A ROBUST, PRECISE AND FLEXIBLE TRACKING ALGORITHM BASED ON IMS AND SWAD

G. Di Caterina and J. J. Soraghan

Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, UK

ABSTRACT

The improved mean shift (IMS) algorithm can effectively track a colour target undergoing fast motion and complete occlusion. Instead template matching based on the sum of weighted absolute differences (SWAD) can track a target precisely, even under partial occlusion. To take advantage of both trackers, this paper presents a novel tracking algorithm that, on one side, further improves the IMS and extends it to a more general and flexible framework; on the other side, the proposed algorithm incorporates SWAD minimisation to increase the precision of the tracking results. Experimental results demonstrate that such an algorithm can robustly and precisely track targets with fast motion and in complete occlusion, more than conventional mean shift, IMS and SWAD-based trackers taken separately.

***Index Terms*—** visual tracking, improved mean shift, sum of weighted absolute differences

1. INTRODUCTION

Target tracking is a key element in many visual systems and several tracking algorithms have been proposed in the literature, as reported in [1]. The improved mean shift (IMS) described in [2] has been shown to be fast and effective in tracking a colour target with fast motion, undergoing prolonged complete occlusion and in complex scenarios. However a weakness of this algorithm, as of the original mean shift tracker introduced in [3, 4], is its low precision during tracking. In other words, the IMS can effectively locate a target in consecutive frames, but the computed target positions are not very precise, with a sort of shaking effect on the overall track. A further limitation of the IMS is that, as it stands, it does not offer much room for improvement, to incorporate additional metrics or techniques for comparing the target model with a target candidate in the current frame.

Template matching based on the sum of weighted absolute differences (SWAD) is a simple, but yet precise tracking algorithm, as demonstrated in [5, 6]. However, although being very efficient over a set of frames wherein the target template is visible and does not change drastically, it dramatically fails when the target has very fast motion and is subject to complete occlusion.

To overcome the weaknesses of both IMS and SWAD-based trackers, while retaining and combining their strengths, in this paper we present a novel tracking algorithm which can precisely track very fast targets, in complex scenarios, also after prolonged complete occlusion. The novelty of the algorithm proposed in this paper is therefore three-fold: first, the IMS is generalised to obtain a more flexible framework, wherein it is easy to incorporate further matching metrics and techniques; second, the IMS initialisation and restart point selection procedures are improved, to make them more data-independent; third, the SWAD is combined with the novel IMS-based framework, to guarantee tracking precision to the overall algorithm.

The rest of the paper is organised as follows. Section 2 describes the novel tracking algorithm, including improved initialisation and SWAD-based stabilisation. Experimental results are reported in section 3, while section 4 concludes the paper.

2. TRACKING ALGORITHM

2.1. Overview

A flowchart of the proposed tracking algorithm is shown in Figure 1. The target to track is directly selected in the initial frame $\mathbf{F}_0$, in position $\mathbf{y}_0$. Then the initialisation step computes the target model, represented by the target histogram $\mathbf{Q}$, from $\mathbf{F}_0$; it also computes an initial value for the threshold τ, which is the smallest distance $d(\cdot) = \sqrt{1 - \rho(\cdot)}$, as in [2], between the target histogram $\mathbf{Q}$ and any other object in the frame, other than the target itself.

After the acquisition of a new frame $\mathbf{F}_i$, the algorithm checks if the position $\mathbf{y}_{i-1}$ of the target in the previous frame $\mathbf{F}_{i-1}$ is defined, i.e. the target has been found in the previous frame. Although this check is trivial for $i = 1$, it is indeed necessary when $i > 1$. In fact, if the target has not been found in $\mathbf{F}_{i-1}$, the algorithm proceeds directly to the computation of candidate restarting points in $\mathbf{F}_i$. If the position in $\mathbf{F}_{i-1}$ is defined, a single step of the conventional MS tracker, referred to as "MS loop" in [2], is executed; such a loop computes the new possible target position $\mathbf{y}_i$ in the current frame, starting from previous position $\mathbf{y}_{i-1}$, with $d(\mathbf{y}_i)$ being the measure of similarity between the target model $\mathbf{Q}$

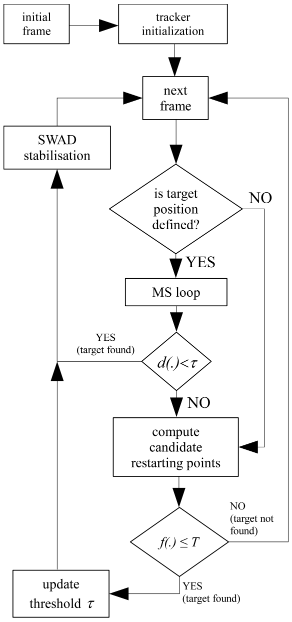


Fig. 1. Flowchart of the tracking algorithm.

and the target candidate found in $\mathbf{y}_i$. The distance $d(\mathbf{y}_i)$ is then compared with the threshold τ, computed as described in the initialisation step in section 2.2. If $d(\mathbf{y}_i) < \tau$, the correct target is found and the algorithm tries to further improve the precision of the track by applying template matching based on SWAD, in a small neighbourhood of $\mathbf{y}_i$, as described in section 2.4. If $d(\mathbf{y}_i) \geq \tau$, one of the three following cases may have occurred:

1. the target found by the MS loop is the correct one, but its appearance has changed, possibly due to light changes or partial occlusion;
2. the actual target has moved substantially from the previous position, so the MS loop has lost the target and the returned position $\mathbf{y}_i$ belongs to the background;
3. the real target is completely occluded, i.e. not visible, and $\mathbf{y}_i$ belongs to the occluding object.

To check which of the three conditions above is true, a set of L restarting points $\{\mathbf{x}_l\}_{l=1..L}$ is computed within the current frame $\mathbf{F}_i$, with a procedure similar to the initialisation step, and described in section 2.3. After that, any additional metric $f(\cdot)$ can be applied to verify whether any of these points belongs to the target, $f(\mathbf{x}_l) \leq T$, or not, $f(\mathbf{x}_l) > T$. If yes, the target is deemed as found, τ is updated and the SWAD-based stabilisation takes place. If $f(\cdot) > T$, the target is not found and the algorithm proceeds directly to the next frame.

2.2. Tracker initialisation

The outputs of the initialisation step are the threshold τ and the target colour distribution $\mathbf{Q}$. While $\mathbf{Q}$ is computed as described in [4], the initial value of τ is obtained as follows.

For a colour target, it is possible to compute in every frame $\mathbf{F}_i$ the distance $d(\cdot)$ between the target model $\mathbf{Q}$ and a candidate distribution $\mathbf{P}(\mathbf{x}_n)$ in a generic point $\mathbf{x}_n$ in $\mathbf{F}_i$. Assuming that the target colour distribution does not change significantly over a certain number of frames, it is clear that the smallest value of $d(\cdot)$ is obtained when $\mathbf{x}_n$ belongs to the actual target. Therefore the distance $d(\cdot)$ between $\mathbf{Q}$ and the background, i.e. any other object in the frame, is always higher than the distance between $\mathbf{Q}$ and the actual target. This suggests that it is appropriate to set τ equal to the smallest distance between $\mathbf{Q}$ and the background. In this way, if $d(\cdot) \geq \tau$, the tracker is likely to have lost the target and most likely is following another object in the background.

To compute the initial value of the threshold τ, the distribution $\mathbf{P}(\mathbf{x}_n)$ should be computed for every point $\mathbf{x}_n$ in the frame $\mathbf{F}_0$ not belonging to the target. As this approach is clearly not practical in terms of computational speed even on very fast machines, the distribution $\mathbf{P}(\cdot)$ is computed only in a small number of candidate points $\{\mathbf{x}_l\}_{l=1..L}$ belonging to the background. The selection of such points is based on the presence in a neighbourhood around $\mathbf{x}_l$ of colours peculiar to the selected target.

As a first step to the selection of $\{\mathbf{x}_l\}_{l=1..L}$, the range of discriminative colours of the target is determined. The position $[r_m, g_m, b_m]$ of the maximum in $\mathbf{Q}$ is given by:

$$[r_m, g_m, b_m] = \underset{r,g,b \in [1,\beta]}{\arg\max} (q_{rgb}) \tag{1}$$

where q_{rgb} is a generic bin in the colour histogram $\mathbf{Q}$. The discriminative colours in the selected target are identified by a set of bins q_{rgb} around $[r_m, g_m, b_m]$ in $\mathbf{Q}$. The distribution probabilities in $\mathbf{Q}$ of the selected range of colours are backprojected onto the initial frame $\mathbf{F}_0$, similar to [7]. Kernel density estimation is then applied to the backprojected probability image, using the Epanechnicov weighting kernel $k(x)$ [8] as in [3]. In the estimated density function obtained, local maxima correspond to points where there is a high concentration of target discriminative colours, i.e. where the visible target is more likely to be. Therefore the L highest local maxima belonging to the background, and not the target, are selected as candidate points $\{\mathbf{x}_l\}_{l=1..L}$. For each point $\mathbf{x}_l$, the MS loop is executed to obtain a value of $d(\cdot)$ associated with $\mathbf{x}_l$. Finally, the initial value of τ is set equal to the smallest $d(\cdot)$ for all the candidate points $\{\mathbf{x}_l\}_{l=1..L}$:

$$\tau = \underset{\{\mathbf{x}_l\}_{l=1..L}}{\arg\min} (d(\mathbf{x}_l)) \tag{2}$$

2.3. Restarting points and additional metric $f(\cdot)$

The computation of the restarting points $\{\mathbf{x}_l\}_{l=1..L}$ is the same as in the initialisation step, with the only difference being that now the L highest local maxima can be selected in any position of the frame $\mathbf{F}_i$.

The flexible aspect of the novel algorithm presented in this paper is represented by the further comparison $f(\cdot) \leq T$, as in Figure 1. Here $f(\cdot)$ can be any metric or matching technique, not even necessarily based on colour. The only assumption is that for any two points, $\mathbf{x}^{tg}$ belonging to the target and $\mathbf{x}^{bg}$ belonging to the background, it holds that:

$$f(\mathbf{x}^{tg}) \leq T < f(\mathbf{x}^{bg}) \tag{3}$$

where T is the separation element, i.e. threshold, between any $\{f(\mathbf{x}^{tg})\}$ and $\{f(\mathbf{x}^{bg})\}$

The usage of $f(\cdot)$ transforms the IMS from [2] into a robust framework, where the MS loop is an efficient procedure to select a reduced number of candidate points $\{\mathbf{x}_l\}_{l=1..L}$, to be tested with any metric $f(\cdot)$ of choice. A simple example for $f(\cdot)$ is $f(\cdot) = d(\cdot)$. In this case $T = \tau$ and the proposed algorithm reduces to the basic IMS, with improved initialisation and SWAD stabilisation.

If it exists any $\mathbf{x}_n \in \{\mathbf{x}_l\}_{l=1..L}$ for which $f(\mathbf{x}_n) \leq T$, the target is considered as found and its new position is $\mathbf{y}_i = \mathbf{x}_n$. The threshold τ is then updated as:

$$\tau = min(\tau, \tau') \tag{4}$$

where τ' is the minimum distance $d(\cdot)$ computed over the subset of the L restarting points not belonging to the target found in position $\mathbf{x}_n$.

2.4. SWAD-based stabilisation

To improve the precision of the tracker over a group of consecutive frames in which the target has been found, a stabilisation step based on the minimisation of the SWAD metric [5] is added to the algorithm, as shown in Figure 1. For this purpose, the algorithm keeps a target template $\mathbf{T}_i$. However, as the stabilisation step is performed only when the target has been found in the current frame $\mathbf{F}_i$ and the target position $\mathbf{y}_i$ is available, $\mathbf{T}_i$ is left undefined when $\mathbf{y}_i$ is not available.

For a set of consecutive frames where $\mathbf{y}_i$ is defined and therefore the target is visible, the template $\mathbf{T}_i$ is re-initialised in the first frame of the set and the SWAD stabilisation is carried out. For the following frames in the set, $\mathbf{T}_i$ is updated using an infinite impulse response filter approach.

A visual representation of such a process is given in Figure 2, where a set of 9 consecutive frames is depicted. Frames where the target is not found are referred to as U-frames ($\mathbf{F}_3$–$\mathbf{F}_4$, $\mathbf{F}_6$), while frames where $\mathbf{y}_i$ is defined are referred to as D-frames ($\mathbf{F}_0$–$\mathbf{F}_2$, $\mathbf{F}_5$, $\mathbf{F}_7$–$\mathbf{F}_8$). As it can be seen, the target template $\mathbf{T}_i$ can be:

- empty – in all the U-frames ($\mathbf{F}_3$–$\mathbf{F}_4$, $\mathbf{F}_6$);
- re-initialised – in the first D-frame after a U-frame ($\mathbf{F}_0$, $\mathbf{F}_5$, $\mathbf{F}_7$);
- updated – in the consecutive D-frames ($\mathbf{F}_1$–$\mathbf{F}_2$, $\mathbf{F}_8$).

SWAD matching is performed on gray scale images, so if the current frame $\mathbf{F}_i$ is a D-frame, a gray scale version $\mathbf{F}_i^{gray}$ of it is computed.

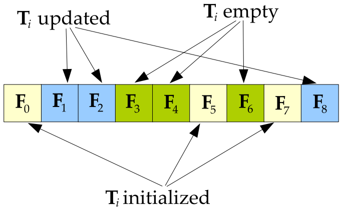


Fig. 2. A set of 9 consecutive frames. Frames $\mathbf{F}_3$–$\mathbf{F}_4$, $\mathbf{F}_6$ are U-frames, i.e. the target is not found. Frames $\mathbf{F}_0$–$\mathbf{F}_2$, $\mathbf{F}_5$, $\mathbf{F}_7$–$\mathbf{F}_8$ are D-frames, i.e. the position $\mathbf{y}_i$ of the target is defined.

Table 1. Target area corner coordinates and relative initial frames.

	TOP-LEFT	BOTTOM-RIGHT	FRAME
S1	(303,741)	(329,757)	#23
S2	(304,743)	(326,759)	#14
S3	(277,654)	(299,676)	#1
S4	(191,76)	(217,88)	#111
S5	(222,611)	(250,629)	#474
S6	(42,137)	(58,153)	#1

When the template $\mathbf{T}_i$ is re-initialised, it is set equal to a portion of $\mathbf{F}_i^{gray}$ around $\mathbf{y}_i$. The dimensions $h_T \times w_T$ of $\mathbf{T}_i$ are equal to the current target size $h_i \times w_i$. The template dimensions stay the same for all the following D-frames, until $\mathbf{T}_i$ is re-initialised.

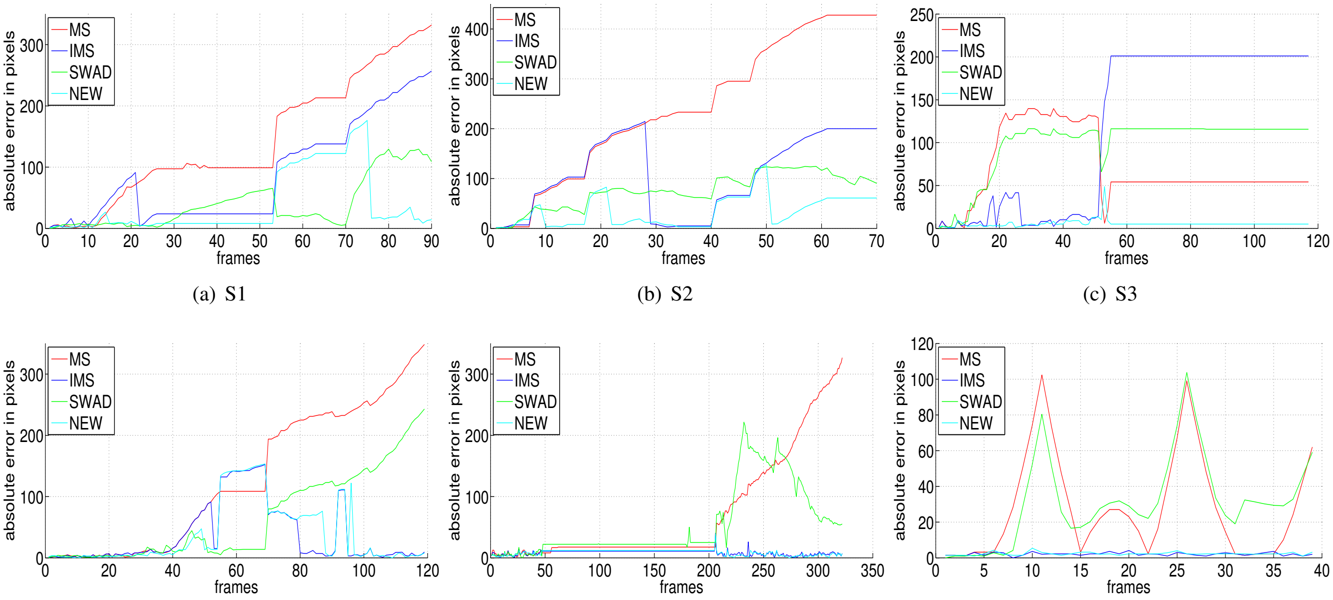
The actual SWAD matching is performed in every D-frames, except when $\mathbf{T}_i$ is re-initialised. By minimising the SWAD coefficient $\psi(v, z)$ over row index v and column index z, the position of the best match $\hat{\mathbf{T}}_{i-1}$ for $\mathbf{T}_{i-1}$ in $\mathbf{F}_i^{gray}$ is obtained. This position becomes the final target position in the current frame $\mathbf{F}_i$. After this, the template is updated as:

$$\mathbf{T}_i = (1 - \alpha)\mathbf{T}_{i-1} + \alpha\hat{\mathbf{T}}_{i-1} \tag{5}$$

where $\alpha \in [0, 1]$ is a blending factor. As we are mainly interested in stabilising the position of the target across single pairs of D-frames, higher weight in (5) can be given to the best match $\hat{\mathbf{T}}_{i-1}$, rather than the previous template $\mathbf{T}_{i-1}$. Therefore the blending factor α should have a value higher than 0.5. In the current implementation of the tracker, it is $\alpha = 0.8$.

3. EXPERIMENTAL RESULTS

To evaluate the performance of the novel algorithm proposed in this paper, with respect to the conventional MS, IMS and SWAD-based tracking, six standard video sequences have been used: (S1) S3-Multiple flow-time14.46-view1, (S2) S3-Multiple flow-time14.52-view1, (S3) S2-L2-time14.55-view4, (S4) S3-Multiple flow-time14.13-view1, (S5) S2-L1-time12.34-view5 from [9], and (S6) the Table Tennis test sequence from [10]. Pixel coordinates, in the format *(row,column)*, of top-left and bottom-right corners of the target areas tracked in the six sequences are reported in Table 1,



(a) S1

(b) S2

(c) S3

(d) S4

(e) S5

(f) S6

Fig. 3. Graphs of the tracking results.

Table 2. Mean absolute error (in pixels).

μ	MS	IMS	SWAD	NEW
S1	143.30	87.88	40.03	38.07
S2	243.24	108.52	76.13	33.69
S3	91.65	12.18	77.52	4.87
S4	121.36	38.55	59.19	37.43
S5	56.30	8.35	54.17	7.10
S6	24.16	2.13	26.08	2.27

Table 3. Standard deviation of error (in pixels).

σ	MS	IMS	SWAD	NEW
S1	101.19	82.22	40.96	51.70
S2	138.80	78.24	35.03	31.49
S3	54.16	12.14	44.19	2.97
S4	108.36	50.72	64.41	49.38
S5	70.15	4.05	59.02	3.21
S6	25.49	0.91	19.24	0.87

alongside the number of frame wherein the target is initially selected. Ground truth for these targets has been gathered through manual labeling.

Here the novel algorithm is indicated with "NEW". As it can be appreciated from the graphs in Figure 3, the MS loses the target in each sequence; this is confirmed by its absolute error diverging, especially in Figure 3(a), (b), (c), (d) and (f). The SWAD drifts and loses the target in S1, S2, S3 and S4, while the fast motion of the ball is the cause of the fail of SWAD in S6. In S5, SWAD can track the target until it goes out of the field of view; when that happens, SWAD fails and it is not able to recover the target, once it is visible again.

On the contrary, IMS and NEW can successfully track the target in all the sequences. However, as it can be noticed from the values of mean absolute error and standard deviation reported in Table 2 and 3, NEW has the lowest μ and σ in S1-S5, while IMS performs slightly better in S6. Nonetheless, in S6, IMS is better only for ≈ 0.13, while NEW is almost twice as accurate in S1-S3. Concerning precision, i.e. standard deviation, NEW is more precise than IMS in all sequence.

These results prove that, as it stands, the novel algorithm presented in this paper is already more accurate and precise than MS, IMS and SWAD, even when it is simply $f(\cdot) = d(\cdot)$, as in its current implementation. It is expected that the tracking performance of the novel algorithm can be further improved by adopting a more accurate metric for $f(\cdot)$.

4. CONCLUSION

In this paper we have a presented a novel tracking algorithm, which is more accurate and precise than the conventional mean shift, IMS and SWAD-based tracker. The proposed algorithm extends the IMS to give the possibility to include within the algorithm other metrics and matching techniques. The resulting tracking framework is therefore more flexible than its predecessor IMS. Higher precision is guaranteed by including a SWAD-based stabilisation step in the novel algorithm. Moreover, initialisation and restarting point computation in the IMS have been further improved. Experimental results have shown that the presented algorithm can accurately and precisely track fast targets, in complex scenarios and after prolonged complete occlusion.

5. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Computing Surveys*, vol. 38, no. 4, pp. 1–45, 2006.

[2] G. Di Caterina and J. J. Soraghan, "An improved mean shift tracker with fast failure recovery strategy after complete occlusion," in *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, 2011, pp. 130–135.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE Conference Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 142–149.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[5] G. Di Caterina and J. J. Soraghan, "Adaptive template matching algorithm based on SWAD for robust target tracking," *IET Electronics Letters*, vol. 48, no. 5, pp. 261–262, 2012.

[6] G. Di Caterina, I. Hunter, and J. J. Soraghan, "DSP embedded smart surveillance sensor with robust SWAD-based tracker," in *Advanced Concepts for Intelligent Vision Systems*. 2012, vol. 7517 of *Lecture Notes in Computer Science*, pp. 48–58, Springer Berlin / Heidelberg.

[7] M. J. Swain and D. H. Ballard, "Indexing via color histograms," in *IEEE International Conference on Computer Vision*, 1990, pp. 390–393.

[8] B.W. Silverman, *Density estimation for statistics and data analysis*, Chapman and Hall, 1986.

[9] PETS 2009, "Benchmark Data," http://www.cvg.rdg.ac.uk/PETS2009/a.html, 2009.

[10] "Xiph.org – derf's test media collection," http://media.xiph.org/video/derf/.