This paper was published at the ACCV Workshop on Detection and Tracking in Challenging Environments, Daejeon, Korea, on 6 November 2012.

Tracking the Untrackable: How to Track When Your Object Is Featureless

Karel Lebeda[1], Jiri Matas[1], Richard Bowden[2]

[1]Center for Machine Perception, Czech Technical University in Prague
[2]Centre for Vision, Speech and Signal Processing, University of Surrey

Abstract. We propose a novel approach to tracking objects by low-level line correspondences. In our implementation we show that this approach is usable even when tracking objects with lack of texture, exploiting situations, when feature-based trackers fails due to the aperture problem. Furthermore, we suggest an approach to failure detection and recovery to maintain long-term stability. This is achieved by remembering configurations which lead to good pose estimations and using them later for tracking corrections.

We carried out experiments on several sequences of different types. The proposed tracker proves itself as competitive or superior to state-of-the-art trackers in both standard and low-textured scenes.

1 Introduction

We present an approach to robustly track objects when they have limited or no visual features (such as distinctive texture). This is difficult as without consistent features many common assumptions used in tracking fail. We overcome this by using a novel formulation based on low level line correspondences which can operate with or without texture while avoiding the aperture problem.

Visual tracking is an active part of computer vision with a number of new approaches in recent years. The basic objective of tracking is, given a sequence of consecutive frames and the annotated pose of the object of interest in the first frame, to estimate the pose of this object in the rest of frames. Current techniques aim to learn the appearance of the tracked object [1, 2] or build a global model joining local trackers [3–5] to a robust frame.

Kalal *et al.*[1] proposed a method for on-line learning from positive and negative examples for tracking and detection (TLD, *track-learn-detect*). While positive samples arise from successful tracking, negative samples are found by contradictions. Other authors improve trackers by globally modelling the target. Matas and Vojir [4] joined local trackers (LK trackers [6]) to a *flock* and let each tracker converge to a feature good to track. Furthermore, they introduced new predictors of local tracker failure to cope with outliers. Similarly, Cehovin *et al.*[3] proposed a *coupled-layer visual model* in their LGT tracker, consisting of a local and global layer. While the local layer describes the target's local visual properties, the global layer encodes the target's global colour, motion and shape in a probabilistic manner. Dupac and Matas [5] used *zero shift points* for tracking – points with approximately even intensity function in their neighbourhood in all the directions. These points are tracked following the *shift field* and are connected hierarchically depending on the size of the neighbourhood.

These techniques works well when sufficient texture of the target object exists, which implies presence of features, which are good to track, e.g. blobs [7], Harris corners [8], or mentioned zero-shift points [5]. Unfortunately, real world scenes often contain objects without sufficient numbers of such features, or these lie on the boundary where the background affects their location and appearance. Without these features, even sophisticated methods like LGT or TLD often fail (see experimental evaluation). Conventional trackers often avoid edges because of the so called *aperture problem*, causing the edge points to be well defined only in one direction (perpendicular to the edge). However, with knowledge of this direction, a line correspondence can be established.

The edge features have been used in a number of previous articles, to solve a problem of tracking an object modelled by either 3D wireframe [9, 10] or by set of 2D edges [11, 12]. However, all of these approaches are based on the fitting of the *user-supplied* model to the image data. We are, on the contrary, trying to learn the object model online from the data, thus one of the challenges is to establish what features can be used to consistently track the object when no a priori information of the appearance is given.

Figure 1 illustrates the aperture problem when tracking a part of contour X_1 in frame f_1 to X_2 in f_2. True correspondences $\{a_2^*, b_2^*\}$ of points $\{a_1, b_1\}$ cannot be found directly as they are not uniquely defined in both parallel (to the edge) and perpendicular directions. When searching perpendicular to the edge, we find incorrect correspondences $\{a_2, b_2\}$ instead. If we assume a small movement between two consecutive frames, then we can expect these points to generate the same corresponding lines $\{k_2, l_2\}$ as true corresponding points. A point correspondence from the intersections of the lines (c_1, c_2) gives the true transformation regardless of the shift along the edge. The transformation

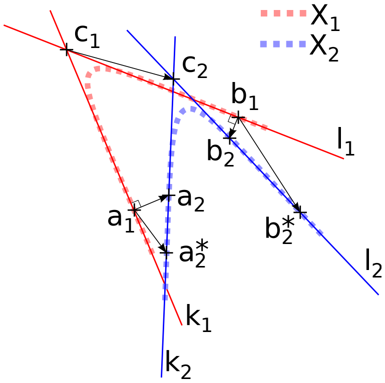

Fig. 1: Establishing edge correspondences (see text for detailed description).

between consecutive frames can be estimated directly from line correspondences, or from point correspondences of their intersections. Our approach is based on this principle.

The structure of this paper is as follows. We describe our tracking algorithm in section 2. In section 3 we address a question of points and lines and correspondences between frames; these correspondences are then used to estimate the inter-frame transformation in section 4. A long-term stability is addressed in section 5. The performance of our algorithm is experimentally evaluated in section 6. Finally, section 7 draws conclusions.

2 Algorithm Overview

The main objective of a tracking algorithm is to find the position of an object of interest in every frame of a video sequence. In other words, to find a transformation T_t from *model space* to the tracked area in every frame f_t. We estimate this transformation by transformation S_t of tracked areas of two consecutive frames f_{t-1} and f_t ($\mathrm{S}_t = \mathrm{T}_t \circ \mathrm{T}_{t-1}^{-1}$). T_1 is supplied by the user in the form of an initial area annotated in the image space. In this work we restrict S_t to a *similarity transformation*.

When the frame f_1 is processed, the initial set of N_1 edge points $\{p_1^{[i]} | i \in \{1, ..., N_1\}\})$ is generated in the model space and transformed to the image space by the user-supplied T_1. Then lines $l_1^{[i]}$ defined by points are computed.

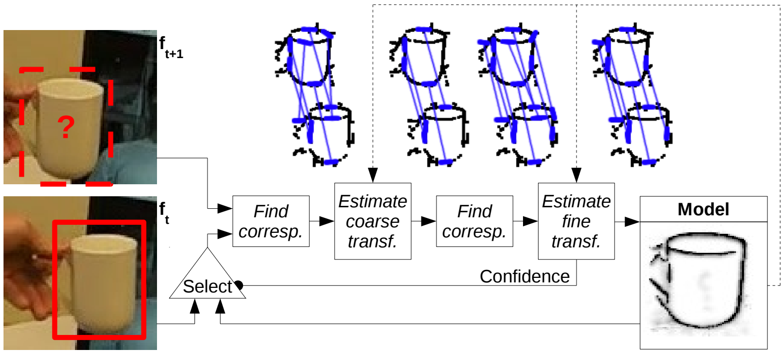

Fig. 2: Overview of the tracking algorithm.

An overview of the iterative tracking procedure is outlined in Figure 2. Firstly, given two consecutive frames f_{t-1} and f_t, initial correspondences $(p_{t-1}^{[i]}; q_t^{[i]})$ are found. There are several ways to do this, we use a *guided edge search* (section 3.1) for coarse correspondences. A smooth movement is assumed and therefore points from the previous frame, moved by the transformation of the previous step $(\mathrm{S}_{t-1}(p_{t-1}^{[i]}))$, can be used as an initial estimate of the new point locations. These correspondences are then used as input to a modified LO-RANSAC (section 4, [13, 14]) to find a coarse estimate of the transformation (S'_t and thus T'_t).

Using the coarse transformation estimate S'_t, new correspondences are computed by moving the points $p^{[i]}_{t-1}$ and employing an *unguided edge search* (see section 3.1). LO-RANSAC is then repeated with these correspondences to refine the transformation to S_t. S_t is usually more precise than S'_t and its number of inliers (which will be retained for the future computations) is higher.

As a measurement of the quality of the estimation, we introduce an *evidence score* E_t of the transformation S_t. E_t measures the fitness of points $p^{[i]}_{t-1}$ to the image of f_t and allows drift or tracking failures to be detected. In such a situation, the algorithm tries to recover and correct its pose.

We define a set of *inliers* of the resulting transformation as a subset of correspondences having an error smaller than or equal to a predefined *error threshold*:

$$\mathcal{I}_t = \left\{ \left(p^{[i]}_{t-1}; q^{[i]}_t\right) \,\middle|\, d\left(p^{[i]}_{t-1}, q^{[i]}_t | \mathrm{S}_t, \mathrm{f}_{t-1}, \mathrm{f}_t\right) \leq \theta;\ i \in \{1, ..., N_{t-1}\} \right\} \ , \qquad (1)$$

where d is a geometric error of corresponding lines defined by $p^{[i]}_{t-1}$ in frame f_{t-1} and $q^{[i]}_t$ in f_t. The points $q^{[i]}_t$ of inliers are retained for the next frame. To ensure a stable number of points we add a set of newly generated points to them. The new points are not cropped strictly to the tracked area and thus allow the model to grow slightly outside the original area.

3 Obtaining and Use of Correspondences

3.1 Search for Edge Correspondence

Unguided. Searching for the nearest strong edge is carried out in the direction of the gradient of image intensity [9]. Candidates for matching edge points are rated according to their magnitude of gradient and distance from the initial position by applying a Gaussian weighting. This process is iteratively repeated to convergence.

Guided. The guided searching of edges works in a different manner as we are not looking for a *strong edge* but for a *similar edge*. Instead of searching only in the direction of gradient, searches are also performed at angles shifted by $\frac{\pi}{20}$ and $\frac{\pi}{10}$ to both sides. The local gradient maxima are extracted and their similarities to the original edge are compared in terms of *change of gradient angle*, *change of appearance* and *spatial proximity*. This process has no iterations, correspondences are found in a single step.

3.2 Creation of Lines

Every line $l^{[i]}_t$ is computed from its defining point $p^{[i]}_t$ and orientation $\alpha^{[i]}_t$ (an angle of the image gradient). As angles of the normal vectors of lines are used in oriented evidence measurement (E_t, see section 4.1) and it is essential to distinguish angles with opposite orientation, normal vectors of lines must have angles in accordance with the image intensity gradient. Thus lines are “oriented”.

3.3 Geometric Error of Line Correspondences

An aim of the algorithm is to find the "best" transformation S_t between two consecutive frames. The "best" usually means the one, which minimises some (robust) function of distance between projected and measured correspondences. But what does it mean for two lines l and l' to be close to each other?

Hartley [15] stated that distance (or geometric error) of lines has to be measured with respect to some point of interest. He suggested to use the distance between a line and line segment. This approach yields usable results. However, in our case it is necessary to calculate intersections between the lines and all four sides of the tracked quadrilateral and computational complexity is prohibitively large.

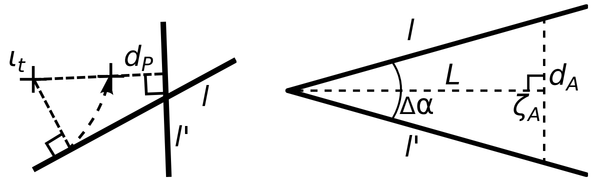

Fig. 3: Geometrical meaning of d_P and d_A.

A faster approach is to see the distance as two independent components – difference of positions d_P and difference of angles d_A, with respect to a given *point of interest* (ι_t, e.g. centre of gravity of tracked area corners, as can be seen in Fig. 3). The error of position is defined as the difference between the perpendicular distances from the lines to ι_t, in normalised homogeneous coordinates as:

$$d_P = \left|\iota_t^T l\right| - \left|\iota_t^T l'\right| \ . \tag{2}$$

The error of angle is defined as the length of the shortest possible line segment with endpoints on the lines, going through point ζ_A, which is equidistant to the lines and its distance to their intersection is equal to L (L can be derived from the size of the tracked area, or set manually).

$$d_A = 2 \cdot L \cdot \tan\frac{\mathit{\Delta}\alpha}{2} \ , \tag{3}$$

where $\mathit{\Delta}\alpha = \alpha_t^{[i]} - \alpha_t^{[j]}$ is the angle between the lines, and finally

$$d = \sqrt{d_P^2 + d_A^2} \ . \tag{4}$$

This approach gives errors similar to Hartley's in significantly lower time (10-fold speed-up with correlation coefficient 0.9). It should be noted that d_P is strongly underestimated in the case of ι_t laying *between* the lines. However, as we are usually concerned with the distance of lines that are close to each other, this condition appears rarely (correspondences are incorrectly classified as inliers less than one percent of the time).

3.4 Transformation from Corresponding Lines

Generally, every line correspondence yields two equations regardless of an estimated transformation (linear in homogeneous coordinates), as well as point correspondences. Nevertheless, in the case of similarity, two lines are obviously not enough (scale ambiguity). The similarity transformation should be computed from at least three line correspondences. Line equations can be directly used, or alternatively equations from point correspondences of intersections.

In contrast to using points, an algebraic error of line correspondence is very different from the geometric error. Therefore the linear least squares solution is not viable. Hence the sum of squares of the geometric errors is minimised by a numerical iterative optimisation.

4 Frame-to-frame Transformation

4.1 LO-RANSAC

The minimal sample is composed of three line correspondences. Intersections of the sampled lines are used for the computation of the hypothesis of S_t as a least squares solution (we have six linearly independent equations and only four degrees of freedom) from the point correspondences.

Standard (LO-)RANSAC use the number of inliers as a measurement of the quality of an estimated transformation. However, in the case of a cluttered background occupying a significant portion of the tracked area, the background-induced transformation may outweigh the correct one. Therefore we propose a different approach, measuring the quality of consistency of two frames, with respect to tracked points and to the evaluated transformation.

For every frame f_t, all the edges are detected by a Canny edge detector [16] and a distance transformation is performed. Evidence $e_t^{[i]}$ of a point $p_{t-1}^{[i]}$ in f_t is given by a modified oriented Chamfer distance [17, 18] of this projection as a product of inverse distance and an orientation weight:

$$e_t^{[i]} = e_{d;t}^{[i]} \cdot e_{A;t}^{[i]} = \frac{1}{1 + \left|\left|\mathrm{S}_t(p_{t-1}^{[i]}) - c_t^{[i]}\right|\right|_2} \cdot \left(\frac{\cos(\alpha_{t-1}^{[i]} + \rho(\mathrm{S}_t) - \alpha_t^{c[i]})}{2} + \frac{1}{2} \right) , \quad (5)$$

where $c_t^{[i]}$ is Canny's edge point in f_t nearest to $\mathrm{S}_t(p_{t-1}^{[i]})$, $\alpha_{t-1}^{[i]}$ is the direction of gradient at point $p_{t-1}^{[i]}$ in f_{t-1}, $\alpha_t^{c[i]}$ is the direction of gradient at point $c_t^{[i]}$ in f_t and $\rho(\mathrm{S}_t)$ is the rotation angle, given by transformation S_t.

Overall evidence of the transformation E_t is then computed as a mean evidence of all the points. To avoid situations when a solution is converging to a local optimum, representing impossible movements, a regularisation term is included as a multiplicative factor. This is a function of a scale change and of an overlap of old and new tracked areas.

$$E_t = \frac{1}{N_t} \sum_{i=1}^{N_t} e_t^{[i]} \cdot \min\left(\frac{\mathit{\Gamma}_t}{\mathit{\Gamma}'_t}, \frac{\mathit{\Gamma}'_t}{\mathit{\Gamma}_t} \right) \cdot \min\left(1, 2 \cdot \mathit{overlap}\right) \ , \quad (6)$$

where $\mathit{\Gamma}_t$ is scale change of S_t and $\mathit{\Gamma}'_t$ is an expected scale change.

4.2 On-line Learning and Using Point Quality

For an area where the points had predicted a correct transformation in previous frames, it has a high probability of having good points in future frames. The image evidence $e_t^{[i]}$ is used as a point quality measurement. The *point quality field* Q_t is learned from these as follows. Q_1 in the first frame is taken directly from detected edge points $p_1^{[i]}$. At the end of processing each frame f_t, the Q_{t-1} is transformed by the estimated S_t and a forgetting factor employed by multiplying by a constant decay. The evidence $e_t^{[i]}$ of points $p_{t-1}^{[i]}$ projected to f_t is then added.

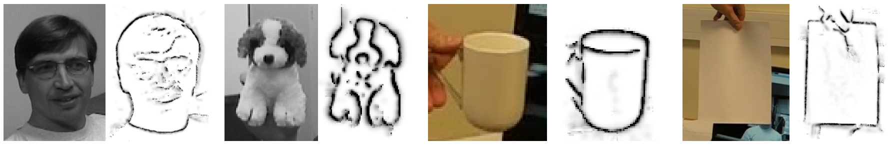

Fig. 4: Examples of images and learned point qualities.

The resulting transformation obtained by LO-RANSAC is noisy despite the minimisation of geometric error in the LO step. This causes inaccuracies in estimated motion and thus drift. To remove this drift, the learned field Q_{t-1} is used. Points from the new frame are back-projected ($\mathrm{S}_t^{-1}(q_t^{[i]})$) and the fit is measured as a mean point quality at their positions. Parameters of the transformation are refined to maximise this fitting score by non-linear iterative optimisation.

5 Long-term Relations

When a sudden decrease of evidence E_t is detected, confidence in the solution will be low and there may be strong drift or total loss of tracking. Then *correction* arises. The procedure of finding correspondences and the transformation is repeated with frame f_1 and initial model M_1 used instead of f_{t-1} and M_{t-1} and possibly with other frames and their models, if previously learned. A comparison of the fitness is carried out in terms of E_t and obtained inlier ratio. One of following situations appears (illustrated in Figure 5).

If the *current estimate* of S_t is the best solution (in terms of evidence score and inlier ratio), the current transformation is kept. Model M_{t-1} is transformed by estimated S_t and updated to get M_t. If the correcting model is a better fit than the current estimate then this *correcting transformation* is used rather than the current estimate and M_t is obtained from the correcting model (e.g. M_1). In this case, the assumption of a smooth movement has to be suppressed, as the correction transformation is not related to an actual movement of the tracked

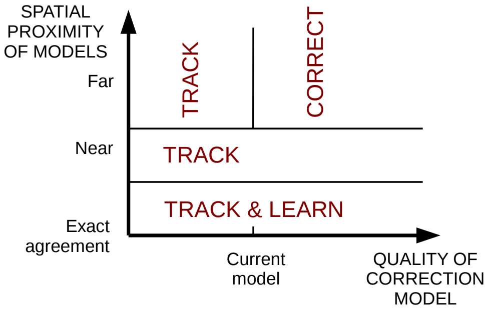

Fig. 5: Possible situations during correction, comparison of active (current) model M_{t-1} and the best correcting model.

object but to a recovery from failure or drift. In the case, when both the current and the best correcting transformations yield similar movement of the tracked area, current estimate is kept. Optionally, this model (as proving itself as leading to the good estimate) may be learned for future corrections.

6 Experimental Evaluation

The performance of the trackers was evaluated on sequences tabulated in Tab. 1 and shown in Figures 6, 8, 10 and 12 (our results superimposed). The first two selected sequences are used for evaluation in a number of previous publications. The latter two are new, obtained specially for their lack of texture. Supplementary material includes videos of all the sequences. Original sequences and the improved ground truth points are made available to the wider community at the website: `http://cmp.felk.cvut.cz/~lebedkar/sequences/`.

To asses trackers' performance, the distance of the centre of the tracked area from its ground truth position was measured as well as an error in the scale estimation (size of the target object; the logarithm of ratio to the ground truth is shown in the graphs, 0 means no error at average). All the measurements are averaged over 20 runs. The results can be seen in Fig. 7, 9, 11 and 13.

Table 1: Used Videosequences.

Name	Resolution	Frames	Colour	Challenges	Prev. Used
Dudek	720×480	1 145	grey	appearance change, occlusion, changing viewpoint	[2, 19]
dog	320×240	1 353	grey	changes in scale, occlusion	[20, 21]
mug	640×480	737	RGB	lack of texture, changes in scale, background	*new*
page	640×480	539	RGB	lack of texture, changes in shape, background	*new*

The performance of the proposed FLOtrack (Feature-Less Object tracker) was compared to several recently published trackers, representing different approaches: LGT by L. Cehovin [3], Flock of Trackers by T. Vojir (FoT [4]) and Z. Kalal's TLD [1]. The same settings were used for all the sequences.

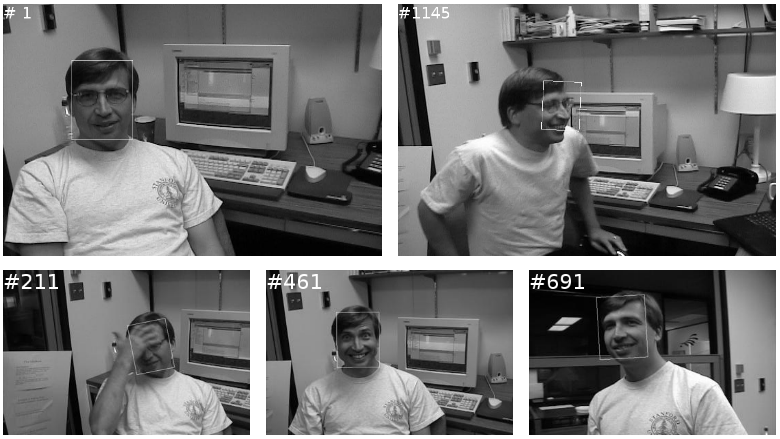

Fig. 6: Selected frames from Dudek sequence.

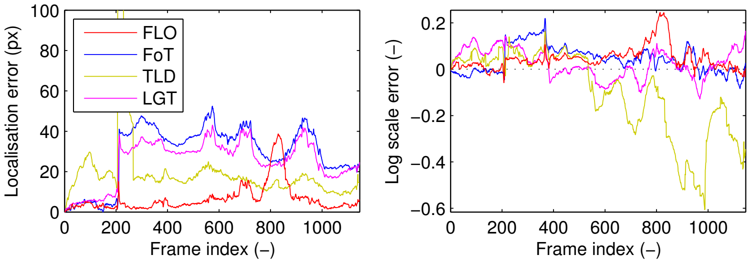

Fig. 7: Dudek sequence evaluation.

6.1 Discussion

Dudek: The most challenging part is at about the 210th frame, when the face is occluded by the right hand. While FLOtrack's pose is corrected in several frames, TLD needs about 50 frames and other trackers never fully recover. FLO also experiences difficulties around frame 800. Here background points influence tracking and cause drift. Nevertheless, FLO recovers in every run.

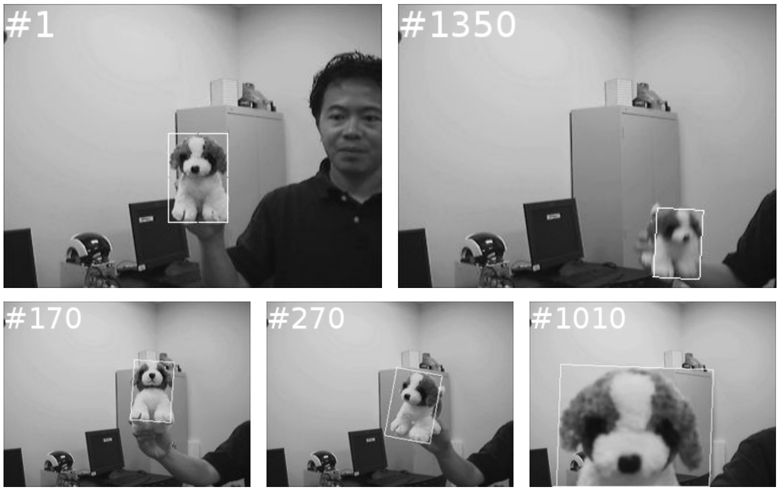

Fig. 8: Selected frames from DOG sequence.

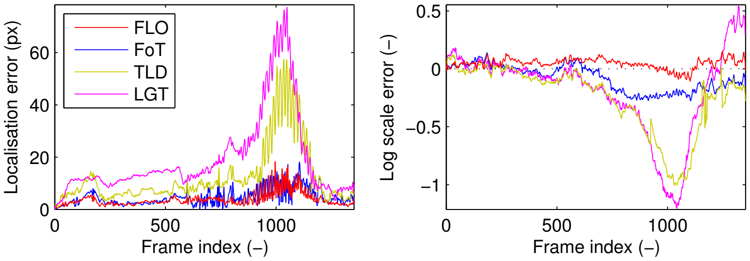

Fig. 9: DOG sequence evaluation.

Dog: In the range of frames between 700th and 1200th the challenges of this sequence are apparent. While FLO has no major problems and FoT experiences only light scale drift, the two others have severe problems, both in localisation and scale estimation.

Mug: Fig. 11 illustrates the inability of the LGT tracker on this scene. With no texture, the points simply drift off the mug and stay at the person's wrist. TLD consistently suffers from underestimation of the tracked area and sometimes loses the object totally. FoT works well in this sequence, while FLO is comparable up to around frame 400 at which point FLO lost tracking in approximately half of the runs, resulting in a poor average score.

Page: Fig. 13 shows performance of the trackers for this sequence. LGT performs similarly to the mug sequence, all the points stabilise at person's hand and wrist. FoT uses only features from fingers and TLD often loses tracking and rarely re-detects even when paper returns to a pose similar to the starting one. FLO experiences difficulties, but still significantly outperforms all the others.

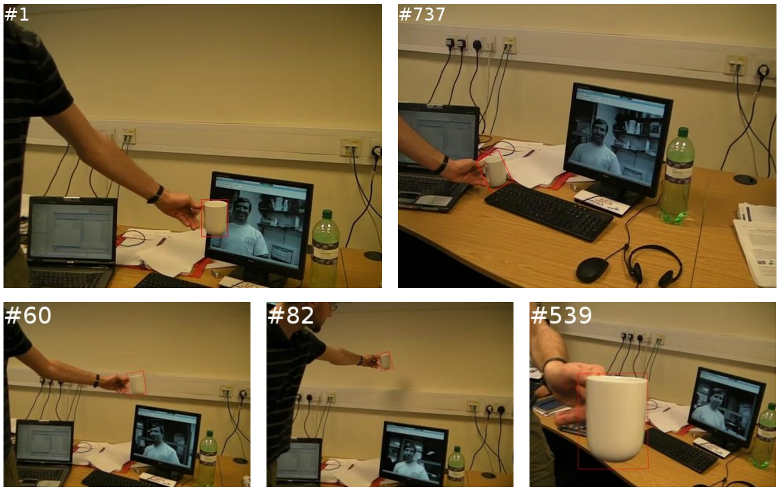

Fig. 10: Selected frames from MUG sequence.

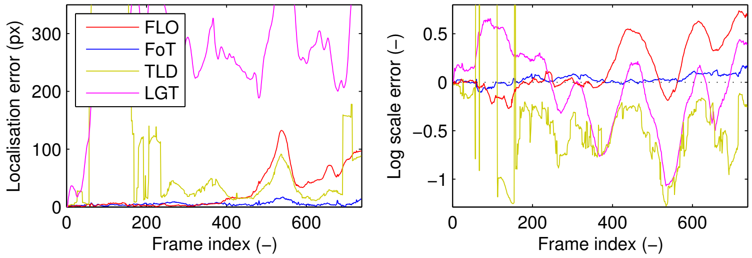

Fig. 11: MUG sequence evaluation.

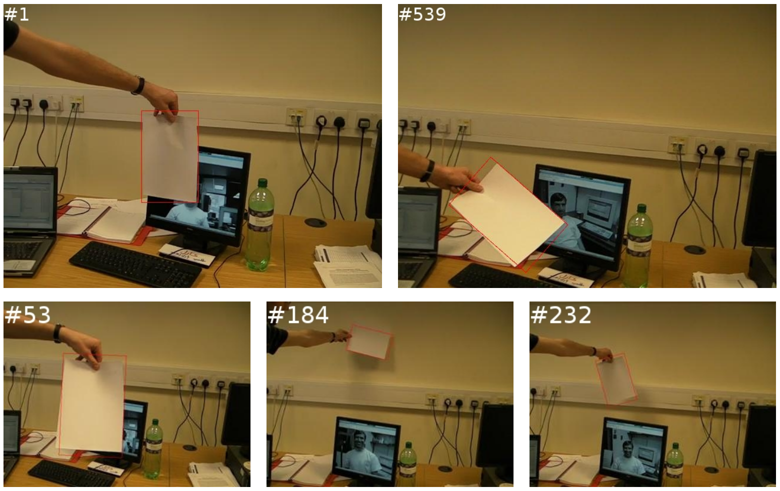

Fig. 12: Selected frames from PAGE sequence.

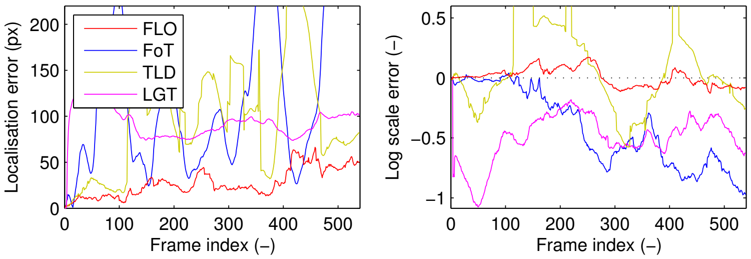

Fig. 13: PAGE sequence evaluation.

Additionally, we carried out qualitative tests of FLOtrack on further sequences with different challenges, such as a strong illumination change and a low resolution. The results are positive as FLO works well even in these conditions (see supplementary material).

7 Conclusion

We proposed and implemented a novel tracking algorithm based on low-level line correspondences with significantly lowered dependency on image features/texture. The tracker gives results competitive or superior to state-of-the-art trackers.

In future work, re-detection should be employed to upgrade to the long-term tracking. To increase stability, a memory holding the history of successful estimation should be increased from just remembering positions of good edgels (in fact learning the object contour) could be extended to hold configurations/combinations of complementary points (e.g. line triplets).

Acknowledgement. The authors were supported by the following projects: SGS11/125/OHK3/2T/13, GACR P103/12/2310 and EPSRC EP/I011811/1.

References

1. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N learning: Bootstrapping binary classifiers by structural constraints. In: Proc. of CVPR. (2010) 49 –56
2. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via On-line Boosting. In: Proc. of BMVC. (2006)
3. Cehovin, L., Kristan, M., Leonardis, A.: An adaptive coupled-layer visual model for robust visual tracking. In: Proc. of ICCV. (2011)
4. Matas, J., Vojir, T.: Robustifying the flock of trackers. In: Proc. of Computer Vision Winter Workshop. (2011) 91–97
5. Dupac, J., Matas, J.: Ultra-fast tracking based on zero-shift points. In: Proc. of ICASSP. (2011) 1429 –1432
6. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of Imaging Underst. Workshop. (1981) 121–130
7. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision **60** (2004) 63–86

8. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. of Alvey Vision Conference. (1988) 147–151
9. Harris, C., Stennett, C.: Rapid – a video rate object tracker. In: Proc. of BMVC. (1990)
10. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Trans. PAMI **24** (2002) 932–946
11. Tsin, Y., Genc, Y., Zhu, Y., Ramesh, V.: Learn to track edges. In: Proc. of ICCV. (2007) 1–8
12. Beveridge, J.R., Riseman, E.M.: How easy is matching 2D line models using local search? (1997)
13. Chum, O., Matas, J., Kittler, J.: Locally optimized RANSAC. In: DAGM-Symposium. (2003) 236–243
14. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
15. Hartley, R.I.: Projective reconstruction from line correspondences. In: Proc. of CVPR. (1994) 903–907
16. Canny, J.: A computational approach to edge detection. IEEE Trans. PAMI **8** (1986) 679–698
17. Olson, C.F., Huttenlocher, D.P.: Automatic target recognition by matching oriented edge pixels. IEEE Trans. Image Processing **6** (1997) 103–113
18. Shotton, J., Blake, A., Cipolla, R.: Multiscale categorical object recognition using contour fragments. IEEE Trans. PAMI **30** (2008) 1270 –1281
19. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. PAMI **25** (2003) 1296–1311
20. Ross, D., Lim, J., Yang, M.H.: Adaptive probabilistic visual tracking with incremental subspace update. In: Proc. of ECCV. (2004) 470–482
21. Chen, M., Pang, S.K., Cham, T.J., Goh, A.: Visual tracking with generative template model based on riemannian manifold of covariances. In: Proc. of Int. Conf. on Information Fusion. (2011) 874–881
22. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)