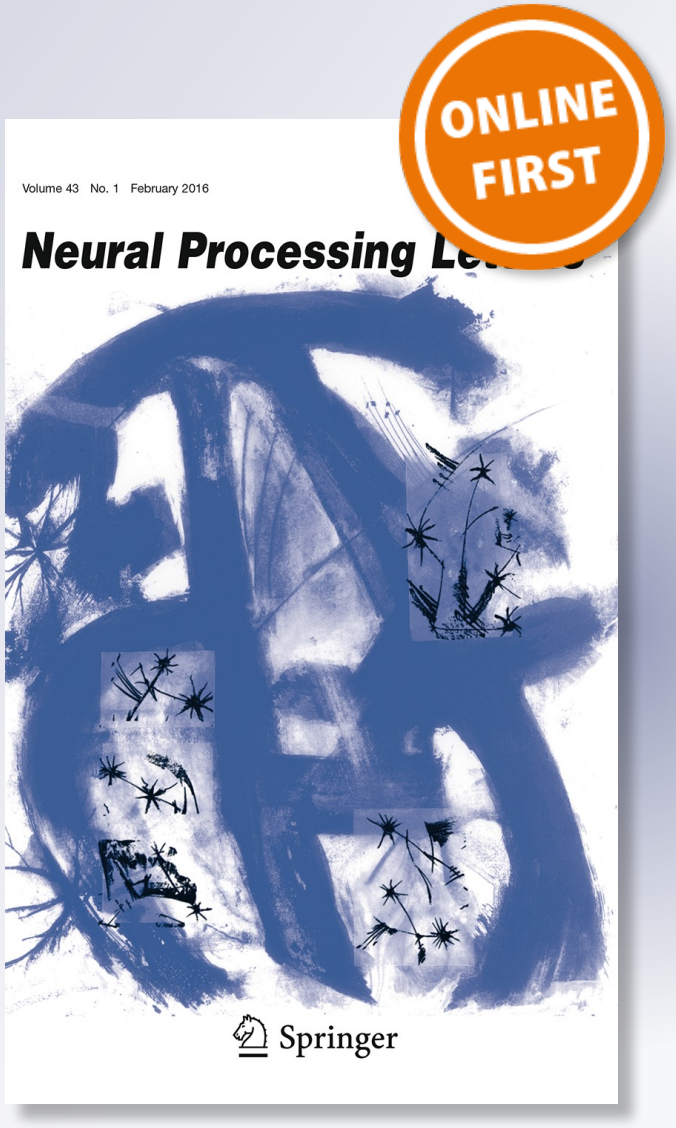
Online Depth Image-Based Object Tracking with Sparse Representation and Object Detection

Wei-Long Zheng, Shan-Chun Shen & Bao-Liang Lu

Neural Processing Letters

ISSN 1370-4621

Neural Process Lett
DOI 10.1007/s11063-016-9509-y



Springer

Neural Process Lett
DOI 10.1007/s11063-016-9509-y

Online Depth Image-Based Object Tracking with Sparse Representation and Object Detection

Wei-Long Zheng[1] · **Shan-Chun Shen**[1] · **Bao-Liang Lu**[1]

Abstract Online object tracking under complex environments is an important but challenging problem in computer vision, especially for illumination changing and occlusion conditions. With the emergence of commercial real-time depth cameras like Kinect, depth image-based object tracking, which is insensitive to illumination changing, gains more and more attentions. In this paper, we propose an online depth image-based object tracking method with sparse representation and object detection. In this framework, we combine tracking and detection to leverage precision and efficiency under heavy occlusion conditions. For tracking, objects are represented by sparse representations learned online with update. For detection, we apply two different strategies based on tracking-learning-detection and wider search window approaches. We evaluate our methods on both the subset of the public dataset Princeton Tracking Benchmark and our own driver face video in a simulated driving environment. The quantitative evaluations of precision and running time on these two datasets demonstrate the effectiveness and efficiency of our proposed object tracking algorithms.

Keywords Object tracking · Depth image · Sparse representation · Object detection

1 Introduction

Online object tracking plays a critical role in many computer vision applications such as activity recognition, human-computer interaction and automated surveillance [29,36]. While much progress has been achieved in recent years, online object tracking is still an important but challenging task, especially under complex environments. The difficulties of object tracking

✉ Bao-Liang Lu
bllu@sjtu.edu.cn

Wei-Long Zheng
weilonglive@gmail.com

[1] Center for Brain-like Computing and Machine Intelligence, Department of Computer Science and Engineering, Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, 200240 Shanghai, China

Published online: 19 February 2016

account for intrinsic (e.g., nonrigid object structure, complex shapes and pose variation) and extrinsic factors (e.g., illumination changing and heavy occlusions) [36].

To address these issues, numerous approaches have been proposed [1,7,31]. Basically, object tracking contains two important components: the appearance modeling and target searching. Tracking algorithms use the appearance models to discriminate the target object and the background and adaptively update the appearance models online. For the appearance modeling, it is very important to balance the invariance and the discrimination. The appearance modeling should be adaptive to the intrinsic variations such as pose changes and shape changes and robust to the extrinsic variations such as varying illumination and heavy occlusions. For target searching, the computation complexity should be considered to find the sampled candidates that best match to the target object in real-time.

Most of visual object tracking are based on color images for their rich information, which is helpful to represent target objects. Generally speaking, there are four types of common visual features extracted from color images: color, edges, optical flow and texture [2,6, 16,20,21]. Various algorithms based on these features perform well in some constrained situations. Pérez et al. introduced a Monte Carlo tracking technique within a probabilistic framework using the principle of color histogram distance [20]. The RGB (red, green, blue) color space is usually used to represent color. However, the differences between the colors in the RGB space do not correspond to the color differences perceived by humans [18]. Moreover, color features are easily influenced by illumination. Object boundaries often locate the areas where image intensities strongly change. Paragios et al. proposed the variational framework for detecting and tracking multiple moving objects using edge detection approach [17], which uses a statistical method based on a mixed model. This framework is robust to illumination changes, but when occlusions occur, the edge based method would lose target.

Projection of the 3D world on a 2D image causes loss of information [36]. The performance of the color image-based tracking methods usually decreases a lot with varying complex illumination. Meanwhile, with the emergence of commercial depth sensors, such as Microsoft Kinect and PrimeSense, it is easier to collect depth images. Unlike color image, the pixel of depth image represents the distance between the points of object and camera, instead of intensity. Depth images can provide additional valuable information for improving the performance of tracking and detection. Moreover, depth images are insensitive to varying illumination, which bypass the drawbacks of color images. Depth image-based approaches gain more and more attention in recent years. In [3], Cai et al. developed a regularized maximum likelihood deformable model fitting algorithms for face tracking. Cao and Lu proposed an online depth image based face tracking method for driving fatigue detection on the assumption that face shape is an ellipse [4]. In order to standardize uniform evaluation criteria for comparing different kinds of algorithms, Song and Xiao constructed one unified benchmark dataset called Princeton Tracking Benchmark (PTB) [25]. Another publicly available RGBD People Dataset [26] contains one sequence with 1,132 frames with people moving. In this paper, we focus on depth image-based object tracking to tackle the problem of illumination changing.

Another critical challenge for object tracking is occlusion. To address the challenges of occlusion is technically difficult in various applications of computer vision [14,34,35]. This difficult is mainly due to the unpredictable nature of the error incurred by occlusion [28]. It can corrupt the representations of the target and introduce unpredictable noise. Numerus approaches are proposed to address this problem [14,34,35]. Among these approaches, the sparse representation based methods have advantages in efficiency and robustness [28,32, 33,37]. The basic idea of sparse representation is that a test image can be represented as a

linear combination of basis images (dictionary), while the weights are relatively sparse. The sparse representation of an occluded test image is a sparse linear combination of basis images plus errors due to occlusion, which can separate the occlusion components from the identity components.

Recently, sparse representation is considered as an efficient solution to object tracking problems. The sparse representation approach can be categorized into three classes: (1) appearance modeling based on sparse coding (AMSC); (2) target searching based on sparse coding (TSSR); (3) the combination of both. Jia et al. proposed a structural local sparse coding model [12]. Mei and Ling solved most challenges like occlusion through a set of positive and negative trivial templates [15]. By transferring object tracking problem to a sparse approximation problem, they proposed a robust algorithm. Wang et al. proposed an online object tracking algorithm with sparse prototypes. Their approach accounts explicitly for data and noise [27]. The existing studies mentioned above have demonstrated that sparse coding is a good solution to color image based object tracking. However, to our best knowledge, limited studies about depth image based tracking using sparse representation are reported in the literature. In this study, we apply sparse representation method to depth image based tracking algorithm in order to tackle the problems of occlusion.

There have been several methods proposed for depth image-based object tracking and detection in the literature. For example, Colombo et al. proposed a feature-based approach to detect salient face features, such as eyes and nose, through an analysis of the curvature of the surface [5]. To handle the noisy input depth data, Cai et al. developed a regularized maximum likelihood deformable model fitting algorithm for 3D deformable face tracking [3]. Xia et al. presented an approach for human detection using a 2D head contour model and a 3D head surface model. They utilized both the edge information and the depth change information in depth images [30].

In this paper, we propose a general object tracking method based on single depth image, which is robust to occlusion and illumination changes. In comparison with color image based methods, our algorithm is less influenced by illumination changes. With sparse coding representation, we can keep tracking the target object until the occlusion area reaches 50 % of the whole target object. This study is the extension of our previous work [23]. We introduce the tracking-learning-detection (TLD) framework to our methods by restarting with detection when failing the track. Moreover, we design a simulated driving environment and collect the driver face videos. We evaluate our methods on both the subset of the benchmark dataset and our own dataset with measures of precision and response time. The promising results demonstrate the efficiency of our methods in real-world applications. Compared with the existing studies, this paper has three major contributions. First, unlike the existing ways proposed for face tracking or detection, we do not assume that the tracking targets are specific. The proposed approach can achieve superior results for generic object tracking. Second, the proposed models in this paper can be updated and trained adaptively, rather than base on some heuristical prior knowledge. Third, most of these previous studies cannot deal with the occlusion situations. On the contrary, we introduce a sparse representation method and employ different update strategies to tackle the occlusion problems.

The rest of this paper is organized as follows. In Sect. 2, we give a detailed descriptions about our tracking method. In Sect. 3, we present qualitative and quantitative results of our tracker on the subset of the public available dataset called Princeton Tracking Benchmark as well as the face videos we collected from a simulated driving environment. Finally we make a conclusion of this paper in Sect. 4.

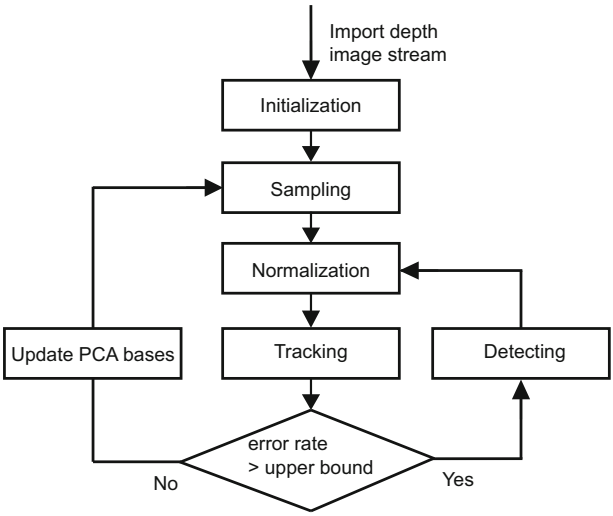


Fig. 1 Workflow of our proposed depth image-based tracking algorithm with sparse representation and object detection

2 Object Tracking with Sparse Representation and Object Detection

Here, we present a novel general framework for object tracking with sparse representation and object detection. Figure 1 shows the workflow of our object tracking algorithm. Firstly, we initialize the tracker by manually calibrating target position, computing PCA bases and setting other parameters such as patch size and bases number. Secondly, we sample the original depth image according to sampling parameters. The samples are size-adjustable to suit for demand of object front-back moving. To speed up the proposed tracking algorithm, we transfer all the samples to the same size patches. By evaluating every patch, we find the patch with the highest posterior probability and return its location as the target. During the process, we compute the occlusion rate by using coefficients of trivial templates. If the occlusion rate exceeds the upper bound, we discard the result and regard it as losing target. Then we startup the detection module with two different strategies: the TLD approach and wider search window approach. If not, we update the PCA bases and go to the next loop.

2.1 Alternative Box Sampling

In sampling stage, we aim to select the candidate patches of target object with respect to current frame. In this paper, we apply an affine image warp to model the taget motion between two consecutive frames. In detail, there are five parameters $(x, y, \alpha, \beta, \theta)$ of the affine transform [19] in tracking sampling stage: x and y denote transformations in plane, α and β are scale variations, and θ is angle rotation. Alternative boxes are uniformly distributed around the target. To adapt to the characteristics of the depth map, we set α and β a little bigger. But too big α and β mean more alternative boxes to be computed and slower processing speed. To speed up the tracking algorithm, we transfer all the samples to the same size by interpolation. The size of candidate patches is fixed as 32×32, which is the trade-off between speed and accuracy. During the tracking process, the tracking sampling alternative boxes are adjustable. Figure 2 shows the candidate patches in the sampling stage.

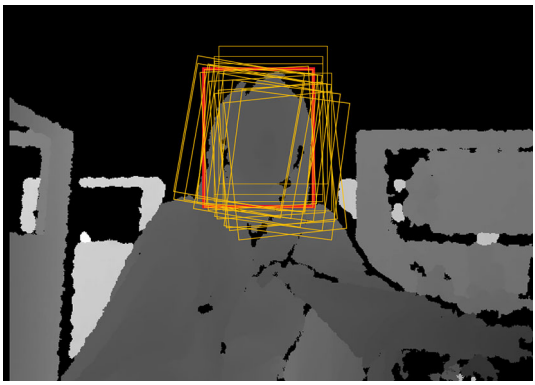


Fig. 2 The candidate patches in the sampling stage

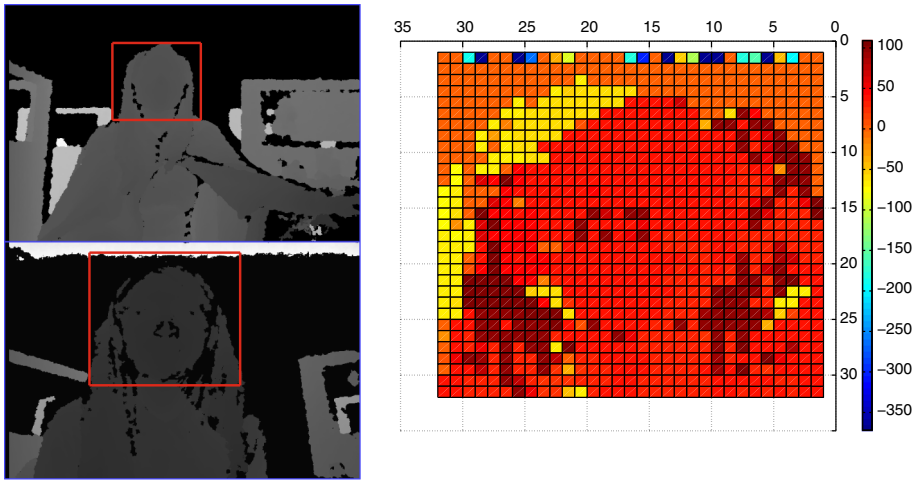


Fig. 3 Pixel value shifts of two frames. *Left top* is the face far from the camera, *left down* is the nearer one, and *right* is the pixel value difference of two patches after interpolating

2.2 Depth Image Normalization

The value of pixels in depth image represents the distance between camera and the point on object. The whole depth image represents the shape of the target. Transformation in the same depth can remain both the same pixel values and pattern. But once target object moves forward or backward, the pattern is remained but the pixel values shift. The pixel values of target objects in color images remain similar when moving forward or backward. This is why color image based methods can utilize absolute pixel values to extract the patterns. However, for depth image, this assumption can not be satisfied.

As shown in Fig. 3, in order to extract depth-invariant patterns, we should normalize the patches with relative pixel values to eliminate the offset. The most common method of normalization is min-max normalization. However, if there are much noises, they often deviate from average values that are the outliers from valid values. In this case, the normalization results are extreme minimum or maximum. The noise in depth images limits the performance of min-max normalization. Therefore, the min-max normalization is not suitable for depth

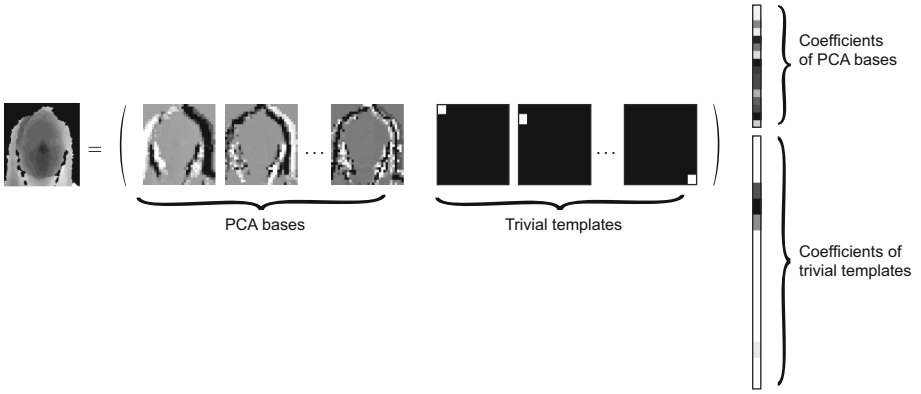


Fig. 4 The framework of sparse representation with subspace learning for object tracking used in this study. Each target appearance can be represented with PCA bases and trivial templates that account for occlusion

images. Here, we adopt the sigmoid filter to normalize the images patches. The sigmoid function is a S-type function as follows,

$$y = f(x) = \frac{1}{1 + e^{\frac{x-\mu}{\sigma}}} \tag{1}$$

In our method, σ equals to 1 and μ is set to the median of each patch. The reason of setting the value of μ to the median of a patch is less sensitive to noise. The sigmoid normalization is insensitive to outliers of the patches in depth images.

2.3 Sparse Representation with Subspace Learning for Object Tracking

Object tracking via incremental subspace learning [9,21] and sparse representation [12,15, 27] has gained much progress and attracted much attention in recent years. However, there are some limitations with these two approaches. For incremental subspace learning approach such as the incremental visual tracking (IVT) method [21], the PCA subspace based representation scheme is sensitive to partial occlusion. For original sparse representation, it does not exploit the rich information that can be compactly represented in subspace learning. Additionally, it requires much computational complexity to solve ℓ_1 minimization problem, which limits the performance of tracking. In this paper, we combine the characteristics of both incremental subspace learning and sparse representation for modeling object appearance based on the methods proposed by Wang et al. [27]. As shown in Fig. 4, each target appearance can be represented with PCA bases and trivial templates that account for occlusion.

In the framework of sparse representation with subspace learning, tracking problem is casted to finding the most likely patch among candidates by

$$y = Uz + e = [U\ I]\begin{bmatrix} z \\ e \end{bmatrix} = Bc \tag{2}$$

where y indicates an observation vector, U denotes a matrix of basis vectors, z represents coefficients of basis vectors, and e is the coefficient of trivial templates. As shown in Fig. 4, the bases consist of a number of PCA basis vectors and trivial templates. We solve Eq. (2) via ℓ_1 minimization:

$$\min_{z,e} \frac{1}{2} \parallel y - Uz - e \parallel_2^2 + \lambda \parallel e \parallel_1 \tag{3}$$

where $\|\cdot\|_2^2$ and $\|\cdot\|_1$ are the ℓ_2 and ℓ_1 norm forms, respectively , and λ is a tradeoff parameter. In Eq. (3), the first term denotes the reconstruct error and the second term represents the error term with arbitrary but sparse noise. For sparse approximation, we use ℓ_1-norm instead of ℓ_0-norm to reduce the computational complexity. In Eq. (3), the coefficients for trivial templates should be sparse, while the coefficients for PCA basis vectors are not sparse since PCA basis vectors are orthogonal. Wang et al. [27] presented an iteration algorithm to compute optimal z and e for each candidate.

Given e_{opt}, the problem of Eq. (3) is equivalent to the minimization of $J(z)$, where $J(z) = \frac{1}{2} \| (y - e_{opt}) - Uz \|_2^2$. The solution of this least squares problem is $z_{opt} = U^\top (y - e_{opt})$. If z_{opt} is given, the minimization of Eq. (3) is equivalent to the minimization of $G(e) = \frac{1}{2} \| e - (y - Uz_{opt}) \|_2^2 + \lambda \| e \|_1$. The global minimization can be obtained as $e_{opt} = S_\lambda(y - Uz_{opt})$ through convex optimization, where $S_\tau(x)$ is a shrinkage operation defined as $S_\tau(x) = sgn(x) \cdot (|x| - \tau)$ [8]. The details of the algorithm solving Eq. (3) are presented in Algorithm 1. After getting the optimal z and e for each candidate, the object tracking problem is transferred to a statistical inference problem. In this framework, the proposed algorithm models partial occlusion explicitly and hence is robust to occlusion. Moreover, this method utilizes subspace representation with less computational complexity.

In order to adaptively handle appearance changes of target objects, it is necessary to update the observation models when tracking. Some target objects can not be utilized to update the observation models directly due to occlusion. Here, we introduce a parameter η to represent the degree of occlusion with the trivial coefficients since the trivial templates account for occlusion. We compute the ratio η between the number of non-zero trivial coefficients and the total number of trivial coefficients. We employ different update strategies according to the degree of occlusion as

$$\eta \begin{cases} < \text{lower threshold: little occlusion, full update} \\ > \text{higher threshold: much occlusion, no update} \\ \text{others: partial occlusion, partial update} \end{cases} \tag{4}$$

Here, we use the incremental PCA method [21] to update the observation model.

2.4 Restarting Object Tracking by Object Detection

As mentioned in the preceding section, we estimate the degree of occlusion by the parameter η. We set a lower threshold and a higher threshold for η. Different values correspond to different tracking results. If η is larger than higher threshold, we view this situation as tracking failure, and we startup the detecting module. Here, we startup the detection module with two different strategies: the TLD approach and wider search window approach.

Algorithm 1 Algorithm for Computing z_{opt} and e_{opt}

Input : An observation vector y, orthogonal basis vectors U, and a small parameter λ.
1: Initialize $e_0 = \mathbf{0}$ and $i = 0$
2: Iterate
3: Obtain z_{i+1} via $z_{i+1} = U^\top (y - e_i)$
4: Obtain e_{i+1} via $e_{i+1} = S_\lambda(y - Uz_{i+1})$
5: $i \leftarrow i + 1$
6: Until convergence or termination
Output : z_{opt} and e_{opt}

Tracking-Learning-Detection (TLD) Approach The framework of TLD was proposed by Zdenek et al. [13] for long-term tracking. The key problem of long-term tracking is the detection of object when it reappears in the images. The TLD framework consists of three modules: detection module, tracking module, and learning module. The tracking module corresponds to tracking target objects based on the relations between two consecutive frames. The detection module aims to detect target objects when tracking fails due to heavy occlusion or disappearance from images. Unlike tracking module, the assumption of detection module is that two consecutive frames are mutually independent. The learning module aims to update and record appearance models of target. The original TLD approach is proposed for color images. Here, we introduce the TLD framework to re-detect target objects in depth images when tracking failures occur.

Wider Search Window Approach On the assumption that when losing the target we still can find it in a wider scope centered on the original position, the sampling scope spreads to three times of the size of the original one. After sampling stage, the rest stages are the same as tracking method. We call this approach as wider search window (WSW) approach. By computing the coefficients of bases and solving the Bayesian task, we find the most likely patch among candidates. We compute error ratio η in detecting module. If η becomes lower than its upper bound, we start the tracking module.

3 Experiments and Results

3.1 Experiment Setup

Our algorithm is implemented in Matlab on a Triple-Core Processor 2.10GHz with 6GB memory. The speed of our algorithm is related to sampling number. More sampling candidate boxes would slow down the processing speed. As a trade-off between computational complexity and accuracy, sampling number is set to 600. To speed up the tracking algorithm, we transfer all the samples to the same size by interpolation. The size of candidate patches is fixed as 32×32. For the parameters of the models, the variance of affine parameters is set to [4, 4, 0.02, 0.1, 0.01, 0.001]. The maximum number of basis vectors is 10 and the number of collected samples for update is 5. The regularization parameter λ for ℓ_1-norm is set to 0.05. The high and low thresholds for model update are 0.6 and 0.2, respectively. We evaluate our methods on both subset of the public dataset Princeton Tracking Benchmark and our own driver face videos in simulated driving environments. We use two accumulative metrics for all frames of videos to evaluate the performance. The first one is the center position error (CPE), which is the Euclidean distance between centers of output bounding boxes and the ground truth. This metric shows how close the tracking results are to the ground truth. The other is the frame per second (FPS), which is to measure the processing speed of different approaches. This metric shows the computational complexity of different methods.

3.2 Experimental Results on the Public Dataset PTB

Song et al. [25] constructed a RGBD dataset of 100 videos, named as Princeton Tracking Benchmark (PTB), which includes deformable objects, various occlusion conditions, moving camera and different scenes. In this study, we select eight typical sequences in PTB as test data to evaluate the performance of our approach. Table 1 shows the detailed descriptions for the selected sequences.

Table 1 The detailed descriptions of the selected sequences from PTB

Test sequence	Frame number	Challenge
Cup	368	Move forward and backward
Ball	117	Illumination change
Bear	281	Severe occlusion
Child	164	No-rigid object tracking
Face_move1	469	Face moving
Face_occ2	387	Partial face occlusion
Face_occ3	262	Full face occlusion
Face_turn	600	Face turning

Fig. 5 The selected image sequences of bear, cup, child and ball are listed from *top* to *bottom* (only the RGB images are presented.)

Table 2 Experimental results with the mean center position errors (CPE) for color image-based and depth image-based methods.

Test sequence	Color image	Depth image (WSW)	Depth image (TLD)
Cup	13.93	**12.83**	13.15
Ball	263.60	14.49	**13.26**
Bear	192.99	46.23	**27.98**
Child	**47.57**	135.34	82.49
Mean	129.51	52.22	**34.22**

WSW and TLD denote two different detection strategies, wider search window and tracking-learning-detection, respectively
The bold numbers denote the best results for each test sequence. Smaller errors indicate higher accuracies

We first evaluate our approach on arbitrary object tracking. Figure 5 shows the selected image sequences of bear, cup, child and ball. Table 2 shows the mean center position errors of different sequences for color image-based and depth image-based methods. From the

Table 3 Experimental results of tracking speed measured by frame per second (FPS) for different sequences

Test sequence	Depth image (WSW)	Depth image (TLD)
Cup	**4.26**	1.78
Ball	**4.06**	1.19
Bear	**3.46**	1.09
Child	**1.79**	0.69
Mean	**3.39**	1.18

experimental results, we can see that the depth image-based methods can achieve better performance than the color image-based approach except for child sequences.

In cup sequences, the challenge is that target object is moving forward and backward. The mean CPEs of color-based and depth-based methods with WSW and TLD are 13.93, 12.83, and 13.15, respectively. The depth-based methods achieve slightly better performance than the color-based one.

In ball sequences, the challenge is that the ball rolls around with illumination changes. In color image sequence, the color-based method loses target in the fortieth frame as the ball rolls to another brighter room. While in depth image sequence, our method keeps tracking the ball through out the whole sequence. Therefore, the errors of depth image-based methods are significantly much lower than the color-based one.

In bear sequences, severe occlusion is the main challenge when a book occludes the target bear for a while. Severe occlusion leads to losing target in color image, and without restarting module in the rest images the color-based method fails in finding the target again. In our proposed methods, we introduce the detecting module to detect the losing target and keep tracking again. In this case, the mean CPEs of color-based and depth-based methods with WSW and TLD are 192.99, 46.23 and 27.98, respectively. These significant results show the efficiency of our detection module in our tracking algorithm.

Finally, no-rigid target tracking in child sequences is the main challenge. From Table 2, we can see that our methods fail to track the target sometimes. It is because the target child is not a rigid object and his movements result in changing of target's shape. Depth image contains most information with shape information. The characteristic nature of depth images make it difficult to track no-rigid objects.

Table 3 shows the results of tracking speed (frame per second) for different sequences. The average tracking speed of depth image-based methods with WSW and TLD are 3.39 and 1.18 frames per second. As we can see, although the depth-based method with TLD achieves higher accuracies than that with WSW, it is slower for tracking with more computational complexity. In the TLD framework, it need additional memory and time cost for different models in tracking module and detection module. Moreover, it should be noted that with optimization improvements and powerful processors, the running times will certainly decrease to satisfy real time requirements in various applications.

We further evaluate our methods on face tracking. Figure 6 shows the selected four face sequences under different conditions (face moving, partial occlusion, full occlusion and face turning). The challenges here are occlusion and head turn. From the results of Table 4, we can see that our proposed methods can achieve comparative performance in face tracking as well as arbitrary object tracking. The performance of depth-based method with TLD is slightly better than that with WSW. However, both methods fail to track faces with high degrees of head turning due to the no-rigid challenge.

Fig. 6 The face sequences under different conditions (face moving, partial occlusion, full occlusion and face turning)

Table 4 Experimental results with the mean center position errors for face tracking

Test sequence	Depth image (WSW)	Depth image (TLD)
Face_move1	**5.53**	6.76
Face_occ2	13.44	**12.49**
Face_occ3	14.17	**13.46**
Face_turn2	87.66	**79.48**
Mean	30.14	**28.05**

3.3 Experimental Results on Driver Face Tracking

Driving fatigue is one of the main causes of road accidents [22,24]. In order to prevent these accidents, the state of drowsiness of drivers should be monitored. It is straightforward to detect and recognize facial expression of drivers as the measurement of fatigue level [4,10,11]. Robust face tracking plays a critical role in these approaches. However, the challenges of complex driving environments including various illumination, occlusion and camera motion degrade the performance of monitoring systems, which limit the real world applications.

In this study, we evaluate the performance of our proposed methods on driver face tracking in a simulated driving system. We have developed a simulated driving system to collect the data as shown in Fig. 7. We record both RGB and depth data using a standard Microsoft Kinect 1.0, and manually annotate ground truth bounding boxes of faces. Here, we evaluate the performance of our proposed tracking algorithms on depth driver face sequences. The

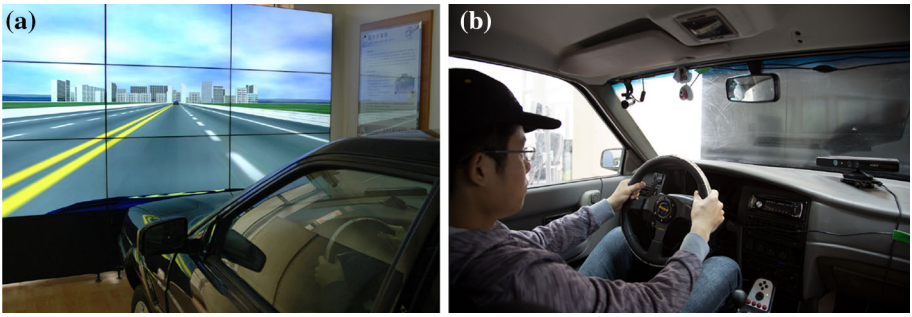


Fig. 7 Our simulated driving system. **a** the virtual driving environment displayed on the screen. **b** the face videos are recorded with Kinect while the participant is driving the virtual vehicle

Fig. 8 The samples of driver face sequences recorded from our simulated driving system

sample driver face sequences are shown in Fig. 8. The dataset contains different conditions in driving environments including various illumination and occlusion with ten-minute time length. The mean CPEs of depth-based methods with WSW and TLD are 11.33 and 16.34, respectively. From the experimental results, we can see that even there exist illumination changing and hand occlusion, our proposed methods using depth images can still achieve comparative tracking performance. These results indicate the efficiency and effectiveness of our proposed methods for real world applications such as driving fatigue detection.

4 Conclusions

In this paper, we have proposed a robust tracking method based on depth image with sparse representation and object detection under challenging conditions like illumination changing and occlusion. We have introduced a framework of combining tracking and detection to leverage precision and efficiency under severe occlusion conditions. The sigmoid normalization algorithm is used to preprocess depth images. For tracking, objects are represented with sparse representation with a set of online updated PCA basis vectors and trivial templates. For detection, we have introduced two detection strategies of WSW and TLD to our tracking algorithm. We have evaluated our methods on both the public available dataset PTB and our own driver face video in a simulated driving environment. The comparative performance with respect to precision and running time demonstrate the effectiveness and efficiency of our proposed methods.

Acknowledgments This work was supported in part by the grants from the National Natural Science Foundation of China (Grant No. 1272248), the National Basic Research Program of China (Grant No. 2013CB329401), and the Science and Technology Commission of Shanghai Municipality (Grant No. 3511500200).

References

1. Avidan S (2004) Support vector tracking. IEEE Trans Pattern Anal Mach Intell 26(8):1064–1072
2. Avidan S (2007) Ensemble tracking. IEEE Trans Pattern Anal Mach Intell 29(2):261–271
3. Cai Q, Gallup D, Zhang C, Zhang Z (2010) 3D deformable face tracking with a commodity depth camera. In: Computer Vision-ECCV 2010, Springer, pp 229–242
4. Cao Y, Lu BL (2013) Neural information processing., Real-time head detection with kinect for driving fatigue detectionSpringer, Heidelberg, pp 600–607
5. Colombo A, Cusano C, Schettini R (2006) 3D face detection using curvature analysis. Pattern Recognit 39(3):444–455
6. Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: IEEE conference on computer vision and pattern recognition, vol 2, pp 142–149
7. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. IEEE Trans Pattern Anal Mach Intell 25(5):564–577
8. Hale ET, Yin W, Zhang Y (2008) Fixed-point continuation for \$ell_1-minimization: methodology and convergence. SIAM J Optim 19(3):1107–1130
9. Hu W, Li X, Zhang X, Shi X, Maybank S, Zhang Z (2011) Incremental tensor subspace learning and its applications to foreground segmentation and tracking. Int J Comput Vis 91(3):303–327
10. Ji Q, Yang X (2002) Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. Real-Time Imaging 8(5):357–377
11. Ji Q, Zhu Z, Lan P (2004) Real-time nonintrusive monitoring and prediction of driver fatigue. IEEE Trans Veh Technol 53(4):1052–1068
12. Jia X, Lu H, Yang MH (2012) Visual tracking via adaptive structural local sparse appearance model. In: IEEE conference on computer vision and pattern recognition, pp 1822–1829
13. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. IEEE Trans Pattern Anal Mach Intell 34(7):1409–1422
14. Martínez AM (2002) Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Trans Pattern Anal Mach Intell 24(6):748–763
15. Mei X, Ling H (2011) Robust visual tracking and vehicle classification via sparse representation. IEEE Trans Pattern Anal Mach Intell 33(11):2259–2272
16. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. Image Vision Comput 21(1):99–110
17. Paragios N, Deriche R (2000) Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Trans Pattern Anal Mach Intell 22(3):266–280
18. Paschos G (2001) Perceptually uniform color spaces for color texture analysis: an empirical evaluation. IEEE Trans Image Process 10(6):932–937
19. Pei SC, Lin CN (1995) Image normalization for pattern recognition. Image Vis Comput 13(10):711–723
20. Pérez P, Hue C, Vermaak J, Gangnet M (2002) Color-based probabilistic tracking. In: European conference on computer vision, Springer, pp 661–675
21. Ross DA, Lim J, Lin RS, Yang MH (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1–3):125–141
22. Sahayadhas A, Sundaraj K, Murugappan M (2012) Detecting driver drowsiness based on sensors: a review. Sensors 12(12):16,937–16,953
23. Shen SC, Zheng WL, Lu BL (2014) Online object tracking based on depth image with sparse coding. In: Neural information processing, Springer, pp 234–241
24. Shi LC, Lu BL (2013) EEG-based vigilance estimation using extreme learning machines. Neurocomputing 102:135–143
25. Song S, Xiao J (2013) Tracking revisited using rgbd camera: Unified benchmark and baselines. In: IEEE international conference on computer vision, pp 233–240
26. Spinello L, Arras KO (2011) People detection in RGB-D data. In: IEEE/RSJ international conference on intelligent robots and systems, pp 3838–3843
27. Wang D, Lu H, Yang MH (2013) Online object tracking with sparse prototypes. IEEE Trans Image Process 22(1):314–325
28. Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
29. Wu Y, Lim J, Yang MH (2013) Online object tracking: A benchmark. In: IEEE conference on computer vision and pattern recognition, pp 2411–2418
30. Xia L, Chen CC, Aggarwal JK (2011) Human detection using depth information by kinect. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW), pp 15–22

31. Yang H, Shao L, Zheng F, Wang L, Song Z (2011a) Recent advances and trends in visual tracking: A review. Neurocomputing 74(18):3823–3831
32. Yang M, Zhang L (2010) Gabor feature based sparse representation for face recognition with gabor occlusion dictionary. In: Computer Vision-ECCV 2010, Springer, pp 448–461
33. Yang M, Zhang L, Yang J, Zhang D (2011b) Robust sparse coding for face recognition. In: IEEE conference on computer vision and pattern recognition, pp 625–632
34. Yang T, Pan Q, Li J, Li SZ (2005) Real-time multiple objects tracking with occlusion handling in dynamic scenes. IEEE conference on computer vision and pattern recognition, vol 1, pp 970–975
35. Yilmaz A, Li X, Shah M (2004) Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans Pattern Anal Mach Intell 26(11):1531–1536
36. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surveys (CSUR) 38(4):13
37. Zhang S, Yao H, Sun X, Lu X (2013) Sparse coding based visual tracking: review and experimental comparison. Pattern Recogn 46(7):1772–1788