probability Maps

Probability maps encode the regions of the images where each object is more likely to appear, assuming that videos are acquired from a single camera and without cuts.

The maps are built online, while the video is being analyzed. For that reason, the maps improve as longer video periods are considered. In order to build the maps, it is assumed that, even though the detectors produce some errors, they usually produce correct results. For each frame, detected object regions are accumulated in a voting map, causing more frequent regions to receive higher values. The probability map M^t at instant t is obtained by:

$$M^t = M^{t-1}\gamma + I_B^t(1-\gamma) \tag{1}$$

where $\gamma \in [0, 1]$ is a given temporal weighting factor and I_B^t is a binary image whose non-zero pixels represent regions where an object was detected. This updating method decreases the influence of older frames over time, thus coping with camera and object movements.

This approach is more suited for videos captured with a static camera. However, it may also be used on video that present not too abrupt movement by first applying a camera stabilization method, such as in [10].

3.2 Object Tracking

This work employs particle filters with the ConDensation algorithm that is implemented in OpenCV[1]. ConDensation uses factored sampling [8] on particle filters models in order to track objects. The particle filter tracking consists in estimating the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ of a set of weighted particles $\{(\mathbf{x}_t^i, w_t^i)\}$, where $\mathbf{x}_t^i$ is the state of particle i and w_t^i its weight, computed from the observed measurements $\mathbf{z}_{1:t}$ until the instant t. The ConDensation approach computes this distribution by generating a new set of n particles by sampling them, with repositions, from the old ones. By assuming $\sum_{j=1}^{n} w_t^j = 1$, the probability of each particle i being chosen in this step is w_t^i. Hence, more likely particles can be sampled several times, while others may not be chosen at all. Then, for each particle, a new state is predicted. The prediction phase involves two steps: drift and diffusion. Drift is a deterministic step, which consists in applying the motion dynamics for each particle. Diffusion, on the other hand, is random and it is used to include noise in the model. The new state of a particle i can be expressed as:

$$\mathbf{x}_{t+1}^i = A\mathbf{x}_t^i + B\mathbf{u} \tag{2}$$

where A is the motion dynamics matrix and $B\mathbf{u}$ is the noise term.

[1] http://opencv.org/

Finally, for each particle i, its weight is computed by:

$$w_{t+1}^{i} = \frac{p(\mathbf{x}_{t+1}^{i}|\mathbf{z}_{t+1})}{\sum_{j=1}^{n} p(\mathbf{x}_{t+1}^{j}|\mathbf{z}_{t+1})} \tag{3}$$

In this work, the state of each particle is a 4-tuple (x, y, h, w) consisting of the centroid (x, y) of the object bounding box as well as its height and width. As it is assumed that the initial states of the objects are unknown, a set of trackers $TR = \{tr_i\}$ is kept, one for each object detected in image I^t at instant t. The set TR is updated at each instant using a two-step approach. First, let $b(.)$ be the bounding box of an object and $A(b(.))$ be the area of $b(.)$. For each object-tracker pair (o_k^t, tr_i^{t-1}), the intersection ratio is computed:

$$r(o_k^t, tr_i^{t-1}) = \frac{A(b(o_k^t) \cap b(tr_i^{t-1}))}{\max\{A(b(o_k^t)), A(b(tr_i^{t-1}))\}} \tag{4}$$

After that, every pair such that $r(o_k^t, tr_i^{t-1}) < \tau_{area}$ is removed and all the remaining pairs are matched using a greedy approach. In order to match pairs in which the bounding box size incorrectly changed abruptly, a second step is performed by computing the distance $d(o_m^t, tr_n^{t-1})$ between every non-matched object o_m^t and tracker tr_n^{t-1}. Pairs are again filtered the same way as before. Finally, a new tracker is created and associated to each object that was not matched. In order to avoid that the number of trackers grow indefinitely, trackers that are not matched on τ_{window} consecutive frames are deleted.

4 Structural Properties

Let $\mathcal{O}_l = (o_1^l, o_2^l, ...o_m^l)$ be the set of detected objects using low-level features and temporal consistency. The goal is to consider high-level information in order to correct $\mathcal{O}_l$. It is assumed that the low-level detector produces an over-detection, i.e. all the desired objects are detected along with some possible misdetections. Therefore, the correction aims at removing inconsistent objects, yielding the new set $\mathcal{O}_h = (o_1^h, o_2^h, ...o_n^h)$ where $n \leq m$. Each object o_i^h is assigned to a class $c_i \in \Omega$ where $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ represents the set of classes, e.g. "person" or "car".

It is assumed that there is a correlation between the behaviors of the objects. Such information is encoded using attributed relational graphs (ARGs). In this work, an ARG is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \Sigma_{\mathcal{V}}, \Sigma_{\mathcal{E}}, C)$ where $v \in \mathcal{V}$ represents one object, and an edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ encodes the relationships between v_i and v_j. $\Sigma_{\mathcal{V}}$ and $\Sigma_{\mathcal{E}}$ are the attributes of $\mathcal{V}$ and $\mathcal{E}$, respectively. C is a $k \times k$ matrix which specifies whether there is a relationship between classes ω_i and ω_j or not. In other words, C is used to generate the set of edges $\mathcal{E} = \{(v_i, v_j) \mid c(v_i) = l, c(v_j) = m, C_{lm} = 1\}$, where $c(v)$ represents the class of vertex v.

By considering that the scenes always present a common set of objects with a common spatial structure, the problem of improving object detection can be reduced to a subgraph matching between the scene and model graphs.

More precisely, let $\mathcal{H}_s$ be a subgraph of the scene graph $\mathcal{G}_s$. The best $\mathcal{H}_s$ is computed as:

$$\arg\max_{\mathcal{H}_s} s(\mathcal{H}_s, \mathcal{G}_m) \tag{5}$$

where $s(\mathcal{H}_s, \mathcal{G}_m)$ is a score function between both graphs, comparing attributes of the two vertex sets and the two edge sets.

5 Results

5.1 Sample Application: Table Tennis Videos

We have chosen to work with table tennis videos, as an example where structural information may be used mainly because of the presence of the table. This object is very important to the game, as everything is organized around it. In that sense, it can be used as a reference to obtain a better global understanding of the scene.

5.2 Object Detection

For this application, the selected objects to detect were the table and the players. The table was detected by backprojection histogram matching [3] in HSV color space. The color model for the table was learned by evaluating several samples of tables. The players, on the other hand, were detected using a background subtraction approach. However, instead of building a background model, motion was detected by the absolute difference between every two consecutive frames. Afterwards, player blobs were obtained by applying a morphological closing with a large structuring element.

5.3 Graph Description

In this example, we set $\Omega = \{player, table\}$ and $C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, indicating that only the relation between table and player is considered. The videos represent singles matches (one player on each side of the table), hence the model graph $\mathcal{G}_m$ is a path of 3 vertices. From the rules of the game, it is known the ball trajectory must include the table. It is also known that from a top view of the game, the ball follows a nearly linear path from one player to another. Therefore, if the players are represented by the points p_1 and p_2 and the table by t, it is expected that the internal angle of the vectors $\overrightarrow{tp_1}$ and $\overrightarrow{tp_2}$ is as close to π as possible.

One common misdetection is to consider other moving objects or people as players. Therefore, the player probability map M_P is used in order to take into account if the player candidate is in a player zone. By taking everything into consideration, the score function is defined as:

$$s(\mathcal{H}_s, \mathcal{G}_m) = i(\mathcal{H}_s, \mathcal{G}_m) \left[\frac{\theta(\overrightarrow{tp_1}, \overrightarrow{tp_2})}{\pi} + \alpha \sum_{j=1}^{n} \mu(b_j, M_P) \right] \tag{6}$$

Table 1. Detection statistics on both databases. The first two rows represent the sum of detected objects along the whole video, while the third one is the ground truth.

	Local		UCF101	
	Table	Players	Table	Players
Detected	378	749	706	541
Hit	374	661	208	346
Expected	378	756	394	394
Hit / Expected	0.99	0.87	0.53	0.88
Hit / Detected	0.99	0.88	0.29	0.64

where

- $i(\mathcal{H}_s, \mathcal{G}_m) = 1$ if the graphs are isomorph, 0 otherwise;
- $\theta(\overrightarrow{tp_1}, \overrightarrow{tp_2})$ is the internal angle function between the two vectors;
- $\mu(b_j, M_P)$ is a relevance function, weighted by a given α, of bounding box b_j belonging to a player zone in player probability map M_P. This function is defined as: $\sum_{x,y \in b_j} \frac{M_P(x,y)}{w(b_j)h(b_j)}$, where $w(b_j)$ and $h(b_j)$ are the width and height of b_j. In this work α is experimentally chosen as 2.

5.4 Databases

The results were obtained from tests performed on two databases: one local database created for this research featuring two videos of amateur table tennis matches under different points of view and another with some videos from the UCF101 [13] table tennis shots. The results were evaluated by considering the hit rate of the detection given by the proposed method. As the videos are not annotated, the evaluation is performed manually by sampling one frame every 30, or around one frame per second, in each of the videos. A detection was considered a hit when $\min\left\{\frac{b_i(o)}{b_d(o)}, \frac{b_i(o)}{b_r(o)}\right\} \geq 0.5$, where $b_d(o)$ is the detected bounding box returned by the proposed method of object o, $b_r(o)$ is the real bounding box, or the smallest box that contains o, and $b_i(o) = b_d(o) \cap b_r(o)$.

Local Database. The local database consists of two videos recorded using a fixed camera of an amateur game viewed from two different points of view. The results in Table 1 show that the detector provided very good results on this database. Figure 1 shows some images of the observed results. As evidenced by the results, the table detector is robust to scale changes caused by the perspective, as well as different lighting conditions. The detector usually presents good results, finding the whole table, with just a few failures of partial detections. The players are also usually correctly detected, even under varying conditions of movement.

UCF101 Table Tennis Shots. This database is composed of 15 types of videos, which remained after removing videos where the table color was not blue. As these videos do not present real matches and only one player, the structural

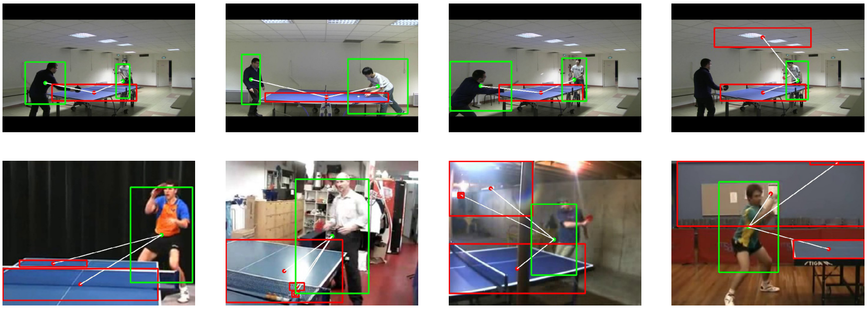

Fig. 1. Detection results for both databases. The first row shows the results for the local database, while the second for UCF101. The method works well even under changes of perspective and challenging lighting conditions. Even when only one player is detected and the structural properties are not used, the player and table are correctly detected among the possible candidates.

descriptor cannot be used, which means that an increase in the number of false positives is expected. The main interest of this database is therefore to check if the correct detection is also present among the false ones. If that is the case, then the correct detection could be found by using the structural descriptor later.

Some results are shown in Fig. 1 and are summarized in Table 1. As it can be seen, the players can be robustly detected even under the several changes present in this database. The table detector, on the other hand, was more highly affected by the changes. This is caused mainly by the large variety of table colors, as well as the very different lighting, which sometimes caused light reflections or shadows. It is worth noting that the tests were performed using only a single general color model, as opposed to learning a different model for each video, which would improve the results. Nonetheless, the detector showed satisfactory results detecting the table in more than half of the time, even under these conditions. As already mentioned, without the structural descriptor, sometimes other misdetections were present among the results, but the most important point is that the method finds the correct one.

6 Conclusion

We presented a framework to detect objects in videos. The detectors combine extraction of low-level features, temporal consistency and structural properties in order to obtain more robust results. They were tested on two databases containing table tennis videos: one created for this project and the challenging UCF101 table tennis shots, showing an average accuracy of over 75% for both table and player detection.

As future works, we intend to improve the detection of the table independently of its color. This could be done by using color quantization to segment the image and by searching for the region *between* [2] the players. Another ongoing research includes the use of action recognition in order to add more information to the process. This would work both ways, because the detection could benefit from information about the actions, as well as the action recognition step could be improved by better detection results.

Acknowledgments. The authors would like to thank FAPESP grants #2013-08258/7, #2012-09741-0, #2011/50761-2, CNPq, CAPES and NAP eScience - PRP - USP.

References

1. Almajai, I., Yan, F., de Campos, T., Khan, A., Christmas, W., Windridge, D., Kittler, J.: Anomaly detection and knowledge transfer in automatic sports video annotation. In: Weinshall, D., Anemüller, J., van Gool, L. (eds.) Detection and Identification of Rare Audiovisual Cues. SCI, vol. 384, pp. 109–117. Springer, Heidelberg (2012)
2. Bloch, I., Colliot, O., Cesar, R.: On the ternary spatial relation between. IEEE Transactions on Systems, Man, and Cybernetics SMC-B **36**(2), 312–327 (2006)
3. Bradski, G.R.: Real time face and object tracking as a component of a perceptual user interface. In: WACV, pp. 214–219 (1998)
4. Choi, W., Chao, Y.W., Pantofaru, C., Savarese, S.: Understanding indoor scenes using 3D geometric phrases. In: CVPR, pp. 33–40 (2013)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
6. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE PAMI **32**(9), 1627–1645 (2010)
7. Gaur, U., Zhu, Y., Song, B., Chowdhury, A.K.R.: A "string of feature graphs" model for recognition of complex activities in natural videos. In: ICCV, pp. 2595–2602 (2011)
8. Isard, M., Blake, A.: CONDENSATION - conditional density propagation for visual tracking. International Journal of Computer Vision **29**(1), 5–28 (1998)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**(2), 91–110 (2004)
10. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. IEEE PAMI **28**(7), 1150–1163 (2006)
11. Morimitsu, H., Hashimoto, M., Pimentel, R.B., Cesar Jr, R.M., Hirata Jr, R.: Keygraphs for sign detection in indoor environments by mobile phones. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 315–324. Springer, Heidelberg (2011)
12. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR, pp. 2161–2168 (2006)
13. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. CoRR abs/1212.0402 (2012)
14. Wang, Z., Shi, Q., Shen, C., van den Hengel, A.: Bilinear programming for human activity recognition with unknown MRF graphs. In: CVPR, pp. 1690–1697 (2013)
15. Widynski, N., Dubuisson, S., Bloch, I.: Fuzzy spatial constraints and ranked partitioned sampling approach for multiple object tracking. Computer Vision and Image Understanding (CVIU) **116**(10), 1076–1094 (2012)