Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Exploiting a depth context model in visual tracking with correlation filter*

Zhao-yun CHEN†[1,2], Lei LUO†‡[1], Da-fei HUANG[1,2], Mei WEN[1,2], Chun-yuan ZHANG[1,2]

([1]*College of Computer, National University of Defense Technology, Changsha 410073, China*)

([2]*National Key Laboratory of Parallel and Distributed Processing, Changsha 410073, China*)

†E-mail: chenzhaoyun@nudt.edu.cn; l.luo@nudt.edu.cn

Received Nov. 10, 2015; Revision accepted June 6, 2016; Crosschecked Apr. 13, 2017

Abstract: Recently correlation filter based trackers have attracted considerable attention for their high computational efficiency. However, they cannot handle occlusion and scale variation well enough. This paper aims at preventing the tracker from failure in these two situations by integrating the depth information into a correlation filter based tracker. By using RGB-D data, we construct a depth context model to reveal the spatial correlation between the target and its surrounding regions. Furthermore, we adopt a region growing method to make our tracker robust to occlusion and scale variation. Additional optimizations such as a model updating scheme are applied to improve the performance for longer video sequences. Both qualitative and quantitative evaluations on challenging benchmark image sequences demonstrate that the proposed tracker performs favourably against state-of-the-art algorithms.

Key words: Visual tracking; Depth context model; Correlation filter; Region growing
http://dx.doi.org/10.1631/FITEE.1500389 **CLC number:** TP391.4

1 Introduction

Visual object tracking is a fundamental task for a wide range of computer vision applications. Applications such as video surveillance, intelligent traffic, robotics navigation, human-computer interaction, and augmented reality require robust and reliable location estimations of a target throughout an image sequence (Yilmaz *et al.*, 2006; Wu *et al.*, 2013; Lee *et al.*, 2014). Despite significant progress in recent years, it remains a challenge to design an all-situation-handled tracker that can handle various challenging factors: occlusion, illumination change, scale variation, cluttering background, etc. (Yilmaz *et al.*, 2006; Yang *et al.*, 2011; Li *et al.*, 2013; Wu *et al.*, 2013; Smeulders *et al.*, 2014). Furthermore, the real-time constraint is another challenge to researchers. Simple tracking models cannot perform well in a complex environment; however, the more sophisticated the tracking model becomes, the higher the computational cost that will arise.

Correlation filter based trackers (Bolme *et al.*, 2010; Henriques *et al.*, 2012; Danelljan *et al.*, 2014a; Li and Zhu, 2014; Zhang *et al.*, 2014; Henriques *et al.*, 2015; Liu *et al.*, 2015; Ma *et al.*, 2015) have been proven to be competitive against others in accuracy and with much higher computational efficiency. Those trackers train a discriminative filter, whose convolutional output can indicate the likeness between a candidate and the target. Although scale-adaptive variants (Danelljan *et al.*, 2014a; Zhang *et al.*, 2014; Li and Zhu, 2014; Liu *et al.*, 2015; Ma *et al.*, 2015) have been proposed, there is no correlation filter variant of flexible adaptability to target's scale and aspect ratio changes.

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61502509, 61402504, and 61272145), the National High-Tech R&D Program (863) of China (No. 2012AA012706), and the Research Fund for the Doctoral Program of Higher Education of China (No. 21024307130004)

ORCID: Lei LUO, http://orcid.org/0000-0002-9329-1411

Occlusion is also an inevitable challenge that a correlation filter must face. Since the context around the target may remain unchanged even if the object is occluded, several trackers adopt the contextual information in an algorithm (Yang *et al.*, 2009; Grabner *et al.*, 2010; Dinh *et al.*, 2011; Zhang and Maaten, 2014; Zhang *et al.*, 2014; Ma *et al.*, 2015). However, these trackers are prone to drifting if the occluders have a similar appearance to the target.

A reliable depth map can provide some valuable information which can significantly improve the tracking performance against occlusion and scale variation. Off-the-shelf depth sensors, such as Microsoft Kinect, make depth information acquisition a reasonably easy task, and the depth information has been widely used in object detection, object segmentation, scene understanding (Chen *et al.*, 2014; Gupta *et al.*, 2014; Hickson *et al.*, 2014), etc. However, little work has focused on the RGB-D tracking method. Several trackers use only a depth map to generate a more robust feature set (Teichman *et al.*, 2013). As will be discussed later, depth information can provide a solution to tackle occlusion and scale variation if used properly.

The tracker proposed in this paper is based on a fast tracking algorithm via spatio-temporal context learning (STC) (Zhang *et al.*, 2014), which integrates context information into correlation filter training. However, STC cannot handle the aspect ratio variation of the target, and have poor performance under various challenging factors such as occlusion and deformation. For that reason, we construct the depth context model for robust estimation of the target location. Moreover, a region-growing method (Adams and Bischof, 1994) based on the depth map is adopted to provide an accurate scale estimation for the target. The main contributions of this study include:

1. A depth context model is constructed for estimation of the target location which is resistant to heavy occlusion, fast motion, and large deformation.

2. A region growing method is adopted to enable the adaptability to the target's scale and aspect ratio changes.

3. A scheme for occlusion detection and learning rate suppression is proposed to improve the performance when occlusion exists.

Based on the benchmark protocol and dataset from Song and Xiao (2013), an experiment on a 20-sequence dataset with various challenging attributes is conducted. The proposed tracker performs favourably, compared to several state-of-the-art algorithms in terms of both tracking accuracy and speed.

2 Related work

Visual object tracking has been extensively studied over the past decade (Yilmaz *et al.*, 2006; Yang *et al.*, 2011; Li *et al.*, 2013; Wu *et al.*, 2013; Smeulders *et al.*, 2014). In this section, we briefly review the methods closely related to our work: (1) correlation tracking, (2) contextual tracker, and (3) RGB-D exploiting.

Correlation tracking: Correlation filters have been widely used in numerous applications such as object detection and recognition (Kumar *et al.*, 2010). Since the convolutional operation is transfered to element-wise multiplication in the Fourier domain, correlation filters have attracted considerable attention recently in visual tracking due to their computational efficiency. The MOSSE tracker (Bolme *et al.*, 2010) is based on learning an adaptive correlation filter by minimizing the output sum of squared error (MOSSE). MOSSE is computationally efficient with a speed of hundreds of frames per second. The CSK tracker (Henriques *et al.*, 2012) extends the correlation filter to kernel space and achieves higher efficiency by using the circulant structure in the representation model. The KCF tracker (Henriques *et al.*, 2015), as an extended version of CSK, is further improved through supporting multi-channel features. The ACT tracker (Danelljan *et al.*, 2014b) proposes a more robust updating scheme and adopts the colour naming feature. However, the above-mentioned trackers cannot tackle the problem of scale and aspect ratio adaptability. To handle the scale change, the SAMF tracker (Li and Zhu, 2014) presents a framework which samples candidates with several pre-defined scale perturbations. The best scale and position are found according to the responses of the correlation filter applied to those samples individually. DSST (Danelljan *et al.*, 2014a) learns separate filters for translation and scale estimation by using HOG features. Recently, Ma *et al.* (2015) proposed a method to combine the discriminative correlation filter with a robust online detector. The correlation filter is responsible for estimating the translation and scale variation of the target, while the online detector is responsible for re-detecting the

target in case of tracking failure.

Contextual tracker: Contextual information is helpful to the performance especially when the target is occluded or leaves the visible image scope (Yang *et al.*, 2009; Grabner *et al.*, 2010; Dinh *et al.*, 2011; Zhang and Maaten, 2014). However, supporting regions with contextual information can be transformed to distracting regions in certain circumstances. When the occluder has a distinct appearance compared with the target, supporting regions as presented in Grabner *et al.* (2010) and Dinh *et al.* (2011) provide valuable cues to cope with occlusion. However, when occluders have a similar appearance, a contextual tracker may be confused and misled to drifting.

RGB-D exploiting: With the widely used depth camera, depth information has been introduced into object detection, object segmentation, scene understanding (Chen *et al.*, 2014; Gupta *et al.*, 2014; Hickson *et al.*, 2014), etc. However, it has not been widely used in visual tracking. The existing RGB-D trackers usually adopt an online training method. Luber *et al.* (2011) combined a novel multi-cue person detector which uses RGB-D data with an online-boosted detector that learns individual target models. Park *et al.* (2011) proposed a textureless object detection and 3D tracking method which automatically extracts information from colour images and the corresponding depth maps on the fly. With the introduction of depth information, more robust features can be generated for tracking. Teichman *et al.* (2013) used a rich feature set, including local image appearance, depth discontinuity, optical flow, and surface normal, to inform the segmentation decision in a conditional random field model. Choi and Christensen (2013) presented a particle filtering approach for 6-DOF object pose tracking using an RGB-D camera. Moreover, new baseline tracking algorithms and benchmarking criteria based on RGB-D image sequences are in demand. Song and Xiao (2013) designed a group of baseline algorithms, and combined RGB-D HOG detection, optical flow, and occlusion handling in the tracking task.

3 Our methodology

We decompose the tracking task into translation estimation, occlusion detection, and scale estimation. Section 3.1 presents how we extend the STC tracker by constructing a depth context model for translation estimation. In Section 3.2, we propose the region growing method for scale estimation. The occlusion detection and a corresponding updating scheme are demonstrated in Section 3.3. The flowchart of our method (Fig. 1) consists of three parts: translation estimation, occlusion detection, and scale estimation. The output bounding box closely encloses the merged region.

Notation: A bold lowercase letter, e.g., $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, denotes a vector. A lowercase letter, e.g., $\boldsymbol{m}, \boldsymbol{h}, \boldsymbol{k}, \boldsymbol{f}$,

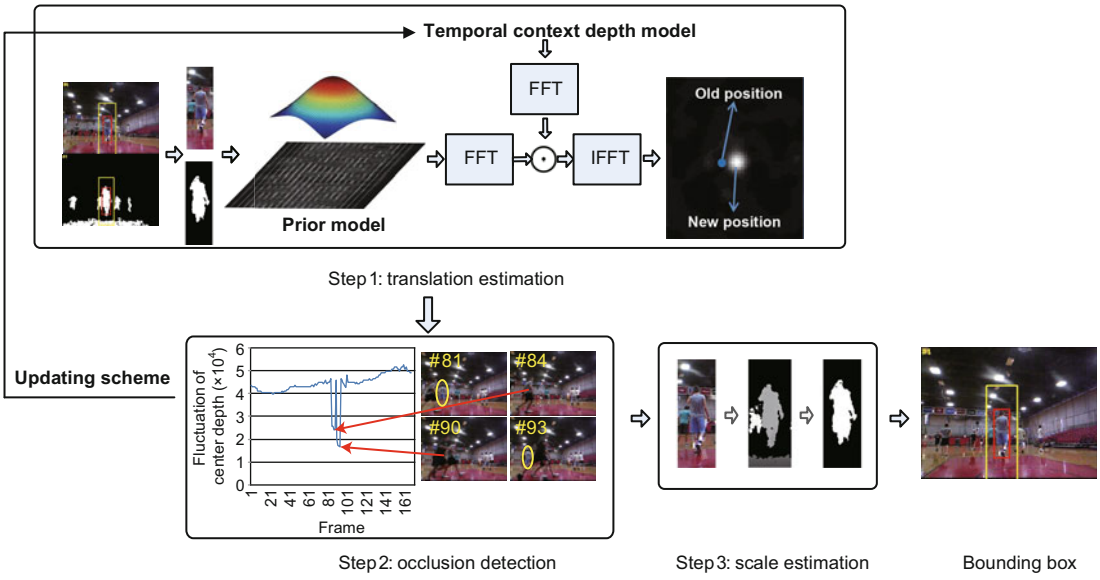

Fig. 1 Workflow of the proposed algorithm

or an uppercase letter, e.g., $\boldsymbol{H}$,$\boldsymbol{P}$, denotes a matrix. $d(\cdot)$ denotes a distance function. $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ are the FFT function and the inverse FFT function, respectively.

3.1 Depth context model

As the basis of our tracker, STC builds a confidence map which estimates the object location likelihood:

$$\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}|o), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^2$ is the object location and o stands for the target object in the scene. By marginalizing the joint probability $\boldsymbol{P}(\boldsymbol{x}|\phi(\boldsymbol{y}), o)$, the object location likelihood function can be decomposed as

$$\begin{aligned}\boldsymbol{m}(\boldsymbol{x}) &= \boldsymbol{P}(\boldsymbol{x}|o)\\ &= \sum \boldsymbol{P}(\boldsymbol{x}, \phi(\boldsymbol{y})|o)\\ &= \sum \boldsymbol{P}(\boldsymbol{x}|\phi(\boldsymbol{y}), o)\boldsymbol{P}(\phi(\boldsymbol{y})|o),\end{aligned} \tag{2}$$

where $\boldsymbol{y}$ is the context patch of $\boldsymbol{x}$ and $\phi(\boldsymbol{y})$ is the context feature set. $\boldsymbol{P}(\phi(\boldsymbol{y})|o)$ represents the prior probability which models the appearance of the local context. $\boldsymbol{P}(\boldsymbol{x}|\phi(\boldsymbol{y}), o)$ denotes the depth context probability which is an indication of the spatial relationship between the object location and its context. The position of the target is detected by searching for the location that maximizes the value of $\boldsymbol{m}(\boldsymbol{x})$. The main problem here is to learn the depth context model and to estimate the confidence map.

$\boldsymbol{P}(\boldsymbol{x}|\phi(\boldsymbol{y}), o)$ in Eq. (2) is defined as

$$\boldsymbol{P}(\boldsymbol{x}|\phi(\boldsymbol{y}), o) = \boldsymbol{h}(\boldsymbol{y}), \tag{3}$$

where $\boldsymbol{h}(\boldsymbol{y})$ is a function with respect to the depth context probability. Eq. (2) is an indication of the spatial relationship between the object location and its context.

The feature adopted in STC is solely the pixel intensity which is weak for modelling appearance. To develop an effective appearance representation, we use the histogram of orientation gradients (HOG) as the feature. Therefore, the prior model is represented by the weighted product of the HOG feature and the Gaussian function:

$$P(\phi(\boldsymbol{y})|o) = b\,\boldsymbol{k}(\boldsymbol{y})\boldsymbol{f}(\boldsymbol{y}), \tag{4}$$

where b is the normalization constant that restricts $\boldsymbol{P}(\phi(\boldsymbol{y})|o)$ ranging from 0 to 1, $\boldsymbol{k}(\cdot)$ denotes the mapping to the HOG feature space, and $\boldsymbol{f}(\cdot)$ denotes the Gaussian matrix, defined as

$$\boldsymbol{f}(\boldsymbol{y}) = \exp\left(-\frac{d(\boldsymbol{z}, \boldsymbol{x})^2}{\sigma^2}\right),\ \forall \boldsymbol{z} \in \boldsymbol{y}, \tag{5}$$

where σ is the scale parameter and $d(\cdot)$ denotes the spatial distance which takes depth information into consideration.

Generally, a bounding box is given in the first frame as the tracking target. The confidence map based on the target location is modeled as

$$\boldsymbol{m}(\boldsymbol{x}) = a\,\exp\left(-\frac{d(\boldsymbol{x}, \boldsymbol{x}')^\beta}{\alpha}\right), \tag{6}$$

where $\boldsymbol{x}'$ is the target location, $d(\cdot)$ denotes the spatial distance, a is the normalization constant, and α and β are the coefficients of the distance function.

Substitute Eqs. (6), (4), and (3) into Eq. (2) and formulate the confidence map as

$$\begin{aligned}\boldsymbol{m}(\boldsymbol{x}) &= a\,\exp\left(-\frac{d(\boldsymbol{x}, \boldsymbol{x}')^\beta}{\alpha}\right)\\ &= \boldsymbol{h}(\boldsymbol{y}) \otimes (b\,\boldsymbol{k}(\boldsymbol{y})\boldsymbol{f}(\boldsymbol{y})),\end{aligned} \tag{7}$$

where $\otimes$ denotes the convolution operator. By adopting the fast Fourier transformation (FFT), Eq. (7) can be transformed to

$$\begin{aligned}&\mathcal{F}\left(a\,\exp\left(-\frac{d(\boldsymbol{x}, \boldsymbol{x}')^\beta}{\alpha}\right)\right)\\ &\quad = \mathcal{F}(\boldsymbol{h}(\boldsymbol{y})) \odot \mathcal{F}(b\,\boldsymbol{k}(\boldsymbol{y})\,\boldsymbol{f}(\boldsymbol{y})),\end{aligned} \tag{8}$$

where $\mathcal{F}$ is the FFT function and $\odot$ denotes the element-wise product. Therefore, the depth context model can be formulated as

$$\boldsymbol{h}(\boldsymbol{y}) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}\left(a\,\exp\left(-\frac{d(\boldsymbol{x}, \boldsymbol{x}')^\beta}{\alpha}\right)\right)}{\mathcal{F}(b\,\boldsymbol{k}(\boldsymbol{y})\boldsymbol{f}(\boldsymbol{y}))}\right), \tag{9}$$

where $\mathcal{F}^{-1}$ denotes the inverse FFT (IFFT). The process of depth context model learning is shown in Fig. 2.

Assuming the target region in the tth frame is fixed, the temporal depth context model $\boldsymbol{H}_{t+1}(\boldsymbol{y})$ in the $(t+1)$th frame is calculated considering the temporal relationship:

$$\boldsymbol{H}_{t+1}(\boldsymbol{y}) = (1-\lambda)\,\boldsymbol{H}_t(\boldsymbol{y}) + \lambda\,\boldsymbol{h}_t(\boldsymbol{y}), \tag{10}$$

where $\boldsymbol{H}_t(\boldsymbol{y})$ is the temporal depth context model in the tth frame, $\boldsymbol{h}_t(\boldsymbol{y})$ is computed from Eq. (9), and λ is the learning rate. According to Eq. (8), the

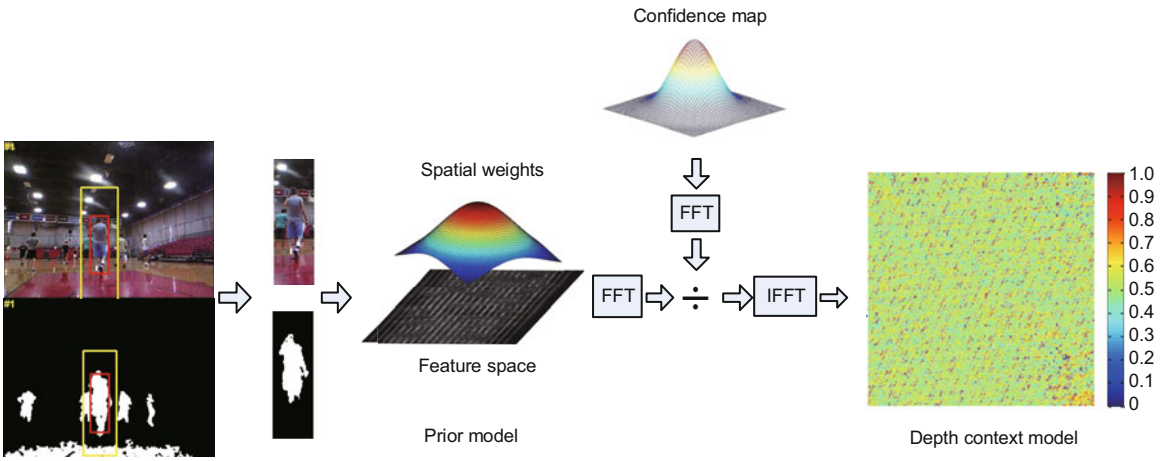

Fig. 2 Process of learning the depth context model

confidence map $\boldsymbol{m}_{t+1}(\boldsymbol{x})$ in the $(t+1)$th frame is estimated as

$$\boldsymbol{m}_{t+1}(\boldsymbol{x}) = \mathcal{F}^{-1}\left(\mathcal{F}\left(\boldsymbol{H}_{t+1}(\boldsymbol{y})\right) \odot \mathcal{F}\left(b\,\boldsymbol{k}_{t+1}(\boldsymbol{y})\,\boldsymbol{f}_t(\boldsymbol{y})\right)\right), \tag{11}$$

where $\boldsymbol{f}_t(\boldsymbol{y})$ is computed according to the target location in the tth frame. The location maximizing the value of $\boldsymbol{m}_{t+1}(\boldsymbol{x})$ is estimated as the target center in the $(t+1)$th frame. In the same way, the temporal depth context model is recomputed based on the new target location according to Eqs. (9) and (10). This process is repeated until the last frame. The flowchart of translation estimation is shown in Fig. 3. The output of translation estimation is prepared for scale estimation and occlusion detection. The upper part of Fig. 3 represents the updating of the temporal depth context model. The lower part stands for target location estimation at the $(t+1)$th frame.

3.2 Region growing method

The variation of the target's scale and aspect ratio is a significant challenge in visual tracking. We propose a region growing method based on the target location from Section 3.1 to estimate the scale and aspect ratio of the target. Region growing is a method for aggregating pixels according to the feature similarity in a region. It starts at a seed region or a seed pixel, and merges other regions or pixels which contain a similar feature to that of the original region. The algorithm will terminate when there is no region or pixel eligible for merging. The key advantage of this method is that the merged output region is not limited by the aspect ratio in previous frames and can provide an accurate scale estimation of the target. Similarity measurement among the pixels or regions can comprise average intensity, colour, texture, etc.

The performance of a region growing method depends on three factors: (1) seed point selection, (2) merging rule, and (3) termination condition. Since the continuity of the target is ensured in the depth map, we adopt a modified region growing method to provide an accurate scale estimation of the target. The method proposed in our tracker is a modified 4-neighbour pixel region growing algorithm. The location of maximal response in Section 3.1 is of high reliability to be the seed point. Generally, there are abrupt gradient changes between the target and its surroundings in the depth map. The merging rule is thus to set a threshold which limits the depth difference between two points. The algorithm will terminate when there is no mergeable point. The output bounding box is a box that closely encloses the merged region. An example of the region growing method is illustrated in Fig. 4.

The region growing process is demonstrated as follows, and the corresponding pseudo-code can be found in Algorithm 1:

1. Push the target center into the stack. The initialized output region contains only the target center.

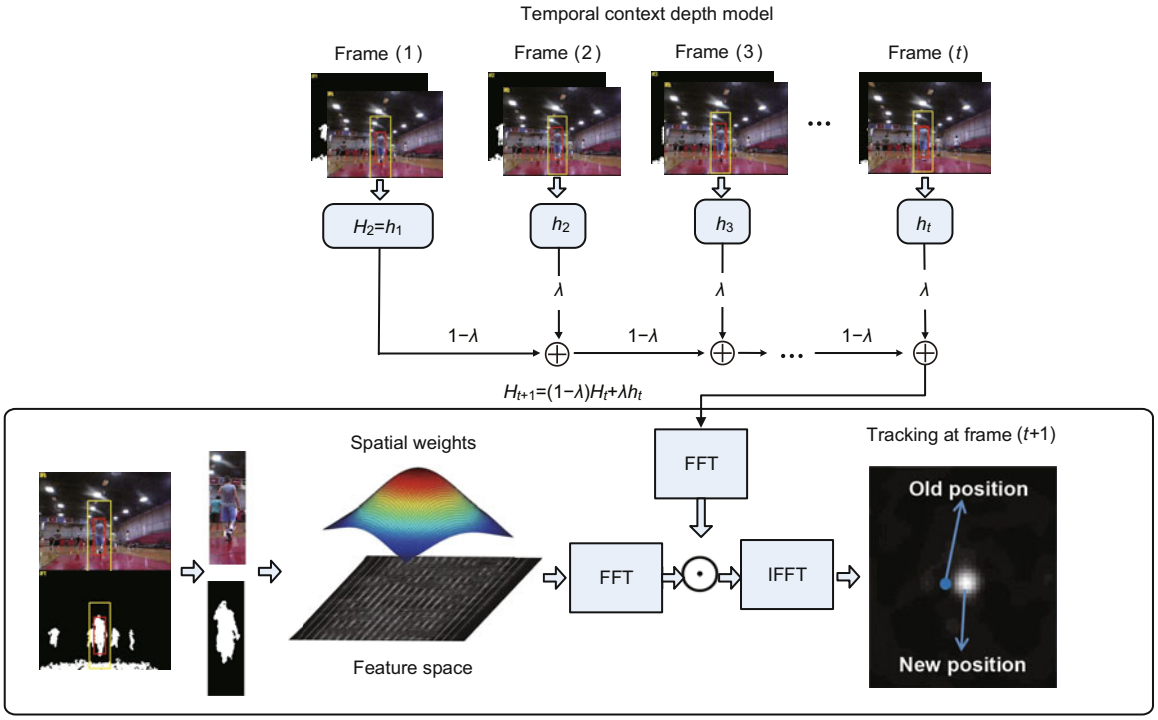

Fig. 3 Main workflow of translation estimation

Fig. 4 A region growing example: (a) RGB context region; (b) depth image of the same area; (c) merged output region of the target

2. Pop the top point from the stack. Examine the neighbours of the point in sequence to check whether they conform to the merging rule. The neighbour whose depth satisfies the merging condition is merged into the output region and pushed into the stack.

3. Check whether the stack is empty. If not, repeat 2.

4. The stack is empty, and the algorithm is terminated.

The region growing method takes advantage of the target continuity in the depth map, and provides the accurate target region growing from the center. Furthermore, when the target is partially occluded, the occluded part will not be included in the merged output region due to the depth gap between the tracking target and the occluder. Since our algorithm is based on the depth, it is robust and accurate not only in scale variation scenes, but also in illumination change and deformation scenarios.

Algorithm 1 Region growing method on the ith frame

Input: $\boldsymbol{x}_i$, the ith frame target location; $d_i(\cdot)$, depth image at the ith frame.
Output: merged output region.
Initialize: stack $\boldsymbol{S}=\{\}$, output $\boldsymbol{R}_i=\{\}$.
Add $\boldsymbol{x}_i$ to stack $\boldsymbol{S}$ and output region $\boldsymbol{R}_i$;
Pop top point $\boldsymbol{q}$ from stack $\boldsymbol{S}$;
Acquire the depth $d_i(\boldsymbol{q})$ of $\boldsymbol{q}$;
while $\boldsymbol{S}$ is not empty **do**
  for the neighbors $\boldsymbol{k}$ of $\boldsymbol{q}$ **do**
    Acquire the depth $d_i(\boldsymbol{k})$ of $\boldsymbol{k}$;
    if $|d_i(\boldsymbol{q})-d_i(\boldsymbol{k})|\,/\,d_i(\boldsymbol{q})<p$ **then**
      Push $\boldsymbol{k}$ into $\boldsymbol{R}_i$ and $\boldsymbol{S}$;
    end if
  end for
end while

3.3 Occlusion detection and updating scheme

Occlusion is one of the most difficult problems in visual tracking. Obviously, it is hard to handle

because the tracker may fail to relocate the target. Although several methods have been proposed to attempt to detect occlusion, such as judging by context (Zhang *et al.*, 2014; Ma *et al.*, 2015) and part-based model (Adam *et al.*, 2006; Ross *et al.*, 2008; Cehovin *et al.*, 2011; Izadinia *et al.*, 2012; Shu *et al.*, 2012; Yang and Nevatia, 2012; Li *et al.*, 2015), a robust enough method is still required to be responsible for occlusion detection, especially for longer video sequences.

Occlusion detection based on a depth map is a good solution that can offer enough robustness for tracking. We present an occlusion detection method here between translation and scale estimations. Following the step of Section 3.1, the detection is according to the depth of the target location. If the occlusion exists on the target location, we will find a new one as a substitute and the new point is selected as the seed point in scale estimation. In addition, a corresponding updating scheme is adopted.

To detect occlusion, the target depth value in each frame is recorded. According to the spatial-temporal continuity, we assume that there is no abrupt change within the target depth sequence when no occlusion occurs. On the contrary, if there is a sudden reduction in target depth, it should be concluded that the object center is occluded, as illustrated in Fig. 5, where in the 84th and 90th frames, the target is occluded, and there is a sudden reduction in the line-chart of center depth. The differential ratio threshold between the depth of target location and the reference depth value is set for occlusion judgement here. The reference depth is set to the average center depth of the latest N frames to avoid random errors.

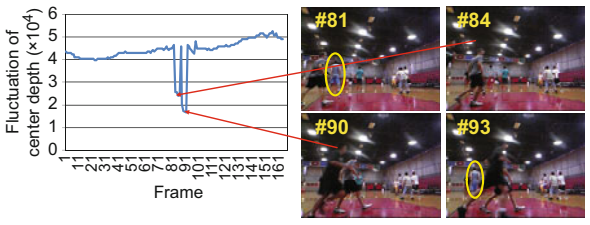

Fig. 5 Occlusion in video and the corresponding abrupt reduction in the line-chart of center depth

When there is no obvious reduction (i.e., the differential ratio does not exceed the threshold) within the depth sequence, it indicates that the target center is not occluded (there are chances that the target is partially occluded). In this case, no additional effort is needed and the scale estimation should be conducted as in Section 3.2.

When a reduction is detected exceeding the threshold, it indicates that the target center is occluded. Therefore, a new location should be found to replace the original one for scale estimation. The scheme we adopt is to do a mask computation between a binary image of the current frame and a binary image of the merged output region from the previous frame. In the binary image of the merged output region from the previous frame, the target pixels are labelled 1 and the others labelled 0. However, in the current frame, the pixel whose depth meets the no-occlusion conditions with the reference depth value is labelled 1 and the rest are labelled 0.

If the pixel set of the computed mask is not empty, the nearest pixel to the previous center is selected as the new target location and also the seed point in Section 3.2. Meanwhile, we record the depth at the new location and adjust the learning rate of the depth context model according to

$$\lambda = \lambda * \frac{|\boldsymbol{x}^* - \boldsymbol{x}|}{\sqrt{S_{\mathrm{h}}^2 + S_{\mathrm{w}}^2}}, \tag{12}$$

where λ is the learning rate, S_{h} and S_{w} are the current height and width of the target, respectively, and $\boldsymbol{x}^*$ and $\boldsymbol{x}$ are the new target location and the previous location, respectively.

If the pixel set of the computed mask is empty, it means that the target is fully occluded. In this case, the center location and depth value of the target in the previous frame remains unchanged, and the learning rate is set to 0. Since the target is fully occluded, there is no need to conduct scale estimation or generate bounding box output in this frame.

The occlusion detection method and updating scheme are shown in Algorithm 2. Our approach is effective in handling partial occlusion and some full occlusion scenarios in which the target does not go out of the context when it reappears.

We provide a brief outline of our tracking process in Algorithm 3. The minimum bound rectangle of the merged output region from Section 3.2 is the final bounding box of the target. Our approach accurately estimates both translation and scale while being resistant to heavy occlusion and other challenging attributes.

4 Experiments

To evaluate our proposed tracker, we compile a set of 20 RGB-D video sequences which are chosen from the Princeton RGB-D tracking dataset (Song and Xiao, 2013), covering challenging factors such as heavy occlusion, cluttering background, drastic deformation, and scale variation. The challenging factors contained in each video are listed in Table 1. However, the tracking ground truths of the video sequences are unpublished. Thus, we mark them manually. As a stereo extension of the STC tracking algorithm, firstly, we compare the performance of our method with that of the original STC (Zhang *et al.*, 2014). Due to the lack of public availability of other RGB-D tracking algorithms, we compare the proposed algorithm with several RGB trackers: Struck (Hare *et al.*, 2011), TLD (Kalal *et al.*, 2012), KCF (Henriques *et al.*, 2015), and DSST (Danelljan *et al.*, 2014a). During the test, the parameters of the proposed algorithms are fixed in all experiments. All experiments are conducted on a conventional Intel i5 desktop computer with a 4 GB RAM.

4.1 Parameter setup

The parameters of the algorithm are initialized to empiric values which are tuned for the best results. Some of the parameters in Section 3.1 are kept the same as in STC. The padding factor of the context is set to 1. It indicates that the width and height of the context are twice those of the target. The coefficients in Eq. (6) are set to $\alpha = 2.25$ and $\beta = 1$. The scale parameter σ in Eq. (5) is set to $(S_{\mathrm{w}} + S_{\mathrm{h}})/2$. The learning rate λ of the temporal depth context model is set to 0.065. Some additional parameters in Sections 3.2 and 3.3 are also initialized. The threshold factor p in the region growing method is set to 0.028, and the differential ratio t in occlusion detection is set to 0.1. In the depth image, the raw depth ranges from 0 to 65 535. To normalise the effect of depth, the depth is shrunk to range from 1 to 600 through a direct linear mapping. The number of frames for calculating the reference depth value in occlusion detection is set to 5. To reduce the frequency effect at the image boundary, the context eigen-matrix multiplies a Hamming window before performing FFT.

4.2 Evaluation

The evaluation criteria employed in our experiments are distance precision (DP) and the success rate (SR) (Everingham *et al.*, 2010; Kristan *et al.*, 2015), both of which are computed based on the manually labelled ground truth. DP is computed as the percentage of frames in the sequence where the center location error is smaller than a certain threshold. The DP values at the threshold of 20 pixels (Everingham *et al.*, 2010) are reported. SR is a little different from that in VOC (Everingham *et al.*, 2010). Based on our observation, when the object is not in the scene or is occluded, the bounding box is

Table 1 Challenging factors within the image sequences

Number	Sequence	Challenge factor	Number of frames
1	Basketball1	Occlusion, scale, deformation, and background clutter	213
2	Book_turn2	Deformation	195
3	Cf_no_occ	Deformation	66
4	Child_no1	Scale and deformation	164
5	Cup_move_1	Scale	370
6	Express3_static_occ	Occlusion and background clutter	197
7	Face_occ5	Occlusion	330
8	Library2.2_occ	Occlusion, deformation, and background clutter	420
9	New_ex_no_occ	Scale, deformation, and background clutter	116
10	One_book_move	Normal tracking	292
11	Static_sign1	Occlusion	201
12	Toy_car_no	Deformation	103
13	Toy_car_occ	Occlusion	154
14	Toy_wg_no_occ	Deformation	158
15	Toy_wg_occ	Occlusion	154
16	Tracking4	Deformation	300
17	Tracking7.1	Occlusion and deformation	234
18	Walking_no_occ	Scale and deformation	118
19	Walking_occ_long	Occlusion and deformation	199
20	Zball_no3	Normal tracking	86

Algorithm 2 Occlusion detection and updating scheme for the ith frame

Input: $\boldsymbol{x}_i$, the target location at the ith frame from Eq. (11); $\boldsymbol{I}_i$, the original image at the ith frame; $d_i(\cdot)$, the depth image at the ith frame; $\boldsymbol{R}_{i-1}$, the output at the $(i-1)$th frame.

Output: $\boldsymbol{x}'_i$, the new ith frame location; λ_i, learning rate.

Initialize: stack $\boldsymbol{S}=\{\}$, output $\boldsymbol{R}_i=\{\}$.

Acquire the depth $d_i(\boldsymbol{x}_i)$ of $\boldsymbol{x}_i$;
Compute the average depth d_{ave} of the target for latest N frames;
if $|d_{\text{ave}}-d_i(\boldsymbol{x}_i)|\,/\,d_{\text{ave}}>t$ **then**
  // **Center occlusion:**
  Mask computation between the binary images of $\boldsymbol{R}_{i-1}$ and $\boldsymbol{I}_i$;
  if mask result is empty **then**
    // **Full occlusion:**
    Retain $\boldsymbol{x}'_i=\boldsymbol{x}'_{i-1}$;
    Set $\lambda=0$;
  else
    Select the nearest point $\boldsymbol{y}$ to $\boldsymbol{x}_i$ as new location $\boldsymbol{x}'_i$;
    Set λ according to Eq. (12);
  end if
end if
if no full occlusion **then**
  Scale estimation according to Section 3.2;
end if

still provided by most trackers. We take that into account and SR in our method is defined referring to Song and Xiao (2013):

$$\text{score}=\begin{cases}\dfrac{\text{area}(\text{ROI}_{T_i}\cap\text{ROI}_{G_i})}{\text{area}(\text{ROI}_{T_i}\cup\text{ROI}_{G_i})}, & \text{both ROI}_{T_i}\text{ and ROI}_{G_i}\text{ exist},\\ 1, & \text{neither ROI}_{T_i}\text{ nor ROI}_{G_i}\text{ exists},\\ 0, & \text{otherwise},\end{cases}\tag{13}$$

where ROI_{T_i} is the target bounding box in the ith frame, while ROI_{G_i} is the ground truth bounding box in the ith frame. When the target is fully occluded, there is no bounding box in the ground truth. At that time if the tracking bounding box does not exist either, the score is set to 1. If either ROI_{T_i} or ROI_{G_i} exists, the score is set to 0. In all experiments, the overlap threshold is set to $R_{\text{t}}=0.5$ (Everingham *et al.*, 2010). When score $>R_{\text{t}}$, the frame is labelled as a success; otherwise it is labelled as a failure. SR is defined as the average SR throughout the video sequences. A speed comparison in terms of the average number of frames per second (FPS) is also provided.

Algorithm 3 Proposed tracking approach: iteration at time step i

Input: $\boldsymbol{I}_i$, image at the ith frame; $\boldsymbol{D}_i$, depth image at the ith frame; $\boldsymbol{x}_{i-1}$, previous target position; $\boldsymbol{H}_i$, temporal depth context model for the ith frame; $\boldsymbol{R}_{i-1}$, output region at the $(i-1)$th frame.

Output: $\boldsymbol{x}_i$, estimated target position; $\boldsymbol{H}_{i+1}$, updated temporal depth context model for the $(i+1)$th frame; $\boldsymbol{R}_i$, output region at the ith frame; O_i, bounding box output at the ith frame.

// **Translation estimation:**
Calculate the featured prior probability $\boldsymbol{P}(\boldsymbol{\phi}|o)_i$ from $\boldsymbol{I}_i$, $\boldsymbol{D}_i$, and $\boldsymbol{x}_{i-1}$ in Eqs. (4) and (5);
Compute the confidence map $\boldsymbol{m}_i$ using $\boldsymbol{P}(\boldsymbol{\phi}|o)_i$ and $\boldsymbol{H}_i$ in Eq. (11);
Set $\boldsymbol{x}_i$ to the target position that maximizes $\boldsymbol{m}_i$;
// **Occlusion detection:**
Detect occlusion based on $\boldsymbol{x}_i$, $\boldsymbol{D}_i$, $\boldsymbol{I}_i$, and $\boldsymbol{R}_{i-1}$ according to Algorithm 2;
Correct the target location $\boldsymbol{x}_i$ and the learning rate λ if necessary;
// **Scale estimation:**
if no full occlusion **then**
  Compute the output region $\boldsymbol{R}_i$ according to Algorithm 1;
else
  Retain the previous output region $\boldsymbol{R}_i=\boldsymbol{R}_{i-1}$;
end if
O_i is the bounding box that closely encloses the output region $\boldsymbol{R}_i$;
// **Model update:**
Calculate the featured prior probability $\boldsymbol{P}(\boldsymbol{\phi}|o)_i$ from $\boldsymbol{I}_i$, $\boldsymbol{D}_i$, and $\boldsymbol{x}_i$ in Eqs. (4) and (5);
Update the depth context model for the ith frame model $\boldsymbol{h}_i$ in Eq. (9);
Update the temporal depth context model $\boldsymbol{H}_{i+1}$ for the $(i+1)$th frame in Eq. (10);

Overall performance: Based on this dataset, the quantitative results for the six algorithms are shown in Table 2. Among the existing trackers, DSST provides the best results with an average SR of 69.6% and our approach improves this performance by 15%. Similarly, in terms of DP, our approach favourably achieves an average DP of 84.3% which outperforms DSST by 16.6%. The ratio of video sequences where our proposed method gets the best or second best performance in terms of SR is 90% and that in terms of DP is 95%. Limited by our experimental platform, our tracker runs at 14.6 frames per second. It is slower than KCF and STC, but still meets the real-time requirement. Fig. 6 shows some screenshots of various trackers.

Table 2 Overall performances

Sequence number	Success rate						Distance precision					
	KCF	DSST	TLD	Struck	STC	Ours	KCF	DSST	TLD	Struck	STC	Ours
1	0.389	0.493	0.115	0.885	0.089	0.751	0.343	0.441	0.231	0.745	0.157	0.732
2	0.261	0.205	0.205	0.205	0.215	0.574	0.185	0.200	0.207	0.265	0.309	0.590
3	0.727	0.773	0.435	0.687	0.354	0.788	0.636	0.803	0.469	0.612	0.297	0.818
4	1.000	0.997	0.116	1.000	0.305	1.000	0.360	0.945	0.135	1.000	0.379	1.000
5	0.827	1.000	1.000	0.827	0.835	0.976	0.770	1.000	1.000	0.879	0.921	1.000
6	0.526	0.617	0.658	0.296	0.133	0.704	0.643	0.658	0.631	0.350	0.154	0.663
7	0.945	0.500	0.939	0.506	0.494	0.967	0.990	0.525	0.954	0.537	0.513	0.994
8	0.374	0.186	0.014	0.060	0.110	0.900	0.207	0.019	0.125	0.217	0.097	0.705
9	0.362	0.991	0.216	0.310	0.647	0.707	0.094	0.991	0.267	0.423	0.687	0.664
10	0.712	1.000	1.000	0.781	1.000	0.928	0.329	1.000	1.000	0.964	1.000	1.000
11	0.766	0.846	0.940	0.169	0.453	0.960	0.861	0.896	0.867	0.158	0.516	0.886
12	0.699	0.456	0.641	0.544	0.485	0.505	0.553	0.476	0.657	0.578	0.579	0.646
13	0.617	0.877	0.773	0.565	0.857	0.571	0.643	0.935	0.795	0.421	0.864	0.805
14	0.930	0.698	0.532	0.804	0.544	0.893	0.791	0.443	0.601	0.503	0.531	0.821
15	0.981	1.000	0.474	0.805	1.000	1.000	1.000	1.000	0.506	0.937	0.989	1.000
16	0.993	1.000	0.437	0.927	0.530	0.995	0.903	0.967	0.439	0.907	0.536	1.000
17	0.325	0.282	0.479	0.620	0.081	0.641	0.338	0.282	0.493	0.505	0.154	0.526
18	0.314	0.771	0.610	0.559	0.678	0.797	0.119	0.975	0.567	0.607	0.796	0.847
19	0.055	0.894	0.070	0.191	0.166	0.754	0.0603	0.879	0.094	0.179	0.264	0.950
20	0.977	0.953	0.957	0.875	0.785	0.989	1.000	1.000	0.964	0.896	0.851	1.000
Average	0.638	0.696	0.537	0.562	0.471	0.846	0.543	0.677	0.560	0.580	0.510	0.843
FPS	140.811	12.159	8.610	6.150	28.700	14.600						

Higher value indicates better performance. Red fonts indicate the best performances. Blue fonts indicate the second best ones. FPS: number of frames per second. References to color refer to the online version of this table

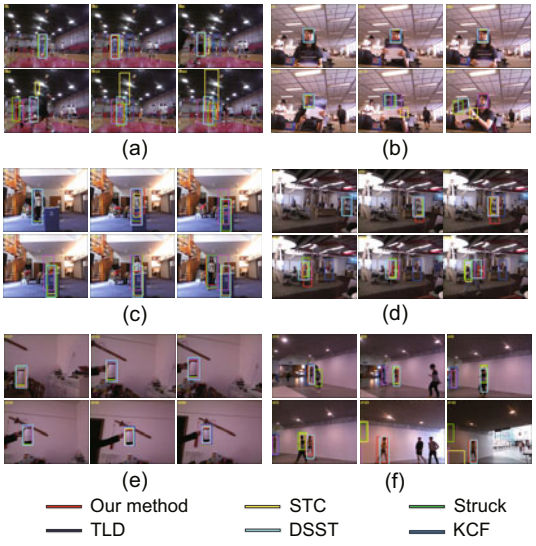

Fig. 6 Screenshots of tracking results: (a) Basketball1; (b) Face_occ5; (c) Tracking7.1; (d) Library2.2_occ; (e) Cup_move_1; (f) Walking_occ_long. References to color refer to the online version of this figure

Occlusion: The correlation filter-based trackers are not able to tackle the occlusion problem, which probably results in misleading of model learning. The proposed approach detects occlusion by the depth map and adjusts the learning rate correspondingly. In the experiments of image sequences with occlusion, the target can be relocated when it reappears as long as it is still within the context region. As shown in Table 3, our approach achieves the best results with an average SR of 82.3% and an average DP of 79.8%. The tracking results on some image sequences with occlusion are shown in Fig. 6.

Scale variation: Struck and KCF cannot adapt to scale variation. STC and TLD provide simple scale updating schemes which perform poorly in challenging tests. DSST trains a separate filter for scale estimation and achieves a reasonable performance with an average SR of 74.9% and an average DP of 75.3% (Table 4). In our method, we adopt a modified region growing method which can describe the contour of the target accurately. The improvements compared with DSST in terms of SR and DP are 8.3% and 6.1%, respectively.

Longer video sequences: We divide the sequences into four groups according to different numbers of frames, and evaluate the trackers on these groups (Table 5). In the first group, the number of frames in each sequence is smaller than 100. Compared with the second tracker DSST, the

Table 3 Performances on image sequences with occlusion

Sequence number	Success rate						Distance precision					
	KCF	DSST	TLD	Struck	STC	Ours	KCF	DSST	TLD	Struck	STC	Ours
1	0.389	0.493	0.115	0.885	0.089	0.751	0.343	0.441	0.231	0.745	0.157	0.732
6	0.526	0.617	0.658	0.296	0.133	0.704	0.643	0.658	0.631	0.350	0.154	0.663
7	0.945	0.500	0.939	0.506	0.494	0.967	0.990	0.525	0.954	0.537	0.513	0.994
8	0.374	0.186	0.014	0.060	0.110	0.900	0.207	0.019	0.125	0.217	0.097	0.705
11	0.766	0.846	0.940	0.169	0.453	0.960	0.861	0.896	0.867	0.158	0.516	0.886
13	0.617	0.877	0.773	0.565	0.857	0.571	0.643	0.935	0.795	0.421	0.864	0.805
15	0.981	1.000	0.474	0.805	1.000	1.000	1.000	1.000	0.506	0.937	0.989	1.000
17	0.325	0.282	0.479	0.620	0.081	0.641	0.338	0.282	0.493	0.505	0.154	0.526
19	0.055	0.894	0.070	0.191	0.166	0.754	0.0603	0.879	0.094	0.179	0.264	0.950
Average	0.543	0.557	0.464	0.412	0.325	0.823	0.537	0.534	0.499	0.423	0.357	0.798

Higher value indicates better performance. Red fonts indicate the best performances. Blue fonts indicate the second best ones. References to color refer to the online version of this table

Table 4 Performances on image sequences with scale variation

Sequence number	Success rate						Distance precision					
	KCF	DSST	TLD	Struck	STC	Ours	KCF	DSST	TLD	Struck	STC	Ours
1	0.389	0.493	0.115	0.885	0.089	0.751	0.343	0.441	0.231	0.745	0.157	0.732
4	1.000	0.997	0.116	1.000	0.305	1.000	0.360	0.945	0.135	1.000	0.379	1.000
5	0.827	1.000	1.000	0.827	0.835	0.976	0.770	1.000	1.000	0.879	0.921	1.000
9	0.362	0.991	0.216	0.310	0.647	0.707	0.094	0.991	0.267	0.423	0.687	0.664
17	0.325	0.282	0.479	0.620	0.081	0.641	0.338	0.282	0.493	0.505	0.154	0.526
18	0.314	0.771	0.610	0.559	0.678	0.797	0.119	0.975	0.567	0.607	0.796	0.847
Average	0.582	0.749	0.512	0.745	0.454	0.832	0.428	0.753	0.538	0.729	0.531	0.814

Higher value indicates better performance. Red fonts indicate the best performances. Blue fonts indicate the second best ones. References to color refer to the online version of this table

Table 5 Performances on image sequences with different lengths

Number of frames	Success rate						Distance precision					
	KCF	DSST	TLD	Struck	STC	Ours	KCF	DSST	TLD	Struck	STC	Ours
<100	0.868	0.875	0.730	0.793	0.598	0.902	0.842	0.914	0.749	0.773	0.610	0.921
100–200	0.561	0.741	0.412	0.511	0.467	0.755	0.445	0.735	0.428	0.506	0.518	0.802
200–300	0.554	0.673	0.657	0.634	0.448	0.823	0.448	0.672	0.671	0.628	0.495	0.797
>300	0.756	0.643	0.575	0.547	0.477	0.955	0.683	0.592	0.612	0.610	0.501	0.911

Higher value indicates better performance. Red fonts indicate the best performances. Blue fonts indicate the second best ones. References to color refer to the online version of this table

performance improvements achieved by our tracker on SR and DP are 2.7% and 0.7%, respectively. In the second group, the number of frames is between 100 and 200. The improvements compared with DSST on SR and DP are 1.4% and 6.7%, respectively. In the third group, the number of frames is between 200 and 300. The improvements compared with DSST on SR and DP are 15.0% and 12.5%, respectively. In the last group, the number of frames is more than 300. The improvements compared with the second tracker KCF on SR and DP are 19.9% and 22.8%, respectively. The above data demonstrates that our tracker is better than the others in each group, especially in longer video sequences. Thus, it can be concluded that our tracker is more practicable for tracking of longer sequences.

Sensitivities to parameters: Some original parameters are nearly the same as those in STC. For the additional parameters, we investigate the relationships between the settings and our tracking performance. SRs of using different settings of the region growing factor p and the occlusion detection ratio t are shown in Fig. 7. The region growing factor p, which limits the depth difference between the two pixels, is the key to the merging rule. SR is relatively high when p is between 0.025 and 0.040. p is

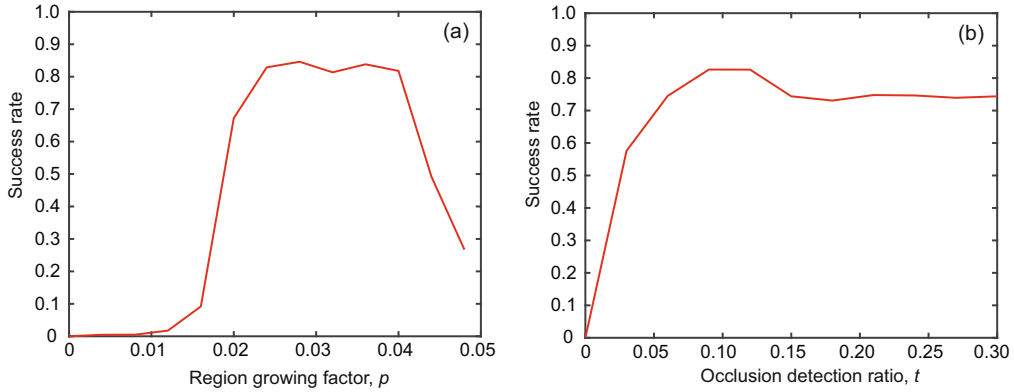

Fig. 7 Overall success rates of our tracker at different settings of the region growing factor p (a) and the occlusion detection ratio t (b)

initialized to 0.028 to achieve the best performance. The occlusion detection ratio t is set for occlusion judgement. Fig. 7 shows that when $t = 0.1$, SR achieves the best performance.

5 Conclusions

Based on the framework of the correlation filter and Bayesian inference, we propose an effective algorithm by exploiting depth information. Our method uses a depth context model to estimate the translation of the target. A modified region growing method is also adopted for scale estimation. Unlike the original updating scheme, we additionally propose an accurate model updating scheme based on occlusion detection using a depth map. Extensive experimental results show that the proposed algorithm outperforms the state-of-the-art trackers in terms of efficiency, accuracy, and robustness.

References

Adam, A., Rivlin, E., Shimshoni, I., 2006. Robust fragments-based tracking using the integral histogram. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.798-805. http://dx.doi.org/10.1109/CVPR.2006.256

Adams, R., Bischof, L., 1994. Seeded region growing. *IEEE Trans. Patt. Anal. Mach. Intell.*, **16**(6):641-647. http://dx.doi.org/10.1109/34.295913

Bolme, D.S., Beveridge, J.R., Draper, B.A., *et al.*, 2010. Visual object tracking using adaptive correlation filters. IEEE Conf. on Computer Vision and Pattern Recognition, p.2544-2550. http://dx.doi.org/10.1109/CVPR.2010.5539960

Cehovin, L., Kristan, M., Leonardis, A., 2011. An adaptive coupled-layer visual model for robust visual tracking. IEEE Int. Conf. on Computer Vision, p.1363-1370. http://dx.doi.org/10.1109/ICCV.2011.6126390

Chen, K., Lai, Y., Wu, Y., *et al.*, 2014. Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information. *ACM Trans. Graph.*, **33**(6):208-219. http://dx.doi.org/10.1145/2661229.2661239

Choi, C., Christensen, H.I., 2013. RGB-D object tracking: a particle filter approach on GPU. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, p.1084-1091. http://dx.doi.org/10.1109/IROS.2013.6696485

Danelljan, M., Häger, G., Khan, F.S., *et al.*, 2014a. Accurate scale estimation for robust visual tracking. British Machine Vision Conf., p.1-11.

Danelljan, M., Khan, F.S., Felsberg, M., *et al.*, 2014b. Adaptive color attributes for real-time visual tracking. IEEE Conf. on Computer Vision and Pattern Recognition, p.1090-1097. http://dx.doi.org/10.1109/CVPR.2014.143

Dinh, T.B., Vo, N., Medioni, G.G., 2011. Context tracker: exploring supporters and distracters in unconstrained environments. IEEE Conf. on Computer Vision and Pattern Recognition, p.1177-1184. http://dx.doi.org/10.1109/CVPR.2011.5995733

Everingham, M., Gool, L.V., Williams, C.K., *et al.*, 2010. The Pascal Visual Object Classes (VOC) challenge. *Int. J. Comput. Vis.*, **88**(2):303-338. http://dx.doi.org/10.1007/s11263-009-0275-4

Grabner, H., Matas, J., Gool, L.V., *et al.*, 2010. Tracking the invisible: learning where the object might be. IEEE Conf. on Computer Vision and Pattern Recognition, p.1285-1292. http://dx.doi.org/10.1109/CVPR.2010.5539819

Gupta, S., Girshick, R.B., Arbelaez, P., *et al.*, 2014. Learning rich features from RGB-D images for object detection and segmentation. ECCV, p.345-360. http://dx.doi.org/10.1007/978-3-319-10584-0_23

Hare, S., Saffari, A., Torr, P., *et al.*, 2011. Struck: structured output tracking with kernels. *IEEE Trans. Patt. Anal. Mach. Intell.*, **38**(10):263-270. http://dx.doi.org/10.1109/TPAMI.2015.2509974

Henriques, J.F., Caseiro, R., Martins, P., *et al.*, 2012. Exploiting the circulant structure of tracking-by-detection with kernels. ECCV, p.702-715. http://dx.doi.org/10.1007/978-3-642-33765-9_50

Henriques, J.F., Caseiro, R., Martins, P., *et al.*, 2015. High-speed tracking with kernelized correlation filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, **37**(3):583-596. http://dx.doi.org/10.1109/TPAMI.2014.2345390

Hickson, S., Birchfield, S., Essa, I.A., *et al.*, 2014. Efficient hierarchical graph-based segmentation of RGBD videos. IEEE Conf. on Computer Vision and Pattern Recognition, p.344-351.

Izadinia, H., Saleemi, I., Li, W., *et al.*, 2012. (MP) 2T: multiple people multiple parts tracker. IEEE Conf. on Computer Vision and Pattern Recognition, p.100-114. http://dx.doi.org/10.1007/978-3-642-33783-3_8

Kalal, Z., Mikolajczyk, K., Matas, J., 2012. Tracking-learning-detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, **34**(7):1409-1422. http://dx.doi.org/10.1109/TPAMI.2011.239

Kristan, M., Pflugfelder, R., Leonardis, A., *et al.*, 2015. The visual object tracking VOT2014 challenge results. IEEE Conf. on Computer Vision and Pattern Recognition, p.191-217.

Kumar, B.V., Mahalanobis, A., Juday, R.D., 2010. Correlation Pattern Recognition. Cambridge University Press, Cambridge.

Lee, D., Sim, J., Kim, C., 2014. Visual tracking using pertinent patch selection and masking. IEEE Conf. on Computer Vision and Pattern Recognition, p.3486-3493.

Li, X., Hu, W., Shen, C., *et al.*, 2013. A survey of appearance models in visual object tracking. *ACM Intell. Syst. Technol.*, **4**(4):58. http://dx.doi.org/10.1145/2508037.2508039

Li, Y., Zhu, J., 2014. A scale adaptive kernel correlation filter tracker with feature integration. ECCV, p.254-265. http://dx.doi.org/10.1007/978-3-319-16181-5_18

Li, Y., Zhu, J., Hoi, S., *et al.*, 2015. Reliable patch trackers: robust visual tracking by exploiting reliable patches. IEEE Conf. on Computer Vision and Pattern Recognition, p.353-361.

Liu, T., Wang, G., Yang, Q., 2015. Real-time part-based visual tracking via adaptive correlation filters. IEEE Conf. on Computer Vision and Pattern Recognition, p.4902-4912.

Luber, M., Spinello, L., Arras, K.O., 2011. People tracking in RGB-D data with on-line boosted target models. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, p.3844-3849. http://dx.doi.org/10.1109/IROS.2011.6095075

Ma, C., Yang, X., Zhang, C., *et al.*, 2015. Long-term correlation tracking. IEEE Conf. on Computer Vision and Pattern Recognition, p.5388-5396.

Park, Y., Lepetit, V., Woo, W., 2011. Texture-less object tracking with online training using an RGB-D camera. 10th IEEE Int. Symp. on Mixed and Augmented Reality, p.121-126. http://dx.doi.org/10.1109/ISMAR.2011.6092377

Ross, D.A., Lim, J., Lin, R.S., *et al.*, 2008. Incremental learning for robust visual tracking. *Int. J. Comput. Vis.*, **77**(1-3):125-141. http://dx.doi.org/10.1007/s11263-007-0075-7

Shu, G., Dehghan, A., Oreifej, O., *et al.*, 2012. Part-based multiple-person tracking with partial occlusion handling. IEEE Conf. on Computer Vision and Pattern Recognition, p.1815-1821. http://dx.doi.org/10.1007/s11263-007-0075-7

Smeulders, A.W., Chu, D., Cucchiara, R., *et al.*, 2014. Visual tracking: an experimental survey. *IEEE Trans. Patt. Anal. Mach. Intell.*, **36**(7):1442-1468. http://dx.doi.org/10.1109/TPAMI.2013.230

Song, S., Xiao, J., 2013. Tracking revisited using RGBD camera: unified benchmark and baselines. IEEE Int. Conf. on Computer Vision, p.233-240.

Teichman, A., Lussier, J.T., Thrun, S., 2013. Learning to segment and track in RGBD. *IEEE Trans. Autom. Sci. Eng.*, **10**(4):841-852. http://dx.doi.org/10.1109/TASE.2013.2264286

Wu, Y., Lim, J., Yang, M., 2013. Online object tracking: a benchmark. IEEE Conf. on Computer Vision and Pattern Recognition, p.2411-2418.

Yang, B., Nevatia, R., 2012. Online learned discriminative part-based appearance models for multi-human tracking. ECCV, p.484-498. http://dx.doi.org/10.1007/978-3-642-33718-5_35

Yang, H., Shao, L., Zheng, F., *et al.*, 2011. Recent advances and trends in visual tracking: a review. *Neurocomputing*, **74**(18):3823-3831. http://dx.doi.org/10.1016/j.neucom.2011.07.024

Yang, M., Wu, Y., Hua, G., 2009. Context-aware visual tracking. *IEEE Trans. Patt. Anal. Mach. Intell.*, **31**(7):1195-1209. http://dx.doi.org/10.1109/TPAMI.2008.146

Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: a survey. *ACM Comput. Surv.*, **38**(4):13. http://dx.doi.org/10.1145/1177352.1177355

Zhang, L., Maaten, L., 2014. Preserving structure in model-free tracking. *IEEE Trans. Patt. Anal. Mach. Intell.*, **36**(4):756-769. http://dx.doi.org/10.1109/TPAMI.2013.221

Zhang, K., Zhang, L., Liu, Q., *et al.*, 2014. Fast visual tracking via dense spatio-temporal context learning. ECCV, p.127-141. http://dx.doi.org/10.1007/978-3-319-10602-1_9