TELKOMNIKA, Vol.16, No.1, February 2018, pp. 302~313
ISSN: 1693-6930, accredited **A** by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v16i1.7271

Long-Term Robust Tracking Whith on Failure Recovery

Khaled Hammemi*[1], Mohamed Atri[2]
[1]Department of Electrical, School of Engineers, University of Monastir
[2]Laboratory of Electronics and Microelectronics, Faculty of Sciences, University of Monastir
*Corresponding author, e-mail: Khaled.hammami@enim.rnu.tn, Mohamed.atri@fsm.rnu.tn

Abstract

This article aims at a new algorithm for tracking moving objects in the long term. We have tried to overcome some potential difficulties, first by a comparative study of the measuring methods of the difference and the similarity between the template and the source image. In the second part, an improvement of the best method allows us to follow the target in a robust way. This method also allows us to effectively overcome the problems of geometric deformation, partial occlusion and recovery after the target leaves the field of vision. The originality of our algorithm is based on a new model, which does not depend on a probabilistic process and does not require a data based detection in advance. Experimental results on several difficult video sequences have proven performance advantages over many recent trackers. The developed algorithm can be employed in several applications such as video surveillance, active vision or industrial visual servoing.

Keywords*: robust tracking, motion, long term, occlusion, matching, failure recovery*

1. Introduction

Perception is the means by which we know our environment. This perception has active aspects, in particular, it can be directed to specific purposes, filtering the data to address only the most relevant ones. Conversely to the effectiveness of human perception, artificial perception is a complex problem that confronts many difficulties, such as changes in illumination, geometric changes or partial or total occultations. In order to to solve these difficulties effectively, artificial perception is inspired by human perception.

In this work, the visual tracking of moving objects is processed in an image sequence. Visual monitoring has been addressed by a large number of studies. However, there are still several challenges to overcome. The most difficult challenge is the occlusion and disappearance of the target. To remedy this challenge, we propose an algorithm that considerably improves the robustness to achieve a persistent visual tracking.

Traditional tracking algorithms lacked the total disappearance object. The Occlusion's robust monitoring has been widely studied by many researchers. The problem lies when the algorithm loses most of the information about the target. Several approaches have been sported by the research community, such as:

1. The center-weighted approach, which solves the partially occluded problem by defining a significant weight for the central pixel.
2. The correspondence approach treats the occlusion by dividing the object into several areas.
3. The predictive approach considers the motion information such as speed and acceleration to predict the object trajectory at the next frame.
4. The probabilistic approach based on the Bayesian theory that treats tracking as a problem to maximize the later Bayesian probability.

A long-term tracking of the target object in video sequences, following the occlusion or when the object leaves the field of vision, accomplished by decompositing into three sub-tasks: tracking, learning and detection. Their components form a general tracker, called TLD. The algorithm estimates the object motion and continuously follows the object to produce smooth trajectories, though it also accumulates errors proposed by [1].

Given an earlier classifier drawn from the first image and the position of the object at time t, the classifier is evaluated at several candidate locations in a search in the t + 1 area. The

Received August 21, 2017; Revised December 11, 2017; Accepted January 3, 2018

confidence map is analyzed to estimate the most likely location of the object. Finally, the classifier is updated in an unsupervised manner using randomly selected patches by [2].

Another approach [3] is to locate precisely the target object at each image in order to prevent tracking errors. Based on a structured SVM framework, it addresses the limitation of previous trackers, such as [4, 5], which separate the target location (Samples Labeling in a Research Region) and updating the model in two separate steps. This dichotomy introduces additional labeling errors to the Update Model because the sample chosen by the classifier may not correspond to the best-estimated object location.

The optical flow approach improves motion monitoring algorithm of several objects with a reformed location by [6]. Another improvement is a long-lasting algorithm, which allows a tracking with panne recovery. After choosing a tracked object in the first frame forward and backward in time, they calculate the distance between these two trajectories. If the distance is greater than the threshold, tracking is likely to fail, however the return of the most recent object model by the detector will reset the tracker. The major problem of the optical flow approach always remains the change of illumination [7]. An algorithm for object traking via prototypes is presented in [8]. The author of [9] present a robust visual tracking with an improved subspace representation model. In [10] the authors discuss an algorithm that makes product rule and weighted sum rule unified into an adaptive framework according to defined features distance. In [11] the authors introduce an algorithm of hybrid tracking through the analysis and experiments associated with the software of sporting video. A real time tracking algorithm based on particle filter with gaussian weighting is presented in [12]. We propose in this work a versatile and generic system of perception (NSSD_DT) based on an active perception strategy (synoptic diagram Figure 01).

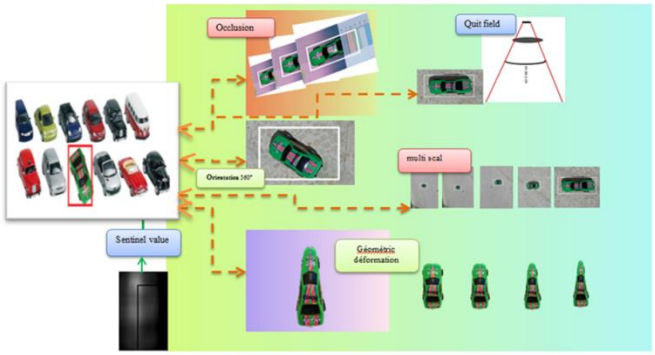

Figure 1. Synoptic Diagram

First, we studied measuring techniques of difference and similarity (SAD, SSD, NSSD, CC, NCC) for matching method on an evaluation bench to choose the most effective in terms of recognition. The study allowed the selection of the normalized square difference function (NSSD).

Then we have dealt with the problem of tracking visual objects which aims to locate a target over time, in particular, we have focused on the difficult scenarios in which the object undergoes important deformations and occlusions, or leaves the field of vision. To achieve this objective, we proposed a robust method based on a similarity sentinel index to update the template when it reaches it. Our tracker has the performance to detect tracking failure and recover after failure. We therefore solve all deformation and occlusion problems to ensure a robust tracking in the long term.

2. Matching Method

The matching model is adopted to detect small parts that corresponds to a template image. This technique is widely used in object detection fields such as vehicle tracking, robotics, medical imaging and in the industry as part of quality control.

The crucial point is to adopt an appropriate "measure" to quantify similarity or matching. However, this method also requires a high computational cost since the matching process involves moving the model image to all possible positions in a larger source image and calculating a numerical index indicating how much the pattern corresponds to the image in that position. This problem is therefore considered as an optimization problem.

The measurement of the correspondence between two images is considered as a metric that indicates the degree of resemblance or dissimilarity between them. This metric may be increasing or decreasing with a degree of similarity. When the metric is specifically indicated as a measure of inadequacy, it is an amount that increases with the degree of dissimilarity.

By sliding, we move the patch one pixel at a time (left to right, up to down). At each location, a metric is calculated, it represents “good” or “bad” match at that location. For each location of Template over source image, we store the metric in the result matrix (R). Each location (xy) in R contains the match metric. The image below is the result R of sliding the patch with a metric NSSD. The brightest locations indicate the highest matches. The location marked by the red circle is the one with the highest value. Thus, that location (the rectangle formed by that point as a corner and width and height equal to the patch image) is considered the match.

Figure 2. Result matrix (R)

$$cor = \frac{\sum_{i=0}^{N-1}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1}(x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1}(y_i - \bar{y})^2}}$$

3. Techniques Related to Matching Model

In matching, the source object can be turned, occluded or set to another scale. Techniques that provide for a distinct model for each scale and orientation, round rigid models. Though, they are too expensive, especially for large Models. The idea is to be robust and fill as much as possible all these deficiencies, with more flexibility and with an optimized cost.

The proposed approach begins with a metric study of two methods, Normalized Correlation and Normalized Square Difference. Afterwards, we propose an improvement of a selected method on the aforementioned preference criterion.

3.1. Normalized Cross Correlation Method and Normalized Square Difference Method

3.1.1. Normalized Cross Correlation Method

The cross-correlation function is an operator that acts on two functions (f (x, y), g (x, y)), each corresponding to an image. This operator has the property of 1 when the two functions are identical and of tending towards -1 when the functions are different. In 2D, to measure the relative displacement of two images along image x and y axes, a correlation algorithm (1), (2) uses this operator, taking as functions f and g respectively portions of the reference and deformed images. The algorithm searches for the values of displacements dx and dy such that g (x + dx, y + dy) maximizes the correlation operator with f. These values are retained as the best displacement estimation of the image g with respect to the image f.

$$R(x,y) = \frac{\sum_{x',y'}(T(x',y').I(x+x',y+y'))}{\sqrt{\sum_{x',y'} 2T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \quad (1)$$

In practice, the algorithm applies this procedure to image series, which are parts of the reference image. It calculates the correlation function between a reference image and a distorted image. The deformed image giving the greatest cross-correlation function with the reference image is retained as the best and thus makes it possible to estimate the displacement at this stage.

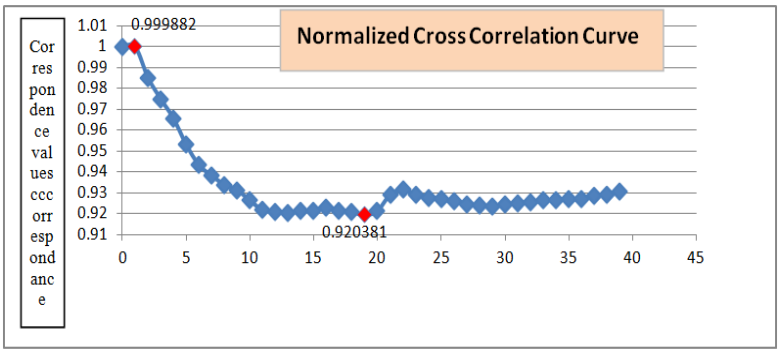

Figure 3. Correspondence values for normalized correlation method

For values varying between 0.999 and 0.930, a correspondence of 44% is obtained.

3.1.2. Normalized Square Sum Difference Method (NSSD)

$$s(x,y) = \frac{\sum_{y=0}^{k-1}\sum_{x'-0}^{w-1}[\,T\,(x',y)-I\,(\,x+x',y+y')]x^2\,]^2}{\sqrt{\sum_{y'=0}^{k-1}\sum_{x'=0}^{w-1}T(x',y')^2\ \sum_{y'=0}^{k-1}\sum_{x'=0}^{w-1}I(x+x',y+y')^2}} \quad (2)$$

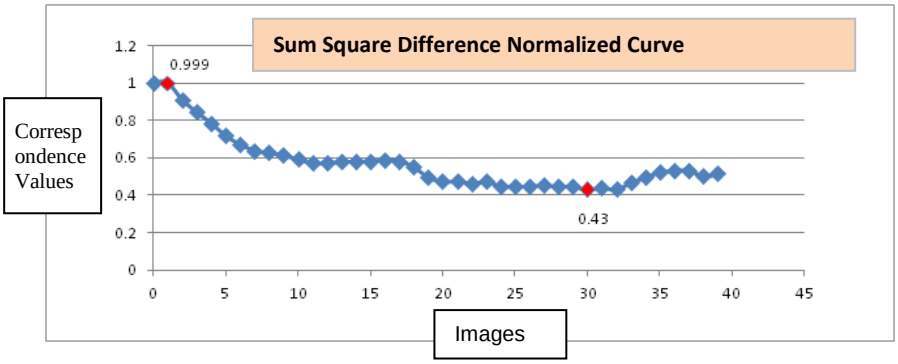

Figure 4. Correspondence values of the sum square difference normalized method

For values varying between 0.999 and 0.430, a good correspondence of 98% hence better matching resistance.

3.1.3 Comparison of matching methods

The normalized cross-correlation method, comparing concordance values to zero (ideal values), for correlation methods, normalized correlations, normalized difference and normalized square difference, compare the matching values to 1 (ideal values) [13].

After testing the six methods, the best results are given by the normalized correlation and the normalized square difference method, above the performance evaluation curves for both methods. For the two selected methods, by fixing the model and modifying the source, we obtain the curves in Figure 3 and 4 of the variations of the correspondence value according to the variation of the source.

In similarity measurements, the NSSD method has less computational cost since it is only a square operation and a subtraction of pixels between the model and the original image. In addition, it takes less time to search the area in an image corresponding to the model. On the other hand, NCC is better than the SSD because it involves a multiplication, a division and a square root operation.

3.2. Basic algorithms: NSSD

The algorithm below represents the matching based on the normalized squared difference method.

The Normalized sum squared difference algorithm
1: Loads an input image (source) and an image patch (template)
2: Perform a matching procedure template using the function matching procedure
4: Locate the location with a likelihood of adaptation
5: Draw a rectangle around the area corresponding to the highest match

4. Proposed Approch

4.1. The NSSD_DT Algorithm

The proposed Matching model NSSD_DT is based on the updating of the template according to a sentinel of recognition. The update is triggered with each change of the source in its geometric form, its scale, Rotation, or occlusion. The tracking begins with an original template. At the first change of the source that exceeds the sentinel index, the updating is done by substitution of the old template by the new one.

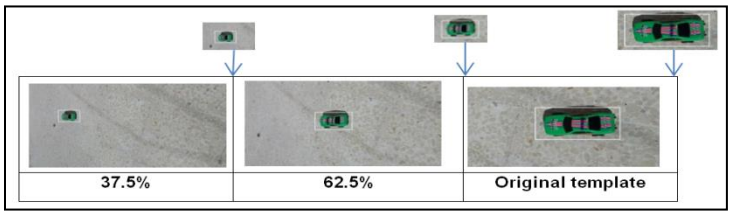

Figure 5. Sentinel principle

The Normalized sum squared difference algorithm with dynamic template

```
1: Load  an input image (source)
2: Load an image Template
3:                    Matching
 4:  If the matching factor <threshold
     {
       Updated Sizes Template
       Restore a new Template
     }
 5 :                  Matching
 6 :  Locate the location with a higher likelihood of adaptation
 7:   Locate by a rectangle the area of highest correspondence

Matching
Perform a Matching Procedure Model
  Template Reconstruction Procedure
    Home
     {Calls parameter Template (j-1)
          Extract Template (j)
          Swap Template (j-1) / Template (j)
     } End
```

4.2. Metric of the NSSD_DT algorithm

For efficient tracking without interruption, we have defined a "Sentinel coefficient" of value monitoring (0.5000). In this section, we discuss the examination of our NSSD_DT algorithm in two parts: The first part, with real-time video source with constant conditions, a single target car "racing" and a plain background, we are served by the raised results as a reference compared to the second metric section. In the second section of the metric, we treated the examination of NSSD_DT with variables resolutions video, variables number of

frames per second, occultation, rotation and change of scale. Certain videos combine two or three variations.

4.3. Metric test with real-time video camera

Test conditions; a single target with a constant background.

This section includes (1) the vertical multiscale tracking test, (2) inclined multi-scale tracking test, tracking with source rotation test, (3) Tacking with geometrical distortion test and (4) Tracking with occlusion test.

4.3.1. Vertical Tracking Multi-scale

With a step of 20° of the size, the algorithm submits to the sentinel match value (0.5000). In order to reach and approach the optimal match value (0.9000), a transposition of the Template is performed.

Table 1. Vertical Tracking Multi-scale

scale	100%	80%°	60%	40%	20%
correspondence	0.8997	0.6524	0.5236	0.3548	0.2765
Recognition Time (s)	0.261	0.254	0.262	0.225	0.245

Figure 5. Vertical Tracking Multi-scale

4.3.2. Inclined multi-scale Tracking

The robustness measurement of the algorithm continues up to 18 % of the moving object.

Table 2. Inclined Multi-scale Tracking

scale	100%	62%°	37%	25%	18%
correspondence	0.8548	0.6458	0.5125	0.478	0.2652
Recognition Time (s)	0.241	0.262	0.255	0.245	0.250

18% 25% 37% 62% 100%

Figure 6. Inclined multi-scale Tracking

4.3.3. Tracking with source rotation

Table 3. Tracking with Source Rotation

Angle	0°	29°	30°	58°	95°	96°	125°	127°
Coefficient of correspondence	0.89956	0.5001	0.89845	0.5002	0.5000	0.89954	0.5100	0.8948
Recognition Time (s)	0.266	0.268	0.258	0.257	02.62	0.261	0.267	0.266
Angle	155	157°	186°	215°	244°	273°	302°	360°
Coefficient of correspondence	0.5002	0.8956	0.5010	0.8947	0.5005	0.8890	0.5004	0.8991
Recognition Time (s)	0.272	0.268	0.259	0.269	0.265	0.256	0.265	0.262

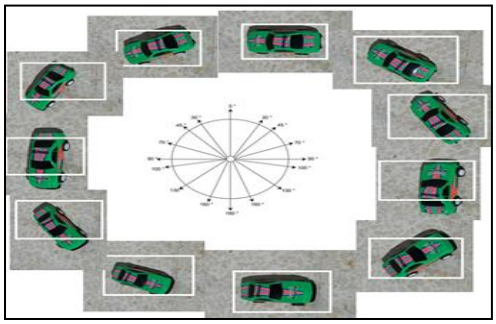

Figure 7. Tracking with source rotation

4.3.4. Tacking with geometrical distortion

Table 4. Tracking with Geometrical Distortion

Angle	0°	15°	30°	45°
correspondence	0.89979	0.6524	0.5236	0.3548
Recognition Time (s)	0.261	0.254	0.262	0.225

Figure 8. Tracking with geometrical distortion

4.3.5. Tracking with occlusion

Table 5.Tracking with Occlusion

occlusion	Original	20%	28%	48%	60%	64%	75%
correspondence	0.90125	0.7226	0.6482	0.4702	0.3625	0.3245	0.2253

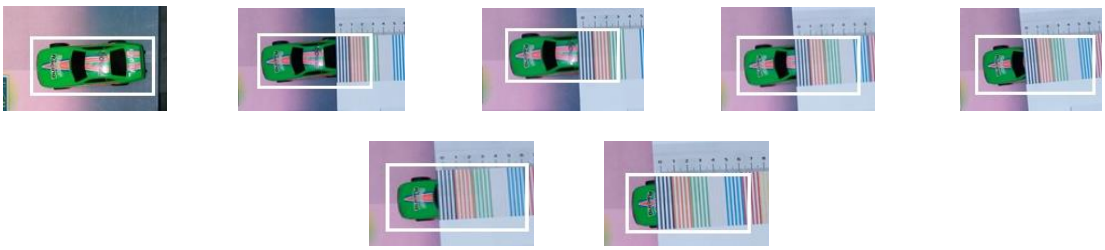

Figure 9. Tracking with occlusion

4.3.6. Comparative Curve of Correspondence tow Algorithme NSSD and NSSD_DT

The values in Table 4 represent the responses of two NSSD and NSSD_DT algorithms that are tested under test leap by applying the same change to the same sequence at 20fps and resolution 720 x 1080 with occlusions.

Figure 10. Video test for two algorithms NSSD and NSSD

Table 6. Correspondence Values for Two NSSD and NSSD_TD Algorithms

frame	10	45	80	140	255	280	310	335	370	405	437	479
Algorithm NSSD	0.8845	0.7023	0.6123	0.5861	0.5425	0.5000	0.2132	0.2332	0.2132	0.2148	0.2198	0.2181
Algorithme NSSD_DT	0.8950	0.8254	0.6578	0.6112	0.5525	0.5000	0.8945	0.7825	0.8561	0.6012	0.5525	0.5000

Images	510	534	570	595	615	640	685	710	742	800	805	900
Algorithm NSSD	0.2112	0.2325	0.2135	0.2157	0.2132	0.2254	0.2198	0.2101	0.2183	2132	0.2192	0.2153
Algorithm NSSD_DT	0.8945	0.7892	0.6560	0.6010	0.5635	0.5000	0.8951	0.7236	0.8301	0.6120	0.5501	0.5000

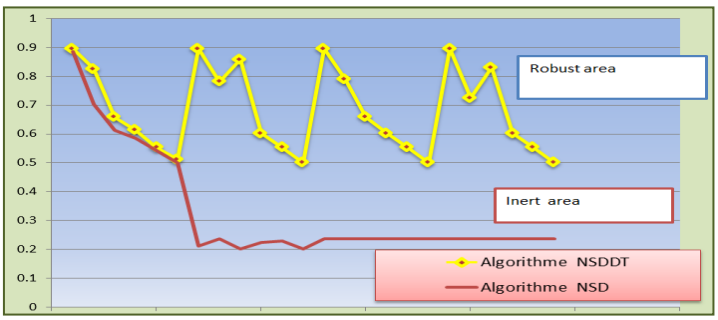

Figure 11. Divergence between the two algorithms NSSD and NSSD_DT

4.4. Evaluation of the results on sequence video

In this section, in order to evaluate our approach, we put our NSSD_DT algorithm implemented in C++ on 8 different videos at 20 Fps: different resolution (v1), rotation of the source and the change of scale (v2, v3, v4) and we resume tracking of race car after leaving the field of vision (v5).

4.4.1 Tracking with fixed template and different resolutions of the source

With a fixed resolution template 1080x1920, we track the targets at different resolution. Table 7 summarizes the results that represents the average of 10 values raised for each resolution. The recognition results are evaluated at an average of 84%.

Table 7. Fixed Template and Different Resolutions of the Source

resolution	Average recognition rate	Average execution time (s)
1080 x 1920 HD	0.8754	0.321
720 x 1080 HD	0.8354	0.295
480 x 720	0.8452	0.261
320 x 480	0.8356	0.257
240 x 320	0.8147	0.243
240 x 144	0.7984	0.222
Average	92.03%	

Figure 12. Different resolution video (v1)

4.4.2 Test avec rotation de la source

The test is applied to three industrial videos; v2 to track the madeleine, v3 to track the car hull on chain and v4 to track the red stylot maintained by the robot bra. Table 8 summarizes the measurements and gives an average of recognition of 87%.

Table 8. Pen Tracking Table with Rotation and Scale Change

video	Fps	Average recognition rate	Average execution time (s)
V2	20	0.7584	0.262
V3	20	0.8455	0.236
V4	20	0.7562	0.256
	Average	87%	

Figure 13. Target Rotation

Figure 14. Target Rotation

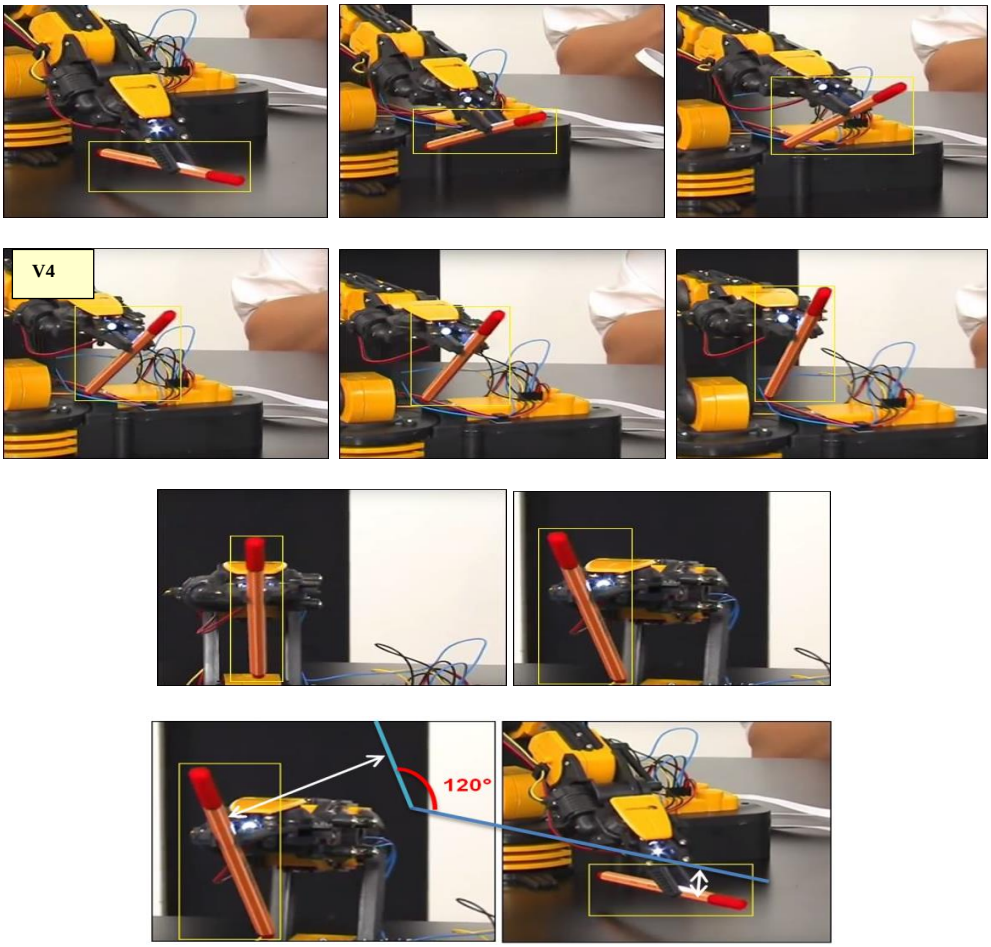

Figure 15. Pen tracking with rotation and scaling

4.4.3. Partial Occlusion and Field of View Output Test

The evaluation of our algorithm is replicated 5 times on this sequence. The following table summarizes the average of recognition and the average of the execution time.

Table 9. The Average Result for Oclusions

Fram	Average recognition rate	Oclusions	Average execution time (s)
50	0.8523		0.2562
75	0.7912		0.2623
93	0.7455		0.2365
120	0.7562		0.2563
144	0.6214	Partiel oclusion	0.2652
176	0.2356	Total oclusion	0.2546
198	0.5784	Partiel oclusion	0.2563
215	0.7587		0.2485
Average	**74%**		

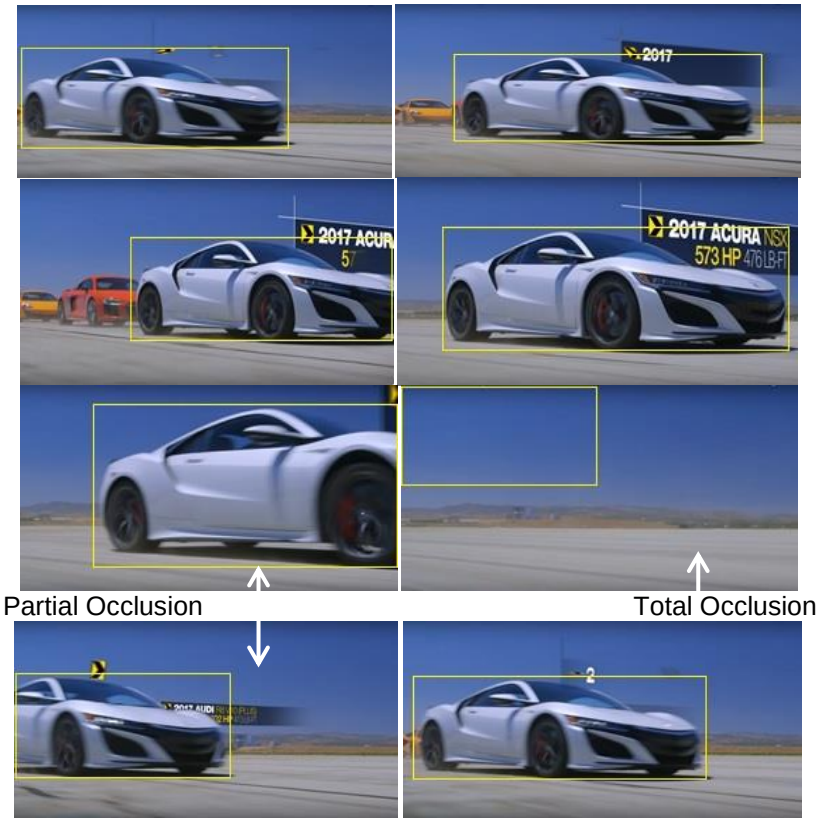

Figure 16. tracking with scale variation and recovery after quit field

4.5. Recap of Results

Based on the various robustness tests of our tracker, we obtain very relevant results with an average overall recognition rate of 84%.

Table 10. Recap of Results

	Recognition tests	Recognition rate
a.	Tracking with different resolutions	92%
b.	Tracking with source rotation	87%
c.	Tracking after quitting the field of view	74%
	General Recognition	84 %

5. Discussion

The principle of tracking is based on a vigilant control on the index of similarity. If the latter reaches the nominal value predefinit (i.e. the sentinel value) then an immediate update of the template is triggered to continue the tracking with similarity closer to the ideal.

This assumes that the value of the sentinel index must be very well chosen, because a value close to 1 affects the strength of the tracking. Besides, a value too low than 0.5 risks the accuracy of target marking. A series of tests for a given subject makes it possible to choose this famous index. Generally a value near to 0.5 gives satisfactory results.

6. Conclusion

In this work, we have dealt with the problem of tracking visual objects. The goal is to locate a target over time. In particular we have focused on difficult scenarios in which the object leaves the field of vision or undergoes important deformations and occlusions. In order to achieve this objective, we proposed a robust method based on a sentinel response index

identifying the state of the object and we update the template when it reaches this index. Our tracker has the performance to detect tracking failure and recover after failure. Therefore, we solve all problems of deformation and occlusion to ensure a robust tracking in the long term.

Acknowledgements

The authors would like to thank the University of Monastir and the electronics and microelectronics research laboratory (LR99ES30) for their support in publishing this work.

References

[1] Z Kalal, J Matas, K Mikolajczyk. *P-N learning: Bootstrapping binary classifiers by structural constraints.* Proc. IEEE Conf. Comput. Vis. Pattern Recognit., San Francisco, CA, USA. 2010: 49–56.
[2] Grabner H, Leistner C, Bischof H. *Promotion semi-surveillée en ligne pour un suivi robuste*. Vision informatique – ECCV. 2008; 234-247.
[3] Bai Y, Tang M. *Un suivi robuste par svm de classement faiblement supervisé. En vision informatique et reconnaissance de motifs (CVPR)*. 2012 IEEE Conference on. 2012: 1854-1861.
[4] Grabner H, Grabner M, Bischof H. *Real-time tracking via on-line boosting*. Proc. BMVC. 2006; 1: 47-56.
[5] Babenko B, Yang MH et Belongie S. Suivi d'objet robuste avec apprentissage en ligne multiple. *Les transactions IEEE sur l'analyse des modèles et l'intelligence de la machine*. 2011; 33(8): 1619-1632.
[6] HAMMAMI K, Salim HENI, Mohamed ATRI. Improvement of Detection and Tracking of Movement. *International Journal of Engineering and Future Technology™*. 2016; 2(1): 3-9.
[7] Daode Zhang, Cheng Xu, Yuanzhong Li. Robust Tracking based on Failure Recovery. *TELKOMNIKA I Telecommunication Computing Electronics and Control.* 2014; 12(2); 1005-1011.
[8] D Wang, H Lu, MH Yang. Online object tracking with sparse prototypes. *IEEE Trans. Image Process*. 2013; 22(1): 314–325.
[9] CHENG, Jing et KANG, Sucheng. Robust Visual Tracking with Improved Subspace Representation Model. T*ELKOMNIKA Telecommunication Computing Electronics and Control*. 2017; 15.
[10] CAO, Jie, GUO, Leilei, WANG, Jinhua, et al. Suivi d'objet basé sur plusieurs fonctions Adaptive Fusion. *Indonesian Journal of Electrical Engineering and Computer Science*. 2014; 12(7): 5621-5628.
[11] Tao ZH. A Tracking Algorithm of Moving Target in Sports Video. *Indonesian Journal of Electrical Engineering and Computer Science.* 2014; 12(10): 7463-7470.
[12] Indah Agustien, Siradjuddin, M Rahmat Widyanto, T Basaruddin. Particle Filter with Gaussian Weighting for Human Tracking. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2012; 10(4): 801- 806
[13] Hisham MB, Yaakob SN, Raof RA, Nazren AA, Embedded NW. *Template Matching using Sum of Squared Difference and Normalized Cross Correlation.* Research and Development (SCOReD), 2015 IEEE Student Conference on. 201: 100-104.