MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning

Mario Edoardo Maresca and Alfredo Petrosino

Department of Science and Technology, University of Naples Parthenope,
Centro Direzionale 80143, Napoli, Italy
marioedoardo.maresca@studenti.uniparthenope.it,
petrosino@uniparthenope.it

Abstract. In this paper we propose a novel framework for the detection and tracking in real-time of unknown object in a video stream. We decompose the problem into two separate modules: detection and learning. The detection module can use multiple keypoint-based methods (ORB, FREAK, BRISK, SIFT, SURF and more) inside a fallback model, to correctly localize the object frame by frame exploiting the strengths of each method. The learning module updates the object model, with a growing and pruning approach, to account for changes in its appearance and extracts negative samples to further improve the detector performance. To show the effectiveness of the proposed tracking-by-detection algorithm, we present quantitative results on a number of challenging sequences where the target object goes through changes of pose, scale and illumination.

Keywords: Tracking by detection, real-time, keypoint-based methods, learning, interest points.

1 Introduction

Despite recent innovations, real-time object tracking remains one of the most challenging problems in a wide range of computer vision applications. The task of tracking an unknown object in a video can be referred to as *long-term tracking* [13] or *model-free tracking* [14]. The goal of such systems is to localize the object (we shall refer to it as *target object*) in a generic video sequence, given only the first bounding box that defines the object in the first frame. Tracking objects is challenging because the system must deal with changes of appearance, illuminations, occlusions, out-of-plane rotations and real-time processing requirements.

In its simplest form, tracking can be defined as the problem of estimating the object motion in the image plane. Numerous approaches have been proposed and they primarily differ the choice of the object representation, that can include: (i) *points*, (ii) *primitive geometric shapes*, (iii) *object silhouette*, (iiii) *skeletal models* and more. For further details, the reader is refereed to [10].

The main challenge of an object tracking system is the difficulty to handle the appearance changes of the target object. The appearance changes can

A. Petrosino (Ed.): ICIAP 2013, Part II, LNCS 8157, pp. 419–428, 2013.

be caused by intrinsic changes such as pose, scale and shape variation and by extrinsic changes such as illumination, camera motion, camera viewpoint, and occlusions. To model such variability, various approaches have been proposed, such as: updating a low dimensional subspace representation [15], MIL based [14] and template or patch based.

Robust algorithms for long-term tracking are generally designed as the union of different modules: a tracker, that performs object motion analysis; a detector, that localizes the object when the tracker accumulates errors during run-time and a learner that updates the object model. A system that uses only a tracker is prone to failure: when the object is occluded or disappears from the camera view, the tracker will usually drift. For this reason we choose to design the proposed framework as the union of only two modules: the detector and the learner. The detector is designed according to a multi-level approach, where multiple keypoint-based methods can be adopted to correctly localize the object, despite changes of illumination, scale, pose and occlusions, within a *fallback model*. The learner updates the training pool used by the detector to account for large changes in the object appearance. Quantitative evaluations demonstrate the effectiveness of our approach that can be classified as a "tracking-by-detection" algorithm, since we track the target object by detecting it frame by frame.

The rest of the paper is organized as follows. Section 2 proposes an outline of the current keypoint-based methods. Section 3 introduces in detail the proposed framework (*Matrioska*): subsection 3.1 and 3.4 analyze the detector and the learning module, respectively. Section 4 shows experimental results.

2 Overview of Known Keypoint-Based Methods

Numerous new keypoint-based methods have been proposed over the recent years (also known as local feature-based or interest point-based). The last technique, to our knowledge, is KAZE [9], published in 2012.

Other methods are (in reverse cronological order): **KAZE** (2012) operating completely in a nonlinear scale space [9]; **FREAK** (2012) inspired by the human visual system and more precisely by the retina [8]; **BRISK** (2011) Binary Robust Invariant Scalable Keypoints [6]; **ORB** (2011) Oriented FAST and Rotated BRIEF [4]; **ASIFT** (2009) fully affine invariant image comparison method [7]; **SURF** (2006) Speeded Up Robust Features [3]; **GLOH** (2005) Gradient location-orientation histogram [5]; **PCA-SIFT** (2004) Principal Components Analysis (PCA) to the normalized gradient patch [2]; **SIFT** (1999) Scale-Invariant Feature Transform [1].

3 Matrioska

In the next sections we describe our novel framework for object detection and tracking (belonging to the category of "tracking-by-detection"). First we will describe the principal components of the proposed detection module: (i) a detector that uses the information of multiple keypoint-based methods, (ii) a filtering

stage with a modified Generalized Hough Transform, and (iii) a scale identification process. Then we will explain the learning module, based on a growing-and-pruning approach. Later we will perform quantitative tests to show that, by using multiple keypoint-based methods, it is possible to achieve both a faster overall detection time and an improved recall. At this stage, we will disable the online learning module to only focus on the outcome of the usage of multiple methods. Then we will enable the online learning module to test Matrioska on a number of challenging video clips that present strong changes in the object appearance. Note that we intentionally choose to not apply any motion analysis, as we only want to focus on the detector and learning components.

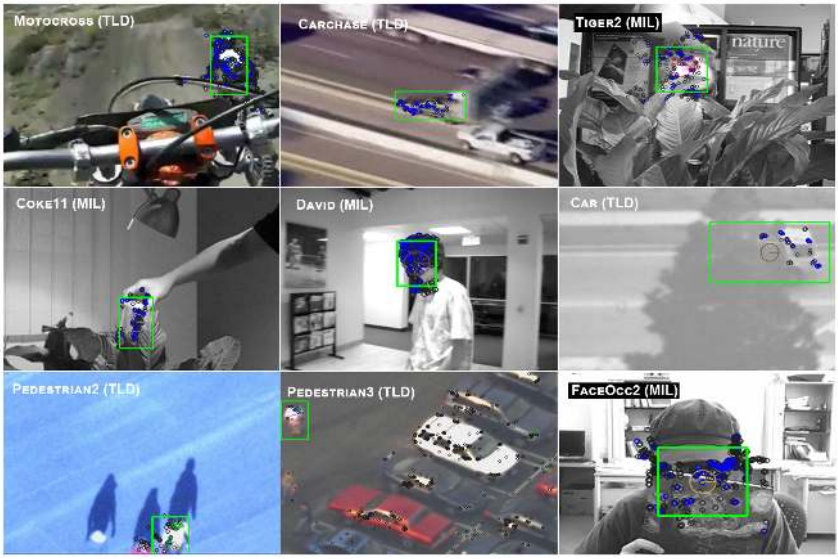

Fig. 1. Snapshots from TLD and MILTrack datasets

3.1 Detection: Combining Multiple Keypoint-Based Methods

As shown in section 2, tracking based on keypoint-based methods gets great interest. The reason for this interest is essentially threefold: (i) they are fast as they only focus on a sparse set of points and (ii) they are inherently robust to a series of challenges (changes in illumination, point of view, rotation, scale and occlusion); (iii) new fast and robust methods are continuously designed.

The development of Matrioska starts from these considerations: we want to achieve real-time performance at an high degree of robustness. We believe that the integration (in a multi-level approach) of various keypoint-based methods represents one of the best ways to achieve these goals. Furthermore, by combining

in a single framework the results coming from different techniques, we are able to take advantage of the strengths of each of them, thus increasing the overall robustness.

Matrioska proceeds as follows:

1. **Detection** of the keypoints from the nth frame with the first registered method in the technique pool.
2. K-nearest neighbor search (**k-NN**) between keypoints (of the same class) of the training pool.
3. A first **outlier filtering** process using the NNDR (Nearest Neighbor Distance Ratio).
4. A second filtering process to discard all matches whose first nearest neighbor was found on a negative sample.
5. A third, more specific filtering, process is performed with the **Generalized Hough Transform** (see section 3.2).
6. The last step involves the **scale estimation** to accurately draw the bounding box according to the parameters obtained by the GHT (see section 3.3).

Steps 1-5 are encapsulated in a *fallback model*: we use the next (higher level) method only if the previous ones were not able to identify the target object. This model ensures that only the sufficient keypoint-based methods are used.

3.2 Outliers Filtering

The main drawback of using multiple keypoint-based methods is the fact that each method will add a considerable amount of new outliers, making the filtering stage a challenging process. Furthermore, we must operate in real-time, therefore the filtering process should be as fast as possible.

In this scenario, filtering outliers with well-known fittings methods, such as RANSAC or LMedS (Least Median of Squares), would not produce good results since the percentage of inliers can fall much lower than 50%. For this reason, we employ a filtering process based on the Generalized Hough Transform (GHT) where each match of keypoints specifies three parameters: 2D object's center and orientation. To estimate the target object center we store, for each trained keypoint, the size of the corresponding training image; therefore, we can project the center of this image on the current frame with a translation and a rotation.

These parameters are sufficient to localize the object if the scale doesn't change during tracking, but this is a strong assumption and generally it doesn't stand. To account for scale changes we could use a GHT with four parameters, adding the scale to the previous three parameters, similar to the solution proposed by D. Lowe [1]. However, this could pose serious limitations on the keypoint-based methods that Matrioska use, since, to identify the right scale bin, we need to use the keypoint's octave in which it was detected. However, we cannot rely on it because not all the methods implement octave scaling, and even if they implement it, we tend to fix the number of octaves to one for performance reasons. Furthermore, the octave number gives a very broad indication

because increasing the scale by an octave means doubling the size of the smoothing kernel, whose effect is roughly equivalent to halving the image resolution. Instead, we want to achieve a higher precision, up to a factor equal to 0.01 of scale changes.

3.3 Scale Identification

The identification of the current object scale is a crucial step and deserves a separate section because it is not directly related to the GHT discussed earlier. We want to achieve a stable and accurate method to correctly identify the scale of the object without relying on the keypoint's octave. In order to satisfy these constraints we study the geometric distance between pairs of keypoints: we compute the ratio between the distance of consecutive pairs of keypoints belonging to one training image and the query image. We repeat the process for each training image having at least two matches. After this process, we obtain the final object size by calculating the mean of all training image sizes scaled by the factor found with the ratio of distances. The final size can be obtained with the following equation:

$$\mathbf{S}_o = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathbf{S}_i}{J-1}\sum_{k=1}^{J-1}\frac{\|\mathbf{P}_k^Q - \mathbf{P}_{k+1}^Q\|}{\|\mathbf{P}_k^{T_i} - \mathbf{P}_{k+1}^{T_i}\|}\right) \tag{1}$$

where

- $\mathbf{S}_o$ and $\mathbf{S}_i$ are two vectors that represent the width and height. $\mathbf{S}_o$ represents the size of the object while $\mathbf{S}_i$ the size of the ith training image.
- N is the number of the training images.
- J is the number of keypoints found on the ith training image.
- $\mathbf{P}_k^Q$ and $\mathbf{P}_k^{T_i}$ are the kth keypoints found on the query image Q and matched to the training image T_i.

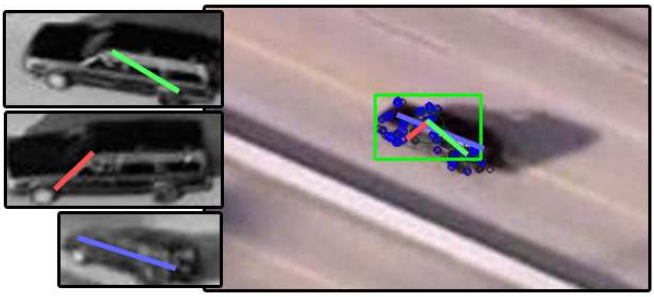

Fig. 2. Consecutive pairs of keypoints belonging to one training image and the query image, represented here as segments of the same color

3.4 Learning: Growing and Pruning

The learning component is imperative to solve one of the toughest challenges of visual tracking: adapting the model to account for changes in the target object appearance (shape deformation, lighting variation, large variation of pose).

In this section we present the proposed schema aiming to update the model (the training pool) used by the detection module (section 3.1) to track the object. Inspired by [15,14,13,11,16] our learning model is an incremental growing and pruning approach; while tracking the object, we must learn both new positive and negative samples that will be added to the training pool.

One of the key factors in learning is the choice of the new positive sample: we must learn a new sample only if the object appearance is undergoing changes. This will: (i) avoid saturating the training pool with duplicated samples, (ii) add valuable information to the detection component and (iii) not let the NNDR discard good matches. To achieve this we must carefully determine the selection criterion. The proposed criterion to choose a new candidate positive sample S_c^+ is a combination of two different conditions: (1) the detection module failed to detect the object in the previous frame, (2) the current best GHT's bin has less than $2V$ votes. To learn a new positive sample one of these conditions must be verified. A similar but simpler process is employed for the negative samples: we learn as negative the keypoints found outside the bounding box when the ratio between the number of positive keypoints and the negatives exceeds a given threshold.

Fig. 3. Some of the positive samples that have passed the selection criterion and have been online learned. The tested video clip is the *Carchase* dataset from TLD.

4 Experimental Results

Matrioska has been tested on several challenging video sequences from TLD [13] and MILTrack [14] datasets. The first tests show how the performance change, adopting different multiple keypoint-based methods. However we only show some possible combinations because testing all possible configurations would not be plausible. Furthermore, our aim is to demonstrate that by using multiple methods in a multi–level approach, it is possible to achieve both a faster overall detection time and an improved accuracy. At this stage of testing, we disabled the online learning module to focus only on the relative meaning of multiple methods.

In other tests we enabled the online learning module to test Matrioska on a number of challenging video clips that present strong changes in the object appearance.

In all tests, the only a priori knowledge was the location of the object at the first frame. Concerning the implementation, our OpenCV C++ single-threaded implementation runs at 25 FPS on an Intel Core i7-920 with a QVGA video stream.

To avoid confusion we adopt the same metric in all sequences to evaluate Matrioska performance: precision $P = correctDetections/detections$, recall $R = correctDetections/trueDetections$ and f-measure, where *correctDetections* represents the number of detection whose overlap with the ground truth bounding box is wider than 25%, if the ground truth is defined. The overlap is defined as $intersection/(GT_{area} + BB_{area} - intersection)$, where GT_{area} is the area of the ground truth and BB_{area} is the area of the bounding box [13].

4.1 Detection with Multiple Methods

For this testing phase we disabled the online learning module to only evaluate the results of using multiple keypoint-based methods. We tested ORB, BRISK, FREAK, SURF and SIFT alone on some sequences and then tried different combinations. Note that our aim is to show the advantage obtained by combining multiple keypoint-based methods rather than running comparative evaluations of single methods (such as [17,18,19]).

Table 1. Tiger2 (MILTrack)

Method(s)	FPS	Recall
ORB	30	0.15
BRISK	20	0.01
FREAK	29	0.23
SURF	6	0.04
SIFT	4	0.05
ORB + FREAK	24	**0.38**
FREAK + SURF	10	0.37

Table 2. Face occlusion 2 (TLD)

Method(s)	FPS	Recall
ORB	26	0.44
BRISK	25	0.44
FREAK	27	0.63
SURF	16	0.57
SIFT	9	0.64
FREAK + SURF	21	0.68
FREAK + SIFT	18	**0.69**

As Tables 1, 2, 3, 4 show, the best results in terms of recall, are obtained with a combination of two methods. Furthermore, table 2 is indicative of the contribution given by the fallback model: we obtained a recall of 0.69, halving the time complexity from 0.11 seconds per frame (9 FPS) required using SIFT only, to 0.055 seconds (18 FPS) using FREAK + SIFT. This is possible because SIFT is adopted by Matrioska only when necessary. Table 4 shows an almost perfect result testing Car dataset (TLD), even without the online learning module enabled: this is due to the fact that the target object does not change its appearance during the sequence and our detector, with a combination of two methods, is enough to obtain a robust performance.

Table 3. Motocross (TLD)

Method(s)	FPS	Recall
ORB	38	0.31
BRISK	24	0.10
FREAK	25	0.05
SURF	9	0.04
SIFT	6	0.05
ORB + SURF	15	0.34
ORB + SIFT	10	**0.40**

Table 4. Car (TLD)

Method(s)	FPS	Recall
ORB	28	0.48
BRISK	30	0.23
FREAK	33	0.54
SURF	15	0.48
SIFT	8	0.67
FREAK + SIFT	14	**0.95**
ORB + FREAK	20	0.87

4.2 Detection and Learning

In this testing phase, we enabled the online learning module of Matrioska to test our approach against the dataset used by TLD and MILTrack. The gain in performance compared to the use of the detection module alone is clear in Table 5. Figure 1 shows snapshots of the tested sequences with examples of detection. The obtained results are better than other state-of-the-art approaches [24,22,25,21,15,23,14,13,26].

It must be remarked that: (i) we could carefully choose for each sequence the most suitable methods to obtain better performance, and we choose to include in the technique pool only ORB and FREAK techniques, independently from the tested sequence, (ii) the use of keypoint-based methods forced us to double the size of the smaller sequence (and relocate the ground truth accordingly), because when the target object is too small we cannot compute a feature vector for each keypoint found inside it (e.g. Tiger2 and Coke11), (iii) we slightly enlarged the first bounding box to be able to detect keypoints near the borders of the target object.

Table 5. Evaluation of Matrioska with online learning enabled. We provided only the first object location and algorithm tracked the target object up to the final frame.

Sequence	Frames	Correct D. / True D.	P / R / F-measure
Car (TLD)	945	854 / 860	0.97 / 0.99 / 0.98
Carchase (TLD)	9928	7551 / 8660	0.97 / 0.87 / 0.92
Motocross (TLD)	2665	1357 / 1412	0.84 / 0.96 / 0.90
Pedestrain2 (TLD)	338	260 / 266	0.94 / 0.98 / 0.96
Pedestrain3 (TLD)	184	145 / 156	0.96 / 0.92 / 0.94
Coke11 (MILTrack)	292	59 / 59	1.00 / 1.00 / 1.00
David (MILTrack)	462	93 / 93	1.00 / 1.00 / 1.00
Face occlusion 2 (MILTrack)	816	163 / 163	1.00 / 1.00 / 1.00
Tiger2 (MILTrack)	365	67 / 73	0.93 / 0.91 / 0.92
Sylverster (MILTrack)	1345	269 / 269	1.00 / 1.00 / 1.00

5 Concluding Remarks

The paper reports a novel framework based on a multi-level approach to address the problem of tracking an unknown object in a video sequence. We used a combination of two modules: (i) a detector that, using keypoint-based methods, can identify an object in presence of illumination, scale, rotation and other changes, and (ii) a learning module that updates the object model to account for large variations of the target appearance. Several tests validated this approach and showed its efficiency. In most cases, we obtained comparable, if not better, results to the current state-of-the-art approaches using only two components (a detector and a learning module).

The integration of a tracker could provide even better overall results, even though with higher computational cost. Instead, to fully exploit the Matrioska framework capabilities, we advice to develop a series of new (keypoint-based) techniques, each based on the analysis of a particular feature (color, shape and more) to obtain fast and simple techniques if used individually, but robust when used together. These techniques, within a fallback model, would ensure also a low computational complexity, as only the sufficient features would be used for the correct detection of the object.

References

1. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60, 91–110 (2004)
2. Ke, Y., Sukthankar, R.: PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 506–513 (2004)
3. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-Up Robust Features (SURF). Comput. Vis. Image Underst. 110, 346–359 (2004)
4. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision, pp. 2564–2571 (2011)
5. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1615–1630 (2005)
6. Leutenegger, S., Chli, M., Siegwart, R.Y.: BRISK: Binary Robust invariant scalable keypoints. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2548–2555 (2011)
7. Morel, J.-M., Yu, G.: ASIFT: A New Framework for Fully Affine Invariant Image Comparison. SIAM J. Img. Sci. 2, 438–469 (2009)
8. Ortiz, R.: FREAK: Fast Retina Keypoint. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 510–517 (2012)
9. Alcantarilla, P.F., Bartoli, A., Davison, A.J.: KAZE features. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 214–227. Springer, Heidelberg (2012)
10. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38 (2006)
11. Kloihofer, W., Kampel, M.: Interest Point Based Tracking. In: 2010 20th International Conference on Pattern Recognition, pp. 3549–3552 (2010)

12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
13. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. IEEE Trans. on Pattern Anal. Mach. Intell. 34, 1409–1422 (2012)
14. Babenko, B., Yang, M.-H., Belongie, S.: Robust Object Tracking with Online Multiple Instance Learning. IEEE Trans. Pattern Anal. Mach. Intell. 33, 1619–1632 (2011)
15. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental Learning for Robust Visual Tracking. Int. J. Comput. Vision 77, 125–141 (2008)
16. Hare, S., Saffari, A., Torr, P.H.S.: Efficient online structured output learning for keypoint-based object tracking. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1894–1901 (2012)
17. Heinly, J., Dunn, E., Frahm, J.-M.: Comparative evaluation of binary features. In: Proceedings of the 12th European Conference on CV, pp. 759–773 (2012)
18. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. Int. J. Comput. Vision 94, 335–360 (2011)
19. Khvedchenia, I.: A battle of three descriptors: SURF, FREAK and BRISK (2012), http://computer-vision-talks.com/
20. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. In: Proceedings of the 2011 International Conference on Computer Vision, pp. 81–88 (2011)
21. Yu, Q., Dinh, T.B., Medioni, G.G.: Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 678–691. Springer, Heidelberg (2008)
22. Avidan, S.: Ensemble Tracking. IEEE Trans. Pattern Anal. Mach. Intell. 29, 261–271 (2007)
23. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: PROST Parallel Robust Online Simple Tracking. In: 2010 IEEE Conference on CVPR, pp. 723–730 (2010)
24. Grabner, H., Bischof, H.: On-line Boosting and Vision. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 260–267 (2006)
25. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised On-Line Boosting for Robust Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
26. Pernici, F.: FaceHugger: The ALIEN Tracker Applied to Faces. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012 Ws/Demos, Part III. LNCS, vol. 7585, pp. 597–601. Springer, Heidelberg (2012)