Accelerated Keypoint Extraction

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

Abstract

Keypoints have become a fundamental feature in image and video processing. This article presents a wavelet-inspired structure to speed up the keypoint extraction process. The method restricts the extraction process to high variance areas solely, deemed likely to provide keypoints. The variance is calculated at different scales according to a hierarchical structure inspired from the Haar wavelets. This structure is also used for a fast access to the regions retained for processing. Experimental results are provided to show the efficiency of the method and the influence of the main parameters.

1. Introduction

Nowadays, keypoints are a feature that has become more and more widespread in image and video analysis [1-8]. Initially developed for robotic [9], their use has extended to domains as various as indexing, compression, image summarization, and object tracking. Keypoints are located at key positions (usually corners or extrema of a given function), making them easy to recover. Moreover, they are enriched by local descriptors in order to increase their robustness to usual transformations (scale changes, illumination changes, rotations, affine transformations…). However, execution time could be an important factor for some of the concerned applications, and their extraction could take up to one second per image.

Our work takes place in the context of the Portivity project [10]. This project aims at developing an interactive television system, which can realize direct interactivity with moving objects on hand-held receivers. Fast annotation of moving objects in videos is an essential component of this system, thereby motivating the need of an efficient generic tracking system, able to deal with various kinds of videos. In our previous work [11,12], a keypoint based tracking system was built up in order to fulfill the specific requirements of this application. In this particular framework, the keypoints and their corresponding descriptors are extracted in a preprocessing step. For the efficiency of this application, it is important that this preprocessing step is as fast as possible. Since most of the execution time is due to the extraction of keypoints, we have investigated the possibility of speeding up this part of the processing.

Our acceleration method relies on the preliminary identification of areas likely to contain keypoints and the restriction of the extraction to these areas. Based on the assumption that keypoints are extracted in high variance areas, we have developed an algorithm inspired from Haar wavelets calculating this variance at different scales. Moreover, we are using this scale structure for a fast access to the interest areas. We have tested this technique with the famous Harris color keypoints [6], but it could be applied to every kind of keypoint extractor fulfilling the above variance hypothesis.

The rest of this article is organized as follows. The second section will describe the algorithm. The third part will be dedicated to the tuning of important parameters. The application for tracking is shown in the fourth part. And the last section will briefly review the contribution of this paper.

2. Algorithm description

The algorithm relies on the variance study at various scales. Each level of the structure is constituted of square blocks systematically including four blocks of the inferior level (or four pixels in the case of the lowest level). Thus, the structure will be a quadtree, each block having a four times bigger surface than each of its sons. For each block *(i,j)* at level n, the variance $V_n[i][j]$ and the three means $M_n^R[i][j]$, $M_n^G[i][j]$, $M_n^B[i][j]$ (one per color channel) are

calculated. They are determined with the following formulas:

$$V_n[i][j] = (\max R[i][j] - \min R[i][j]) + (\max G[i][j] - \min G[i][j]) + (\max B[i][j] - \min B[i][j]) + \max P[i][j]$$

with (for $X = \{R,G,B\}$):

$$\max X[i][j] = \max (M_{n-1}^X[i][j], M_{n-1}^X[i][j+1], M_{n-1}^X[i+1][j], M_{n-1}^X[i+1][j+1])$$

$$\min X[i][j] = \min (M_{n-1}^X[i][j], M_{n-1}^X[i][j+1], M_{n-1}^X[i+1][j], M_{n-1}^X[i+1][j+1])$$

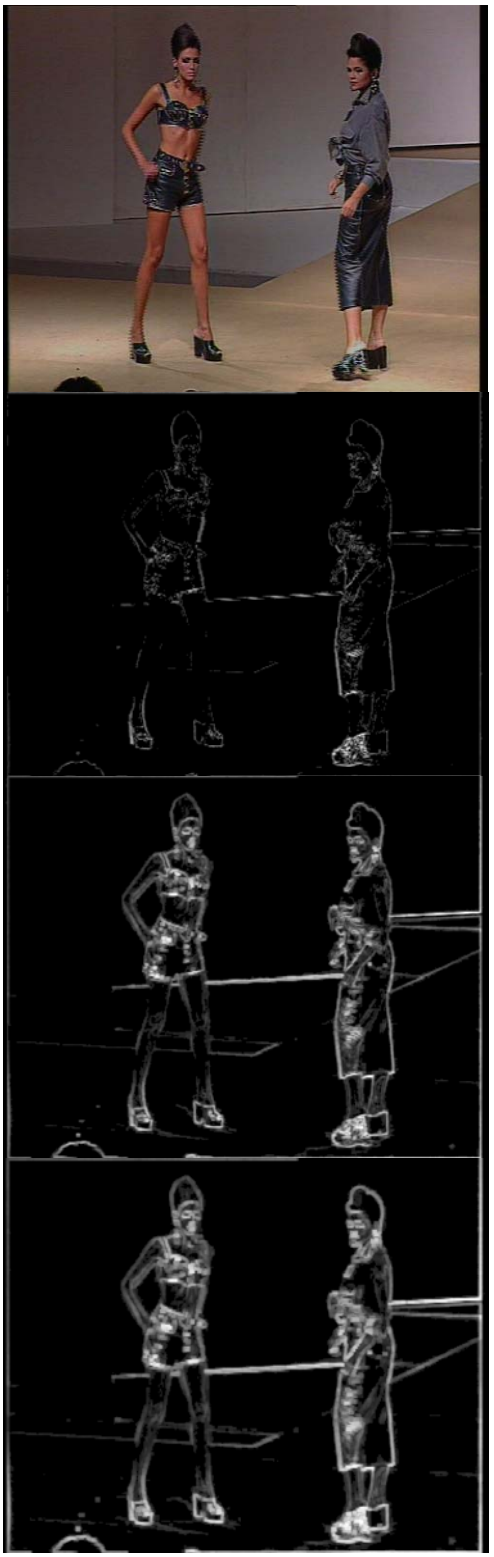
$$\max P[i][j] = \max (V_{n-1}[i][j], V_{n-1}[i][j+1], V_{n-1}[i+1][j], V_{n-1}[i+1][j+1])$$

$$M_n^X[i][j] = (M_{n-1}^X[i][j] + M_{n-1}^X[i][j+1] + M_{n-1}^X[i+1][j] + M_{n-1}^X[i+1][j+1]) / 4$$

The parameter $\max P[i][j]$ represents the variance information from the previous scales. So, the value $V_n[i][j]$ of each block will be the best possible variance, cumulated on many scales according to the inferiors paths of the tree structure. The thinner scales being more important than the coarser ones, their corresponding variance are weighted in consequence. Hence, the value $V_n[i][j]$ informs us about the need to look down in the tree (see figure 1). Indeed, if this value is below a certain threshold S, none of the included pixels will be considered as a potential candidate for keypoint extraction. This threshold defines the compromise between the algorithm rapidity and the amount of retrieved keypoints.

The main known issue of Haar wavelets is the problem of side effects. Indeed, due to the block break down, the information split up at the border of two blocks won't be detected. In order to tackle this problem, essentially appearing at the bottom of the structure, the analysis of the first 2×2 blocks is performed on a p pixels dilated area (So a 4×4 pixels block is studied for $p=1$). This improvement is made at the cost of a subsequent computation effort. The figure 2 shows the pixels selected by the algorithm for an ulterior keypoint extraction with different values of p.

Figure 1: Spatial support of the tracked features (from [7]).



(a) **(b)** **(c)** **(d)**

Figure 2: illustration of the Neighborhood parameter *p* influence. White pixels are pixels retained for keypoint extraction. (a) Analyzed image. Results for (b) *p=0* (c) *p=1* (d) *p=2*.

3. Parameters setting

In order to assess this algorithm, we have compared the Harris keypoint extractor with and without this quadtree structure. This evaluation is based on two criteria: the execution time difference and the percentage of identical keypoints. Two keypoints are considered identical if their position on the image is the same. We have studied the influence of the *p* parameter, the structure level number, as well as the block selection threshold *S* on the algorithm efficiency. The tests have been made on a set of 340 different images coming from 5 different videos. Some images are mostly homogeneous, others present high variations. Of course, our algorithm will perform better for homogeneous images, for which the useless treatment of quasi-uniform areas will be avoided. What we have tried to set up with these experiments is a set of parameters which provides satisfying results for every type of image without a priori knowledge about their content. Results are presented in figure 3, 4, and 5. We state that the quadtree level number does not seem to have, on average, any influence on the algorithm's results (see figure 2). This result could be explained by the fact that, for an image with lots of variations, this structure will be uselessly computationally expensive because the whole image should be analyzed anyway. On the contrary, for an image with large uniform areas, it will be profitable. An a priori knowledge about the complexity of the treated images will lead to an optimization of the structure.

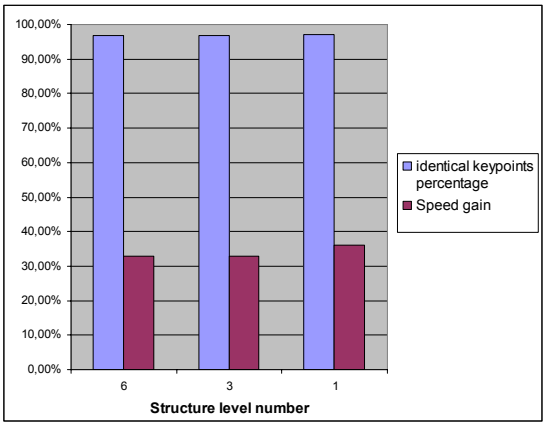


Figure 3: Influence of the structure level number on the algorithm. The measures are the speed gain and the percentage of identical keypoints compared to the Harris extractor only. Each test is the mean of three executions for *p* equal from 0 to 2 and *S=20*.

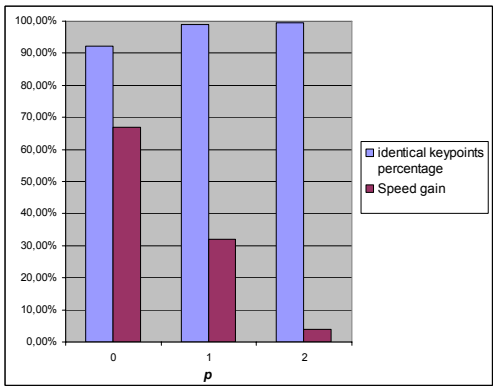


Figure 4: Influence of the *p* parameter on the algorithm. The measures are the speed gain and the percentage of identical keypoints compared to the Harris extractor only. Each test is the mean of three executions for a structure of 1, 3, or 6 levels and *S=20*.

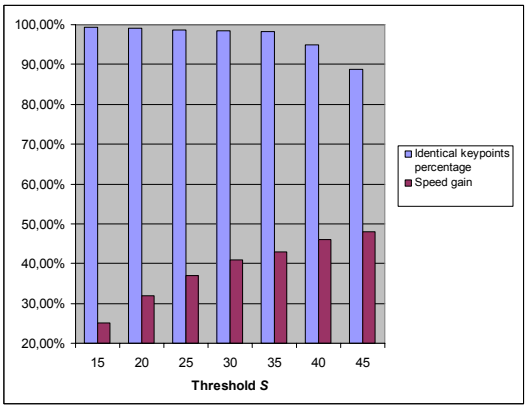


Figure 5: Influence of the threshold *S* on the algorithm. The measures are the speed gain and the percentage of identical keypoints compared to the Harris extractor only. Each test is done with a 3 levels structure and *p=1*.

The others parameters *p* and *S* have a big influence on the results. They increase the efficiency of the algorithm, for a higher computational cost. Fixing *p* to 1 and *S* to 30 seems to be a good compromise, yielding 98.48% of identical keypoints for a 42% reduced execution time (see figure 4 and 5). The threshold *S* could be tuned according to the desired accuracy.

4. Application to object tracking

Despite of these encouraging results, we have observed a high standard deviation in the percentage of identical keypoints detailed in figures 3, 4, and 5. So, the question of the influence of this variation on tracking quality has arisen. Thus, some experiments have been conducted on our tracking system [11,12] to

further validate the efficiency of the structure. The tests were conducts on 14 shots of length varying from 30 to 120 frames. The difficulties encountered are manifold: motion blur, cluttered background, low contrast, fast motion, irregular motion, camera motion, object deformations, and occlusions. Moreover, objects of all sizes are tracked. The objects are identified with a bounding box that is compared with a ground truth in order to evaluate the tracking quality. The similarity between the two bounding boxes, in percentage, is given by the following formula measuring the overlap and the distance between the centers:

$$d_1(X,Y) = \frac{X \cap Y}{X \cup Y} \times \left(1 - \frac{d(C_X, C_Y)}{2D}\right)$$

with *d(a,b)* the Euclidian distance, *D* the distance between the center and one corner of the bounding box. A shot tracking quality is the mean of these similarities for all the sequence frames. And the average tracking quality is the mean of this quality measure on all the shots. We use the tracking quality without the structure as a reference. The results for various threshold *S* settings are presented in figure 6.

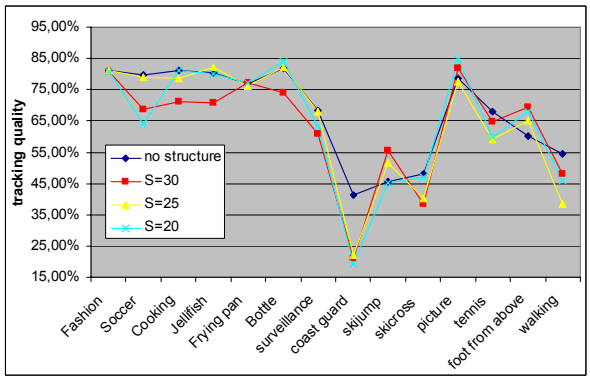


Figure 6: tracking quality with and without the use of the structure for 14 videos sequences. Various threshold *S* settings are shown.

We see that the structure worsen the tracking results but only for the shots were the tracker already fails to correctly localize the object (see the "coast guard", "skicross", and "walking" sequences). If we restrict the benchmark to the 11 sequences where the objects are successfully tracked, the structure remains reliable. Nevertheless, a maximum threshold *S* equal to 25 is needed to keep the tracking accurate.

5. Conclusion

We have presented in this article, a Haar wavelets inspired quadtree structure for keypoint extraction acceleration. This tree structure analyzes the variance at different scales to find the areas likely to contain keypoints. It is also used for a fast access to these areas. We have shown the influence of the main parameters in term of speed gain and amount of keypoints retrieved. Some results validating its use for a tracking application have also been presented.

In the offing, we plan to associate this structure to some other speed improvements in order to use keypoints for a real-time tracking application.

6. References

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

[2] D G. Lowe, Object recognition from local scale-invariant features**,** *International Conference on Computer Vision,* Corfu, Greece, pp. 1150-1157, September 1999.

[3] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions"**,** *British Machine Vision Conference*, volume 1, pp. 384-393, London, UK, September 2002.

[4] Mikolajczyk K., Tuytelaars T., Schmid C., Zisserman A., Matas J., Schaffalitzky F., Kadir F., Van Gool L., 2005-2, A comparison of affine region detectors, *International Journal of Computer Vision, Volume 65, Number ½.*

[5] K. Mikolajczyk, C. Schmid, «Indexation à l'aide de points d'intérêt invariants à l'échelle » *Journées ORASIS GDR-PRC Communication Homme-Machine*., May 2001.

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

[7] N. Sebe, M.S. Lew, Salient Points for Content-based Retrieval, *British Machine Vision Conference* (BMVC'01), pp. 401-410, Manchester, UK, 2001.

[8] T. Tuytelaars et L. Van Gool, "Matching Widely Separated Views based on Affine Invariant Regions"**,** *Int. Journal on Computer Vision*, 59(1), pp. 61-85, 2004.

[9] Moravec, H.P, 1980, Obstacle avoidance and navigation in the real world by a seeing robot rover, *Tech. Rept, CMU-RI-TR-3, The Robotic Institute, Carnegie-Mellon University, Pittsburgh, PA.*

[10] Stoll, Gerhard, "porTiVity: new Rich Media iTV Services for handheld TV", 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies', London, 1st December 2005

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

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