Visual Coin-Tracking: Tracking of Planar Double-Sided Objects

Jonáš Šerých and Jiří Matas

CMP Visual Recognition Group, Dept. of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

Abstract. We introduce a new video analysis problem – tracking of rigid planar objects in sequences where both their sides are visible. Such *coin-like objects* often rotate fast with respect to an arbitrary axis producing unique challenges, such as fast incident light and aspect ratio change and rotational motion blur. Despite being common, neither tracking sequences containing coin-like objects nor suitable algorithm have been published.

As a second contribution, we present a novel *coin-tracking benchmark* containing 17 video sequences annotated with object segmentation masks. Experiments show that the sequences differ significantly from the ones encountered in standard tracking datasets. We propose a baseline coin-tracking method based on convolutional neural network segmentation and explicit pose modeling. Its performance confirms that coin-tracking is an open and challenging problem.

1 Introduction

Visual tracking is one of the fundamental problems in the field of computer vision. Given a video sequence and some defined object, e.g. by its location in the first frame, the task is to find its pose in each frame of the sequence. Until recently, standard visual tracking datasets like [17] or [23] have been only annotated using bounding boxes and subsequently, state-of-the-art trackers usually represented the objects pose as a rotated or axis-aligned bounding box. Recently, tracking-by-segmentation, also called video object segmentation, has gained on popularity, thanks to the introduction of segmentation-annotated datasets like DAVIS [20] and YouTube-VOS [24]. Here, object pose is a segmentation mask.

Visual tracking is an active research field; tracker performance improves significantly every year [16,15]. Nevertheless, a particular class of every-day objects remains challenging even for state-of-the-art methods, namely, rigid flat double-sided objects like cards, books, smartphones, magazines, coins[1], tools like knives, hand saws, sport equipment like table tennis rackets, paddles etc. Such objects often rotate fast producing unique challenges for trackers like fast incident light and aspect ratio change and rotational motion blur.

[1] Hence the problem name.

In this paper, we introduce an annotated *coin-tracking dataset*[2], CTR dataset in short, containing video sequences of coin-like objects. We then show that the proposed dataset is fundamentally different from the standard ones [15,23]. Finally, we propose a baseline coin-tracking method, called CTR-Base, that outperforms classical state-of-the-art trackers in experiments on the CTR dataset.

2 Coin-Tracking Dataset

We define coin-tracking as tracking of rigid, approximately planar objects in video sequences. This means that at any time only one of the two sides - *obverse* (front) and *reverse* (back) - is visible. Unlike general objects, the rigidity and planarity of the coin-like objects means that the boundary between their two sides is always visible, except for occlusions by another object and position partially outside of the camera field of view. In this settings, the currently invisible side is fully occluded by the visible side and the visible side does not occlude itself at all. The state of a coin-like object is thus fully characterized by a visible side identification and a homography transformation to a canonical frame together with a possible partial occlusion mask.

However, because the objects in the CTR dataset are often symmetric, reflecting the real world coin-like object properties, the homography transformation might not be uniquely identifiable and thus we characterize the object state by a segmentation mask instead. Notice that unlike in standard general tracking sequences, where the exact extend of the tracked object is often not well defined due to the ambiguity of the initialization bounding box or segmentation, there is an unambiguous correspondence between a segmentation mask and a physical object in the case of coin-tracking.

Recent video object segmentation datasets [20,24] represent the object pose by segmentation as well, nevertheless, they contain mostly outdoor sequences of animals, people and vehicles. Therefore, there is a significant domain gap between these datasets and the proposed coin-tracking problem. Other datasets for tracking planar object exist, such as [18,5], but they only contain sequences with single side of the planar object visible. Moreover, in most cases the objects are fixed and the camera moves around them. This induces both different dynamics and appearance changes in the sequences as discussed in section 2.1.

The are multiple levels of tracking of coin-like objects. In the simplest form, the tracker is initialized by a template of each side of the object and the object pose on the first frame of the sequence. One could also initialize the tracker on the first side only and require it to discover the reverse side without supervision. Moreover, a full 6D pose output (rotation and translation) together with a complete object surface reconstruction (including even the initially occluded parts of the object) could be required for sequences with known camera calibration.

The introduced CTR dataset contains 17 video sequences of coin-like objects, with total of 9257 frames and segmentation ground truth masks on every fifth frame. See Fig. 1 for examples of the sequences in the CTR dataset.

[2] Available at `http://cmp.felk.cvut.cz/coin-tracking`.

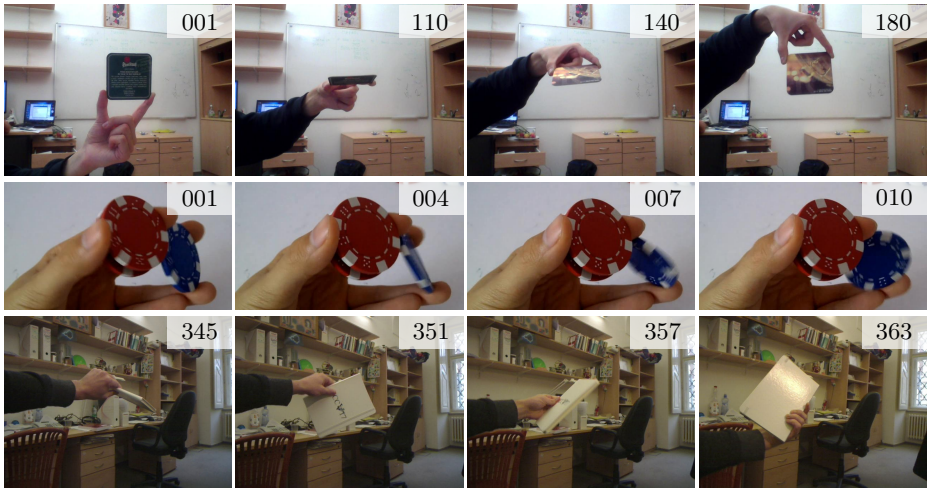


Fig. 1: Examples from the coin-tracking dataset (frame number in the top-right corner). Notice the effects of the out-of-plane rotation – fast illumination change, blur and significant aspect ratio change of the objects.

2.1 A Comparison with Other Datasets

The main motivation for introducing a new tracking dataset is its difference from the currently available tracking sequences. In this section we show some of the novel aspects of the proposed dataset.

The planar object tracking datasets [5,18] are the closest to the CTR dataset, but they only contain a single sided view of the object; the viewing angle range is limited. In most of the sequences the tracked object is fixed to the background behind it, e.g. a poster fixed on a wall and the object motion in the sequence is induced by the camera motion only. On the contrary, the camera is static or close to static in many of the CTR sequences and it is the object that causes the motion. This difference is important since the two situations introduce different challenges to the visual tracking task.

When a planar object is fixed and a camera moves around it, the perceived out-of-plane rotation is relatively slow as the camera needs to move along a long arc in order to change the viewing angle significantly. On the other hand, when the main part of the perceived motion of the object in the sequence is caused by the physical motion of the object itself, as it is the case in the proposed sequences, the object out-of-plane rotation happens faster as it is physically easy to rotate coin-like objects.

Most state-of-the-art trackers, e.g. the winners of the VOT2018 tracking challenge [15] – MFT [1] and UPDT [2], represent the object pose as axis-aligned or rotated bounding box, while the aspect ratio change modeling is not common. Later in this section, we show that both the range and the speed of aspect ratio change in the CTR sequences is higher than in the VOT [14] and OTB [23] track-

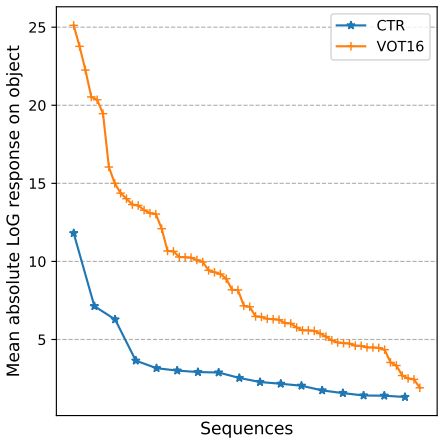


Fig. 2: Comparison of object "textureness" in the proposed CTR and VOT 2016 datasets, measured by the absolute value of Laplacian of Gaussian $\sigma = 0.8$ averaged over the tracked object pixels.

ing datasets. Besides causing significant aspect ratio changes, the 3D rotation of the coin-like objects often induces fast changes of illumination as the object plane normal direction relative to the light sources changes rapidly. Apart from these differences, the objects in the CTR dataset are also less textured than the ones appearing in standard visual tracking datasets as discussed in the next section.

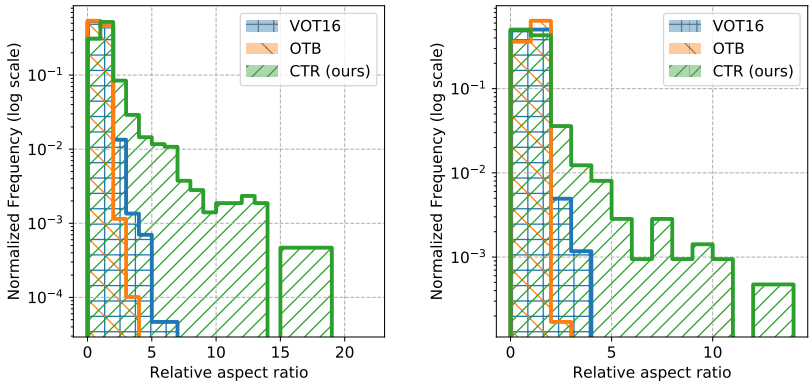
Textureness. As a measure of object textureness, we computed the Laplacian of Gaussian (LoG) responses and averaged their absolute values over the object pixels and all frames. Fig. 2 shows that the typical object textureness in the CTR dataset is significantly lower than on the VOT 2016 dataset [14]. The lack of texture prevents tracking to be implemented by classical methods for homography estimation based on key-point correspondences.

Aspect ratio change. One of the unique properties of the coin-tracking dataset is the presence of strong changes in object aspect ratios, not usually encountered in the standard visual tracking datasets as shown in the following two experiments. In order to compute the aspect ratio statistics, we first compute minimal (rotated) rectangle bounding the ground truth segmentation mask on each frame. The aspect ratio (1) of the resulting rectangle with sides a, b is defined as

$$r(a,b) = \max\left(\frac{a}{b}, \frac{b}{a}\right) \tag{1}$$

We define the relative change in aspect ratios of two rectangles A, B with sides a_1, a_2 and b_1, b_2, respectively, as (2)

$$\Delta r(A,B) = \max\left(\frac{r(a_1,a_2)}{r(b_1,b_2)}, \frac{r(b_1,b_2)}{r(a_1,a_2)}\right) \tag{2}$$



(a) relative to the first frame. (b) relative to frame t-5.

Fig. 3: Histogram of aspect ratio changes

The maximum of the two ratios is chosen because only the magnitude of the aspect ratio change matters.

Aspect ratio change relative to the first frame. We have computed aspect ratio changes $\Delta r(R_1, R_t)$ between the bounding rectangle on the first frame and each of the other annotated frames in the sequence. We then represent each tested dataset (VOT2016, OTB, CTR) by a histogram of these aspect ratio changes in all the dataset sequences as shown in Fig. 3a. Notice that although the VOT2016 and OTB datasets are not restricted to rigid objects, i.e. their segmentation masks can change shape arbitrarily during the sequences, the CTR dataset contains significantly bigger changes in the aspect ratios.

Aspect ratio change speed. In the proposed CTR dataset, the change in object aspect ratio is also faster than in the other compared datasets as shown in Fig. 3b. Instead of computing the aspect ratio change with respect to the first frame, the change is computed relative to the previous frame. Notice that because the CTR dataset does not contain ground truth segmentation masks on every frame, but only on every fifth, we measure $\Delta r(R_{t-5}, R_t)$ on all three datasets.

2.2 Evaluation Metric

We address the simplest form of the coin-tracking task, in which the tracker is initialized by an image of the front side of the tracked object on the first frame and an image of the back side later in the sequence, together with the respective ground truth segmentation masks.

We use *intersection over union* (IoU) as the evaluation metric – it is the standard metric for evaluating both segmentation and bounding box quality. In order to deal with frames with empty ground truth segmentation, i.e. with

the object fully occluded or fully outside of the view, we augment the scoring function such that these frames do not contribute into the per-sequence total as proposed in [15].

3 The Baseline Coin-Tracking Method

Standard trackers represent the object by a bounding box and are thus unable to capture the perspective transformations common for coin-like objects. Trackers based on key-point correspondences can estimate homographies, but the low textureness of CTR objects prevents their use. Convolutional neural networks recently used for video object segmentation, e.g. [3,12,22], classify pixels as object or background taking into account large context thanks to large receptive fields of the neurons in the final layers. They do not consider the underlying homography transformations, but the segmentations capture the object extent in the image with high granularity.

Most video object segmentation methods use a deep neural network trained offline for general object segmentation. The network is then fine-tuned for tracking of a particular object at the initialization. One of the significant challenges in visual tracking is object appearance change and changes in the background in the video sequence. Because of this, trackers usually have to perform some kind of *online adaptation* to prevent performance deterioration soon after initialization. A simple adaptation scheme for video object segmentation has been proposed in OnAVOS [22], where the pixels classified as object with high confidence are treated as new object appearance examples. Background examples are taken from the parts of the image over a certain distance from the object. However, the online adaptation requires lengthy fine-tuning of the segmentation neural network on each frame, making the method slow.

An alternative approach has been proposed in FAST-VOS [6], where the segmentation is done by k-nearest neighbor search in an embedding space learned offline by a CNN. Instead of fine-tuning the embedding network on the first frame or later during online adaptation, the FAST-VOS method inserts dense embeddings into a k-NN classifier index. This makes the adaptation to a particular object faster and easier to interpret, compared to the network fine-tuning methods. The online adaptation proposed in [6] is similar to the original method in [22], selecting high confidence pixels – all of their $k = 5$ neighbors agree with the label – for the model update.

With all this in mind, we propose a baseline tracking method CTR-Base, which is based on the tracking-by-segmentation FAST-VOS [6] method. After an input frame is segmented using the k-NN classifier, we explicitly model the object pose and possibly perform online adaptation.

3.1 Object Pose Estimation

We have performed experiments with the adaptation scheme of FAST-VOS but it did not work well on the coin-tracking sequences. The adaptation has quickly

drifted and led to a complete failure of the tracker, either segmenting almost all of the background as the object or vice versa. Our experiments with distance-threshold based background adaptation as in [22] as well as experiments with other heuristics based on analysis of the connected components and other properties of the segmentation mask were not successful either. We hypothesize that one of the reasons that those adaptation techniques work reasonably well on the DAVIS dataset, but fail on the coin-tracking task, might be the length of the sequences. The mean number of frames in the DAVIS 2017 sequences is only 69.7 [21] while the mean number of frames in the coin-tracking sequence in the CTR dataset is 544, with several sequences as long as 1000 frames. The robustness of the online adaptation scheme is crucial on sequences of such length.

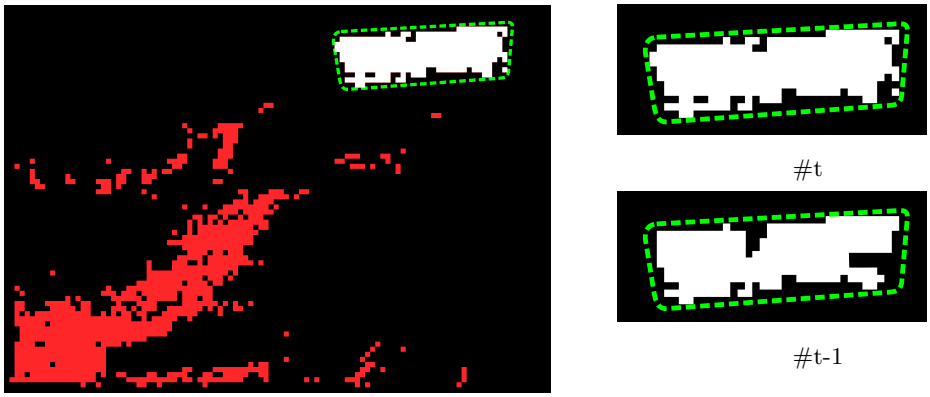


Fig. 4: Homography score computation. Left: the segmentation mask split into pixels inside (white) the object pose hypothesis (dashed green) and the rest (red). Right: Object visibility mask for the current and the last frames.

In order to address the online adaptation in coin-tracking more robustly, we explicitly model the object pose using the homography to the ground-truth canonical frame. Both the object and the background pixel online adaptation is controlled by the agreement between the segmentation output by the k-NN classifier and the estimated pose model.

Objective Function. In each video frame, we search for the homography $\mathbf{H}_{*\to t}$ mapping the object on a ground truth frame into the current one, optimizing the objective function s, Eq. 5, composed of four parts computed as follows. First, we map the segmentation mask from the ground truth frame into the current frame using the homography. This splits the segmentation mask in the current frame into two parts, one inside and the other one outside of the hypothesized object contour as shown in Fig. 4. The s_{obj} part of the score function is set to the fraction of the segmentation mask located inside the contour, indicating

the fraction of the segmentation explained by the object. This part of the score function penalizes segmentation outside of the object with the pose given by $\mathbf{H}_{*\to t}$.

The s_{cover} part of the score function s is the fraction of the pixels inside the hypothesized object contour being classified as the object. This part penalizes homographies mapping the object contour such that it is not well covered by the segmentation. Notice, however, that in the case of partial occlusion by other object, the segmentation should not cover the whole object. Since the occlusion mask is changing relatively slowly in CTR sequences, the s_{occl} component of the score function s is the IoU overlap of the current and last visibility mask, which is transformed to the current frame by $\mathbf{H}_{t-1\to t} = \mathbf{H}_{*\to t}\mathbf{H}^{-1}_{*\to t-1}$. This prefers homographies with a small occlusion change with respect to the previous frame.

Finally, the appearance score $s_{\text{appearance}}$ is the zero-offset coefficient of the zero-normalized cross-correlation (ZNCC) score

$$s_{\text{appearance}} = \frac{1}{2} + \frac{\sum\limits_{x,y\in O}(I_t(x,y)-\mu(I_t))(I_*(x,y)-\mu(I_*))}{2\sqrt{\sum\limits_{x,y\in O}(I_t(x,y)-\mu(I_t))^2 \sum\limits_{x,y\in O}(I_*(x,y)-\mu(I_*))^2}} \tag{3}$$

of the object image in the current frame and the template from the ground-truth frame, where $I_t(x,y)$ and $I_*(x,y)$ are the image values at coordinates $[x,y]$ in the current frame and the ground truth frame projected using the homography $\mathbf{H}_{*\to t}$ respectively and

$$\mu(I) = \frac{1}{|O|}\sum_{x,y\in O} I(x,y) \tag{4}$$

with O being the set of points segmented as object in both the ground truth and the current frame. The rationale behind introducing the appearance score is that it helps distinguishing a correct homography in case of objects with symmetric shape or partial occlusions. The final score, Eq. 5, of the homography is the product of these four components giving a number in 0-1 range:

$$s = s_{obj} \cdot s_{cover} \cdot s_{occl} \cdot s_{appearance} \tag{5}$$

Notice that compared to summing the score components, taking their product highlights drops in any of the score components and thus it is preferable for making our adaptation method conservative.

Optimization. Since the cost function described above is not differentiable, we use a probabilistic optimization procedure based on simulated annealing for finding $\mathbf{H}_{*\to t}$ for each frame. The optimization is initialized using either the homography found in the previous frame or using optical flow from the previous frame, in which case we uniformly sample 4 points from inside the object and transform them by the flow field to get 4 correspondences necessary for estimating the inter-frame homography. This is repeated 50 times and the $\mathbf{H}_{*\to t}$

maximizing the score function is chosen as the initialization of the following iterative optimization procedure.

In each step of the optimization a random homography matrix is sampled by randomly perturbing 4 control points at the corners of the object bounding box and computing the homography from the resulting 4 correspondences. Next, the homography score s is computed and compared to the current best score, s^*. The $\mathbf{H}_{*\to t}$ hypothesis is accepted as the current estimate of the optimum with probability

$$p(s, s^*, T) = \begin{cases} 1 & \text{if } s > s^*, \\ e^{-\frac{s^* - s}{T}} & \text{otherwise,} \end{cases} \tag{6}$$

where the T is decreasing in each iteration, allowing jumps from local minima but with decreasing probability during the optimization procedure. We also decrease the control point perturbation σ in each of the 350 iterations.

Depending on the ratio of pixels being classified as belonging to the obverse or the reverse side of the object, the optimization procedure is run against the respective ground truth frame. Finally, when the score of the best found homography is low, the tracker switches into a *lost* state and stays in it until a successful re-detection of the object.

The re-detection procedure is the same as the optimization described above, except for spending more time (400 iterations) sampling for the initialization pose and not using the information from the previous frame. The previous visibility mask used in computation of s_{occl} is replaced by the full object mask.

3.2 Online Adaptation

The proposed homography optimization procedure reduces the overall speed of the tracker, but we have observed that it finds a good solution reliably, unless the segmentation is grossly incorrect, enabling us to use online-adaptation on the long sequences in the CTR dataset. In particular, no online adaptation is attempted when the tracker is in the *lost* state, reducing the probability of making incorrect adaptation.

If the tracker is in the *tracking* state, new background and object embedding examples are added into the segmentation k-NN classifier. To stay on the safe side, only the pixels that are far from the object boundary and were incorrectly classified (with respect to the hypothesized object pose) are used as new background examples. Moreover, these pixels must not be connected to the object by the segmentation mask, otherwise they are not used for adaptation even if they are very far from the image.

For the new object examples, we select the pixels classified as background by the segmentation k-NN classifier that are not connected to the object edges, in other words only closed 'holes' in the object segmentation are adapted.

Altogether, the proposed online adaptation technique allows for conservative online adaptation, not making severe mistakes that would lead to complete failure of the tracker, as shown in the experiments in section 4.2.

3.3 Implementation details

We use a DeepLabv3+ [4] segmentation head on top of MobileNetv1 [10] backbone architecture. The MobileNet backbone was pretrained[3] on ImageNet [7], then trained for semantic segmentation on PASCAL VOC 2012 [8] enriched by the *trainaug* augmentations by [9]. We have used the Adam [13] optimizer with batch size 5 and initial learning rate of 7×10^{-4} decaying to 10^{-6} according to the *poly* schedule with decay power 0.9 for 53000 iterations. Finally, using the augmented triplet loss proposed by [6], we have fine-tuned the network for 492000 iterations on the YouTubeVOS dataset [24] to output dense 128-dimensional embeddings useful for segmentation by k-NN classifier. Given an $H \times W$ image, the network produces a per-pixel 128-D embeddings with output stride 4 (resolution $\frac{H}{4} \times \frac{W}{4}$). We use FAISS [11] library[4] with a flat L2 index for speeding up the nearest neighbor searches used in the segmentation. For the optical flow computation, we use ContinualFlow [19].

The method runs at around 7 seconds per frame at 1280×720 resolution with the majority of time spent optimizing the pose. The runtime drops without losing much performance when the pose optimization is done on lower resolution.

4 Experiments

In this section we show that the proposed CTR-Base method outperforms general state-of-the-art trackers on the CTR dataset and retains good performance on the POT-210 [18] dataset. Then we demonstrate that the homography-based pose modeling prevents the CTR-Base tracker from making fatal mistakes.

4.1 Baseline Experiment

In the standard visual tracking formulation, the tracker is initialized by the ground truth object pose, which can be represented by axis-aligned bounding box, rotated bounding box or segmentation mask [15,20,24]. This means that standard state-of-the-art trackers cannot be directly evaluated on the coin-tracking task in which the tracker is initialized on one frame from each side of the object. On the other hand, the coin-tracking task can be viewed as a long-term tracking on single side, enabling us to evaluate state-of-the-art long term trackers MBMD [25] and DaSiam_LT [26] – the winners of the VOT 2018 [15] long-term tracking challenge on the CTR dataset. Moreover, the VOT long-term tracking challenge requires a tracker confidence output on each frame, which allows us to run each tracker two times - once initialized from the obverse and once from the reverse side, merging the results by picking the one with higher tracker confidence. We have represented the axis-aligned bounding box outputs of the long-term trackers as segmentation masks and evaluated using the IoU metric. The results are shown in Tab. 1.

[3] Code and weights available at `https://github.com/tensorflow/models/`
[4] Available at `https://github.com/facebookresearch/faiss`

The proposed CTR-Base method significantly outperforms both state-of-the-art bounding box trackers and a bounding box oracle, which outputs the bounding boxes of the ground truth segmentation masks. Computing IoU from the bounding boxes might not seem fair, but the performance gap demonstrates the need of representing the tracked object by segmentation, even with relatively compact objects present in the CTR dataset.

sequence	MBMD	DaSiam_LT	bbox oracle	CTR-Base (ours)
beermat	0.70	0.18	0.78	**0.83**
card1	0.72	0.71	0.73	**0.79**
card2	0.71	0.68	0.79	**0.93**
coin1	0.60	0.62	0.71	**0.80**
coin3	0.32	**0.46**	0.63	0.38
coin4	0.33	0.41	0.56	**0.65**
husa	0.35	0.40	0.51	**0.73**
iccv_bg_handheld	0.27	0.31	0.54	**0.33**
iccv_handheld	0.32	0.39	0.55	**0.50**
iccv_simple_static	0.37	0.31	0.51	**0.65**
iccv_static	0.34	0.40	0.55	**0.67**
pingpong1	**0.42**	0.38	0.64	0.33
plain	0.44	0.50	0.60	**0.74**
statnice	0.53	0.57	0.67	**0.87**
tatra	0.47	0.54	0.66	**0.86**
tea_diff_2	0.54	0.57	0.61	**0.87**
tea_same	0.53	0.52	0.63	**0.85**
Mean over all frames	0.47	0.44	0.63	**0.70**

Table 1: The evaluation of the IoU overlap metric on the proposed CTR dataset. Notice that the CTR-Base method outperforms both state-of-the-art long-term trackers and the bounding box oracle.

In order to further test the CTR-Base method, we evaluated it on the POT-210 [18] dataset, converting the ground – object corners – to segmentation (not modeling occlusions). The mean IoU (mIoU) is 0.81, showing that our method generalizes to POT-210 well. The best results were achieved on the *out-of-view* and the *perspective distortion* subsets of [18] with mIoU 0.89 and 0.88 respectively, while the worst on the *motion blur* subset with mIoU of 0.71.

4.2 Results on confident frames

The mean IoU score computed only on the frames where the CTR-Base method is in the *tracking* state, i.e. online adaptation is allowed, improves from 0.70 to 0.88. This shows that the proposed tracker can correctly detect its own failures and only adapt when tracking reliably. Overall the tracker spends 47% of the frames in the *tracking* state as shown in Tab. 2.

sequence	beermat	card1	card2	coin1	coin3	coin4	husa	iccv_bg_handheld	iccv_handheld	iccv_simple_static	iccv_static	pingpong1	plain	statnice	tatra	tea_diff_2	tea_same	**average**
IoU ×100	89	89	96	82	94	84	87	90	85	85	83	67	88	89	92	92	86	88
frames in *tracking* state %	89	68	93	64	02	21	69	17	15	29	28	17	42	46	34	87	47	47

Table 2: The IoU score of the CTR-Base tracker evaluated only on the frames, where it is in the confident *tracking* state and the online adaptation is enabled. Notice that indeed the tracker is confident on the frames, where it performs well.

5 Conclusion

We have introduced a novel video analysis problem – coin tracking – and presented a novel tracking CTR dataset consisting of 17 sequences of *coin-like* objects and ground truth segmentations. We have shown its dissimilarity to other tracking datasets. Besides studying the special properties of coin-like objects, the CTR dataset may benefit both training and the evaluation of general trackers, including video object segmentation methods, because it contains objects classes different from the ones encountered in the available datasets. Sequences in CTR are long, making online adaptation more challenging.

We have proposed a baseline CTR-Base tracking method that enables robust online adaptation through explicit modeling of the tracked object pose and failure detection. The proposed CTR-Base method outperforms state-of-the-art long-term trackers on the CTR dataset in terms of the IoU while generalizing well to the POT-210 dataset [18].

Finally, the advanced variants of the coin-tracking task described in section 2, like the unsupervised back side discovery or full surface reconstruction, are challenging and open topics left for future research.

Acknowledgements. This work was supported by Toyota Motor Europe HS, by CTU student grant SGS17/185/OHK3/3T/13 and Technology Agency of the Czech Republic project TH0301019.

References

1. Bai, S., He, Z., Xu, T.B., Zhu, Z., Dong, Y., Bai, H.: Multi-hierarchical independent correlation filters for visual tracking. arXiv preprint arXiv:1811.10302 (2018)
2. Bhat, G., Johnander, J., Danelljan, M., Shahbaz Khan, F., Felsberg, M.: Unveiling the power of deep tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 483–498 (2018)

3. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 221–230 (2017)
4. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
5. Chen, L., Ling, H., Shen, Y., Zhou, F., Wang, P., Tian, X., Chen, Y.: Robust visual tracking for planar objects using gradient orientation pyramid. Journal of Electronic Imaging **28**(1), 1–16 (2019)
6. Chen, Y., Pont-Tuset, J., Montes, A., Van Gool, L.: Blazingly fast video object segmentation with pixel-wise metric learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1189–1198 (2018)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (June 2009)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
9. Hariharan, B., Arbelaez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: International Conference on Computer Vision (ICCV) (2011)
10. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
11. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data (2019)
12. Khoreva, A., Benenson, R., Ilg, E., Brox, T., Schiele, B.: Lucid data dreaming for object tracking. In: The DAVIS Challenge on Video Object Segmentation (2017)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014)
14. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojíř, T., Häger, G., Lukežič, A., et al.: The Visual Object Tracking VOT2016 Challenge Results, pp. 777–823. Springer International Publishing (2016)
15. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Cehovin Zajc, L., Vojir, T., Bhat, G., Lukezic, A., Eldesokey, A., et al.: The sixth visual object tracking vot2018 challenge results. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
16. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernández, G., Vojir, T., Hager, G., Nebehay, G., Pflugfelder, R.: The visual object tracking vot2015 challenge results. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 1–23 (2015)
17. Kristan, M., Matas, J., Leonardis, A., Tomáš, V., Pflugfelder, R., Fernandez, G., Nebehay, G., Porikli, F., Čehovin, L.: A novel performance evaluation methodology for single-target trackers. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(11), 2137–2155 (2016)
18. Liang, P., Wu, Y., Lu, H., Wang, L., Liao, C., Ling, H.: Planar object tracking in the wild: A benchmark. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 651–658. IEEE (2018)
19. Neoral, M., Šochman, J., Matas, J.: Continual occlusion and optical flow estimation. In: Asian Conference on Computer Vision. pp. 159–174. Springer (2018)

20. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016)
21. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675v2 (2017)
22. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for video object segmentation. British Machine Vision Conference (BMVC) (2017)
23. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(9), 1834–1848 (2015)
24. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: The European Conference on Computer Vision (ECCV) (2018)
25. Zhang, Y., Wang, D., Wang, L., Qi, J., Lu, H.: Learning regression and verification networks for long-term visual tracking. arXiv preprint arXiv:1809.04320 (2018)
26. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 101–117 (2018)