16^{th} *Computer Vision Winter Workshop*
Andreas Wendel, Sabine Sternig, Martin Godec (eds.)
Mitterberg, Austria, February 2-4, 2011

Robustifying the Flock of Trackers

Jiří Matas and Tomáš Vojíř
The Center for Machine Perception, FEE CTU, Prague
Karlovo namesti 13, 121 35 Praha 2, Czech Republic
{matas, vojirtom}@cmp.felk.cvut.cz

Abstract. *The paper presents contributions to the design of the Flock of Trackers (FoT). The FoT trackers estimate the pose of the tracked object by robustly combining displacement estimates from local trackers that cover the object.*

The first contribution, called the Cell FoT, allows local trackers to drift to points good to track. The Cell FoT was compared with the Kalal et al. Grid FoT [4] and outperformed it on all sequences but one and for all local failure prediction methods.

As a second contribution, we introduce two new predictors of local tracker failure - the neighbourhood consistency predictor (Nh) and the Markov predictor (Mp) and show that the new predictors combined with the NCC predictor are more powerful than the Kalal et al. [4] predictor based on NCC and FB.

The resulting tracker equipped with the new predictors combined with the NCC predictor was compared with state-of-the-art tracking algorithms and surpassed them in terms of the number of sequences where a given tracking algorithm performed best.

1. Introduction

Tracking is an important task in computer vision. Given two consecutive frames and the position of an object in the first frame, the task is to estimate the pose of the object in the second frame. In a video sequence, tracking is the task of estimation the full trajectory of the object.

There are many approaches addressing the problem. This paper focuses on the so-called *Flock of Trackers* (FoT). The Flock of Trackers is a tracking approach where the object motion is estimated from the displacements, or, more generally, transformation estimates, of a number of independent local trackers covering the object.

Each local tracker is attached to a certain area specified in the object coordinate frame. The local trackers are not robust and assume that the tracked area is visible in all images and that it all undergoes a simple motion, e.g. translation. The Flock of Trackers object motion estimate is robust if it is obtained by a combination of local tracker motions which is insensitive to failures.

This idea was utilized in the Median-Flow (MF) [4] tracker and was shown to be comparable to state-of-the-art trackers. In [4], the FoT is based on local trackers placed on a regular grid, i.e. the local trackers cover the object uniformly. The object motion, which is assumed to be well modelled by translation and scaling, is estimated by the median of a subset of local tracker responses.

Theoretically, the median is robust up to 50% of outliers for translation and $100 \times (1 - \sqrt{0.5})\%$ for scale estimation which is based on pairs of correspondences. In practice, the outlier tolerance is often higher since the outlier do not "conspire" and are not all above or below the median. Nevertheless, in challenging tracking scenarios, the inlier percentage was not sufficient and the median failed.

In order to robustify the FoT, [4] proposed several local tracker filtering methods which perform the task of finding and removing probable outliers before the median estimation. The outlier subset of local trackers is selected by a failure-predicting procedure which takes into account the following quantities: the normalised cross-correlation of the corresponding patches (NCC), the sum of squared differences (SSD) and the consistency of the so called forward-backward procedure (FB). The forward-backward procedure runs the Lucas-Kanade tracker [5] twice, once in the forward direction and then in the reverse direction. The probability of being an oulier is a function of the distance of the starting point and the

point reached by the FB procedure.

The paper present two contributions to the design of the Flock of Trackers. First we show that superior tracking results are achieved if the failed local trackers are not reset to the grid position and they are allowed to track those areas of the object they drifted to. This process can be viewed as a new method for selecting "good points to track" on the object which can be described as "good points to track are those the tracker drifted to".

Second, we propose two new predictors of local tracker failure - the neighbourhood consistency predictor (Nh) and the Markov predictor (Mp). The Nh predictor is based on the idea that a correct local tracker will return a displacement similar to its neighbours. The Markov predictor exploits temporal consistency, a local tracker that performed well in the recent past is likely to perform well on the current frame and vice versa.

We show the new predictors combined with the NCC predictor are more powerful than the Kalal et al. [4] predictor based on NCC and FB. Moreover, the new predictors are efficiently computed, at a cost of about 10% of the complete FoT procedure whereas the forward-backward procedure slows down tracking approximately by a factor of two, since the most time consuming part of the process, the Lucas-Kanade local optimization, is run twice. With the proposed precise failure predictors, a FoT with much higher robustness to local tracker problems is achieved with negligible extra computational cost.

The rest of the paper is structured as follows. Section 2 discussed placement of the local trackers in FoT and present the cell placement strategy. Section 3 propose two new predictors of local tracker failure. Finally, Section 4 evaluate proposed improvements and compare resulting MF tracker with the state-of-the-art tracking algorithms and conclusion is given in Section 5.

2. Local tracker placement in FoT

The task of object tracking is usually decomposed into two steps. First, interesting points to track are found (e.g. "the good features to track" [8]). Next, the selected points are tracked. In the case of FoT, Kalal et al. [4] omit the first step and chooses the points to evenly cover the object of interest - the local trackers are laid out on a regular grid. It is clear that not all local trackers will be placed at location suitable for tracking. The poorly placed local trackers drift away from the original position on the grid. The Grid FoT (as proposed in [4]) resets, after estimating the global object motion, all local trackers to their original place in grid.

We argue that this is a suboptimal approach, because local trackers reinitialized to the same position unsuitable for tracking will drift again. Instead, we propose the Cell FoT, where each local tracker is allowed to "find" a suitable offset from its default position in the grid, see Fig. 1. The local trackers are forced to stay in their cells, and thus guaranteeing to evenly cover the tracked object, but within the cell the local trackers are let to assume the best position for tracking.

To avoid tracking points that are near each other, which could make the local trackers dependent and likely to fail simultaneously, the cells within which a local tracker must stay may not completely cover the object, as depicted in Fig.1. In preliminary experiments, we observed that the cell parameters c_w and c_h have a noticable influence on the results. However, to keep the method simple, in experiments reported in this paper they are set to the grid resolution, i.e. the local trackers are allowed to assume any position on the object (the cells cover the object).

The improvement of the FoT achieved by the Cell method is demonstrated in experiments in Section 4.1.

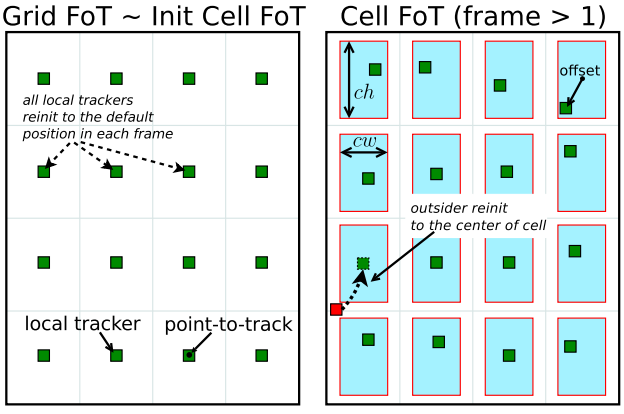


Figure 1: A comparison of the Grid and the Cell FoT. In the Grid FoT, in every frame, local trackers are placed on a regular grid. In the Cell FoT, after the first frame, the location the local tracker drifted to is tracked - the offset with respect to the grid positions is stored. Only local trackers that drifted away form their cells are reset.

3. New failure prediction methods

In this section, two novel methods for the local tracker failure prediction are presented together with a method that combines them with a predictor based on normalised cross-correlation and achieves very high accuracy at a very low computational cost. This new Median-Flow tracker with the new T_{Σ} combined predictor is evaluated in Section 4.2. The section is structured as follows: Section 3.1 describes the Neighbourhood constraint, Section 3.2 present an orthogonal predictor based on temporal local trackers behaviour.

3.1. Neighbourhood consistency constraint predictor

The idea of the neighbourhood constraint predictor Nh is that the motion of neighbouring local trackers is very similar, whereas failing predictors return a random displacement. The idea corresponds to the *smoothness assumption* which is commonly used in optic flow estimation, e.g. [7].

The Nh predictor was implemented as follows. For each point i, a neighbourhood N_i is defined. In all experiments, N_i was the four neighbourhood of i (three points are used on the edge, two in the corner of the grid). The neighbourhood consistency score S_i^{Nh}, i.e. number of the neighbourhood local trackers that have similar displacement, is calculated for each point i as follows:

$$S_i^{Nh} = \sum_{j \in N_i} [\| \Delta_j - \Delta_i \|^2 < \varepsilon]$$

$$\text{where} \quad [expression] = \begin{cases} 1 & \text{if } expression \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

and where ε is the displacement difference threshold, and Δ_i is the displacement of local tracker i. A local tracker is defined consistent if $S_i^{Nh} \geq \theta$. The value of θ was set to 1. The displacement difference threshold ε was set to 0.5 pixels. The process is visualised in Fig. 2).

3.2. The Markov predictor

The Markov predictor (Mp) exploits a simple model of the past performance of a local tracker. The model is in the form of a Markov chain (Fig. 3) with two states, state = {*inlier=1*, *outlier=0*}.

The predicted state of the local tracker depends on its state in the previous time instance and the transition probabilities. Transition probabilities are computed incrementally, from frame to frame. Each local

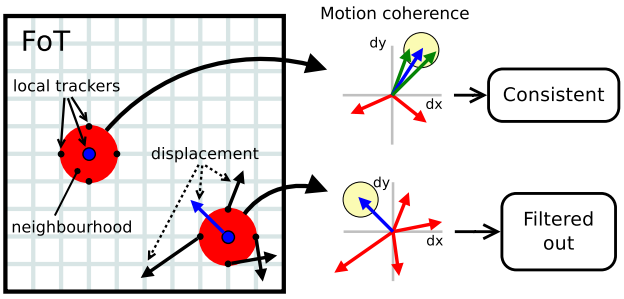


Figure 2: The neighbourhood consistency tracking failure predictor. The neighbourhood (red area) of the local tracker (blue circle) is explored for motion coherence. Blue arrow - center local tracker displacement, green arrows - local trackers with coherent motion, red arrows - local tracker with incoherent motion.

tracker i in time t is modeled as transition matrix $\mathbf{T}_t^i$ described in eq. 2.

$$\mathbf{T}_t^i = \begin{bmatrix} p^i(s_{t+1}=1 \mid s_t=1) & p^i(s_{t+1}=1 \mid s_t=0) \\ p^i(s_{t+1}=0 \mid s_t=1) & p^i(s_{t+1}=0 \mid s_t=0) \end{bmatrix} \tag{2}$$

where s_t is the current state of the local tracker and sums in columns are equal to 1. Prediction that certain local tracker would be inlier (or outlier) is dual task to next state prediction in Markov chain. To predict next state in time t+1 we compute probability of crossing to state 1 with apriori of current state. This is done according to equation 3.

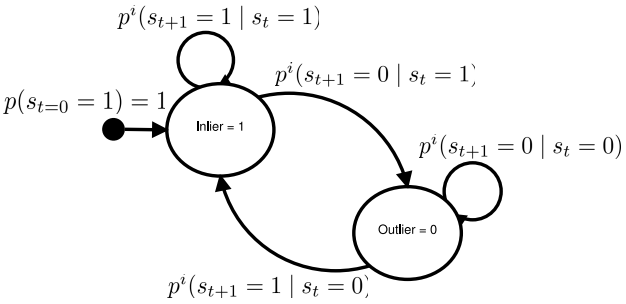


Figure 3: The state diagram of the Markov chain for the local tracker in the generalized form of two states probabilistic automaton with transition probabilities p^i, where i identifies the local tracker.

$$\begin{bmatrix} p^i(s_{t+1}=1) \\ p^i(s_{t+1}=0) \end{bmatrix} = \mathbf{T}_t^i \times \begin{bmatrix} p^i(s_t=1) \\ p^i(s_t=0) \end{bmatrix} \tag{3}$$

where $p(s_t=1)=1$ if current state is inlier, 0 otherwise (likewise for $p(s_t=0)$). The left side of equation 3 are then probabilities that next state would be

inlier (outlier) (e.g. if $p(s_{t+1} = 1) = 0.6$, then we considered local tracker as inlier in 60% of cases).

Model update is equal task to estimation of transition probabilities $p^i(s_{t+i} = 1 \mid s_t = 1)$ and $p^i(s_{t+i} = 1 \mid s_t = 0)$. These probabilities are updated in each frame as follow :

$$\begin{aligned} p^i(s_{t+1} = 1 \mid s_t = 1) &= \frac{n^i_{11}}{n^i_1} \\ p^i(s_{t+1} = 1 \mid s_t = 0) &= \frac{n^i_{01}}{n^i_0} \end{aligned} \tag{4}$$

where n_1 (n_0) are relative frequency for the local tracker *i* being inlier (outlier), and n_{11} (n_{01}) are relative frequency for event that the local tracker *i* was inlier (outlier) in the time *t* and inlier in the time *t*+1, for $t \in \langle 0, t \rangle$. The current state of the local tracker being inlier (outlier) is obtained by fitting the local tracker correspondence to estimated global motion (by MF) and thresholding their errors.

4. Performance evaluation

The performance of the proposed FoT was tested on challenging video sequences with object occlusion (or disappearance), illumination changes, fast motion, different object sizes and object appearance variance. The sequences are described in Tab. 1.

In the experiments, the predictor of neighbourhood consistency (Nh) and the Markov predictor (Mp) were run as explained in Section 3. The sum of squared differences (SSD), normalized cross-correlation (NCC) and the forward-backward predictor (FB) rank local trackers by their score and treat the top 50% as inliers. Predictors are denoted by the names of their error measure, except for the combination Mp+NCC+Nh which is abbreviated to Σ.

A frame is considered correctly tracked if the overlap with ground truth is greater than 0.5, with the exception of experiment 4.4 where influence of the initialization of the tracker was assessed. Since in this case the bounding boxes are randomly generated and may not fully overlap the object, the threshold was lower to 0.3, see Fig. 6.

4.1. Cell FoT-MF vs. Grid FoT-MF

Two version of the FoT that differ by local tracker placement — the Cell FoT-MF and the Grid FoT-MF — were compared on sequences presented in Tab. 1.

The tests were carried out for all individual predictors and for most combination of local tracker failure predictors; some combinations, e.g. SSD and NCC, make no sense since the predictions are highly correlated.

The performance was first measured by the length of correctly tracked sub-sequences (Tab. 2) and, on a single sequence, by the overlap of the estimated object pose and the ground truth bounding box (Fig. 4).

According to both criteria, the Cell FoT outperform Grid FoT for almost all sequences and failure prediction methods. We therefore conclude that the Cell FoT is superior to the Grid FoT.

Finally, the Grid and the Cell trackers were compared by the fraction of local trackers that are inliers to the global object motion, see Tab. 3. Again, the Cell method dominates. The interpretation of this result is not straightforward since the median and mean inlier rates are calculated on the whole correctly tracked sub-sequences, which are different for different method.

4.2. Comparison of failure prediction methods

We compared performance of individual predictors and combinations FB+NCC (as proposed in [4]) and Σ within the MF tracker on sequences presented in Tab. 1. Performance was measured in terms the length of the subsequence that was correctly tracked and by the number of sequences where a given tracker failed last (Tab. 2 last row). All parameters for Nh was fixed for all sequences, as described in Section 3.1. The proposed local tracker failure predictor Σ outperform FB+NCC on all tested sequences.

4.3. Comparison of T_{FB+NCC} and T_Σ speed

The MF tracker is intended for real-time performance and thus the speed of local tracker predictor is important. The experiment was performed on a subset of sequences listed in Tab. 1 (where the trackers successfully tracked the whole sequence) and then the results were averaged. Speed was measured as the average time needed for frame-to-frame tracking (Tab. 4). Processing time for I/O operations, including image loading, and other tasks not relevant to tracking, was excluded. The MF with Σ predictor performs 58% faster than FB+NCC. Moreover, the Σ overhead is negligible compared to reference MF tracker (i.e. MF without any predictor).

4.4. Robustness to bounding box initialization

For a tracker, it is highly desirable not to be sensitive to the initial pose specified by the object bound-

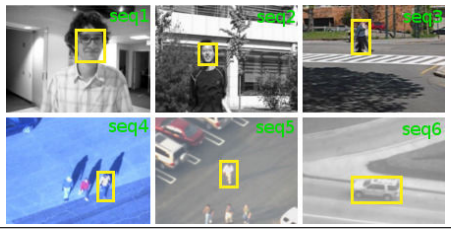
Sequence	name	frames	First appeared in	preview
1	David	761	D.Ross et al., IJCV'08 [6]	
2	Jumping	313	Q. Yu et al., ECCV'08 [9]	
3	Pedestrian 1	140	S. Avidan, PAMI'07 [1]	
4	Pedestrian 2	338	Q. Yu et al., ECCV'08 [9]	
5	Pedestrian 3	184	Q. Yu et al., ECCV'08 [9]	
6	Car	945	Q. Yu et al., ECCV'08 [9]	

Table 1: The description of test sequences and sample images with the selected object of interest.

Sequence	$T^g_\emptyset/T^c_\emptyset$	$T^g_{\text{SSD}}/T^c_{\text{SSD}}$	$T^g_{\text{NCC}}/T^c_{\text{NCC}}$	$T^g_{\text{FB}}/T^c_{\text{FB}}$	$T^g_{\text{Nh}}/T^c_{\text{Nh}}$	$T^g_{\text{Mp}}/T^c_{\text{Mp}}$	T^g_Σ/T^c_Σ	$T^g_{\text{FB+NCC}}/T^c_{\text{FB+NCC}}$
1	296/476	63/529	133/479	**761/761**	**761/761**	597/700	453/**761**	**761/761**
2	36/36	78/**79**	49/56	45/76	33/34	36/36	76/76	36/36
3	14/12	20/26	29/33	34/38	15/27	28/27	125/**140**	45/49
4	90/90	33/33	90/90	**264**/90	90/90	90/90	153/**264**	90/90
5	**52/52**	**52/52**	**52/52**	**52/52**	**52/52**	**52/52**	**52/52**	**52/52**
6	389/345	290/374	**510/510**	**510/510**	**510/510**	**510/510**	**510/510**	**510/510**
best	1/1	1/2	2/2	4/3	3/3	2/2	2/5	3/3

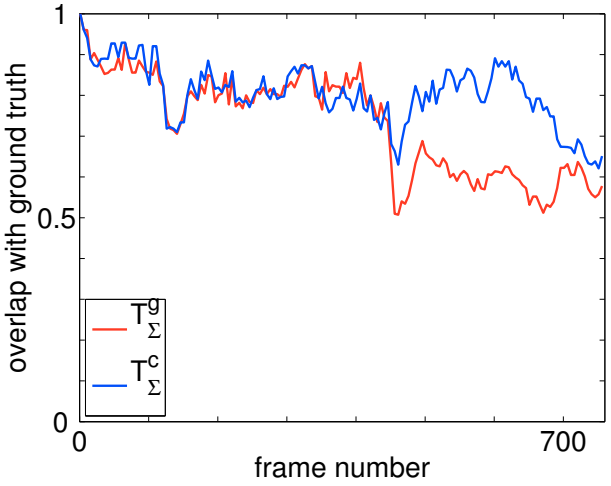
Table 2: A comparison of the Grid (T^g) and the Cell (T^c) FoT-MF with different local tracker failure predictors in term of the length of the successfully tracked subsequences. Best results for each sequence are in bold. Row *best* shows the number of sequences where the tracking method perform best.The total number of a frames in the sequences are listed in Tab. 1 The T^c_Σ clearly dominates.

Sequence	$T^g_\emptyset/T^c_\emptyset$	$T^g_{\text{SSD}}/T^c_{\text{SSD}}$	$T^g_{\text{NCC}}/T^c_{\text{NCC}}$	$T^g_{\text{FB}}/T^c_{\text{FB}}$
1	0.41(0.45)/**0.51(0.55)**	0.12(0.13)/**0.52(0.55)**	0.26(0.25)/**0.50(0.53)**	0.57(0.61)/**0.60(0.64)**
2	0.64(0.76)/**0.66(0.77)**	0.47(0.45)/0.47(0.45)	**0.55(0.63)**/0.52(0.55)	**0.60(0.70)**/0.50(0.54)
3	0.33(0.33)/**0.35**(0.33)	**0.40(0.43)**/0.39(0.40)	0.41(0.40)/**0.43(0.43)**	0.42(0.40)/**0.43**(0.40)
4	0.52(0.53)/**0.53(0.54)**	0.47(0.48)/**0.54(0.56)**	0.52(0.51)/**0.53(0.52)**	0.52(0.53)/0.53(0.52)
5	0.68(0.71)/0.68(**0.72**)	0.70(**0.74**)/0.70(0.73)	0.67(0.69)/**0.68(0.73)**	0.68(0.71)/0.68(**0.73**)
6	0.74(0.81)/**0.75**(0.81)	0.74(0.81)/0.74(0.81)	0.74(0.81)/**0.75(0.82)**	0.74(0.81)/0.75(0.82)
# better	0/**6**	2/2	1/**5**	1/**4**
Sequence	$T^g_{\text{Nh}}/T^c_{\text{Nh}}$	$T^g_{\text{Mp}}/T^c_{\text{Mp}}$	T^g_Σ/T^c_Σ	$T^g_{\text{FB+NCC}}/T^c_{\text{FB+NCC}}$
1	0.58(0.62)/**0.61(0.66)**	0.47(0.50)/**0.53(0.57)**	0.56(0.60)/**0.59(0.63)**	0.54(0.57)/**0.56(0.59)**
2	**0.68**(0.78)/0.65(0.78)	0.65(0.76)/**0.66**(0.76)	0.48(**0.49**)/0.48(0.44)	**0.65(0.79)**/0.64(0.76)
3	0.40(0.40)/0.41(0.39)	0.40(**0.39**)/0.40(0.38)	0.34(0.33)/**0.35(0.35)**	0.43(0.43)/0.43(0.43)
4	0.53(0.52)/**0.54(0.55)**	0.52(0.51)/0.52(**0.52**)	0.51(0.51)/**0.52**(0.51)	0.50(0.49)/**0.51**(0.49)
5	0.68(0.72)/0.68(0.72)	0.68(0.72)/0.68(0.72)	0.67(0.71)/**0.68**(0.71)	0.67(0.70)/0.67(0.70)
6	0.74(0.81)/**0.75**(0.81)	0.74(0.81)/**0.75(0.82)**	0.74(0.80)/**0.75(0.82)**	0.74(0.81)/**0.75(0.83)**
# better	1/3	1/4	1/5	1/3

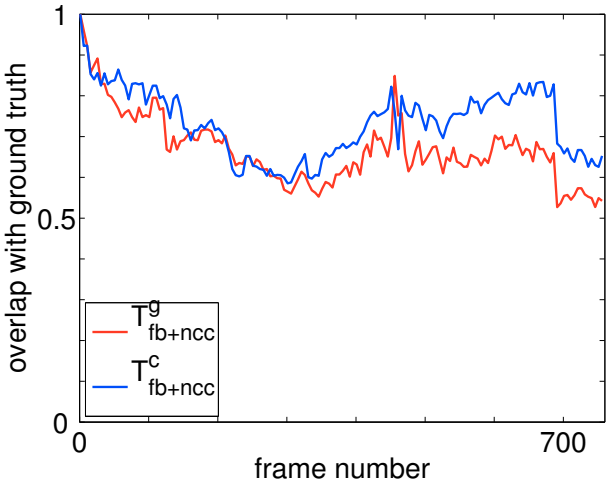
Table 3: A comparison of the Grid (T^g) and the Cell (T^c) FoT-MF in terms of inliers rates, i.e. the fraction of local trackers consistent with the estimated global object motion. Entries are in the following format: *mean*(*median*) of inliers rates for the correctly tracked sub-sequences. Row *# better* shows the number of sequences where the grid/cell methods perform better then the other one.

ing box as it is often selected manually, with unknown precision.

If part of the bounding box does not cover the object, the Mp predictor should soon discover that the local trackers are consistently in the outlier set. The Mp predictor can be used to define the object more precisely, e.g. as the set of cells that are likely to be inliers, according to Mp. Also, with Mp, there



(a)



(b)

Figure 4: A comparison of the Grid and the Cell trackers with (a) T_Σ (b) $T_{\text{FB+NCC}}$ in terms of the overlap with ground truth as a function of time for the sequence 1 (David).

Method	f[Hz]	T [ms]
$T_\emptyset \approx T_{\text{SSD}}$ [ref]	227	4.41
$T_{\text{FB+NCC}}$	131	7.63
T_Σ	207	4.83

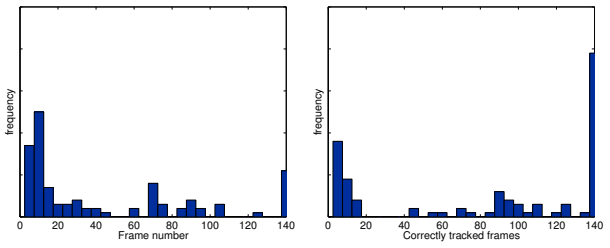
Table 4: A comparison of the speed of tracking failure prediction methods for the MF tracker.

is hope that the global tracker will be insensitive to initialization.

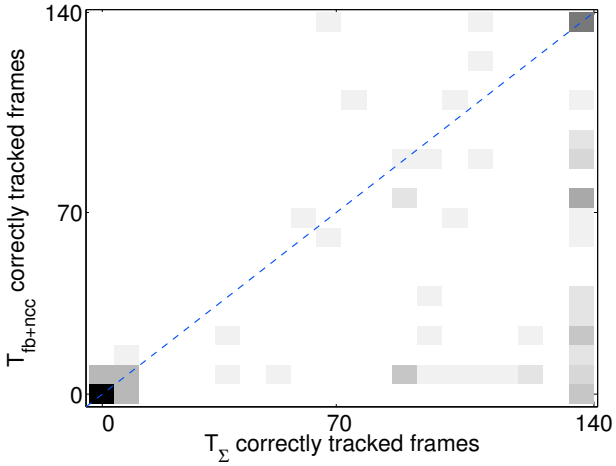
This experiment tested this assumption on the challenging sequence 3 (Pedestrian 1). We generated randomly 100 initial bounding boxes overlapping the object of interest (Fig. 6) and count the correctly tracked frames (Tab. 5). In this experiment, the frame was declared correctly tracked if overlap with ground truth was greater than 0.3. The T_Σ tracker perform about 90% better than $T_{\text{FB+NCC}}$ and was able to track the object correctly up to frame 85 in average. Figs. 5a and 5b show the histograms of the number of correctly tracked frames and fig. 5c show 2D histogram of the corresponding numbers of correctly tracked frames.

	Score	mean (median)
$T_{\text{FB+NCC}}$ [ref]	4493	45 (21)
T_Σ	8438	84.4 (99.5)

Table 5: Evaluation of filtering methods in terms of the number of correctly tracked frames with randomly initialized bounding box (see. Fig. 6). Score is the total number of correctly tracked frames, the mean and the median of the same quantity is presented in the right column.



(a) (b)



(c)

Figure 5: Histograms of the number of correctly tracked frames for (a) $T_{\text{FB+NCC}}$, (b) T_Σ and the (c) 2D histogram of the corresponding numbers of correctly tracked frames for different bounding box initializations.

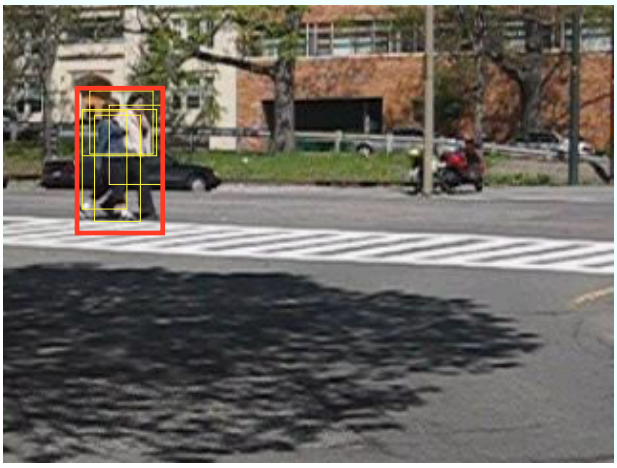


Figure 6: Exmaples of randomly generated initial bounding boxes (yellow) which were randomly generated within the red rectangle.

4.5. Comparison with state-of-the-art approaches

The proposed method was compared with the state-of-the-art algorithms. Results of the experiment are presented in Table 6. The key results is presented in the bottom (denoted *best*). The number is defined as number of sequences where the given tracking algorithm perform best. Results for algorithms [6, 3, 1, 2] were obtained from [4]. The experiment follows [4] - each frame was considered tracked correctly if overlap with ground truth was bigger than 0.5. Object initialization was done by the ground truth. The proposed Σ predictor with MF tracker outperform state-of-the-art algorithms and proof to be superior in speed and robustness to FB+NCC (as was shown in sections 4.3, 4.4), which perform similarly.

sequence	[6]	[3]	[1]	[2]	[4]	T_Σ
1	17	n/a	94	135	**761**	**761**
2	75	**313**	44	**313**	170	76
3	11	6	22	101	**140**	**140**
4	33	8	118	37	97	**264**
5	50	5	**53**	49	52	52
6	163	n/a	10	45	**510**	**510**
best	0	1	1	1	3	**4**

Table 6: A comparison of the proposed MF T_Σ tracker with recently published tracking methods.

5. Conclusions

This paper presented an improvement of the Flock of Trackers, the so called Cell FoT that allows local trackers to drift to points good to track. The Cell FoT was compared with the Grid FoT [4] and outperformed it on all sequences but one and for all local failure prediction methods.

As a second contribution, two new local tracker failure predictors were introduced - the neighbourhood consistency predictor and the Markov predictor. Together with the NCC predictor, they formed a very strong predictor Σ. The Σ predictor was compared with the FB+NCC predictor in the framework of the Median Flow. The Σ predictor outperformed the FB+NCC in all criteria, ie. in terms of speed, the length of correctly tracked sequences and the robustness to bounding box initialization.

The MF tracker equipped with Σ predictor was compared with state-of-the-art tracking algorithms and surpassed them in terms of the number of sequences where a given tracking algorithm performed best.

Acknowledgements

The authors were supported by Czech Science Foundation Project P103/10/1585 and by Czech Government reseach program MSM6840770038

References

[1] S. Avidan. Ensemble tracking. *PAMI*, 29(2):261–271, 2007. 5, 7

[2] B. Babenko, M.-H. Yang, and S. Belongie. Visual Tracking with Online Multiple Instance Learning. In *CVPR*, 2009. 7

[3] R. Collins, Y. Liu, and M. Leordeanu. On-line selection of discriminative tracking features. *PAMI*, 27(1):1631 – 1643, October 2005. 7

[4] Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-Backward Error: Automatic Detection of Tracking Failures. *ICPR*, 2010. 1, 2, 4, 7

[5] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *IJCAI*, pages 674–679, April 1981. 1

[6] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008. 5, 7

[7] P. Sand and S. J. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80(1):72–91, 2008. 3

[8] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593 – 600, 1994. 2

[9] Q. Yu, T. B. Dinh, and G. G. Medioni. Online tracking and reacquisition using co-trained generative and discriminative trackers. In *ECCV*, pages 678–691, 2008. 5