The Matrioska Tracking Algorithm on LTDT2014 Dataset

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
alfredo.petrosino@uniparthenope.it

Abstract

We present a quantitative evaluation of Matrioska, a novel framework for the detection and tracking in real-time of unknown object in a video stream, on the LTDT2014 dataset that includes six sequences for the evaluation of single-object long-term visual trackers. Matrioska follows the approach of tracking by detection: the detector localizes the target object in each frame, using multiple keypoint-based methods. To account for appearance changes, the learning module updates both the target object and background model with a growing and pruning approach.

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* [1] or *model-free tracking* [2]. The goal of such systems is to localize the object (we will 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, but they mainly differ in the choice of the object representation, that can include: (i) *points*, (ii) *primitive geometric shapes*, (iii) *object silhouette*, (iv) *skeletal models* and more. For further details we refer the reader to [3].

The main challenge of an object tracking system is the difficulty to handle the appearance changes of the target object. The appearance changes can 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 [4], MIL based [2] 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/background 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 the proposed framework is the union of only two modules: the detector and the learner. The detector can use multiple keypoint-based methods to correctly localize the object, despite changes of illumination, scale, pose and occlusions, within a *fallback model*. The learning module updates the training pool used by the detector to account for large changes in the object appearance. Quantitative evaluations demonstrate the effectiveness of this approach that can be classified as a "tracking-by-detection" algorithm, since it tracks the target object by detecting it frame by frame.

1.1. Related work

Recently a number of surveys ([5], [6], [7]) compared the performance of many visual trackers. Different outcomes were proposed, especially because VOT challenge did not require a re-detector module in case of tracker drifts.

We present a short summary of the most recent trackers. PLT (single scale pixel based LUT [26]) runs a classifier at a fixed single scale to determine the top scoring bounding box. An online sparse structural SVM is used to select a small set of features, and a probabilistic object-background segmentation is used to adjust the weight during the training. FoT (Flock of Trackers [9]) estimates the object motion using local trackers covering the object.

Local-Global Tracking LGT [10] and LGT++ [11] combine the target global and local appearance by interlacing two layers. EDFT (Enhanced Distribution Fields for Tracking [12]) derives an enhanced computational scheme by employing the connection between histograms and channel representations. ALIEN (Appearance Learning In Evidential Nuisance [17]) resides on local features to detect and track. HoughTrack [13] and PixelTrack [14] use a voting procedure to detect the object. CMT (Consensus-based Matching and Tracking [15]) uses a voting procedure exploiting the information given by the matching and tracking procedure. Matrioska [8] also resides on local features and uses a voting procedure. Other interesting approaches are: LT-FLO [16], GSDT [18], AIF [19], DFT [20] and ORIA [21].

2. Matrioska Overview

Matrioska is composed by two modules: detector and learning. The detector can use multiple keypoint-based methods (such as ORB [23], FREAK [24], SIFT [25] and more) to correctly localize the object frame by frame exploiting the strengths of each method. We showed how the joint use of multiple keypoint-based methods can enhance the robustness while keeping real-time performance with the use of a *fallback* strategy. According to this strategy only the sufficient keypoint-based methods will be used to detect the object in relation to the difficulty of the detection on each frame.

The learning module, on the other hand, updates the training pool used by the detector to localize the object in presence of strong appearance changes (shape deformation, lightning variation and scale and pose changes) using a growing-and-pruning approach: while tracking the object, the system learns new positive and negative samples (keypoints) identified by the detector, as Figure 1 shows. To avoid performance degradations, when the number of keypoints is greater than a threshold, the pruning procedure removes 20% of the samples (typically the oldest).

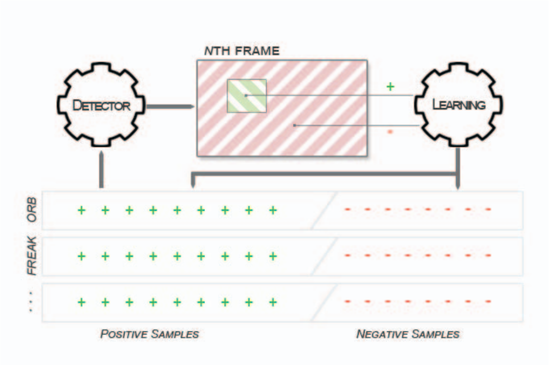

Figure 1: Matrioska uses two components: the detector and the learning module. The detector localizes the target object using the information given by the learning module and the latter updates the training pool with both positive and negative samples (keypoints) identified by the detector.

2.1. Outliers Filtering

Matrioska has been enhanced for testing on LTDT2014 dataset using a slightly different outlier filtering method. Filtering outliers is one of the toughest challenge that arises using keypoint-based methods. Many well-known robust fitting methods, such as RANSAC or LMedS (Least Median of Squares) cannot handle large percentage of outliers (greater than 50%). For this reason we use a clustering procedure that produces good results even in presence of high percentages of outliers [25].

Specifically, we use a slightly different procedure than the original Matrioska: to achieve a higher accuracy we cluster features in a four-dimensional accumulator space (instead of a three-dimensional space) using every pair of keypoints found inside the current frame. A single pair of keypoints specifies four parameters: 2D object center coordinates, orientation and scale.

The scale factor can be easily estimated by calculating the ratio between the distance of the pair of keypoints of the model image on the distance of the respective keypoints in the query image. After all pairs of keypoints voted their parameters, the most voted bucket, in the accumulator space, is used to localize the target object if it contains at least four votes.

The accumulator space can be seen as a sparse matrix, therefore to efficiently implement it we use a hash table where the four parameters are combined to calculate the hash index. Collisions are solved with a chaining approach.

3. Performance Evaluation

We present the performance evaluation of Matrioska on the dataset provided in LTDT2014. The dataset contains six long-term sequences: motocross, volkswagen, carchase, LiverRunCropped, NissanSkylineChaseCropped and Sitcom. They contain the typical challenges encountered with long-term sequences, such as: scale changes, appearance changes and partial or total occlusion. We use the same parameters for all sequences, and because we aim to achieve, along with a good robustness, a very fast processing speed we register in the Matrioska's technique pool only ORB keypoints. This is a risky choice because ORB is clearly not as robust as SIFT, but on the other hand is several order of magnitude faster than SIFT. Our motivations are: (i) for long term sequences a real-time processing speed is mandatory and

(ii) we want to show how Matrioska can achieve a very solid performance without using computational expensive keypoint-based methods. To evaluate the performance, both the distance score and the bounding box overlap metrics are used.

3.1. Carchase

The first sequence, *carchase*, contains 9928 frames. The main challenges of this sequence is the scale changes that the target object undergoes during the chase, and the strong appearance change at about the frame #6100. Figure 2 shows some snapshots of the sequence with successful tracking, red points represent negative keypoints, while blue represent positive ORB keypoints. Green segments show the pair of keypoints used for voting the object center in the accumulator space.

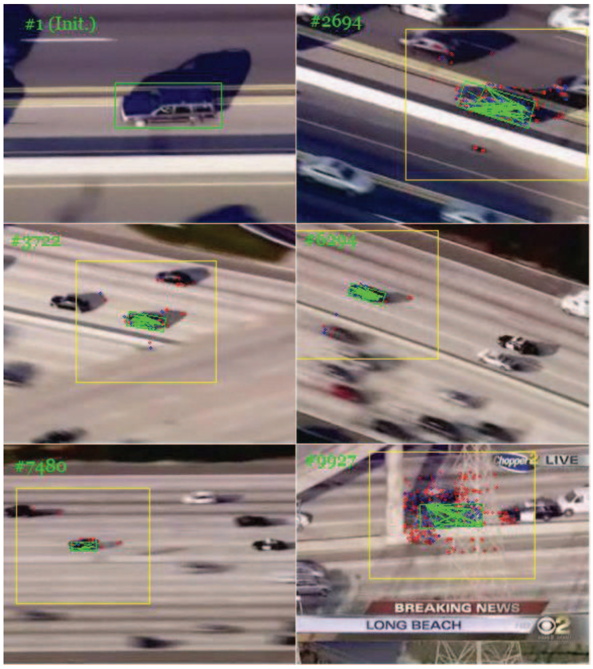

Figure 2: Snapshots from the *carchase* sequence. The target object is correctly localized despite the presence of strong scale and appearance changes and a partial occlusion. Green segments show the pair of keypoints used to cast the object center votes in the accumulator space.

3.2. Volkswagen

The second sequence, *volkswagen*, contains 8576 frames. The main challenge of this sequence is the small size of the first appearance of the target object. This can be tricky because: (i) a single patch of a single keypoint covers the entire object, or, worse, it can describe the background instead of the object and (ii) very few keypoints may be detected inside the object. For this reasons we reduce the patch size used by ORB to 13x13 pixels, note that in this situation the use of multiple keypoint based methods would be very helpful.

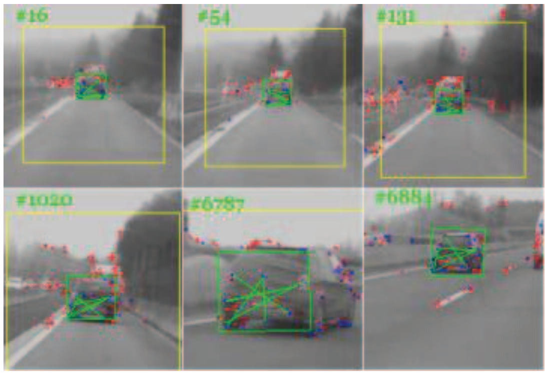

Figure 3: Snapshots from the *volkswagen* sequence. The target object is correctly localized despite the small size of the target object in the first frames.

3.3. Motocross

The third sequence, *motocross*, contains 2665 frames. The main challenge of this sequence is the appearance and scale changes of the motorbike. This sequence might seem difficult, but in practice it is not, due to the presence of an uniform background that clearly separates the target object from the rest of the scene, making a tracker drift very unlikely.

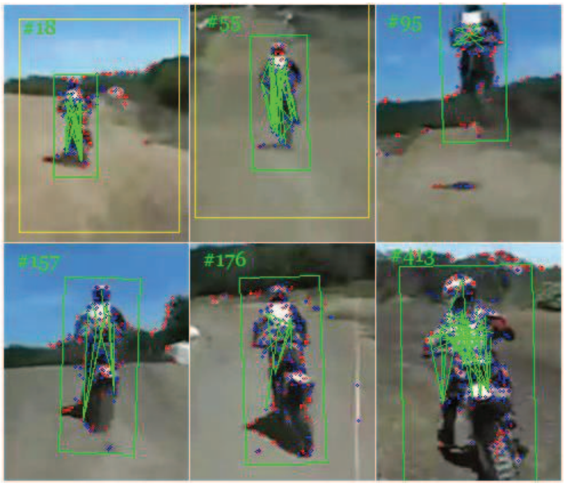

Figure 4: Snapshots from the *motocross* sequence. The target object is correctly localized despite the appearance changes.

3.4. LiverRunCropped

The fourth sequence, *LiverRunCropped*, contains 29598 frames, it is the largest sequence of the dataset. This is, along with *volkswagen*, the most difficult sequence of the dataset. The main challenge is the poor quality of the video and the low number of keypoints found inside the target object in the first frames.

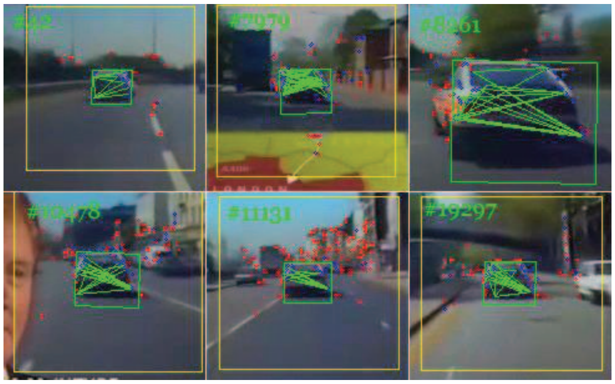

Figure 5: Snapshots from the *LiverRunCropped* sequence. The target object is correctly localized despite the appearance changes.

3.5. NissanSkylineChaseCropped

The fifth sequence, *NissanSkylineChaseCropped*, contains 3742 frames. The main challenge of this sequence is the scale changes of the car. In terms of difficulty this sequence is very similar to *volkswagen,* but this time the object is much better defined and many more keypoints can be detected and described.

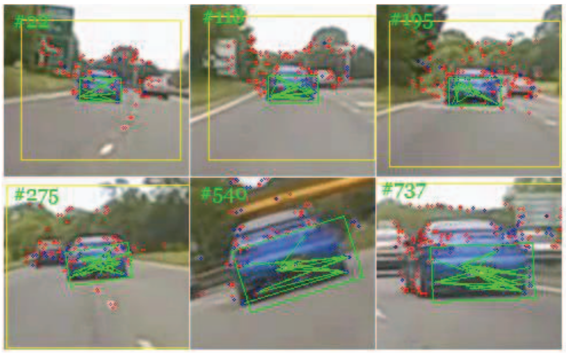

Figure 6: Snapshots from the *NissanSkylineChaseCropped* sequence.

3.6. Sitcom

The sixth sequence, *Sitcom*, contains 3898 frames. The main challenge of this sequence is the out-of-plane rotation of the target object (in this case a face). Figure 7 shows some snapshots of Matrioska localizing the target object.

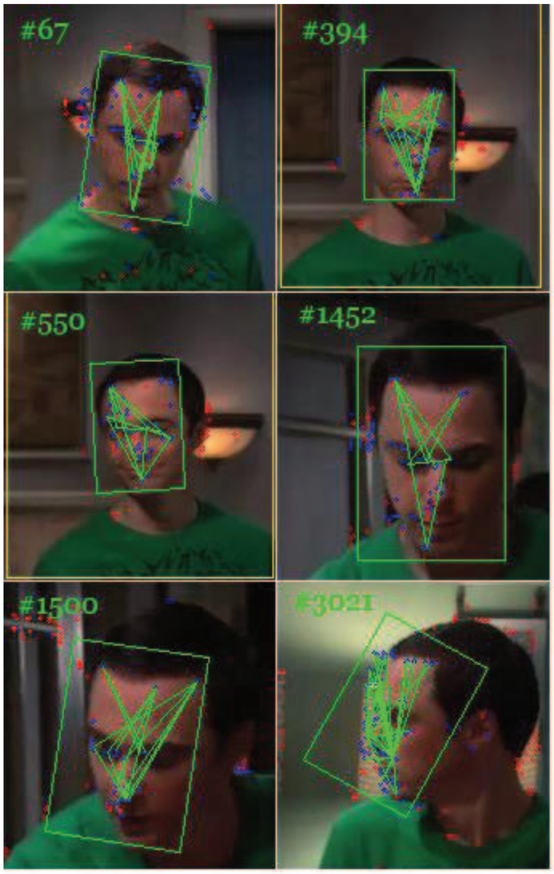

Figure 7: Snapshots from the *Sitcom* sequence.

3.7. Failure Cases

Figure 9 and Figure 8 show typical failure cases of our approach. Matrioska uses keypoint-based methods to detect the target object in each frame, hence it inherits their weaknesses. Failure cases can include: low quality sequences, texture-less objects, motion blur, repetitive patterns and small objects. Other minor failure cases include: out-of-plane rotations and non-rigid deformations.

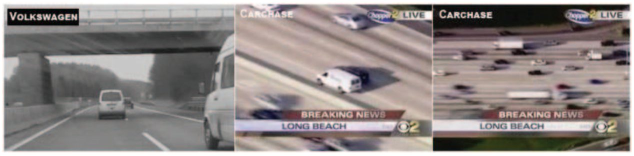

Figure 8: Typical failure cases for *volkswagen* and *carchase* sequences.

Figure 8 shows snapshots from *volkswagen* and *carchase* sequences. Typical failure cases, for these sequences, include: very small object size (frame #7105 for volkswagen and frame #8087 for carchase) and a strong partial occlusion (frame #9212 for carchase). Small objects represent a challenge because a single keypoint covers the entire object and the system cannot estimate the right pose of the object.

Figure 9 shows snapshots from *LiverRunCropped*, *NissanSkylineChaseCropped* and *Sitcom* sequences. Typical failure cases, for these sequences, include: a low illumination (frame #20302 for the first sequence), a motion blur (frame #392 for the second) and a strong pose change (frame #626 for the *Sitcom* sequence). Low illumination and motion blur are challenging because the keypoints detected inside the target object are not sufficient for a correct localization.

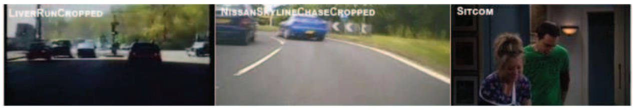

Figure 9: Typical failure cases for *LiverRunCropped*, *NissanSkylineChaseCropped* and *Sitcom* sequences.

3.8. Experiments

We report the results obtained with the dataset in terms of recall of both the bounding box overlap Φ and the center distance δ. The bounding box overlap is defined as:

$$\Phi = \frac{BB_T \cap BB_{GT}}{BB_T \cup BB_{GT}}$$

where BB_T is the bounding box of the tracker and BB_{GT} is the bounding box of the ground truth.

Table 1 shows the performance of Matrioska. It is interesting to note the solid performance obtained by the algorithm using only a fraction of the processing time of other algorithms. This is possible due to the use of the ORB keypoints inside the framework.

Sequence	Frames	Overlap	Distance	FPS
Carchase	9928	0.642	0.850	55
Volkswagen	8576	0.757	0.842	48
Motocross	2665	0.749	0.792	52
LiverRunCropped	29598	0.598	0.610	56
NissanSkylineCC	3742	0.854	0.894	42
Sitcom	3898	0.486	0.534	39

Table 1: Matrioska performance with LTDT sequences. We use the default threshold for both overlap (0.5) and center distance (20) recall.

For completeness we report the results, in Table 2, obtained by the algorithm with the VOT2013 challenge [5] (held in conjunction with ICCV2013) aimed to benchmark short-term trackers. For the complete set of the experiments and further details we refer the reader to [5]. For all the experiments we used a 2.67 GHz Intel i7-920 processor.

Tracker	Baseline experiment
PLT	**5.26**
FoT	**7.85**
LGT++	**9.99**
EDFT	**10.1**
SCTT	**10.6**
CCMS	**11**
AIF	**11.1**
Matrioska	**11.5**
LGT	**11.6**
DFT	**11.9**
LT-FLO	**11.9**
GSDT	**11.9**
STRUCK	**12.6**
IVT	**13**
ASAM	**13.2**
ORIA	**14.1**
PJS-S	**15**
SwATrack	**15.8**
TLD	**16.4**

Table 2: VOT2013 baseline experiment. Matrioska used a combination of ORB and SURF for this challenge.

4. Conclusions

In this paper we evaluated the performance of Matrioska with six long-term sequences. We demonstrated how this algorithm reaches state-of-the-art accuracy while requiring only a fraction of the processing time of other trackers. This is possible by using computationally efficient techniques such as ORB. Even if some failure cases can be evidenced (e.g. low quality sequences, texture-less objects, motion blur, repetitive patterns and small objects) Matrioska well behaves in most cases.

We also reported the results obtained for the VOT2013 challenge aimed to benchmark short-term trackers. In this case we used a combination of ORB and SURF and demonstrated how Matrioska can handle both short and long-term sequences.

Regarding future developments, to avoid failure cases as much as possible, it would be interesting to integrate a new module to handle texture-less objects and low quality sequences.

References

[1] Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-Learning-Detection. IEEE Trans. on Pattern Anal. Mach. Intell. 34, 1409--1422 (2012).

[2] 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).

[3] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. 38 (2006).

[4] 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).

[5] Matej Kristan, R. Pflugfelder, Ales Leonardis, Jiri Matas, Fatih Porikli, Luka Cehovin, Georg Nebehay, Fernandez Gustavo, Tomas Vojir, and Et Al. The visual object tracking VOT2013 challenge results. In Workshop on the VOT2013 Visual Object Tracking Challenge, pages 98–111, December 2013.

[6] Yi Wu; Jongwoo Lim; Ming-Hsuan Yang, "Online Object Tracking: A Benchmark," Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on , vol., no., pp.2411,2418, 23-28 June 2013.

[7] Arnold W. M. Smeulders, Dung M. Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, Mubarak Shah, “Visual Tracking: An Experimental Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, no. PrePrints, p. 1, , 2013.

[8] Mario Edoardo Maresca, Alfredo Petrosino: MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning. ICIAP 2, volume 8157 of Lecture Notes in Computer Science, page 419-428. Springer, (2013).

[9] T. Vojir and J. Matas. Robustifying the flock of trackers. In Comp. Vis. Winter Workshop, pages 91–97. IEEE, 2011.

[10] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. IEEE Trans. Pattern Anal. Mach. Intell., 35(4):941–953, 2013.

[11] J. Xiao, R. Stolkin, and A. Leonardis. An enhanced adaptive coupled-layer LGTracker++. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013, 2013.

[12] M. Felsberg. Enhanced distribution field tracking using channel representations. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013, 2013.

[13] Godec, M.; Roth, P.M.; Bischof, H., "Hough-based tracking of non-rigid objects," Computer Vision (ICCV), 2011 IEEE International Conference on , vol., no., pp.81,88, 6-13 Nov. 2011.

[14] S. Duffner, C. Garcia. PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. International Conference on Computer Vision (ICCV 2013), Sydney, Australia. pp. 2480-2487. 2013.

[15] Georg Nebehay and Roman Pflugfelder. Consensus-based matching and tracking of keypoints for object tracking. In Winter Conference on Applications of Computer Vision. IEEE, March 2014.

[16] K. Lebeda, R. Bowden, and J. Matas. Long-term tracking through failure cases. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013, 2013.

[17] Federico Pernici, Alberto Del Bimbo, "Object Tracking by Oversampling Local Features," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 99, no. PrePrints, p. 1, , 2013.

[18] J. Gao, J. Xing, W. Hu, and X. Zhang. Graph embedding based semi-supervised discriminative tracker. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013, 2013.

[19] W. Chen, L. Cao, J. Zhang, and K. Huang. An adaptive combination of multiple features for robust tracking in real scene. In Vis. Obj. Track. Challenge VOT2013, In conjunction with ICCV2013, 2013.

[20] L. Sevilla-Lara and E. G. Learned-Miller. Distribution fields for tracking. In Comp. Vis. Patt. Recognition, pages 1910–1917. IEEE, 2012.

[21] Y. Wu, B. Shen, and H. Ling. Online robust image alignment via iterative convex optimization. In Comp. Vis. Patt. Recognition, pages 1808–1814. IEEE, 2012.

[22] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. In Readings in computer vision: issues, problems, principles, and paradigms, Martin A. Fischler and Oscar Firschein (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA 714-725, 1987.

[23] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G., "ORB: An efficient alternative to SIFT or SURF," Computer Vision (ICCV), 2011 IEEE International Conference on, vol., no., pp.2564,2571, 2011.

[24] Alahi, A.; Ortiz, R.; Vandergheynst, P., "FREAK: Fast Retina Keypoint," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , vol., no., pp.510,517, 16-21 June 2012.

[25] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision. 60, 91—110, 2004.

[26] Cher-Keng Heng; Yokomitsu, S.; Matsumoto, Y.; Tamura, H., "Shrink boost for selecting multi-LBP histogram features in object detection," Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on , vol., no., pp.3250,3257, 16-21 June 2012.