MTAP manuscript No.
DOI

Prototype-based budget maintenance for tracking in depth videos

Sari Awwad · Massimo Piccardi

Received: date / Accepted: date

Abstract The use of conventional video tracking based on color or gray-level videos often raises concerns about the privacy of the tracked targets. To alleviate this issue, this paper presents a novel tracker that operates solely from depth data. The proposed tracker is designed as an extension of the popular Struck algorithm which leverages the effective framework of structural SVM. The main contributions of our paper are: i) a dedicated depth feature based on local depth patterns, ii) a heuristic for handling view occlusions in depth frames, and iii) a technique for keeping the number of the support vectors within a given "budget" so as to limit computational costs. Experimental results over the challenging Princeton Tracking Benchmark (PTB) dataset report a remarkable accuracy compared to the original Struck tracker and other state-of-the-art trackers using depth and RGB data.

Keywords Depth videos · tracking · Struck tracker · local depth patterns · budget maintenance · Princeton Tracking Benchmark dataset

1 Introduction

The aim of video tracking is to extract the trajectories of a chosen set of targets. However, given that tracking is typically performed on color or gray-level video, it also allows the identification of the tracked targets in many cases. While this is desirable in video surveillance scenarios, it may prove inappropriate in applications where privacy is paramount such as patient monitoring in hospitals and care facilities. Even if post-processing can be applied to obfuscate personal traits, the collection of appearance data in the first instance poses a potential threat to privacy that has to be balanced with the needs for operation and security.

In recent years, the release of sensors such as Microsoft Kinect has made it possible to acquire depth videos inexpensively. A significant trend in tracking research has

S.Awwad · M. Piccardi
University of Technology, Sydney (UTS, PO Box 123, Broadway, NSW, Australia)
E-mail: {Sari.Awwad@student., Massimo.Piccardi@}uts.edu.au

become the use of depth data in addition to RGB data to disambiguate occlusions and overcome illumination artifacts [4, 17, 25]. However, the possibility to perform general tracking solely from depth videos has received only partial attention to date. The challenges posed by pure depth tracking are major since conventional trackers rely on the targets' appearance and texture to provide correct data association. While in many applications it is possible to successfully fit and track skeletal models on depth data (see, for all, Shotton *et al.* [22]), skeletal tracking is mostly designed for interaction with co-operative human users and is prone to failure in the presence of severe view occlusions. For these reasons, the aim of this paper is to investigate tracking of a generic target's global motion based on depth data alone.

One of the most popular approaches for video tracking is known as *tracking-by-detection* [1,3,6,10,11,26]. Its main idea is to frame tracking as a target classification problem and learn the classifier in an online and unsupervised manner. In this category, the Struck tracker from Hare *et al.* [12] has recently attracted much attention since it leverages the efficient, discriminative framework of structural SVM and has reported a remarkable accuracy in a number of evaluations [14, 24, 31]. Its main rationale is to use patches of the video frames as the support vectors of an SVM, maintaining the set dynamically and within a "budget" so as to not compromise real-time operation. For these reasons, we have decided to adopt it as the base for our depth tracker. However, in initial tests with depth video, Struck showed some limitations that motivated the extensions that are the focus of this paper.

One of the main challenges in tracking from depth data is the design of features effective at tracking single targets through severe occlusions. In our experiments, existing features such as appearance histograms and Haar features did not seem as effective as they are on RGB data. For this reason, the first contribution of this paper is a novel depth descriptor based on the recently proposed *local depth patterns* [35]. Another significant challenge in tracking-by-detection is the efficient update of the target's classifier. As frames get processed, Struck applies heuristic rules to update the set of support vectors that define the detector. Such a set must respect a budget, i.e., an upper bound on the number of vectors, to ensure real-time performance. In an initial evaluation of Struck on depth videos, we have noticed a rapid proliferation of support vectors possibly due to the typical distributions and range of depth values. Therefore, in this paper, we propose to curb the number of support vectors by techniques inspired by *prototype selection* approaches [21]. These approaches have proven effective and flexible in many other domains and in this paper we show how they lead to improvements in tracking accuracy. As last contribution, we introduce a simple heuristic allowing to resolve target occlusions caused by static objects and other targets. An initial version of this work was published as a short conference paper at ACM Multimedia 2015 [2]: in this journal extension, we i) introduce the concept of prototype selection for the budget maintenance of the SVM classifier, ii) extend the experimental analysis to explore the main factors of influence for performance such as the search radius and budget size and iii) expand the experimental discussion with comparative cases of success and failure. In addition, all materials are presented together in a self-contained presentation.

Experiments have been carried out over two datasets: a simulated hospital environment created by these authors, and the recent Princeton Tracking Benchmark

(PTB) [25]. The first dataset consists of depth videos from staged visits to a patient lying on a hospital bed. The second dataset consists of 95 depth and RGB videos varying in target type (humans, animals and non-deformable objects), scene type, presence of occlusion and bounding box distribution. Figure 1 displays samples of depth frames from these two datasets. The experimental results show the remarkable improvements obtained with the proposed extensions and the competitive performance against state-of-the-art trackers.

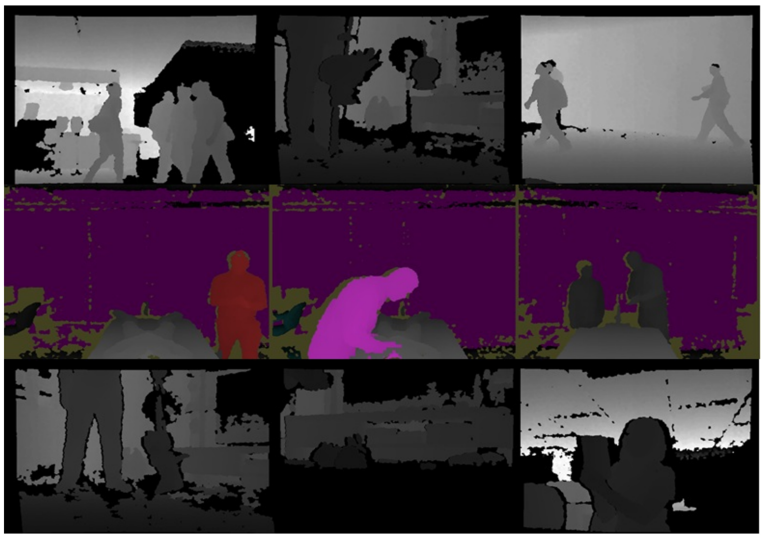


Fig. 1: Depth frame examples from our simulated hospital scenario (middle row, in pseudo-colors) and the Princeton Tracking Benchmark dataset (top and bottom rows).

2 Related work

Since their inception, consumer depth cameras have found increasing adoption in computer vision and multimedia. The widespread availability of depth data has led to the proposal of several dedicated features which, in most cases, are adaptations of pre-existing appearance features. For instance, [18] has proposed the HON4D feature which is a histogram of oriented 4D normals suitable for recognizing activities from depth video. Xia and Aggarwal have proposed a modification of the popular STIP detector and descriptor in [32]. In [15], the authors have proposed a simple range-depth feature computed around the location of skeletal joints. In [35], Zhao *et al.* have introduced a depth-based version of local binary patterns [19]. In addition to these works on features, depth data have found significant use as an additional modality for tracking: for instance, [33] leverages point-cloud clustering of depth pixels; [4, 17] use depth-based hierarchical clustering for tracking both individuals and groups; while [25] have used the depth information to resolve occlusions between targets. Following [25, 35], in this paper we propose to adopt a dense local descriptor aggregating depth values from the target's area.

In machine learning, an important problem is the learning of a classifier under a "budget" constraint, aiming at speeding up both the training and the classification [23]. This problem is even more urgent in tracking-by-detection where the online learning of the classifier must be performed within real-time constraints. In the case of SVM, the budget constraint limits the number of support vectors that can be used in the classifier. The general strategy for adding a support vector is to add it at its first appearance, while the decision to remove it is more critical and arbitrary. Therefore, several approaches have been proposed for removal: [7,28] remove samples based on a random selection; [9] removes the oldest support vectors; while [8, 29, 30] and also Struck [12] remove the support vector that causes the minimum L^2 norm change to the SVM primal model. In this work, we have decided to follow a different approach based on the notion of *prototype selection* [21]. The most common use of prototypes is for the embedding of non-vectorial objects such as strings, sets and graphs. Prototypes are typically selected from a given object set based on various centrality or uniformity measures [21]. In this paper, we have decided to evaluate three different prototype selectors to remove the support vector that is possibly the most redundant or otherwise an outlier inside the current set. The experimental results presented in Section 5 show the effectiveness of this approach.

3 The Struck tracker: overview

The Struck tracker was proposed in [12] as a principled improvement to tracking-by-detection approaches. It leverages the framework of structural SVM [27] to provide a prediction for the movement of a target, y, from its current position, p_t. By noting as x_t the frame at time t, Struck provides the prediction as the inference of a generalized linear model:

$$\bar{y} = \underset{y}{\operatorname{argmax}}\, w^\top \phi(x_t, y) \tag{1}$$

where feature function $\phi(x_t, y)$ computes a feature vector from a patch of pixels inside frame x_t centred at position $p_t + y$, and product $w^\top \phi(x_t, y)$ assigns it a score. In other terms, $w^\top \phi(x_t, y)$ is a joint model for the displacement and appearance of a target.

The challenge with maintaining the parameter vector, w, is that it has to be adapted at every new frame and in an unsupervised manner. This is achieved by framing this learning problem as a structural SVM objective and providing a heuristic for its online update. The structural SVM primal objective is expressed as:

$$\begin{aligned}
&\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \quad s.t. \\
&w^\top \phi(x_i, y_i) - w^\top \phi(x_i, y) \geq \Delta(y_i, y) - \xi_i, \\
&\quad i = 1 \ldots N,\ \forall y \in \mathcal{Y}
\end{aligned} \tag{2}$$

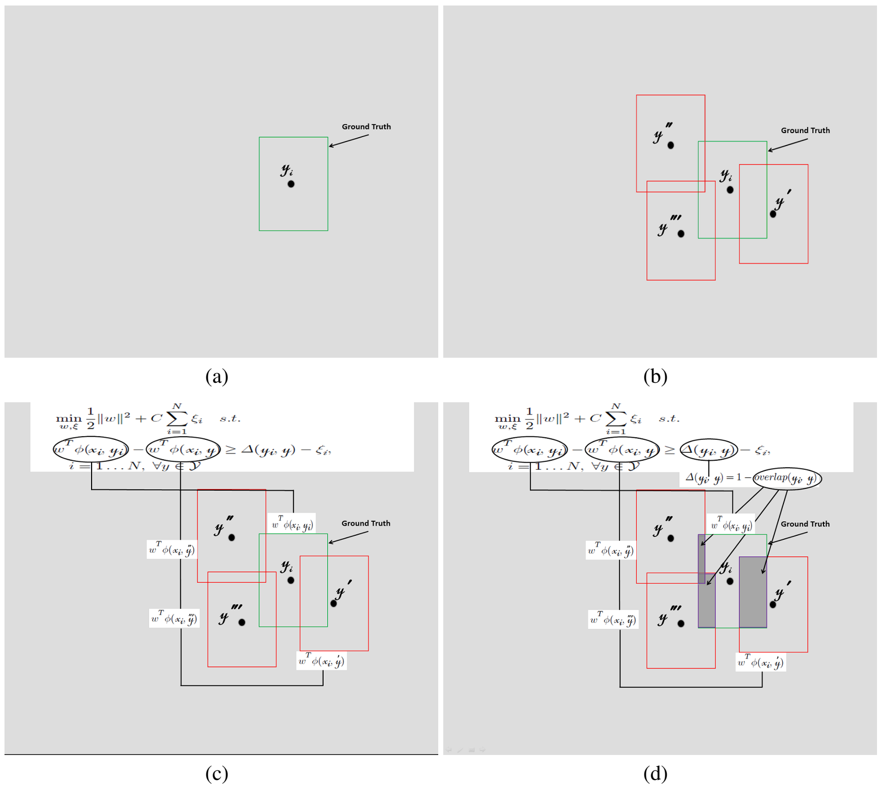

Ground Truth
y_i
y''
y'''
y'
(a)
(b)
(c)
(d)
$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad s.t.$
$w^T\phi(x_i, y_i) - w^T\phi(x_i, y) \geq \Delta(y_i, y) - \xi_i,$
$i = 1 \ldots N,\ \forall y \in \mathcal{Y}$
$\Delta(y_i, y) = 1 - overlap(y_i, y)$
$w^T\phi(x_i, y_i)$
$w^T\phi(x_i, y'')$
$w^T\phi(x_i, y''')$
$w^T\phi(x_i, y')$

Fig. 2: The main steps of Struck: a) the estimated ground-truth bounding box at frame i (a positive support vector); b) other bounding boxes around the ground truth (negative support vectors); c) the score, $w^\top \phi(x, y)$, of all bounding boxes is computed; d) the constraints in (2) impose that the score of the true displacement, y_i, is greater than that of any other displacement, $y \neq y_i$, by an amount set by the chosen loss function, $\Delta(y_i, y)$. At its turn, $\Delta(y_i, y)$ is chosen to be complementary to the overlap between bounding boxes y_i and y.

The objective in (2) is the standard SVM primal objective balancing an upper bound over the empirical loss, $\sum_{i=1}^{N} \xi_i$, with a regularization term, $\|w\|^2/2$. For brevity of notations, x_i notes the feature vector associated with displacement y_i. The constraints in (2) impose that the score assigned to the true displacement of the target, y_i, is greater than that assigned to any other displacement, $y \neq y_i$, by an amount decided by a chosen loss function, $\Delta(y_i, y)$. At its turn, the loss function is set to reflect the overlap between two bounding boxes centred, respectively, on the target's true location, y_i, and predicted location, y:

$$\Delta(y_i, y) = 1 - overlap(y_i, y) \qquad (3)$$

Figure 2 shows the main steps of Struck. The challenges with the SVM problem in (2) are that the actual ground truth is unknown, and that the model requires updating at every new frame. To this aim, Struck predicts the ground-truth labeling of a new sample, y_i, based on the current model, and uses a heuristic to select samples and labelings for the weight updates that we briefly describe in the following. As shown in [12], the primal SVM objective (2) can be turned into this equivalent dual problem:

$$\begin{aligned}
&\max_{\beta} -\frac{1}{2}\sum_{i,y}\sum_{j,y'}\beta_i^y\beta_j^{y'}\phi(x_i,y)^\top\phi(x_j,y') - \sum_{i,y}\beta_i^y\Delta(y_i,y)\\
&s.t.,\; i = 1\dots N:\\
&0 \leq \beta_i^{y_i} \leq C,\; \beta_i^y \leq 0\;\forall y \neq y_i,\; \sum_y \beta_i^y = 0
\end{aligned} \tag{4}$$

The maximization in (4) takes place over a vector of variables, β, where β_i^y denotes the variable for sample $i = 1\dots N$ and labeling $y \in \mathcal{Y}$. Such variables have different sign constraints: those for the estimated ground truth, $\beta_i^{y_i}$, are constrained to be positive, while the others, $\beta_i^y, y \neq y_i$, negative. Therefore, we refer to the respective vectors, (x_i, y_i) and $(x_i, y \neq y_i)$, as "positive" and "negative" vectors for short in the following.

The solver for (4) is an SMO (sequential minimal optimization) algorithm that sequentially selects a sample, x_i, and a pair of its β coefficients for update [20]. The two chosen coefficients, renamed as β_i^{y+} and β_i^{y-}, are modified, respectively, as $\beta_i^{y+} + \lambda$ and $\beta_i^{y-} - \lambda$, with $\lambda \geq 0$, so as to preserve the sum-to-zero constraint of (4). The choice of the two coefficients is performed by identifying the direction with the highest directional derivative of the objective function, in order to gain maximum benefit from the update. By noting as $g(y)$ the derivative along β_i^y, we select $y_+ = \text{argmax}_y\, g(y)$ and $y_- = \text{argmin}_y\, g(y)$: in this way, moving "toward" $g(y_+)$ by $+\lambda$ and "against" $g(y_-)$ by $-\lambda$ guarantees moving along the direction with the highest derivative for any possible pair of dimensions for this sample.

If the update modifies one of the β_i^y coefficients from an initial value of zero to a different value, its (x_i, y) vector is included in the current working set of support vectors. When the budget is eventually reached, an existing support vector is selected for removal so as to minimize the change in L^2 norm to the primal model, $w = \sum_{i,y}\beta_i^y\phi(x_i,y)$. The reader can refer to [5, 12] for further details.

The last component of the tracker is feature vector $\phi(x, y)$. As options, Struck provides:

– a 192-D Haar-like feature vector extracted from a grid centred at displacement y;
– a 256-D feature vector of spatially re-scaled raw pixels;
– a 480-D feature vector obtained from the concatenation of 16-bin intensity histograms computed on a four-level pyramid.

4 Extensions for depth tracking

In this section, we present the proposed extensions to Struck consisting of a novel depth descriptor (Section 4.1), a technique for support vector removal based on prototype selection (Section 4.2), and an occlusion handling procedure (Section 4.3).

4.1 Local depth features for tracking

In this work, we have decided to explore a local depth feature recently proposed for activity recognition in depth video. The feature, called local depth pattern (LDP), resembles the popular local binary patterns [19] in that it computes differences between cells of a local patch [35]. While this feature has proved effective for activity recognition, its performance for tracking cannot be anticipated since these two tasks rely on very different characteristics of the target.

To form our tracking feature (named LDP for tracking, or LDPT for short), we divide the target's bounding box into an $HD \times VD$ grid of LDPs. As values for HD and VD, we typically select 3 and 4, respectively. At its turn, each LDP contains a 3×3 grid of cells. Given that the bounding box has variable size, the size in pixels of the LDP and its cells adjust accordingly. The value of each LDP is obtained by concatenating the differences between the average depth of each of its cells with every other. Therefore, the total size of the LDPT feature is:

$$size(LDPT) = HD \times VD \times \binom{3*3}{2} \tag{5}$$

for a total of 432 dimensions. Algorithm 1 shows the detailed steps for computing an LDPT feature.

Algorithm 1 The algorithm for computing the proposed LDPT feature.

```
Input: Bounding box
Output: LDPT feature
 1: {initializes the LDPT feature to an empty set:}
    $LDPT = \emptyset$
 2: loop r = 1 : VD
 3:     loop c = 1 : HD
 4:         {initializes the LTD(r,c) descriptor to an empty set:}
            $LDP(r,c) = \emptyset$
 5:         loop i = 1 : 9
 6:             {loops over all cells in the LTD descriptor}
 7:             loop j = i + 1 : 9
 8:                 {computes the difference with every other cell:} $diff(i,j) = |avgdepth(i) - avgdepth(j)|$
                    $LDP(r,c) = concatenate(LDP(r,c), diff(i,j))$
 9:             end loop
10:         end loop
            $LDPT = concatenate(LDP(r,c))$
11:     end loop
12: end loop
```

4.2 Support vector removal based on prototype selection

Given a set of vectors or non-vectorial objects and a distance over them, prototype selection aims to find the sub-set of the objects that maximizes chosen properties such as the centrality in the set, uniform spread of the prototypes, proximity to the set boundaries and others. In our case, we aim to use prototype selection to determine the positive support vector (i.e., an estimated target) that is the most “disposable” according to these properties. In particular, we evaluate three different selection strategies, namely the *median*, *centre* and *marginal* support vectors [21]:

$$sv^*_{median} = \underset{i}{\operatorname{argmin}} \sum_j d(i,j) \tag{6}$$

$$sv^*_{centre} = \underset{i}{\operatorname{argmin}} \max_j d(i,j) \tag{7}$$

$$sv^*_{marginal} = \underset{i}{\operatorname{argmax}} \sum_j d(i,j) \tag{8}$$

where $d(i,j)$ represents the distance between two positive support vectors, sv_i and sv_j. The median support vector is defined as the support vector minimizing the sum of the distances from the remaining vectors, while the centre support vector minimizes the maximum distance from them. Both these selection strategies aim to remove a support vector that is “central” in the pool and therefore less likely to prove discriminative. On a different rationale, the marginal support vector maximizes the sum of the distances from the other vectors, and we remove it under the assumption that it may prove an outlier. The chosen distance also plays an important role in the selection: to this aim, we have evaluated three distances: 1) a simple Euclidean distance between the feature functions of support vectors sv_i and sv_j (d_u); 2) a distance weighted by the β coefficients of the two vectors (d_w); and 3) a distance weighted by their square root (d_s):

$$d_u(i,j) = \|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{9}$$

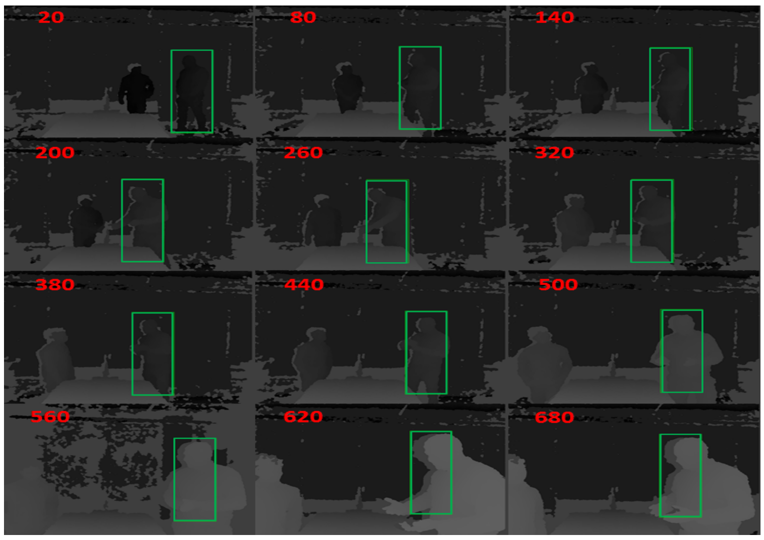
$$d_w(i,j) = \beta_i^{y_i} \beta_j^{y_j} \|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{10}$$

$$d_s(i,j) = \sqrt{\beta_i^{y_i} \beta_j^{y_j}} \|(\phi(x_i, y_i) - \phi(x_j, y_j\| \tag{11}$$

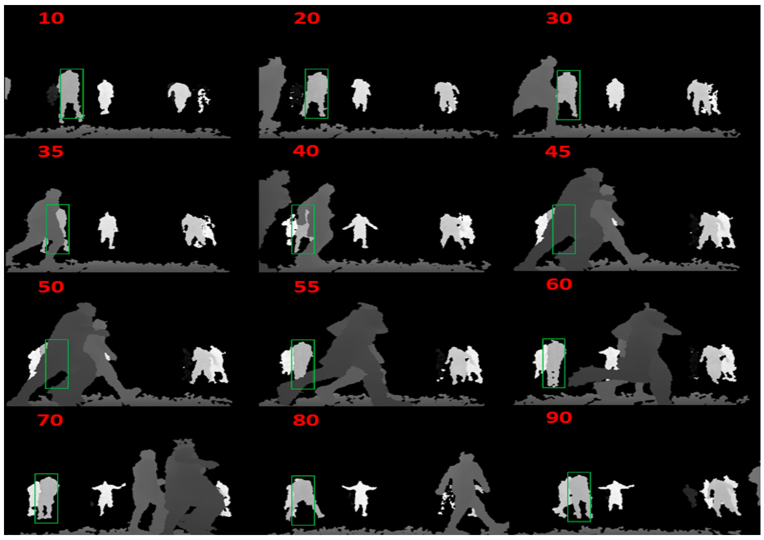
Algorithm 2 shows the main steps of the support vector removal procedure.

Algorithm 2 The proposed algorithm for support vector removal.

```
Input: Current $SV = \{sv_1, \ldots, sv_n\}$, with $n >$ size(budget)
Output: New $SV = \{sv_1, \ldots, sv_m\}$, with $m <=$ size(budget)
1: while size($SV$) $>$ size(Budget)
2: $SV_C = central\_positive\_support\_vector(SV)$
3: loop 1 : size(SV)
4:     $SV_N = corresponding\_negative\_support\_vector(SV_C)$
5:     $SV = SV \setminus \{SV_N\}$
6: end loop
7: $SV = SV \setminus \{SV_C\}$
8: end while
```



(a)



(b)

Fig. 3: Examples of occlusion handling in a) the hospital simulation and b) PTB datasets.

4.3 Occlusion handling

View occlusions from static objects and other targets are likely the main challenge of tracking. While the weakness of depth data is their lack of appearance features, their strength is the possibility to provide reliable target discrimination based on their distance from the camera. Therefore, in the proposed tracker we have built an occlusion detector that flags an occlusion whenever the depth of the candidate target, d_t, differs from its historical average, d_{avg}, more than a given threshold θ. This threshold is set in centimeters (to 50 cm) so as to have a uniform threshold value across the entire depth range. The measurements are computed at the centre of the respective bounding boxes and the historical average is maintained as a running average of update coefficient λ, updated only in the absence of detected occlusions:

$$occlusion = |d_t - d_{avg}| > \theta \tag{12}$$

$$d_{avg}(updated) = \begin{cases} \lambda d_t + (1-\lambda) d_{avg} & \text{if } occlusion = 0 \\ d_{avg} & \text{otherwise} \end{cases} \tag{13}$$

Figures 3.a and 3.b show examples of successful occlusion handling in a video from our hospital simulation dataset and a challenging basketball video from the PTB dataset. The videos with the full results can be visualized from Dropbox [1].

5 Experiments

5.1 Datasets

The proposed tracker has been evaluated both qualitatively and quantitatively using a hospital simulation dataset collected by these authors and the recent Princeton Tracking Benchmark (PTB) dataset [25]. Our hospital simulation dataset consists of 26 depth videos that stage simulated visits to a patient lying on a hospital bed. These videos are characterized by ample back-and-forth target movement and static occlusions and have been used for qualitative evaluation only [2]. The work on the hospital environment was motivated by a collaboration with clinical researchers from the Intensive Care Unit of Sydney's Royal Prince Alfred Hospital, who provided guidance on the simulation and set the privacy requirement for the video footage. The PTB dataset was released as part of an ICCV 2013 publication to offer a unified, challenging benchmark for tracking in RGB and depth data. It consists of 95 videos varying in target type (humans, animals and substantially rigid objects such as toys and human faces), level of background clutter (plain living rooms, cafes, sport courts etc), and type of occlusions (different durations, appearance changes during occlusions, similarity between targets and occluders etc). The dataset comes accompanied by an evaluation website [3] managed by the benchmark's authors which allows for an unbiased accuracy evaluation. The evaluation protocol considers three types of tracking errors: Type I errors that occur when the target is visible, but the tracker's output is far off from the target (wrong detections); Type II errors that occur when the target is invisible but the tracker still outputs a bounding box (false detections); Type

[1] https://www.dropbox.com/s/8codeji5lnzkg22/hospital.avi?dl=0,
https://www.dropbox.com/s/dzuock30489st1u/occlusion.avi?dl=0

[2] the dataset can be downloaded from https://drive.google.com/file/d/0B9cAe42oTaT_aUs4ckVrazQ1OXM/

[3] http://tracking.cs.princeton.edu/submit.php

Table 1: Accuracy comparison for the proposed tracker and other trackers on the Princeton Tracking Benchmark.

Algorithm	**target type**			**target size**		**movement**		**occlusion**		**motion type**		**average**
	human	**animal**	**rigid**	**large**	**small**	**slow**	**fast**	**yes**	**no**	**passive**	**active**	
Struck (Depth videos), Haar	0.31	0.32	0.36	0.29	0.36	0.36	0.32	0.21	0.49	0.36	0.32	0.34
Struck (Depth videos), raw pixels	0.34	0.44	0.42	0.37	0.41	0.44	0.37	0.29	0.54	0.43	0.37	0.40
Struck (Depth videos), histogram	0.38	0.46	0.44	0.45	0.40	0.51	0.39	0.31	0.57	0.52	0.38	0.44
HOG (Depth videos) from [25]	0.43	0.48	**0.56**	0.47	0.50	0.52	0.47	0.38	0.63	0.54	0.48	0.50
Proposed tracker (no occl. handling, no prototype selection)	0.39	**0.61**	0.54	0.46	0.51	**0.58**	0.45	0.32	**0.69**	0.56	0.46	0.51
Proposed tracker	**0.45**	0.59	0.54	**0.49**	**0.53**	0.57	**0.49**	**0.39**	0.68	**0.57**	**0.49**	**0.53**
Struck (RGB videos) from [25]	0.35	0.47	0.53	0.45	0.44	0.58	0.39	0.30	0.64	0.54	0.41	0.46
OF tracker (RGB videos) from [25]	0.47	0.47	0.63	0.47	0.47	0.57	0.52	0.47	0.62	0.63	0.49	0.53
RGBD tracker (RGB and depth) from [25]	0.74	0.63	0.78	0.78	0.70	0.76	0.72	0.72	0.75	0.70	0.82	0.74

Table 2: Comparison of average accuracy with different prototype selection techniques and distances.

Prototype selector	**Euclidean** (d_u)	**Weighted** (d_w)	**Square root** (d_s)
Median	0.49	0.51	0.52
Centre	0.49	0.52	**0.53**
Marginal	0.48	0.49	0.50

III errors that occur when the target is visible but the tracker fails to produce any output (missed detections). Accuracy figures are divided by target type, target size, movement, occlusion and motion type.

5.2 Experimental results

Our experiments aim to compare the proposed tracker with the original Struck tracker and other state-of-the art trackers. The qualitative evaluation on the hospital simulation dataset is generally very positive, with the target (a visiting clinician) successfully tracked in all videos. The original Struck tracker instead tends to lose the target in the presence of large static occlusions.

The quantitative evaluation on the PTB dataset provides the test-bed for a rigorous and current performance analysis. Table 1, top part, reports the accuracy comparison for the proposed depth tracker against other trackers using only depth data. These include Struck with different types of features and a tracker based on HOG features [25]. The results in Table 1 show that the proposed tracker outperforms the other trackers in 7 categories out of 11. The introduction of the LDTP feature alone achieves an average accuracy improvement of 7 percentage points over Struck with the best feature (histogram; 0.51% vs 0.44%). The addition of the proposed prototype selection approach together with the occlusion handling heuristic achieves a further improvement of 2 percentage points.

For comparison, the bottom part of Table 1 reports the performance of trackers using RBG data. The proposed tracker outperforms Struck operating on RGB data in almost every category (10 out of 11). This result is remarkable in that it shows that depth tracking with suitable features can outperform RGB tracking at a parity of targets and scenes. In turn, this proves that depth tracking is a viable approach to tracking under privacy-preserving operating conditions. It is also important to add that the performance of Struck on RGB data was reported in [25] as the best out of a pool of popular trackers including TLD [16], CT [34], MIL [3], semi-B [10] and VTD [13]. The only RGB tracker that outperforms our depth tracker in a few categories is the tracker proposed by the authors of the benchmark itself (*OF tracker*, Table 1). Remarked improvements over depth tracking alone is only achieved by fusion of depth and RGB information (*RGBD tracker*, Table 1).

Figure 4 shows cases of success and failure for the proposed tracker and the original Struck tracker. Figure 4.a shows a case where Struck wrongly swaps the target with another passer-by due to a temporary occlusion. In the same case (Figure 4.b), the proposed tracker continues to track correctly thanks to the effective detection and handling of the occlusion. Figure 4.a shows a failure from the proposed tracker due to the sudden fast motion of the target (a small dog). Since the model is a joint model for the movement and appearance of the target, abrupt changes are the main potential cause of failure. The same sequence shows that the proposed tracker withstands another major occlusion around frame 30.

To explore the sensitivity to parameters, Figure 5.a compares the accuracy achieved using different features as a function of the *search radius* for the target. The search radius determines the maximum distance over which the displacements are computed,

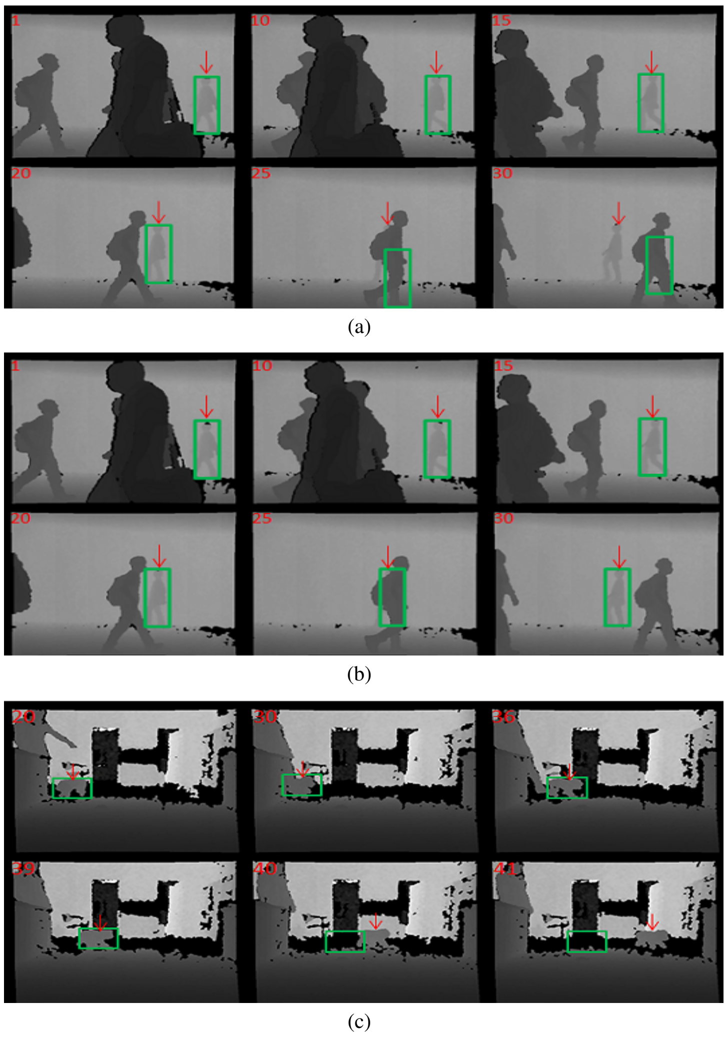
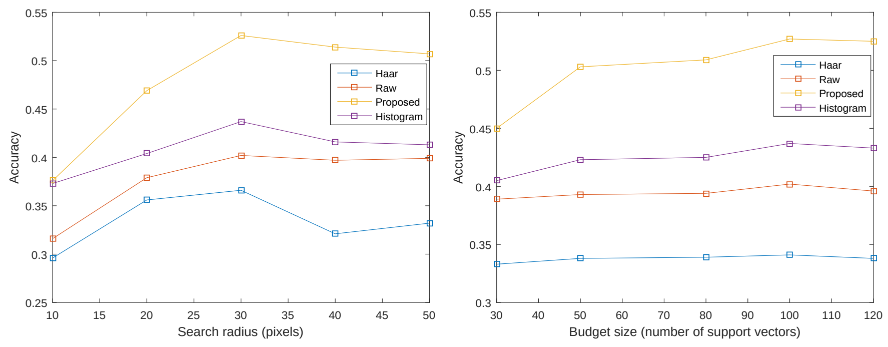


Fig. 4: Cases of success and failure for the proposed tracker and the original Struck tracker.

and in this experiment it has been made vary between 10 and 50 in steps of 10 while leaving all the other parameters unchanged. The plot shows that the highest accuracy for every value of the search radius is achieved with the proposed LDPT features. The accuracy shows no increase beyond a radius of 30 which is a desirable result given that the computational time increases with larger radii. Likewise, Figure 5.b compares the accuracy achieved using different features as a function of the *budget size*. In this experiment, the budget size has been made vary between 30 and 120 in steps of 10 while, again, leaving all the other parameters unchanged. Figure 5.b shows that



(a) (b)

Fig. 5: Comparison between the proposed tracker and the original Struck tracker with various features; a) by varying the search radius; b) by varying the budget size.

also in this case the proposed features achieve the highest accuracy for every value in the range, with a maximum for a budget size of 100. These results further validate the usefulness and operational robustness of the proposed approach.

Eventually, Table 2 shows a comparison of the average accuracy for different prototype selection techniques in combination with different support vector distances. The results show that the use of weighted distances is generally preferable and that the distance with the square root of the weights' product, d_s, achieves the best accuracy in combination with the centre-based prototype selection.

6 Conclusion

In this paper, we have proposed various extensions to the popular Struck tracker to improve its tracking performance on depth videos. The extensions include a dedicated depth feature based on local depth patterns, a heuristic for handling occlusions in depth frames, and a technique for maintaining the number of support vectors within a given budget to limit computational costs. The experimental results over the challenging Princeton Tracking Benchmark show that:

- the lack of appearance information typical of depth videos has not proved a major impediment for achieving an interesting tracking accuracy. Rather, in the experiments tracking from depth data has outperform tracking from RGB data at a parity of targets and scene (Table 1);
- the proposed tracker has achieved a higher accuracy than existing results on depth data in 7 categories of the benchmark out of 11, and on average (Table 1);
- The proposed extensions have lead to an average improvement of 9 percentage points over Struck with the best feature (Table 1). Amongst the various prototype selection methods and distances, centre-based selection and the distance weighted

by the square root of the vectors' weights have reported the best accuracy (Table 2).

On a more general note, the depth frames from our simulated hospital scenario displayed in Figures 1 and 3 confirm that depth data do not disclose significant identification clues of the targets. We expect this to prove a key factor for acceptance of real-time monitoring in privacy-sensitive environments.

References

1. S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.
2. S. Awwad, F. Hussein, and M. Piccardi. Local depth patterns for tracking in depth videos. In *23rd ACM International Conference on Multimedia*, MM '15, pages 1115–1118, 2015.
3. B. Babenko, Ming-Hsuan Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 983–990, 2009.
4. F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti. Fast and robust multi-people tracking from rgb-d data for a mobile robot. In *Intelligent Autonomous Systems 12*, pages 265–276, 2013.
5. A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with larank. In *24th International Conference on Machine Learning*, ICML '07, pages 89–96, 2007.
6. M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, Sept 2011.
7. G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2-3):143–167, 2007.
8. L. Cheng, S. Vishwanathan, D. Schuurmans, S. Wang, and T. Caelli. Implicit online learning with kernels. *Advances in Neural Information Processing Systems*, 19:249, 2007.
9. O. Dekel, S. Shalev-Shwartz, and Y. Singer. The forgetron: A kernel-based perceptron on a budget. *SIAM Journal on Computing*, 37(5):1342–1372, 2008.
10. H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *10th European Conference on Computer Vision: Part I*, ECCV '08, pages 234–247, 2008.
11. Y. Guo, Y. Chen, F. Tang, A. Li, W. Luo, and M. Liu. Object tracking using learned feature manifolds. *Computer Vision and Image Understanding*, 118:128–139, 2014.
12. S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *IEEE International Conference on Computer Vision (ICCV)*, pages 263–270, 2011.
13. J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, pages 1269–1276, 2010.
14. X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel. A survey of appearance models in visual object tracking. *ACM transactions on Intelligent Systems and Technology (TIST)*, 4(4):58, 2013.
15. C. Lu, J. Jia, and C.-K. Tang. Range-sample depth feature for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 772–779, 2014.
16. M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *IROS*, pages 3844–3849, 2011.
17. M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with rgb-d data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pages 2101–2107, 2012.
18. O. Oreifej and L. Zicheng. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 716–723, 2013.
19. M. Pietikäinen, G. Zhao, A. Hadid, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Number 40 in Computational Imaging and Vision. Springer, 2011.
20. J. C. Platt. Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, 1999.
21. K. Riesen and H. Bunke. *Graph classification and clustering based on vector space embedding*. World Scientific, 2010.

22. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, 2011.
23. K. Singer. Online classification on a budget. *Advances in Neural Information Processing Systems*, 16:225, 2004.
24. A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: an experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.
25. S. Song and J. Xiao. Tracking revisited using RGBD camera: unified benchmark and baselines. In *IEEE International Conference on Computer Vision (ICCV)*, pages 233–240, 2013.
26. S. Tran and L. Davis. Robust object tracking with regional affine invariant features. In *IEEE 11th International Conference on Computer Vision*, ICCV, pages 1–8, 2007.
27. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.
28. S. Vucetic, V. Coric, and Z. Wang. Compressed kernel perceptrons. In *Data Compression Conference, 2009. DCC'09.*, pages 153–162, 2009.
29. Z. Wang, K. Crammer, and S. Vucetic. Multi-class pegasos on a budget. In *27th International Conference on Machine Learning (ICML-10)*, pages 1143–1150, 2010.
30. J. Weston, A. Bordes, and L. Bottou. Online (and offline) on an even tighter budget. In *10th International Workshop on Artificial Intelligence and Statistics*, pages 413–420, 2005.
31. Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 2411–2418, 2013.
32. L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2013, pages 2834–2841, 2013.
33. H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.
34. K. Zhang, L. Zhang, and M.-H. Yang. Real-time compressive tracking. In *European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 864–877, 2012.
35. Y. Zhao, Z. Liu, L. Yang, and H. Cheng. Combining rgb and depth map features for human activity recognition. In *2012 Asia-Pacific Signal Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–4, 2012.