A Superior Tracking Approach: Building a Strong Tracker through Fusion

Christian Bailer[1], Alain Pagani[1], and Didier Stricker[1,2]

[1] German Research Center for Artificial Intelligence, Kaiserslautern, Germany
{Christian.Bailer,Alain.Pagani,Didier.Stricker}@dfki.de
[2] University of Kaiserslautern, Germany

Abstract. General object tracking is a challenging problem, where each tracking algorithm performs well on different sequences. This is because each of them has different strengths and weaknesses. We show that this fact can be utilized to create a fusion approach that clearly outperforms the best tracking algorithms in tracking performance. Thanks to dynamic programming based trajectory optimization we cannot only outperform tracking algorithms in accuracy but also in other important aspects like trajectory continuity and smoothness. Our fusion approach is very generic as it only requires frame-based tracking results in form of the object's bounding box as input and thus can work with arbitrary tracking algorithms. It is also suited for live tracking. We evaluated our approach using 29 different algorithms on 51 sequences and show the superiority of our approach compared to state-of-the-art tracking methods.

Keywords: Object Tracking, Data Fusion.

1 Introduction

Visual object tracking is an important problem in computer vision, which has a wide range of applications such as surveillance, human computer interaction and interactive video production. Nowadays, the problem can be robustly solved for many specific scenarios like car tracking [18] or person tracking [2,27]. However, object tracking in the general case i.e. when arbitrary objects in arbitrary scenarios shall be tracked can still be considered as widely unsolved. The possible challenges that can occur in an unknown scenario are too various and too numerous to consider them all with reasonable effort within one approach – at least with todays capabilities. Classical challenges are for example illumination changes, shadows, translucent/opaque and complete occlusions, 2D/3D rotations, deformations, scale changes, low resolution, fast motion, blur, confusing background and similar objects in the scene.

As the evaluation in [29] and our comparison in Table 1 shows, each tracking algorithm performs well on different sequences. An on average good algorithm might fail for a sequence where an on average bad algorithm performs very well. For example in Table 1 the on average best algorithm SCM [35] fails in the *lemming* sequence, while the on average second worst algorithm SMS [8] outperforms every other algorithm on this sequence. This shows that different tracking algorithms master different challenges that can occur in general object tracking and that an approach which combines the strengths of different algorithms while avoiding their weaknesses could outperform each single algorithm by far.

A possibility for this combination is the fusion of the tracking results of different algorithms into one result. In this paper we show that we can actually clearly outperform single algorithms with this approach. Furthermore, we show that our fusion approach can generate good results for many more sequences than the globally best tracking algorithm. Moreover, our fusion approach often outperforms even the best tracking algorithm on a sequence by up to 12% in *success score* in our tests. Thanks to trajectory optimization our fusion result is also continuous and smooth as expected from standard tracking algorithms – we even outperform tracking algorithms in this regard. For the best results trajectory optimization has to run offline, but we can also obtain very good results with pure online fusion. Online means that only tracking results of the current and past frames can be considered to create the fusion result for the current frame. This makes it suitable for live tracking. In offline fusion the tracking result for a whole sequence is known beforehand. Moreover, we present a short runtime evaluation and we show the robustness of our approach towards bad tracking results. Our approach is very generic and can fuse arbitrary tracking results. As input it needs only tracking results in the form of rectangular boxes.[1]

2 Related Work

In this section we give a general overview of fusion approaches for object tracking. For an overview of common object tracking algorithms we refer to recent state of the art review articles [32,6] and general tracker evaluation articles [29,30].

Fusion of tracking algorithms can be performed actively with feedback to the tracking algorithms or passively without feedback. As tracking algorithms are usually not designed to receive feedback and to be corrected active fusion requires specific tracking methods that work hand in hand with the fusion approach. One such approach is PROST [25] which combines optical flow tracking, template tracking and a tracking by detection algorithm in a very specific way. Thus, the three component algorithms can only be replaced very similar methods. Further active fusion approaches are VTD [19] and VTS [20]. These also require special tracking algorithms which fit into the proposed probability model and the tracking algorithms need to be defined by an appropriate appearance and motion model to be compatible with their approach. It is also possible to integrate common tracking algorithms into an active model. However, active fusion with many tracking algorithms is extremely complex as the fusion approach has to consider the specifics of each algorithm. Furthermore, feedback in the form of position correction is problematic as it forces tracking algorithms to follow one truth that might be incorrect and thus leads to permanent tracking failure.

[1] However, optional labeled training data can improve the results of our approach.

In contrast, passive approaches work with arbitrary tracking algorithms as long as these provide outputs that are compatible with the fusion approach. To our knowledge, the only existing passive approach is part of our previous work [4]. In this work the aim was to create a user guided tracking framework that produces high quality tracking results with as little user input as possible. The framework allows the user to select the object in different frames and to track the sequence with several different tracking algorithms. One feature to keep the effort small was a fusion approach that allows the user to check one fused result instead of several tracking results. In the fusion part of [4], we first search the biggest group of tracking result boxes that have all an overlap above a threshold to each other and then we average these boxes. If there are two groups with the same size we prefer the group with the greater overlap to each other.

3 Fusion of Tracking Results

In this section we describe our fusion approach. The input for our approach are M tracking results T_j, $j \in [1...M]$ for an object in a sequence. Each tracking result consist out of N bounding boxes $b_{i,j}$, $i \in [1...N]$ – one for each frame i in the sequence. For online/live tracking N is incremented for each new frame. Each tracking result T_j is considered to be created by a different tracking algorithm j. The fusion result of our approach that we call T^* also consists of one rectangular box for each frame. Our fusion approach works online, but some parts provide a better performing offline version as well (if mentioned).

3.1 The Basic Approach

A common approach in data fusion is majority voting. Our previous work [4] is also based on this idea. However, in tracking this requires a threshold parameter that defines if two result boxes vote for the same position. In our experiments, such thresholds showed to be very sequence dependent. Instead our approach is based on the idea of attraction fields, which does not need thresholds. The closer a fusion candidate is to a tracking result box the stronger it is attracted by it. The sum of attractions for all tracking results can be seen as an energy function that we want to maximize to find the fusion result. Attraction is computed in a 4 dimensional space to consider not only the object position but also the object size. The 4 dimensions are the x and y position of the box center and the box's width w and height h. The distance d between two boxes b and c is computed as:

$$d(b,c) = \left\| (d_x(b,c), d_y(b,c), d_w(b,c), d_h(b,c))^T \right\|_2 \tag{1}$$

$$= \left\| \left(2\frac{c_x - b_x}{c_w + b_w}, 2\frac{c_y - b_y}{c_h + b_h}, 2\alpha\frac{c_w - b_w}{c_w + b_w}, 2\alpha\frac{c_h - b_h}{c_h + b_h} \right)^T \right\|_2 \tag{2}$$

α is a constant which determines the influence of scale to the distance. It has no influence if the two boxes have the same size. The attraction function (or energy) for a candidate box c in a frame i defined as:

$$a_i(c) = \sum_{j \in M} \frac{1}{d(b_{i,j}, c)^2 + \sigma} \tag{3}$$

σ is a constant that not only avoids infinite attraction for a zero distance, but also reduces attraction increase close to zero. This prevents a perfect match to a box $b_{i,j}$ from getting a higher overall attraction than a position with good agreement to many close-by boxes. Thus, a well chosen σ is useful for noise reduction. In order to find the fusion result box $c_i^* \in T^*$ that gets the greatest attraction for a frame i we first test all tracking result boxes $R_i := \{b_{i,1} ... b_{i,M}\}$ for how much attraction they get and keep the one with the greatest attraction. Then we perform gradient ascent starting from that box to determine c_i^*.

3.2 Tracker Weights

Different tracking algorithms perform on average different well and it is reasonable to trust algorithms more if they perform on average better. We can consider this by adding weights to tracking algorithms, if ground truth labeling for some sequences is available to determine the weights. If we call G_s^i the ground truth labeling for a sequence s at frame i, the weight w_j for a tracking algorithm j is determined as:

$$w_j = \sum_{s \in S} \sum_{i \in N} \frac{1}{d(G_i^s, b_{i,j}^s)^2 + \sigma} \tag{4}$$

S is the set of all sequences from which we determine weights. Normalization is not necessary as long as all w_j are determined on the same frames of the same sequences. The weighted attraction function is then:

$$a_i^w(c) = \sum_{j \in M} \frac{{w_j}^2}{d(b_{i,j}, c)^2 + \sigma} \tag{5}$$

3.3 Trajectory Optimization

The current approach is computed on a frame by frame basis and thus ignores possible discontinuities in the tracking trajectory. As the correct trajectory likely does not have such discontinuities it is desirable to avoid them and to find a continuous trajectory. To this aim we define the energy function E_T for the whole trajectory T as an extension of the frame-based energy of Equation (5):

$$E_T = \sum_{i \in N} \overline{a_i^w}(c_i) + \beta p(c_{i-1}, c_i), c_i \in R_i := \{b_{i,1} ... b_{i,M}\}, T := \{c_1 ... c_N\} \tag{6}$$

where c_i is the fusion candidate box for a frame i on the trajectory candidate T. β weights the importance of continuity versus the best local score. Trajectory optimization cannot determine energies for boxes which do not belong to a tracking result. Thus for a valid trajectory T, the boxes $c_i \in T$ must be chosen from the set of tracking result boxes R_i. $\overline{a_i^w}$ is the normalized attraction that can, like a_i^w, be defined for single frames as:

$$\overline{a_i^w}(c) = \frac{a_i^w(c)}{\max\limits_{b_{i,j} \in R_i} a_i^w(b_{i,j})}, \ R_i := \{b_{i,1} ... b_{i,M}\} \tag{7}$$

The normalization ensures that the algorithm considers each frame with the same attention and does not favor simple frames with a concentration of attraction. If weights are not available $\overline{a_i^w}$ can also be replaced by the corresponding $\overline{a_i}$ by using a_i (Equation 3) instead of a_i^w. The function p is a function designed in order to penalize tracking result switches in a frame. It is 1 in frames where the trajectory T keeps following one tracking result T_j and close to 1 if the tracking result which T follows is changed but the trajectories of the old and new results are very close to each other in the corresponding frame. For discontinuities i.e. a leap to a distant trajectory it is close to zero. The function is defined as:

$$p(c_{i-1}, c_i) = \frac{\sigma}{d(c_i^\times, c_i)^2 + \sigma}, \ c_{i-1} = b_{i-1,j} \Leftrightarrow c_i^\times = b_{i,j} \tag{8}$$

$c_i^\times$ is the bounding box following c_{i-1} in the tracking result T_j where c_{i-1} is originating from, while c_i can belong to another tracking result on a tracking result switch in the frame. We do not use any motion model like e.g. a Kalman filter in p as we expect from the tracking algorithms themselves already to have motion models *i.e.* by choosing an algorithm we indirectly also choose a motion model. Instead we determine the cost of switching the tracking algorithm in a frame with our normalized distance d between the trajectories of two algorithms. To find the trajectory T^* that maximizes E_T within a reasonable time we use a dynamic programing based approach with $N \times M$ energy fields determined as:

$$E(0, j) = \overline{a_i^w}(b_{0,j}) \tag{9}$$

$$E(i, j) = \overline{a_i^w}(b_{i,j}) + \max_{j_2 \in M} p(b_{i-1,j_2}, b_{i,j}) + E(i-1, j_2) \tag{10}$$

Energy fields have to be calculated in increasing frame order starting with frame 0. All costs fields for a frame can be calculated in parallel. For online trajectory optimization T^* consists of the boxes with the highest energy in each frame. For the offline version the last frame is determined in the same way. The full offline trajectory of T^* can then be found by replacing "max" with "arg max" in Equation (10) to calculate $W(f, i)$. $W(f, i)$ can then be used as a lookup table for the way back on T^* starting from the last frame.

After finding the best trajectory, gradient ascent as described in Section 3.1 is performed here es well with Equation (5). We limit the maximal descent distance as we only want to use it for noise removal and not to destroy our found trajectory. The descent is limited to:

$$maxDescent = \delta \frac{b_w + b_h}{2} \tag{11}$$

with b_w and b_h being the width and height of a box $c_i \in T^*$ and $\delta = 0.05$.

3.4 Tracking Algorithm Removal

It might be advantageous to remove bad tracking results before fusion as they may disturb the fusion process more than they support it. In this section we present two removal approaches. A local one which removes different tracking algorithms for each sequence independently and a global one that removes tracking algorithms for all sequences at once.

The Local Approach: The idea behind the local approach is that there are only a few good tracking results for each sequence and we can identify them by looking at the average attraction a tracking result gets in a sequence. To this aim we calculate the performance P_j for each tracking result j in a sequence:

$$P_j = \sum_{i \in N} a_i^w(b_{i,j}) \tag{12}$$

Then we exclude the γ worst results with the smallest value P_j from the fusion. γ is a global parameter which is set equally for all sequences. $\gamma = |T| - 1$ is a special case which forces our approach to pick a single best tracking result for a sequence. Local removal can be used offline as well as online. In the online version all P_j must be recalculated every frame as N grows with each new frame.

The Global Approach: The global removal approach has the advantage that algorithms are removed permanently i.e. we do not need them for tracking anymore. To find candidates for global removal we first divide a training dataset into 10 parts of similar number of sequences and then perform experiments with all 10 permutations of 9 different parts, each. First we calculate for each experiment the success rate (see Section 4). Then we test for each experiment the removal of single tracking algorithms starting with the algorithm with the smallest w_j and proceeding in increasing order. If the success rate raises through removal of an algorithm, it will stay removed. Otherwise it will be added again. Algorithms that are removed in at least 7 experiments will be removed permanently. The reason why we do not simply use only the full training set to determine removal is that the removal procedure is extremely instable. For many algorithms the exchange of one or very few sequences already makes a big difference. Only if we perform several experiments we can identify algorithms that can be removed safely. Global removal is compatible with all online and offline fusion approaches, but requires in contrast to local removal labeled training data.

4 Evaluation Data and Methodology

To evaluate our fusion approach we use tracking results and ground truth data provided by Wu et al. [30]. They provide tracking results for 29 different online tracking algorithms on 51 different sequences with 32 different initializations on each sequence which are in total 47,328 tracking results. 20 initializations are used for "temporal robustness initialization" (TRE) of the tracking algorithms.

This means that the tracking algorithms are not only initialized in the first frame of each sequence but at 20 different temporal positions in each sequence. The first initialization of TRE starts from the first frame, which is the classical approach in object tracking. It is used for "one-pass evaluation" (OPE). The other 12 initializations they used for "spatial robustness evaluation" (SRE). This means that the initial tracking box is compared to the ground truth either scaled (scale factors: 0.8, 0.9, 1.1, 1.2), shifted (left, right, up and downwards shift) or shifted and scaled by the factor 1.1 at the same time. Nevertheless, SRE is evaluated with the unmodified ground truth. The authors argue that SRE is interesting because tracking initialization may be performed by an object detector that is not as accurate as a human initialization. In their work, Wu et al. [30] utilized OPE, TRE and SRE as independent datasets for a detailed evaluation of all 29 tracking algorithms. To take advantage of their whole data we will evaluate our fusion approaches also on these datasets. To our knowledge, there are only very few offline tracking algorithms in single object tracking. As a result, the best performing algorithms are usually almost all online. Therefore, we use these datasets created by online algorithms also to evaluate our offline approaches.

Evaluation Methodologies: For comparability we also use the same evaluation methodologies: a precision and a success plot. *Precision* measures the center location error in pixel space between the tracking box and the ground truth data. It is used already for a long time for evaluation in object tracking, but has some drawbacks. First of all, it does not normalize for object pixel size. A negligible center location error for a big object in a high resolution sequence can mean tracking failure for a small object. Secondly, the error can still increase if the object is already lost. Furthermore, it does not consider whether the object size is determined correctly. By using a precision plot which thresholds precision (it shows the percentage of frames that are below an error threshold) the problem of increasing error on object loss can be reduced but not completely solved. Anyhow, we think it is sufficient to compare location accuracy without scale accuracy for the provided dataset as the variability of object sizes stays within an acceptable limit. A measure that does not suffer from these problems is *success* which measures the overlap between a tracking and a ground truth box:

$$O(a,b) = \frac{|a \cap b|}{|a \cup b|} \tag{13}$$

The overlap is the intersection divided by the union of the two box regions. Overlap is normalized for object pixel size, penalizes if the size of a tracking box is different to the ground truth and the error will not increase if the object is lost. The successes plot measures the percentage of frames that have an overlap above a threshold. We call the area under curve (AUC) of the success plot *success score*. The success score is at the same time also the average overlap over all frames. To make sure that tracking result fusion is performed without influence from its ground truth data we determine the weights w_j and global algorithm removal in a cross validation manner, where we optimize for success score. The parameters α, β and σ we set to 4, 20 and 0.03, respectively, for all our tests. These parameters showed to be very robust i.e. the best values for them are very similar for OPE, TRE and SRE and our approach still shows a similar performance if we vary the values of these parameters in a relativity big range (supplemental material Fig.3). In our tests γ showed to be less robust and dataset dependent. To still prove its usefulness we optimize it for each dataset with cross validation. For SRE we use OPE to find γ, the weights w_j and to perform global algorithm removal. This simulates the effect of good training data but bad initializations from an object detector on real data. Cross validation is applied here as well.

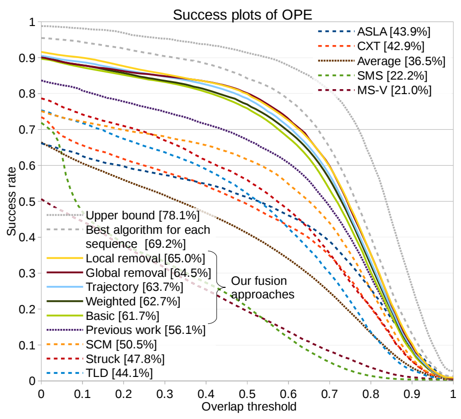
5 Results

In this section we present and discuss the evaluation results for our fusion approach. Further results and details can be found in our supplemental material. Success and precision plots can be seen in Figure 1 for OPE and Figure 2 for SRE and TRE. These plots contain different curves:

- The five best and two worst tracking algorithms according to [30].[2]
- The average of the curves of all 29 tracking algorithms.
- The fusion result of our previous work [4].
- Fusion results for our basic approach (Section 3.1), weighted approach (Section 3.2), trajectory optimization approach (Section 3.3) and for local and global removal (Section 3.4) based on our trajectory optimization approach for fusion. Due to space constraints we only show the offline versions of trajectory optimization and local removal as they provide slightly better results than the online versions and as they are sufficient for many practical applications.
- The "Best algorithm for each sequence" curve. It is determined by choosing always the best performing tracking algorithm for each sequence according to the success score. Note that this curve is not attainable without knowing the ground truth, and is only given as reference.
- The "Upper bound" curve. It is determined by taking for each frame in each sequence always the best tracking result box with the biggest overlap to the ground truth. Here again, this curve is not attainable without knowing the ground truth, and is only given as reference.

The numbers in brackets in the figures show, similar to [30], the value at location error threshold = 20 for the precision plot and the success score (AUC) for the success plot.[3] We show results of individual tracking algorithms as dashed lines in color and results of our fusion approaches as solid lines. Further lines are doted. Gray lines are not attainable without knowing the ground truth. Getting close to the "Upper bound" curve is very unrealistic for a fusion approach, as with a large amount of tracking results it is likely that there are results close to the ground truth only by chance. Nevertheless, it shows that there is at least a theoretical potential to get above the "Best algorithm for each sequence" curve.

[2] For plots of all 29 tracking algorithms we refer to the supplemental material of [30].
[3] Numbers differ slightly from [30] as we calculated them exactly by average overlap.



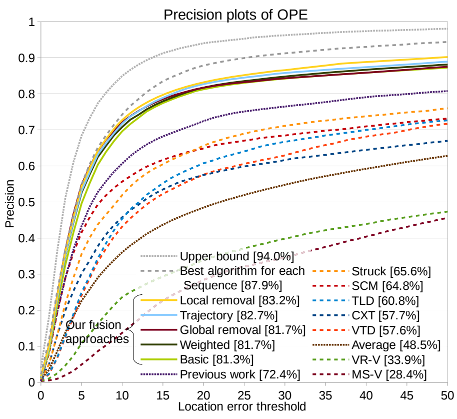


Fig. 1. Fusion results of OPE. Best viewed in color. See text for details.

Success and Precision Performance: As can be seen in Figure 1 and 2 our basic approach clearly outperforms the best tracking algorithm as well as our previous work [4] in all success and precision plots. Such good results are remarkable, given the fact that the fusion result is also influenced by many tracking algorithms that perform clearly worse than the best tracking algorithm. In every plot the weighted approach outperforms the basic approach and the trajectory optimization approach again outperforms the weighted approach. Global removal outperforms trajectory optimization in success score – for what it was optimized. However, in the success and precession plots the performance varies depending on the position. The performance for local removal differs for the three datasets.

OPE: On OPE local removal outperforms all other fusion approaches. The curve for local removal is even close to the "Best algorithm for each sequence" curve.[4] For a threshold lower than 5 pixels the curves in the precision plot are even almost the same. In the success plot which additionally considers scale correctness the approach does not get that close. This is likely because scale is often not correctly determined when position is determined correctly. We believe that this happens because many tracking algorithms are not able to determine the scale and thus they all vote for the scale of initialization.

SRE: On SRE local removal slightly underperforms the trajectory optimization approach. As reason we found that surprisingly the best γ for SRE is only 2, which is probably related to the poor initializations in SRE. However, as we take γ for SRE from OPE a γ of around 17 is used in cross validation.

TRE: On TRE tracking algorithms are mostly initialized in the middle or at the end of a sequence, which results in very short tracking results. Because of this, the difference between the best tracking algorithms and the "Average" tracking

[4] It is not outperformed because of a few sequences where fusion fails. See Table 1.

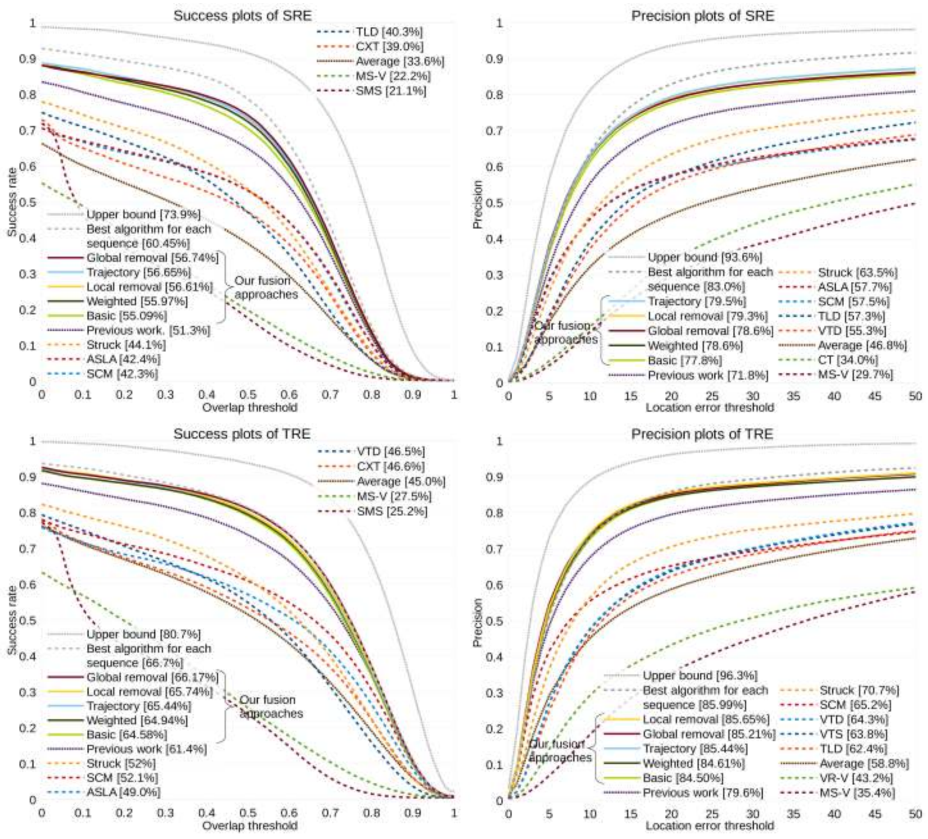
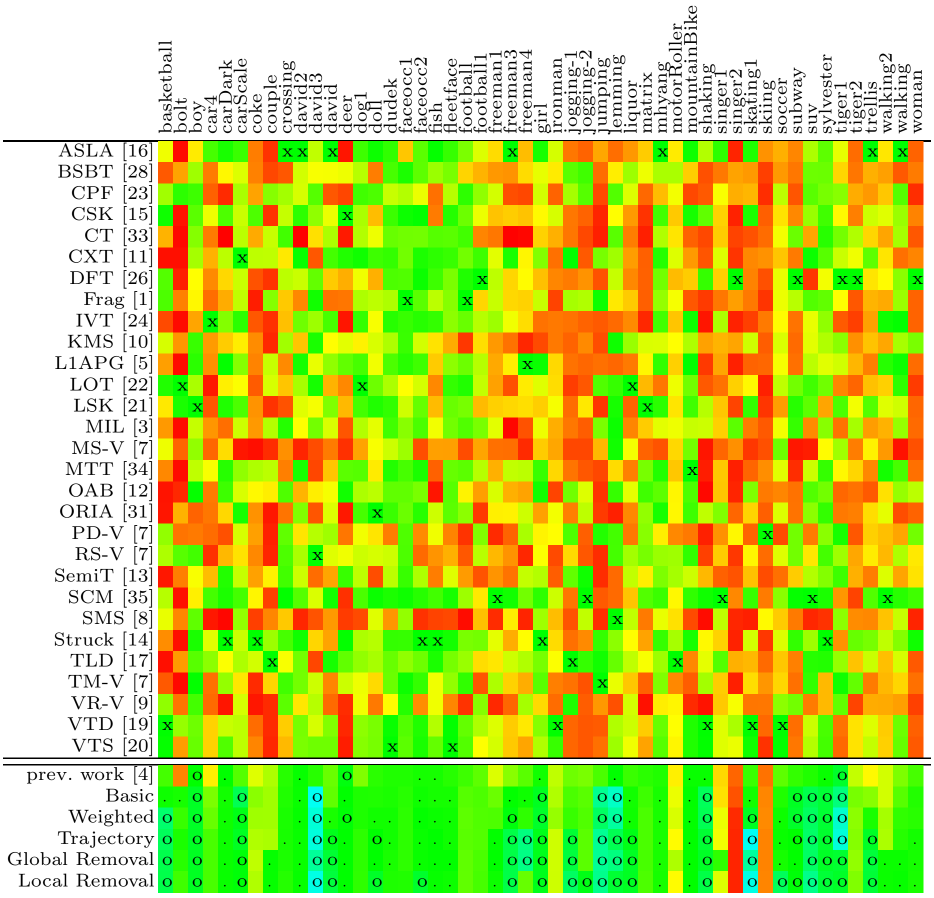


Fig. 2. Fusion results of SRE and TRE. Best viewed in color. See text for details.

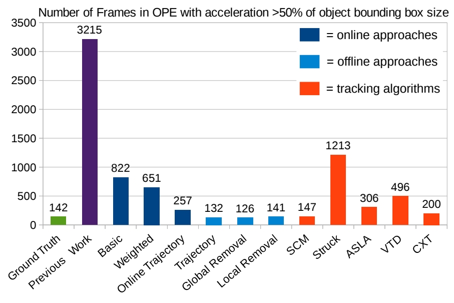
algorithm curve is clearly smaller than on OPE, as good algorithms can take less advantage from short results. Similarly, the gain for our weighted, trajectory optimization and local removal approach is smaller. On the other hand, our basic approach and global removal approach seem not to be negatively affected by the short sequence effect as there is an advantage similar to OPE. As a result, our approaches are even very close to the "Best algorithm for each sequence" curve and even outperform it at some locations. The best γ for TRE is 14.

Performance on Single Sequences: Table 1 shows the performance of our fusion approach on single sequences of OPE (SRE and TRE in supplemental material). Our previous work [4] outperforms the best tracking algorithm only in 3 sequences while already our basic approach outperforms the best tracking algorithm in 11 sequences. The weighted approach, trajectory approach and global and local removal approaches outperform the best algorithm even in 15, 20, 18 and 22 sequences, respectively. Our previous work [4] has at least 95%

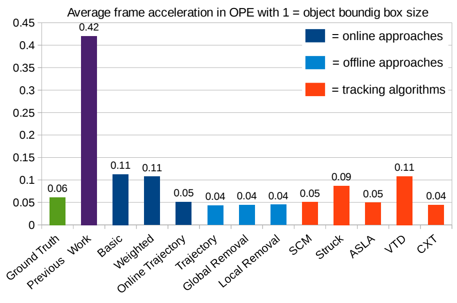
Table 1. Comparison of tracking results and fusion results for each sequence of OPE. The heatmap is normalized so that the best tracking result on a sequence is green. Red is the worst possible result. Cyan means that the fusion result is up to 12% (full cyan) better than the best tracking result. "x" marks the best tracking algorithm for a sequence, "o" fusion results that outperform the best algorithm and "." fusion results with at least 95% of the success score of the best algorithm. The heatmap is calculated by success score (see text for details). Tables for TRE and SRE can be found in our supplemental material. Best viewed in color.



of the success score of the best tracking algorithm in 12 sequences, our basic approach in 25, the weighted approach in 27, the trajectory approach in 33 and the global and local removal approaches in 35 and 34 sequences, respectively. This shows that our fusion approaches can provide for most sequences results very close or even better than the best tracking algorithm. Hereby, we can also clearly outperform our previous work [4] and our extended approaches clearly outperform our basic approach. Furthermore, our approaches often not only outperform the best tracking algorithm, but they are even up to 12% (18% SRE,



(a)



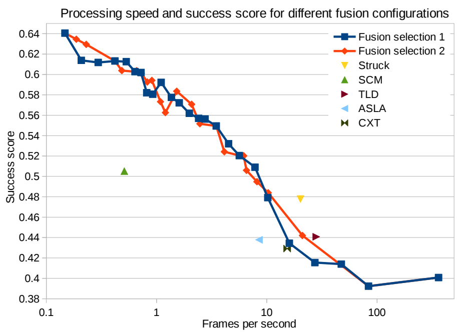
(b)

Fig. 3. Trajectory continuity evaluation on the OPE dataset. See text for details.

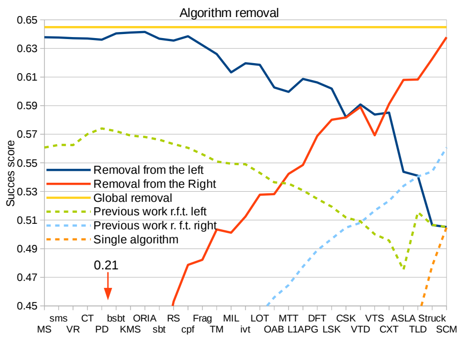
33% TRE) better in success score. However, there are also sequences like *skiing* and *singer2* where fusion performs poorly. The reason is probably that there are few algorithms that clearly outperform all other algorithms (*skiing*: 1 algorithm, *singer2*: 2 algorithms). Fusion cannot deal very well with such situations where very few algorithms outperform all others. The sequences where fusion performs poorly are the reason why our approach does not outperform the "Best algorithm for each sequence" curve in Figure 1.

Continuity and Smoothness of Trajectory: A good tracking trajectory should be continuous and smooth. Figure 3(a) show the number of frames on all sequences of OPE where there is a per frame acceleration greater than half of the size of the object bounding box. This happens even in the ground truth as there are a few sequences with extreme accelerations (in 0.48% of all frames). However, high accelerations that are not in the ground truth and thus not in the object trajectory can be considered as outliers or discontinuities. As expected, the frame based approaches show many discontinuities in their trajectories. Nevertheless, our basic and weighted approach perform much better than our previous work [4], but show still many discontinuities. Our trajectory optimization approach shows here its greatest strength. Thanks to its ability to consider past frames, online trajectory optimization performs much better than the other online approaches. For the offline version the numbers are even very close to the numbers of the ground truth. Our removal approaches which use offline trajectory optimization for fusion show similar results. The trajectories of some tracking algorithms like SCM and CXT also show only a few discontinuities. In contrast, Struck and VTD have by far too many discontinuities.

This shows that our trajectory optimization approach not only provides a trajectory continuity, which is similar to that of tracking algorithms, but even outperforms most of them in this regard. Figure 3(b) shows the average acceleration. Our trajectory optimization approaches (online and offline) have here even a smaller value than the ground truth. This shows that the trajectories of our trajectory optimization approaches are in general even smoother in velocity



(a)



(b)

Fig. 4. a): Processing speed, success score comparison for different tracking algorithms and fusion selections. b): The performance with removal of the worst or best tracking algorithms, compared to our global removal approach. All solid non dashed results are created with our trajectory optimization approach. See text for more details.

than the ground truth. This does not mean that they are better, but we think error by smoothness is preferable over many other error source.

Processing Speed: A possible drawback of a fusion approach could be its high runtime as it requires several tracking algorithms to run. As processing speed for tracking algorithms we take the average frame per second numbers from [30]. Figure 4(a) shows the processing speed of our approach with different subsets of algorithms. We construct the subsets by selecting algorithms in two different ways. Starting with one algorithm, we build the next subset by adding the next best algorithm to the set. The next best algorithm is the one with the greatest

$$\text{frames per second} \times \text{success score}^X \tag{14}$$

that is not yet in the set. We use $X = 2$ and $X = 4$ for the two selections, respectively (See supplemental material for more details). Concerning processing speed we cannot outperform fast and good tracking algorithms like TLD [17] and Struck [14] as there are only few faster algorithms in the dataset that mostly do not perform very well. Perhaps, with more very fast algorithms it might be possible. However, for a processing speed of ten frames per second or less fusion clearly outperforms every tracking algorithm.

Removal: Figure 4(b) shows the removal of the best (from right) or worst (from left) tracking algorithms. We perform this test with our trajectory optimization approach and with our previous work [4] for comparison. Although, the worst algorithms perform really poor, fusion only slightly suffers from them and still benefits from relatively bad algorithms like ORIA [31]. This interesting effect is true for both fusion approaches. However, our previous work needs a minimal number of tracking results to get a stable result probably because it uses majority voting. It performs worse than the best tracking algorithm SCM [35] when fusing the best 6 tracking algorithms. To avoid suffering from bad tracking results global removal can be used. It outperforms the peak of removal from the left. Removal from the right shows that the best algorithm SCM [35] (success score 0.505) can already be outperformed with the 15 worst algorithms that all have a performance below the average (best is IVT [24] with success score 0.358).

We also determined the probabilities that algorithms are removed by global removal. In doing so, we discovered that some algorithms like SMS, Frag and LOT are very removal resistant while others like CSK and VTS can be easily removed despite better average performance. We think easily removed algorithms cannot utilize their strengths in fusion as these are already widely covered by other algorithms, but their weaknesses still harm. On the other hand, removal resistant algorithms likely provide more original/unique strengths that are more useful for fusion. We think that the probabilities are not only interesting for evaluating the usefulness of tracking algorithms for fusion, but they are also an interesting way of estimating the originality/diversity of tracking algorithms. For the probabilities and a more detailed discussion see our supplemental material.

6 Conclusions

In this paper we presented a new tracking fusion approach that merges the results of an arbitrary number of tracking algorithms to produce a better tracking result. Our method is based on the notion of attraction, and the result that maximizes the attraction of all the trackers is chosen as global tracking result. We presented different variants of the method, including a weighted combination of trackers and an approach that favors continuous trajectories throughout the sequence. The latter method is solved using dynamic programming. In a complete evaluation we showed that our method clearly outperforms current state of the art tracking algorithms. On most tested sequences, our method even produces better results that the best algorithm for that specific sequence. We introduced further improvements using tracker removal techniques that remove tracking results before fusion either locally or globally. In addition we presented two new criteria for evaluating trackers. One measures originality/diversity in the behavior by utilizing global removal. The other one measures the continuity of the trajectory. We showed that our approach outperforms existing algorithms in continuity – most of them even with online fusion.

We think that the awareness that fusion of tracking algorithms actually improves the tracking performance will help to improve tracking methods in general. It shows that the combination of several good tracking ideas can clearly outperform single ideas if the methods are combined in the right way. In our future work we will investigate this property further to write generic object tracking algorithms that work well in general.

Acknowledgements. This work was partially funded by the Eurostars-Project VIDP under contract number E!5558 (FKZ 01AE1103B, BMBF 01QE1103B) and by the BMBF project DENSITY (01IW12001).

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 798–805. IEEE (2006)
2. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
3. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 983–990. IEEE (2009)
4. Bailer, C., Pagani, A., Stricker, D.: A user supported tracking framework for interactive video production. In: Proceedings of the 10th European Conference on Visual Media Production (CVMP). ACM (2013)
5. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1830–1837. IEEE (2012)
6. Chate, M., Amudha, S., Gohokar, V., et al.: Object detection and tracking in video sequences. Aceee International Journal on Signal & Image Processing 3(1) (2012)
7. Collins, R., Zhou, X., Teh, S.K.: An open source tracking testbed and evaluation web site. In: IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, pp. 17–24 (2005)
8. Collins, R.T.: Mean-shift blob tracking through scale space. In: Proceedings of 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, p. II-234. IEEE (2003)
9. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
10. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 564–577 (2003)
11. Dinh, T.B., Vo, N., Medioni, G.: Context tracker: Exploring supporters and distracters in unconstrained environments. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1177–1184. IEEE (2011)
12. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. BMVC 1, 6 (2006)
13. 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)
14. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 263–270. IEEE (2011)
15. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 702–715. Springer, Heidelberg (2012)
16. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829. IEEE (2012)
17. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 49–56. IEEE (2010)

18. Koller, D., Weber, J., Malik, J.: Robust multiple car tracking with occlusion reasoning. Springer (1994)
19. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1269–1276. IEEE (2010)
20. Kwon, J., Lee, K.M.: Tracking by sampling trackers. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1195–1202. IEEE (2011)
21. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1313–1320. IEEE (2011)
22. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1940–1947. IEEE (2012)
23. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
24. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. International Journal of Computer Vision 77(1-3), 125–141 (2008)
25. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 723–730. IEEE (2010)
26. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1910–1917. IEEE (2012)
27. Shu, G., Dehghan, A., Oreifej, O., Hand, E., Shah, M.: Part-based multiple-person tracking with partial occlusion handling. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1815–1821. IEEE (2012)
28. Stalder, S., Grabner, H., Van Gool, L.: Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1409–1416. IEEE (2009)
29. Wang, Q., Chen, F., Xu, W., Yang, M.H.: An experimental comparison of online object-tracking algorithms. In: SPIE Optical Engineering+ Applications, pp. 81381A–81381A. International Society for Optics and Photonics (2011)
30. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418. IEEE (2013)
31. Wu, Y., Shen, B., Ling, H.: Online robust image alignment via iterative convex optimization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1808–1814. IEEE (2012)
32. Yang, H., Shao, L., Zheng, F., Wang, L., Song, Z.: Recent advances and trends in visual tracking: A review. Neurocomputing 74(18), 3823–3831 (2011)
33. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 864–877. Springer, Heidelberg (2012)
34. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2042–2049 (2012)
35. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845. IEEE (2012)