A TRACKING REPOSITIONING ALGORITHM USING KEYPOINT LABELING

Rémi Trichet, Bernard Mérialdo
Institut Eurecom
BP 193, 06904 Sophia Antipolis, France
{remi.trichet, Bernard.Merialdo}@eurecom.fr

ABSTRACT

This paper reports some work undertaken during the development of a generic object tracking application based on a keypoint model. In our previous work, we proposed a keypoint labeling algorithm to distinguish object from background keypoints. This article is the continuation of this work and deals with the problem of handling object deformation using the keypoint labels. In consequence, we introduce an algorithm to refine the bounding box position according to the surrounding keypoint labels. Experimental results are shown to validate this theory.

1. INTRODUCTION

Our tracking system was developped in the context of the portivity project which is developing a converged rich media iTV system, integrating broadcast and mobile broadband delivery to portables and mobiles and which will enable the end-user to link rich media information with moving objects within TV programmes. This particular framework brings up some constraints to our application. For the annotation process to be efficient, the tracking has, first, to be as near to real-time as possible. Moreover, in order to limit the user required intervention, the object needs to be coarsely located within a bounding box. And finally, the tracker has to be generic, handling all types of videos and thus, all kinds of difficulties such as illumination changes, blur, affine transformations, fast object motion, occlusions, and so forth.

During the last few decades, a multitude of approaches have been developed to tackle the object tracking issue. The simplest of these, the block matching technique [1] proceeds in two steps. It starts by isolating the object of interest from the background. The object is then cut into rectangular blocks. The algorithm is based upon the correlation between such blocks in successive frames. The very fast results are balanced by a low tolerance to most of the difficulties and a rather imprecise localization of the object. In contrast, methods that track the object boundary [2] offers an accurate detection, but, it is obvious that highly distinctive contours are required. They use deformable contours, like “snakes”, that behave badly in the case of small or fast moving objects. Mesh-based methods [3] are a good compromise, in terms of precision, between the first two methods. They offer good object delimitation for a low computational cost. Their efficiency is directly dependent on the node detection process.

Approaches relying on a particular cue to achieve the tracking usually choose color or motion. Color information has the advantage of being immediately available, and thus easy to exploit. Segmentation based methods [4] or histograms perform in real-time but are highly sensitive to illumination changes and occlusions. On the contrary, gradient ascent methods, and especially the mean-shift algorithm [5][6] are nowadays considered as yielding the best results. The mean-shift approach first aims to build up a similarity map with the salient colors of the object. The local resemblance between the object and the studied area of the image is modeled with Gaussians probability density functions. The similarity map is then constructed by summing these kernels. The most probable position of the object will finally be obtained with a gradient ascent starting at its last known position. This approach is particularly well adapted for the tracking of small and fast objects and efficiently deals with occlusions. However, at least one salient color is required for the algorithm to work properly.

Motion is a feature used jointly with other characteristics by most of the applications. However, techniques relying on this cue are drastically different and constitute a separate branch. Some methods dubbed “predictive” analyze the anterior displacements of the object to predict the next. As regards Bayesian methods, they evaluate the potential displacements in terms of probability. They generally use Markov random fields, or particle filters [7][8].

All of these methods are specific to a given application and designed to deal with the stemming difficulties. The keypoint algorithms [9-12] overcome this flaw. These points are localized at strategic positions of the image, such as corners or extrema of a given function and enriched by local descriptors. They have been proved robust to usual transformations and are designed to deal with partial occlusions. In consequence, they have been the subject of intensive studies in the past few years [13-16]. Their advantages make the keypoints a promising tool for a generic tracking system. Nevertheless, their computation is expensive. But, in our specific context of an annotation system, this extensive calculation can be completed off-line.

We previously developed a generic keypoint-based object tracking system [17][18] in order to fulfill our project requirements. The principle consists in extracting the keypoints and their corresponding descriptors for consecutive frames in order to assess the global bounding box displacement thanks to its included keypoint motion vectors. Moreover, we also propose a keypoint labeling algorithm [19] in order to differentiate object keypoints from those of the scenery (see Figure 1).

Although our motion model remains reliable in the case of uniform motion, it suffers of some limitations when the object motion is not homogeneous on its surface: different motion vectors associated with different parts of the object are involved. This kind of situation usually takes place in the presence of deformable objects but can also arise with rigid ones under complex motion (rotation for instance). Differentiating background displacements from those of the object is then a much more delicate issue. Moreover, since the keypoint density is not generally homogeneous, a part of the object having a higher quantity of keypoints will bias the global object motion toward this part.

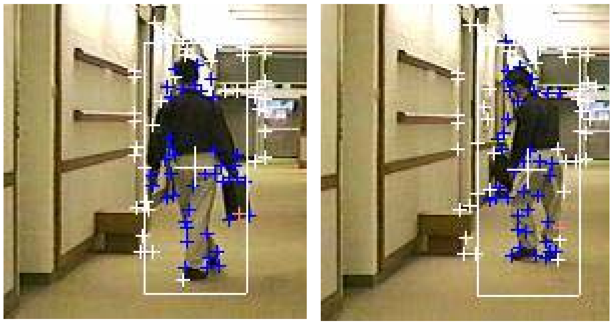
However, we have other data in order to support the bounding box repositioning which are the keypoint labels. Given that the motion model is sufficiently reliable, we can use it to assess the global object motion, and refine the bounding box repositioning in relation to the label of the surrounding keypoints.

The rest of the paper is organized as follows. The principles of the labeling algorithm are described in Section 2 as a pre-requisite for the bounding box position refinement algorithm detailed in Section 3. Section 4 shows experimental results studying the algorithm behavior under various conditions. Finally, Section 5 presents our conclusions.

2. LABELING ALORITHM

In the scope of an object tracking application, the object is often coarsely located within a bounding box. Due to this approximation, part of the information included in the bounding box, and hence, falsely considered as identifying the object, will thereby perturb the algorithm. Despite this coarse localization, the accuracy of the bounding box positioning remains, like for every tracking algorithm, one of our priorities. Thus, we propose a keypoint labeling algorithm in order to prevent the scenery information to be part of the tracking process.

Labeling the keypoints consists in differentiating object keypoints from those of the background in order to avoid the scenery influence like for a common background subtraction problem. The principle relies on allocating an "object" or "background" flag to each point. Only the "object" keypoints will latter be used to estimate the object motion. A classical labeling algorithm will consider that keypoints inside of the bounding box are "object" and those outside are "background". To improve the tracking performance, we refine this process by introducing for each keypoint a likelihood to belong to the object or the scenery. This likelihood will be a real value ranging from 0 to 1 determined on a four-features basis: the label of the matched keypoint, the color, the motion, and the position in relation to the bounding box. In order to increase the reliability of the labeling algorithm, the label allocation of keypoints for which there is a lack of information (some keypoints may remain unmatched) is postponed. An example of labeling is shown in Figure 1.



<u>Figure 1:</u> Labeling for frames 30 and 60 of the "surveillance" sequence. The "object" keypoints are in blue, those of the "background" in white, undetermined keypoints are in red.

However, in the case of a non cluttered background where all the extracted keypoints will certainly belong to the object, the basic algorithm mentioned above will be more efficient. The best solution to this issue being an algorithm adapting to the scenery, we have set up a clutter assessment measure based on the background keypoint rate. For a detailed version of the algorithms, please refer to [19].

3. POSITION REFINEMENT ALGORITHM

This section is dedicated to the algorithm description. The mechanism relies on the allocation of labels to the immediate inner and outer areas of the bounding box frame. The label of an area, that we call "quality" is calculated as the mean of all the included keypoints in the area. The principle consists in evaluating the quality of areas that are adjacent to the bounding box frame, for various sizes. If an exterior (respectively interior) area is deemed belonging to the object (respectively to the scenery), the bounding box is moved in consequence. It is obvious that the precision of the approach strongly relies on accurate keypoint labeling (which has been proven in [19]). This modification most of the time resulting from a local displacement of the object, and in particular of object contours, we choose not to integrate it in the object global motion computation.

Two factors influence this quality measure, first of all, by the number of considered keypoints, from which the measure reliability directly depends, and second, by the fact that this measure will be biased by our labeling algorithm. Indeed, this algorithm will consider that the area where the labels are uncertain is widespread inside of the bounding box, and restricted on the outside. Thus, there will be more "background" keypoints inside of the bounding box than "object" keypoints on the outside. The optimization of the bounding box in relation with the keypoint labels will then trend to shrink it.

Considering these behaviors, we have set up a bounding box optimization algorithm. For each of the four sides of the bounding box, we test the possibilities of a two- or four-pixel narrowing (or dilatation). If the area quality is higher than a given threshold *Tquality*, the shift is retained as a candidate. Afterwards, the best candidate is applied. The shrinking and dilating algorithm are the following:

```
Bounding box narrowing (initial threshold Tquality = 2×S)
    Add = 0.5-quality;
    If (Add > Tquality/nb)
        Save the modification;
        Tquality = add×nb;

Bounding box dilatation (initial threshold Tquality = S)
    Add = quality-0.5 ;
    Si (Add > Tquality/nb)
        Save the modification;
        Tquality = add×nb;
```

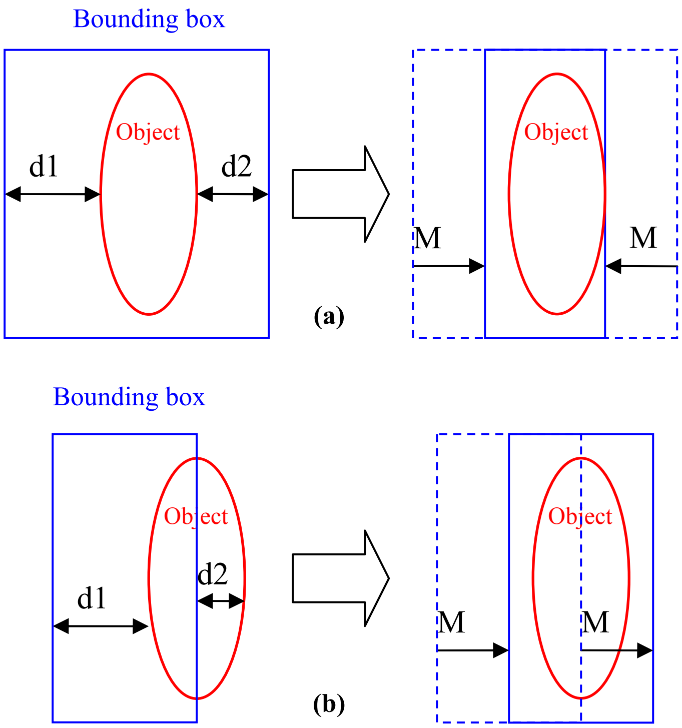
,where *nb* is the number of analyzed keypoints,

After this step, information about shrinking and dilating for each border of the bounding box is available. In order to limit excessive deformations, each axis is treated independently and the following repositioning rules are applied:

1- If the *d1* and *d2* border deformations are in opposite directions, the bounding box is then shrunk or dilated by *M* pixels according to the smallest magnitude of the two detected shifts. $M = min(abs(d1), abs(d2))$.

2- Else (the *d1* and *d2* deformations are in the same direction), the bounding box is then recentred according to these deformations by applying a displacement $M = (d1+d2)/2$ to each of these two borders.

This process can then change the scale or recenter the bounding box. A conceivable variant will be to perform the two modifications for the same frame, first, the scale shift, and then the center adjustment. However, we prefer not doing it in order to counter potential errors. The shift between two images being minimal, if recentering the bounding box is really needed, it will be redetected at the next frame. Figure 2 illustrates this process.



<u>Figure 2:</u> Bounding box optimization example based on the labeling (a) narrowing (b) recentering. *d1* and *d2* represents the detected deformations and *M* the performed modification.

The efficiency of this algorithm entirely relies on the adjustments of the *Tquality* threshold. It is initialized at a value twice more important for narrowing than for dilatation in order to favor the former in relation to the latter. Moreover, this threshold adapts itself in accordance to the number of keypoints involved. The higher the keypoint number, the more reliable the measure considered, and lower the threshold will be. For each retained modification, the threshold is updated in order to consider only the modifications with a better quality.

This algorithm is also adaptable in the context of non cluttered environment (few or none of the extracted keypoints belongs to the scenery). It is then based on the number of keypoints present in the analyzed area without considering their labels. If an area inside of the bounding box has no keypoints, a consequent edge reduction is memorized. And, if an area outer of the bounding box has more than one keypoint, we save a corresponding border dilatation. The priority is given to the dilatation rather than to the narrowing, and then to the candidate with the highest amplitude.

By applying the following rules, similar to those enunciated for a cluttered environment, it is possible to obtain a bounding box optimization algorithm reliable for non cluttered environments.

$$Reduction\ M = min(abs(d1)-p,\ abs(d2)-p)$$
$$Dilatation\ M = min(abs(d1)/d,\ abs(d2)/d)$$
$$Recentering\ M = (d1+d2)/2*d$$

The purpose of the number of pixels *p* and the *d* factor is to increase the reliability of the transformations by minimizing the modifications. Only the modifications detected on several consecutive frames, and so deemed reliable, will be fully compensated. We use *p=4* and *d=2*. The main advantage of this variant is the absence of parameters to adjust.

Nevertheless, a generic algorithm has to face cluttered environments as well as those that are uniform. It is also possible for the clutter to change during the video sequence (for instance, a character passing in front of a tree). Our algorithm has then to adapt as a function of the scenery. In order to do that, we use the clutter measure discussed in Section 2 of this article [19]. The threshold determining the choice of the algorithm to use is fixed to 5% clutter, like for the labeling algorithm [19]. This simultaneous use of the two algorithm variants leads however sometimes to lapses when the clutter rate is oscillating around the threshold. Indeed, the behavior of the two variants of the algorithm will be very different for a rather still configuration of the scenery. In order to harmonize the behaviors, we restrict the transformations from the algorithm operating in non cluttered environment to the recentering of the bounding box.

Figures 6 and 9 show examples of the algorithm for, respectively, bounding box rescaling and repositioning under cluttered environment. The method effectiveness for recentering the bounding box on the object in non cluttered scenery is illustrated by examples 7 and 8.

4. EXPERIMENTAL RESULTS

Two sets of tests have been conducted separately. First, we have experimented this algorithm in a cluttered environment for two types of keypoint having different densities (Harris [9][10] and Harris-Laplace [11] keypoints) as well as for different values of the *Tquality* parameter. The results are presented in Figures 3 and 4. We have also tested the algorithm in the case of non cluttered scenery on 3 video sequences. The results, shown in Figure 5, highlight the enhancement brought up by the label-based bounding box optimization algorithm in this context.

Whilst the results with the Harris keypoints are encouraging, those with Harris-Laplace keypoints are disappointing. This is due to a too weak keypoint density for the algorithm to be efficient. Moreover, the quality threshold, which is the only important parameter, has no drastic influence on the algorithm results, proving that the method is not constrained by parameter tuning. Nevertheless, this threshold represents the compromise between security and the amount of shifts. With a high quality threshold, only reliable changes will be made, missing small repositioning. On the contrary, with a low value of the quality threshold, lot of shifts will be made, but at the risk of some mistakes.

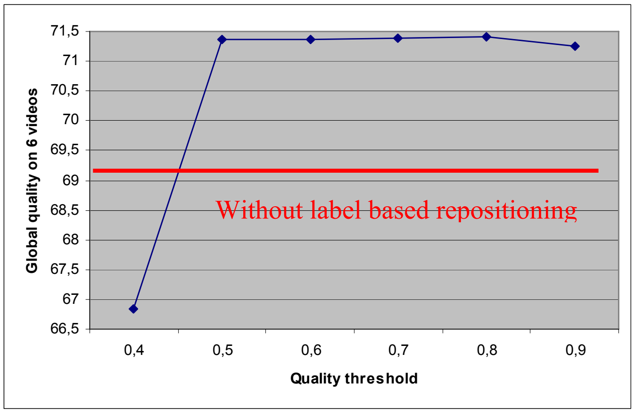


Figure 3: **Results for the bounding box optimization algorithm in relation to the Harris keypoint labels. Tests on 6 video sequences with cluttered background.**

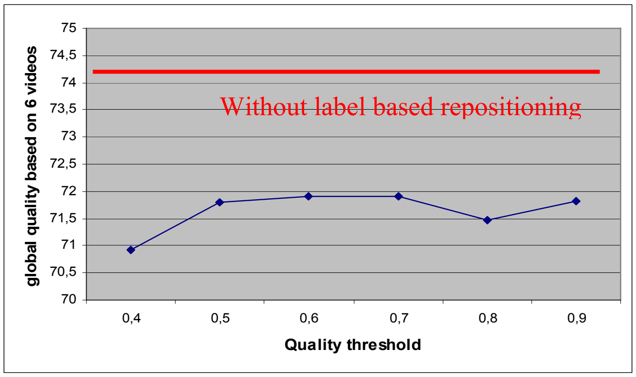


Figure 4: **Results for the bounding box optimization algorithm in relation to the Harris-Laplace keypoint labels. Tests on 6 video sequences with cluttered background.**

In the case of non cluttered environment, the algorithm improves the tracking system performances whatever the used keypoints are.

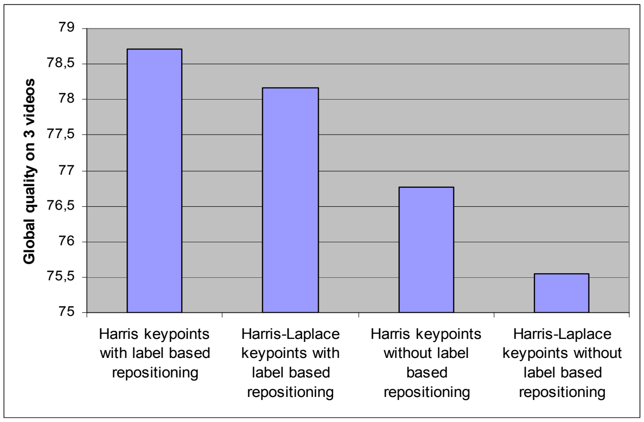


Figure 5: Results for the bounding box optimization algorithm in relation to the Harris and Harris-Laplace keypoint labels. Tests on 3 videos sequences with non cluttered background.

Notice that although this algorithm will be able to rectify minor trajectory errors from the motion model, it cannot replace it and will be useless in the case of global motion estimation failure. This process will handle object deformations but will be unable to deal with object loss or occlusions which have to be treated by some other algorithms.

5. CONCLUSION

This paper has presented a bounding box position optimization algorithm. This technique takes as input a label that is allocated to each keypoint. The process uses the label of the keypoints surrounding the bounding box frame in order to correctly center and scale it, if required. It has been shown to be a success but needs a sufficient keypoint density to be effective.

The results have proved that, in the case of correct global motion estimation, this algorithm is yielding a much more accurate bounding box positioning.

6. REFERENCES

[1] Gyaourova A., Kamath C., and Cheung S.-C., Block matching for object tracking, LLNL Technical report,. UCRL-TR-200271, October 2003.

[2] Techmer A., Contour-based motion estimation and object tracking for real-time applications, In International Conference on Image Processing, volume 3, pp. 648-651, Thessaloniki, Greece, 2001.

[3] Valette S, Magnin I, Prost R, Mesh-based video objects tracking combining motion and luminance discontinuities criteria, Signal Processing archive, Vol 84 , pp. 1213-1224, Issue 7, July 2004.

[4] Cavallaro A, Steiger O, Ebrahimi T, Tracking video objects in cluttered background, TCSVT, 15(4), pp.575-584, April 2005.

[5] Comaniciu D., Meer P., Mean Shift: A Robust Approach Toward Feature Space Analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24(5): 603-619, 2002.

[6] Jaffré G., Crouzil A, Non-rigid object localization from color model using mean shift, ICIP (3), 317-320, 2003.

[7] Isard M. and MacCormick J., BraMBLe: A Bayesian Multiple-Blob Tracker Proc Int. Conf. Computer Vision, vol. 2, 34-41, 2001.

[8] Pupilli, M., and Calway, A., Real-Time Camera Tracking Using a Particle Filter, In Proceedings of the British Machine Vision Conference, BMVA Press, 2005.

[9] C. Harris et M.J. Stephens, A combined corner and edge detector, In Alvey vision conference, pp147-152, 1988.

[10] P. Montesinos, V. Gouet, and R. Deriche, Differential invariants for color images, International conference on pattern recognition, 1998.

[11] K. Mikolajczyk, C. Schmid, indexation à l'aide de points d'intérêts invariants à l'échelle, Journées ORASIS GDR PRC, Communication Homme-Machine – May 2001.

[12] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F.Schaffalitzky, F. Kadir, L. Van Gool, A comparison of affine region detectors, International Journal of Computer Vision, Volume 65, Number ½, 2005.

[13] P. Gabriel, J.-B. Hayet, J. Piater, J. Verly. Object Tracking Using Color Interest Points, in Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance (AVSS'05), 2005.

[14] M. Kölsch et M. Turk. Hand Tracking with Flocks of Features, In Video Proc. CVPR IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[15] M. Brown, R. Szeliski and S. Winder. Multi-Image Matching using Multi-Scale Oriented Patches, International Conference on Computer Vision and Pattern Recognition (CVPR2005), pp 510-517, 2005.

[16] V. Garcia and E. Debreuve and M. Barlaud. Region-of-interest tracking based on keypoint trajectories on a group of pictures, In Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing (CBMI), Bordeaux, France, June 2007.

[17] R. Trichet and B. Mérialdo, "Generic Object Tracking for Fast Video Annotation", VISAPP, Barcelona, Spain, 2007.

[18] R. Trichet and B. Mérialdo, "Probabilistic Matching Algorithm for Keypoint Based Object Tracking Using a Delaunay Triangulation", WIAMIS, Santorini, Greece, 2007.

[19] R. Trichet and B. Mérialdo, "Keypoint Labeling for Background Substraction in Tracking Application", ICME, Hannover, Germany, 2008, Waiting for acceptance.

[20] Zhengyou Z. Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting, RR-267, Projet ROBOTVIS, Sophia Antipolis, October 1995.

[21] Charles V. Stewart, Robust Parameter Estimation in Computer Vision, SIAM review, 1999.

[22] E. Malis, E. Marchand. Experiments with robust estimation techniques in real-time robot vision. In IEEE/RSJ Int. Conf. IROS'06, pp. 223-228, Beijing, China, October 2006.

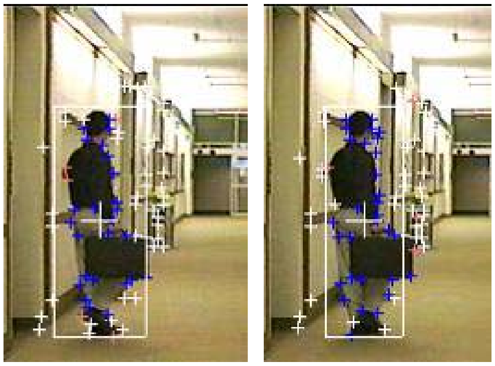


Figure 6: bounding box rescaling. Frames 7 and 8 of the "surveillance" sequence. The "object" keypoints are in blue, those of the "background" in white, undetermined keypoints are in red.

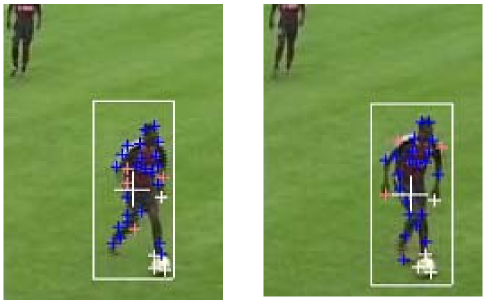


Figure 7: bounding box recentering. Frames 18 and 21 of the "soccer" sequence. The "object" keypoints are in blue, those of the "background" in white, undetermined keypoints are in red.

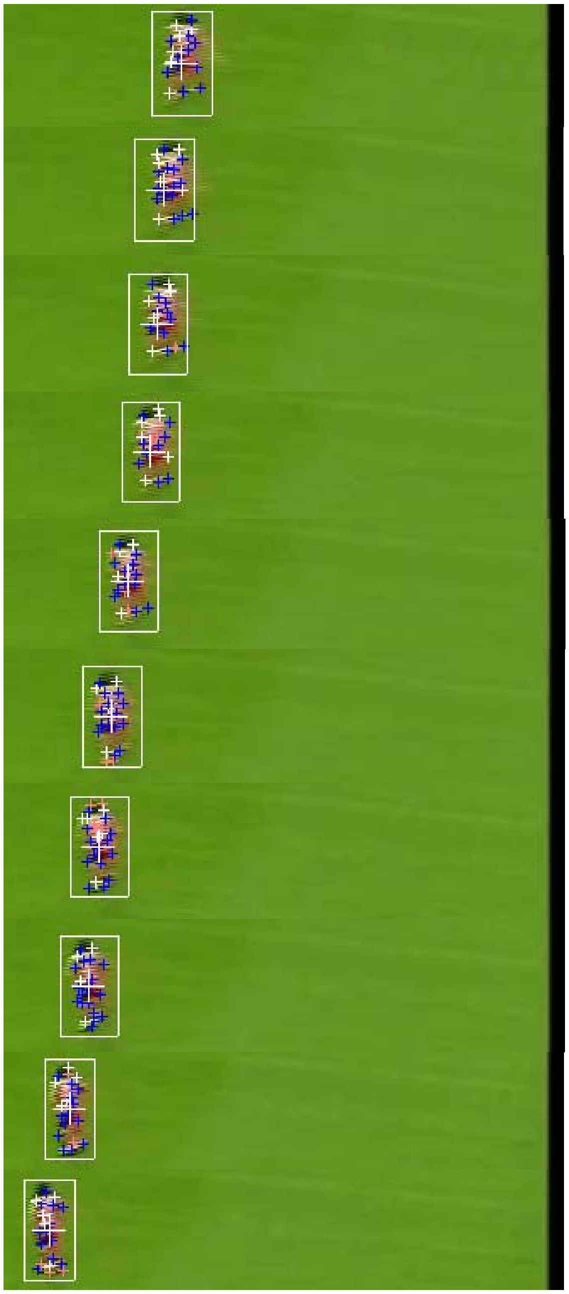


Figure 8: Tracking of a blurred football player thanks to the bounding box recentering. Frames 20 to 30 of the "football from above" sequence. T The "object" keypoints are in blue, those of the "background" in white, undetermined keypoints are in red.

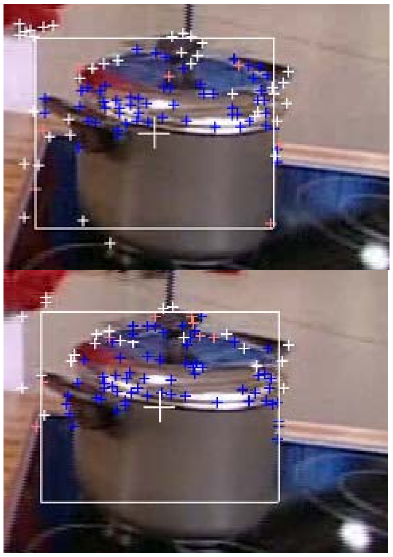


Figure 9: bounding box repositioning 4 pixels to the right. Frames 35 and 37 of the “cooking” sequence. The “object” keypoints are in blue, those of the “background” in white, undetermined keypoints are in red.