Multimed Tools Appl (2016) 75:1459–1480
DOI 10.1007/s11042-014-2305-7

Realtime and robust object matching with a large number of templates

Chaoqun Hong · Jianke Zhu · Jun Yu · Jun Cheng · Xuhui Chen

Received: 8 November 2013 / Revised: 9 September 2014 / Accepted: 29 September 2014 /
Published online: 19 November 2014

Abstract Most of conventional object matching methods are based on comparing local features, which are too computational demanding. Recently, Dominant Orientation Templates (DOT) were proposed to solve the efficiency issue. Although DOT obtains promising results, it still suffers the problem of wasting too many bits in representation and fragility when partial occlusion occurs. As the number of templates increase, the performance will decrease. Therefore, we propose a compact DOT representation with a fast partial occlusion handling approach. Instead of using seven orientations in the original implementation, we employ single orientation of the highest gradients for the proposed compact DOT representation (C-DOT). Consequently, the size of feature vectors is reduced from 8 bits to 3 bits. To efficiently tackle the partial occlusion, we introduce the C-DOT similarity map to store the matching scores of individual grids in each sliding window, which is used to further infer the occlusion map. The experimental results demonstrate that the proposed method outperforms DOT.

Keywords Realtime object matching · Dominant orientation templates · Compact representation · Partial occlusion handling

C. Hong · X. Chen
School of Computer and Information Engineering, Xiamen University of Technology,
Xiamen, China

J. Zhu
College of Computer Science, Zhejiang University, Hangzhou, China

J. Yu (✉)
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
e-mail: zju.yujun@gmail.com

J. Cheng
Shenzhen Institues of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

J. Cheng
The Chinese University of HongKong, Shatin, HongKong, China

1 Introduction

Object matching in large image collections and videos is now the burning issue in the research on computer vision [9, 22, 23, 25, 29, 33, 40, 41]. Differently from conventional content-based image matching, it aims at not only finding images related to the query object, but also providing actual location information[21, 48, 49].

There are plenty of schemes for object matching, such as sliding window searching [27, 42], branch-and-bound searching [17–19] and so on. Sliding window searching is a simple scheme, which slides the searching window in the testing image while comparing the features in the sliding window with the query object. Branch-and-bound is a classical algorithm for object matching. Specifically, it divides the images into smaller subimages and disposes those that are impossible to contain the query object. This operation runs recursively and finally we can find the subimage that is sufficiently small and likely to contain the query object.

No matter which scheme is chosen, further analysis always works on local patches since the object is only a part of the whole image. Therefore, the global features, such as color, shape, texture and so on, do not work very well in such case. On the other hand, local features are computed at every pixel using its neighborhood [5]. SIFT [24] and SURF [2] are the most representative feature descriptors, which reveal more representation power than global features for describing parts of images. Therefore, they are widely used in object matching [38, 39, 46], and could even achieve free-shape object matching [51] through combining with edge detection. Moreover, they are usually employed to train classifiers for the object matching. However, these local feature approaches are based on the statistics of local structures that typically involve with the heavy computational cost. Thus, it is hard to achieve the realtime performance for the object matching task [7, 8, 11, 12, 52].

To overcome the above issue, Dominant Orientation Templates (DOT) was proposed [10], which not only retains the merits of statistical methods, but also can take advantages of the grid representation in the sliding windows with the bitwise operations to greatly reduce the computational time on extracting the features and matching the object. However, there still exist some limitations on DOT method. First of all, it wastes some bits to represent the dominant orientations of gradients. A template matching model usually needs a lot of templates. Therefore, the memory consumption will increase as the number of templates increase. Moreover, it employs a naive scheme to compute the global score within the whole sliding window, which results in the reduction of performance when partial occlusion occurs.

In this paper, we propose an improved DOT descriptor method for object matching, which focuses on solving the problems mentioned above. Our contribution lies in two-fold. Firstly, we introduce a compact representation which only uses the orientations of strongest gradients in each grid. In contrast to the DOT representation, fewer bits are needed for our presented C-DOT method. We name it as C-DOT. Secondly, we calculate the similarity map for each sliding window based on the results of comparing C-DOT. This map consists of the matching score of each grid and reflects the visibility of the object. Therefore, it can be further employed to handle partial occlusion problem.

The rest of this paper is organized as follows. Section 2 reviews some previous approaches related to local features and template matching in object matching. In Section 3, we introduce compact representation of DOT. Section 4 presents our occlusion handling approach. Section 5 evaluates the effectiveness of our method by conducting the comprehensive experiments. Finally, we conclude this paper and prospect the future work in Section 6.

2 Related work

Object matching has already received intensive attentions. which is closely related to the feature extraction and template matching. In the following, we will briefly review the literature on these topics.

2.1 Feature extraction

Feature extraction is usually employed as a preprocessing step for most of object matching methods [5], which can significantly affects the overall performance in the real-world applications. Being capable of capturing the repeated regions, local feature is widely used in the various applications. Generally speaking, the common scheme of the conventional local feature extractors are made of two parts: feature detection and feature description. Feature detection tries to find the local patches that reflect the characteristics of regions while feature description distinguishes one region from other regions.

Scale Invariant Feature Transform (SIFT) [24] is the most successful local feature extractor, which is mainly based on the Harris corner detector and the 128-dimensional histogram of orientations. Although the discrimination performance of SIFT is promising, its computational complexity is far too high. Specifically, it usually takes seconds to extract features from an image of VGA size (640×480).

Inspired by the SIFT method, lots of research efforts have been devoted to finding the feature descriptors with low computational cost. Ke and Sukthankar [13] employ Principal Component Analysis (PCA) to reduce the dimensionality of SIFT descriptor. PCA-SIFT obtains even more robust results than the original SIFT implementation, which requires an extra step to project the extracted SIFT descriptors onto the low dimensional feature space. Speeded Up Robust Features (SURF) [2] uses Haar wavelet response, and reduces the dimensionality of feature vector from 128 to 64. Moreover, SURF demonstrates the good performance with less time and space consumption. Wu [45] took advantage of GPU hardware and implemented SiftGPU to significantly improve the speed to extract the SIFT features. Although PCA-SIFT, SiftGPU and SURF improve the speed for extracting the local features, these methods are still not fast enough to fulfill the requirement of realtime object matching in video sequences.

Some fast corner detectors are also proposed to facilitate realtime object tracking. FAST [30] and Faster [31] are extremely fast corner detectors, which are adapted to a realtime object matching system by Simon Taylor et al. [36]. Unfortunately, they needs an offline training process.

This work is also related to the spatial-temporal features which are employed in video analysis. To find the local features in videos, a lot of spatial-temporal descriptors were proposed [3, 15, 43], which are widely used in action recognition and video copy detection [20, 26, 44, 47, 50]. They are not designed for the realtime object retrieval, since there may be very few spatial-temporal features being detected in a static scene containing a lot of objects.

2.2 Template matching

Being capable of using very simple scheme to deal with different objects [27], template matching has been proved to be promising for object matching. The key idea of template matching is to the combine naive brute-force sliding window searching with fast matching.

It is hard for Branch-and-bound scheme to combine with template matching since it involves with the very complicated settings for the templates with the different sizes of the subimages during the matching process.

According to template matching scheme, the most critical problem is the feature selection. Early template matching methods are dependent on contours and employ Chamfer distance as the similarity measure [6]. The major drawback of these methods is that the performance is greatly affected by the fragile contour extracting methods with the binary thresholding.

Furthermore, some advanced template matching methods are presented by using the image gradients [34]. One of the most successful features is Histograms of Gradients (HOG) [4], which captures the local distributions of image gradients computed on a regular grid. However, extracting the HOG features involves with the high computational cost. Moreover, it is not easy to adapt it to the realtime applications. Ouyang et al. [28] computed rectangle sums and orthogonal Haar transform (OHT) based on integral images, which is faster than the naive brute-force searching methods. Some rotation-invariant features, i.e., Rotation-Invariant Fast Features (RIFF) [35] and Fourier Coefficients of radial projections (Forapro) [14], achieve the good matching performance. As most of them need a full re-computation of features in a sliding window, such features are too computational demanding to be integrated into the fast updating scheme of histograms in sliding windows [42].

Most recently, Dominant Orientation Templates (DOT) reveals the promising direction [10]. Differently from HOG feature, it quantizes the orientations of gradients by grid and only keeps the most significant orientations. Benefited from the compact feature representation, it utilizes the bit-wise operations and a part updating scheme similar to [42] to further improve the speed. Therefore, DOT is able to achieve real-time performance on a regular PC.

Although DOT shows good performance, there are still some limitations on it. We try to address these issues in this paper. First of all, it takes too many bits to represent the templates. There are only 8 types of the templates after computing the gradients for each grid in the image. Generally speaking, 3 bits is enough to represent a number of which the maximum value is 8. Therefore, we can reduce the memory consumption by using less bits to represent the templates. Additionally, it uses SSE2 instructions to compute the similarities of several templates simultaneously [16]. This improves the speed of template matching while discards the local scores. Although we could tune the threshold in DOT implementation to match the occluded regions, this will lead to a lot of false positive detection. Some grids in the false regions may also match the templates. As a result, these regions may get similar scores comparing to the occluded regions. The original DOT implementation cannot distinguish these two kinds of regions. Therefore, it is hard for DOT to handle partial occlusion. Differently from DOT method, we retain the local scores to build similarity maps. Thus, the visible parts are inferred from these similarity maps, which could be further employed to retrieve the occluded regions.

3 Compact dominant orientation templates

In this section, we present the proposed compact dominant orientation templates method. Firstly, we briefly review the original DOT method. Secondly, we study the compact representation for the DOT feature. Thirdly, we provide some detailed comparisons between DOT and compact DOT.

3.1 Review of DOT

We summarize the process of object matching using DOT method in Fig. 1. First of all, the query target is divided into several grids of the same size during the training process. Note that the size of the target and the size of sliding window are exactly the same. The size of grids needs to be set to a proper value, which should not be too large or too small. If the grid size is too large, the condition of similarity will be too strict. On the other hand, the DOT feature cannot capture useful information for the small grid size. Usually, the size of each grid is set to 7 × 7. Then, the orientations of gradients are computed at each grid and the extracted orientations are quantized into 7 bins. To this end, each grid is represented by an 8-bit unsigned integer. The bits in this integer are set to 1 if the corresponding orientations are dominant. The dominant orientation is set when the magnitude is larger than a threshold. If the last bit is set to one, it means that there is no dominant gradient in this grid. These integers are listed in the pre-defined order to form the template feature for a target. In the matching process, the whole image is also divided into the grids. Comparing to the DOT feature extracted from templates, only the orientation with the strongest gradient at each grid is kept. As the window slides, the histogram in this window will be dynamically assembled with the pre-computed gradient orientations and compared against the template with an energy function. Formally speaking, the energy function is represented by:

$$\delta(do(I, c + R) \in \mathbb{L}(T, R)) = 1 \text{if} L \otimes D \neq 0 \tag{1}$$

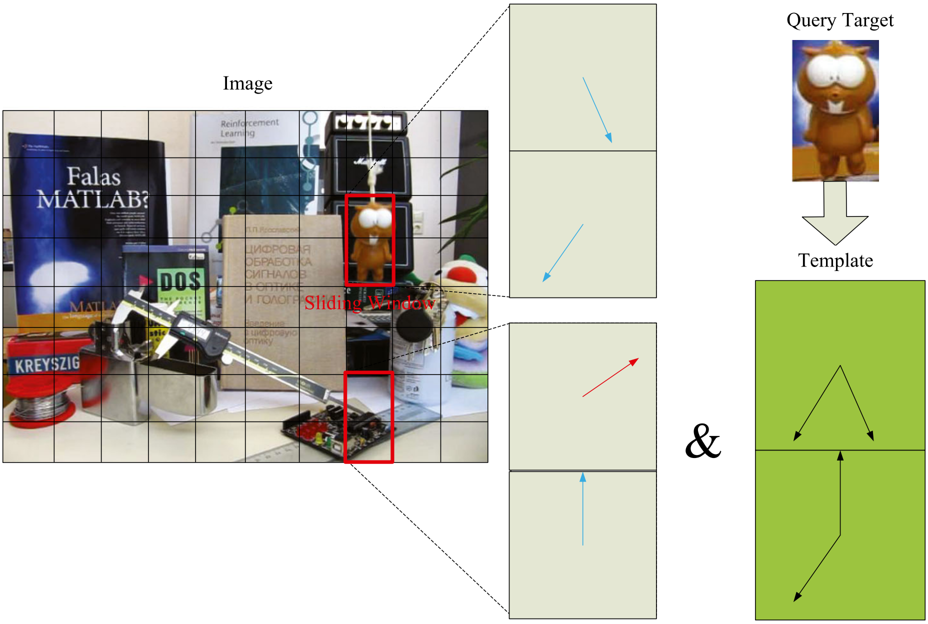


Fig. 1 The matching process using DOT. The orientation histograms in templates are compared with the histogram in each sliding window. The matched orientations are depicted by the blue arrows while the unmatched orientations are plotted by the red arrows. If the total number of matched orientation is large enough, a true detection will be reported in this window. The division of grids and features in this figure are only for illustration purpose, which do not reflect the actual data

In this function, $do(I, c + R)$ returns the orientation of the strongest gradient in the region R shifted by c in the input image I. L and D are features representing dominant gradient orientations in a grid of templates and sliding windows respectively. $\otimes$ is the bitwise AND operation. If $L \otimes D \neq 0$, the result of bitwise AND operation is non-zero, which means that the orientation of a grid in current sliding window matches one of the dominant orientation of the grid in the corresponding position of templates. In this way, the two grids are similar. $\mathbb{L}(T, R)$ can be written as follows:

$$\mathbb{L}(T, R) = o : \exists M \in \Lambda \mathrm{suchthat} o \in DO(\mathbb{W}(T, M), R) \tag{2}$$

DO is defined by:

$$DO(T, R) = \begin{cases} \mathbb{S}(T, R) & \mathrm{if} \mathbb{S}(T, R) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \tag{3}$$

with

$$\mathbb{S}(T, R) = \mathrm{ori}(T, l) : l \in \max \mathrm{mag}_k(R) \wedge \mathrm{mag}(T, l) > \gamma \tag{4}$$

where $\max \mathrm{mag}_k(R)$ is a set of locations for the top k strongest gradients in R and $\mathrm{mag}(T, l)$ gets the magnitude on location l in target T. In this way, $\mathrm{ori}(T, l)$ can get the dominant orientation on location l. Moreover, $DO(T, R)$ obtains a set of orientations of the strongest gradients in T. $\perp$ denotes that there is no reliable gradient information in the region. $\mathbb{W}(T, M)$ computes the warped image set Λ for T with a transformation M, which is used to achieve the invariance to small translation and rotations. The energy function employs the bitwise AND operation to obtain the number of matched orientations. The higher the number, the more similar the two histograms are. Finally, the algorithm sorts the numbers of different windows. The target is supposed to appear in the window with the highest number.

To further improve the matching speed, DOT takes advantage of three SSE2 instructions. Specifically, AND operation is employed to check whether two DOTs are similar. Moreover, it compared the results of checking similarity with zero. In addition, it fetches the most significant bits and gets the number of similar grids by looking up the table. Although DOT is quite simple, it is still powerful and achieves the promising results in practice.

3.2 Compacting DOT

In the object matching process, the performance heavily depends on templates. Therefore, the implementation artificially creates a lot of templates in order to account for various translations and rotations. For the DOT method, it aims at handling small translations and rotations by synthesizing the artificial templates using the original templates. With the default setting, it creates templates using the translations within $[-21, 21]$ and rotations for every 10° in 360°. Thus, there will be around 1,500 templates generated. Even with a clustering scheme, the total number of templates is still a few hundreds, which costs lots of memory to store these templates. Therefore, a compact representation for DOT is needed.

Although DOT is a kind of very simple representation, it still wastes too much memory space. As mentioned in the previous sub-section, only one bit is used in each byte of the DOT representations for the scanned images in the matching process. This is shown in Fig. 1. Consequently, there are only 8 types of DOT for these 1-orientational representations. Only 3 bits are actually needed to represent them. In this paper, we propose to just use 3 bits instead of 8 bits to represent the dominant orientations. Therefore, the new representation can compact DOT in order to reduce the memory requirement. We name this compact representation as C-DOT. The corresponding representations of DOT and C-DOT are shown in Table 1.

This idea seems quite straightforward. However, there are some issues to be solved for C-DOT representation:

- The DOT of the query target could contain 7 gradient orientations at most. That means the number of possible DOTs for the query target is 2^8. They cannot be compacted into 3 bits.
- Even if the DOT of the query target uses only one orientation, AND operation will not work when comparing the new histograms. Counting the total number of 1's in AND results will not lead to the correct similarity score.
- Differently from 8-bit representation, 3-bit representation cannot fill the whole byte. In this way, both a new bit alignment strategy and a new look-up table are needed.

For the first problem, we could maintain only one orientation for the DOT of the query target. This may reduce the chance that the orientations in the sliding window match the orientations in the templates. Therefore, the performance of DOT will be affected. According to our experimental results, 1-orientation templates approach may reduce the performance, and such reduction is not that significant. We will study it throughly in Section 5.

To solve the second problem, we use bitwise XOR operation instead of bitwise AND operation. When initializing the DOT of the query target and computing gradients of the images, we calculate their complementary values. For example, if we get a DOT on zero degree, the DOT for the query target will be '000' and the gradient of the image will be '111'. If two DOTs match, we will get '111'. Thus, we can compute the similarities between templates and sliding windows by counting the number of consecutive three 1's. According to the definition of C-DOT, (1) could be rewritten as:

$$\sigma(do(I, c + R) \in \mathbb{L}(T, R)) = 1 \mathrm{if} L \oplus D = 111 \tag{5}$$

$\oplus$ is bitwise XOR operation. Counting the number of consecutive three 1's is very time-consuming. Therefore, all the possible values of counting results are stored in the look-up table. This table will be computed beforehand. In this way, we could fetch the similarities instantly with the XOR results.

As for the third problem, we could put the new 3-bit representation in different size of units and have to leave the last few bits unused. In this way, the 3-bit C-DOT feature for one grid may appear in two consecutive byes. Due to the setting of SSE2 instructions, the size could be a byte (8 bits), a word (16 bits), a double word (32 bits) or a quadruple word (64 bits). The size affects the size of the look-up table (LUT) and the utilization rate of bits. The

Table 1 The Correspondence of DOT and C-DOT

DOT	C-DOT
00000001	000
00000010	001
00000100	010
00001000	011
00010000	100
00100000	101
01000000	110
10000000	111

utilization rate of bits further affects the processing speed. If the utilization rate is higher, the length of the bit vector will be reduced and the speed will be improved. We demonstrate this effect with a 128-bit vector which could be processed by SSE2 instructions. The effect is shown in Table 2. The size of LUT is not equal to 2^{bits} since the last few bits are wasted. Inserting the results of XOR into the lookup tables, we can get the number of similar grids in each unit.

3.3 Analysis of C-DOT

In contrast to the original DOT method, our proposed C-DOT approach has several advantages. The most important merit of the C-DOT representation is that the vector length of C-DOT is shorter than the original DOT. Therefore, the memory requirement is reduced. Moreover, the total number of bit-wise operations is greatly reduced in the energy function of C-DOT. The reduction of memory requirement may be not so ideal. Take C-DOT with 8-bit units as an example. Theoretically speaking, it costs only half of the memory required by DOT. However, we need to realign the bytes to fit the length that could be concurrently computed in the SSE2 instructions. The length is $128/8 = 16$. In this way, the last few bytes are wasted. In DOT, if 1 byte is wasted, the space for 1 feature is wasted. In C-DOT, if 1 byte is wasted, the space of 2 features is wasted. Therefore, C-DOT may waste more space in byte realignment.

Another advantage of C-DOT is that such compression is totally lossless. According to Table 1, the one-to-one correspondence between DOT and C-DOT does not lose any information during the compression process. It means that the effect on performance brought by C-DOT actually comes from the usage of templates with only 1 gradient orientation.

Comparing to DOT, C-DOT also loses the advantage of parallel computing the similarities of multiple grids. The bits in C-DOT representation are not well-aligned to fill the whole byte. The operations of retrieving the most significant bits in bytes make no sense. Therefore, we cannot make use of SSE2 instructions as DOT, and the results of looking up tables will be more than that in DOT. In this way, we have to use more ADD operations to get the number of similar grids after XOR operations and looking up tables. This will increase the computational cost since ADD costs more computational time than DOT. Fortunately, this issue can be avoided in the occlusion handling scheme introduced in the following section, in which we don't need to sum the scores of individual grids.

4 Occlusion handling

In this section, we present the occlusion handling method for the C-DOT representation. We first illustrate the key idea of the proposed occlusion handling approach, and then introduce the similarity map to detect occluded areas. Finally, we address the details of our implementation.

Table 2 The influence of unit sizes

Bits	Size of LUT	Contained Grids	Wasted Bits
8	2^6	$2 \times 16 = 32$	$2 \times 16 = 32$
16	2^{15}	$5 \times 8 = 40$	$1 \times 8 = 8$
32	2^{30}	$10 \times 4 = 40$	$2 \times 4 = 8$
64	2^{63}	$21 \times 2 = 42$	$1 \times 2 = 2$

4.1 Proposed occlusion handling approach

Generally speaking, there are always a lot of objects in the real-world scenes and the layout of these objects is very complicated. Moreover, occlusion among the objects commonly occurs in the object matching task, which severely deteriorates the performance of object matching systems. Therefore, there is a need of the effective occlusion handling schemes to deal with this issue in order to improve the performance of object matching.

The proposed occlusion handling approach is inspired by the idea of HOG-LBP [37], in which Wang et al. constructed an occlusion likelihood map for each ambiguous scanning window by utilizing the response of each block of the HOG feature to the global detector. As this map indicates the visibility of each block, it is then segmented by mean-shift clustering algorithm to estimate the occluded regions and the un-occluded regions. Finally, the part detectors are applied to obtain the final classification on the current scanning window. This method depends on the classification scores of SVM, which is typically very computational expensive.

In contrast to HOG-LBP, the proposed occlusion handling approach makes use of the energy function of DOT to reduce the computational cost. There is a flow in the original implementation of energy function. The energy function of DOT performs AND operation on 128 bits in parallel and counts the total number of 1's using two other operations. Therefore, the similarities of 16 grids are computed with the lookup table at the same time. It improves the speed of the matching process, however, the local scores are lost and only a global decision will be made. Although we can loose the detecting threshold to make DOT tolerant to some partial occlusions, it may increase the chances of false detections. Thus, we have to adapt it to improve the performance in the case of the partial occlusions.

4.2 Occlusion handling with dot similarity maps

Based on the above idea of occlusion handling, we try to modify the results of lookup tables for C-DOT. Instead of matching targets and sliding windows by summing the similarity scores of individual grids, we compute the similarity for each grid separately. As a result, we not only know how many grids are similar, but also can obtain the list of similar grids. Take 8-bit C-DOT as an example. If we get 00111111 after XOR operation, we will get 2 after looking up tables without occlusion handling while we will get [0, 1] with occlusion handling. By the similarity score for each grid, we construct a similarity map for the scanning window. We name this map as a DOT similarity map.

HOG-LBP employs mean-shift to segment the likelihood map. With proper settings, mean-shift could cluster the grids with relatively high scores and relatively low scores. In this way, HOG-LBP is able to distinguish occluded regions and un-occluded regions. Such scheme does not work with our DOT similarity map. The situation of DOT is different from it since the DOT similarity map is a binary image. In DOT representation, the energy function only indicates whether two grids are similar rather than how similar they are. If there exist some noises, two similar grids may not be matched. On the other hand, some different grids may fortunately matched. In these situations, 0 or 1 will be returned and no intermediate values between 0 and 1 will be returned. If they are shown on the similarity map, we could see some small holes. To fill these small holes, we apply the median filter on the DOT similarity map to remove these noisy grids. Median filter is defined as:

$$m(x, y) = M(f(x+i, y+j))|i| < w, |j| < r \tag{6}$$

Based on intensive experiments, matched grids are usually dispersed in false windows while they are usually concentrated in true windows. From this observation, we can employ a very fast connected component detector to find the largest connected component in this map, which manifests the largest visible part of the query target. If the size component is above an empirical threshold, we determine that the query target exists in this window.

4.3 Implementation details

We summarize the details of the proposed occlusion handling approach in Fig. 2. In the training process, artificially templates are created with variations on target images. Templates are represented by C-DOT according to the previous section. In the matching process, testing images are are also represented by C-DOT. A matching window slides in each testing image to find candidate positions of the target. To reduce the processing time, we adopt the same detection threshold as the original DOT, which is defined as T_w. Moreover, we only construct the DOT similarity maps for the windows of which the overall scores are larger than T_w. Therefore, lots of windows that are apparently not similar to the query target will be disposed directly. In our experimental settings, T_w is usually smaller than the one used in the original DOT implementation in order to avoid missing some true regions with the small global scores. Then, we retrieve and refine the similarity map step by step according to the idea described in the previous subsection. With the similarity map, we propose two criteria to measure whether a connected component should be a good detection. One is the area ratio of this component, which computes the ratio between the area of the component and the total area of the window. We denote it as the area score. The other is the similarity per grid inside this component, which calculates the ratio between the sum of similarities inside the component and the area of the component. We denote it as the valid score. Both of them are dependent on the thresholds T_a and T_v. T_a reflects how much occlusion the system can handle. Moreover, T_v reflects how much noise the system can handle. If they are larger than T_a and T_v respectively, we determine that there may be a good detection in this window.

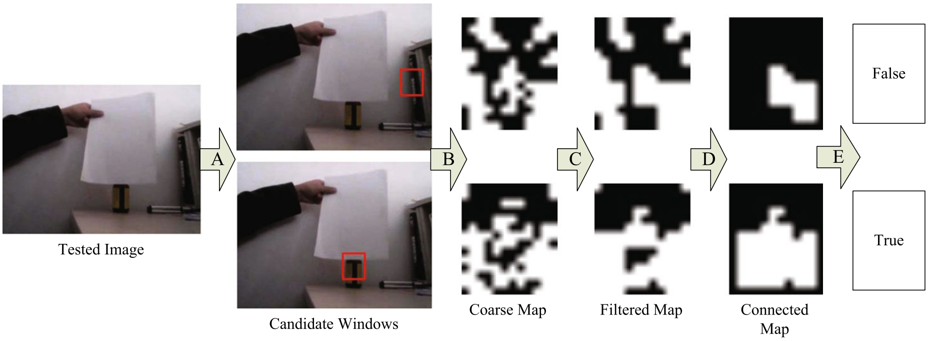


Fig. 2 The flowchart of the proposed occlusion handling approach. A: All sliding windows in the tested image are filtered by T_w; B: The sliding windows that pass the step A are chosen as candidate windows. The similarity maps are computed for them. After this step, the similarity maps are still coarse since they may be affected by noises; C: The coarse maps are filtered by Median Filter to get rid of some noises; D: The filtered maps are further processed with a connected component detector. E: The candidate windows are selected with criteria T_a and T_v . We can get the final decision by sorting them with the matching scores

This window is considered to be a candidate window. Finally, the decision is made by sorting the scores of candidate windows. The candidate window with the top score is supposed to be the position of target. We have evaluated the different scores during the detection process. In the proposed approach, we employ the valid score while the global score is use in the original DOT implementation.

In the above occlusion handling scheme, We only compute local scores while global scores of comparing features will not be calculated. Therefore, we could reduce the computation of a large number of ADD operations in C-DOT. On the other hand, the proposed occlusion handling scheme is highly portable. We could also extend the original DOT implementation a little bit to make it suitable for occlusion handling.

5 Experimental evaluation

In this section, we present the details of our experimental implementation and report the results of performance evaluation on object matching. We demonstrate that the proposed approach is effective to handle the partial occlusions in the object matching.

5.1 Experimental setup

To evaluate the efficacy of the object matching system, we evaluate it on several recent datasets, such as MILtrack [1] and PROST [32]. There are three QCIF (320×240) sequences in MILtrack dataset and four CIF (640 × 480) sequences in PROST dataset. The information on these datasets is summarized in Table 3. In these sequences, the ground truths of only one in five frames are given. All the experiments are performed on a regular PC with Intel i7-2640M CPU and 4GB RAM. Unless it is expressly stated, we create templates using the translations within $[-21, 21]$ and rotations for every 10° in 360°. In this way, the number of templates is 1548. In our experiments, we always turn on SSE2 optimization for the original DOT.

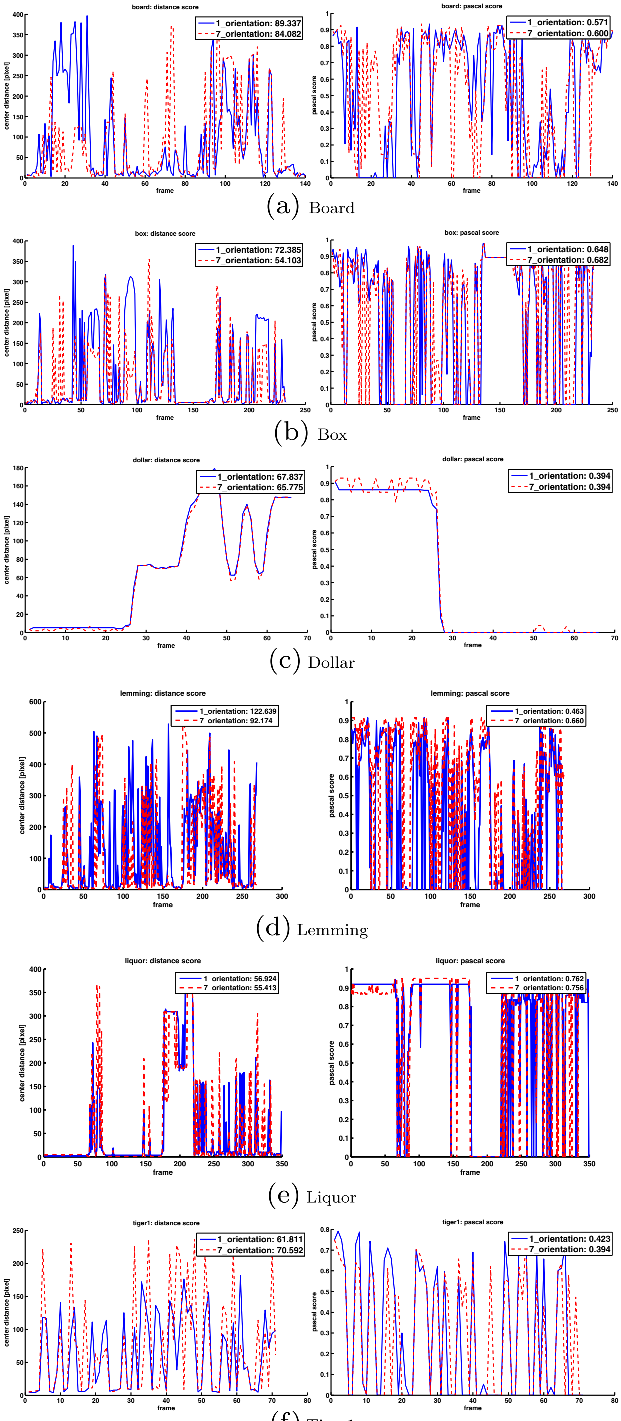
Additionally, we take advantage of the evaluation tool developed in PROST testbed [32], which provides two evaluation metrics. One is the distance score that calculates the mean distance of the tracking rectangle to annotated ground truth. The smaller the distance score is, the better the performance is. The other is the PASCAL score which is defined as follows:

$$score = \frac{area(ROI_D \cap ROI_{GT})}{area(ROI_D \cup ROI_{GT})} \tag{7}$$

The PASCAL score mainly measures the overlap of the detected bounding box ROI_D and the ground truth bounding box ROI_GT. The larger the PASCAL score is, the better

Table 3 The sequences in the datasets

Name	Number of Frames	Size of the Target
Board	698	185 × 153
Box	1161	86 × 112
Dollar	328	62 × 98
Lemming	1336	69 × 113
Liquor	1741	72 × 209
Tiger1	354	38 × 42
Tiger2	366	34 × 39



(a) Board

(b) Box

(c) Dollar

(d) Lemming

(e) Liquor

(f) Tiger1

Fig. 3 The performance comparisons of 1-orientation templates and 7-orientation templates

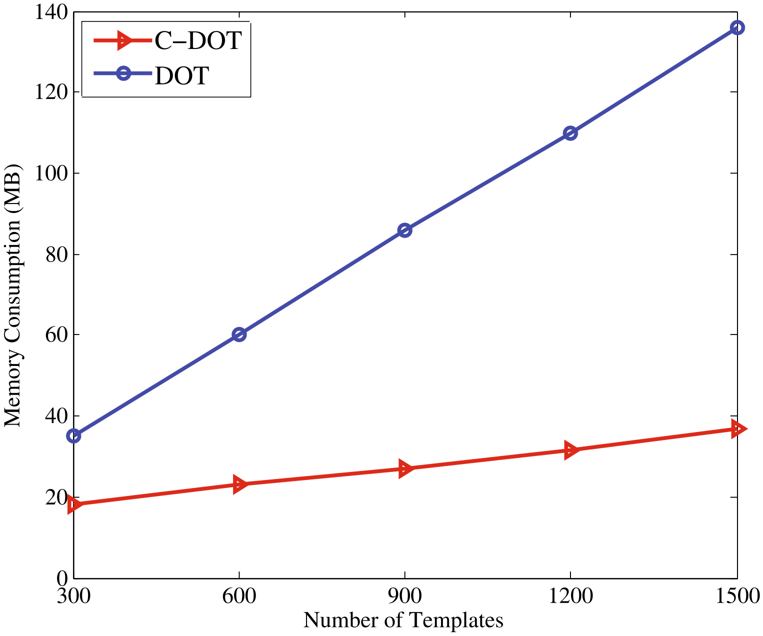


Fig. 4 The performance comparisons of 1-orientation templates and 7-orientation templates

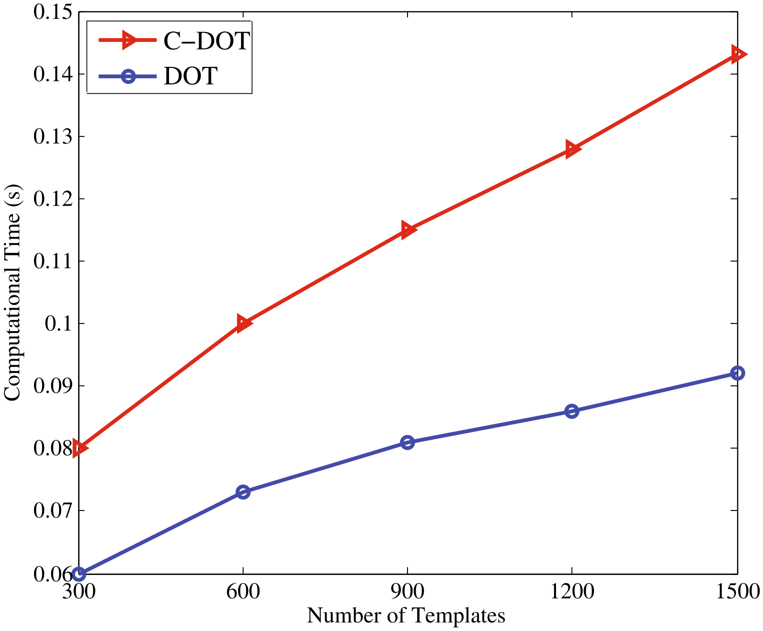
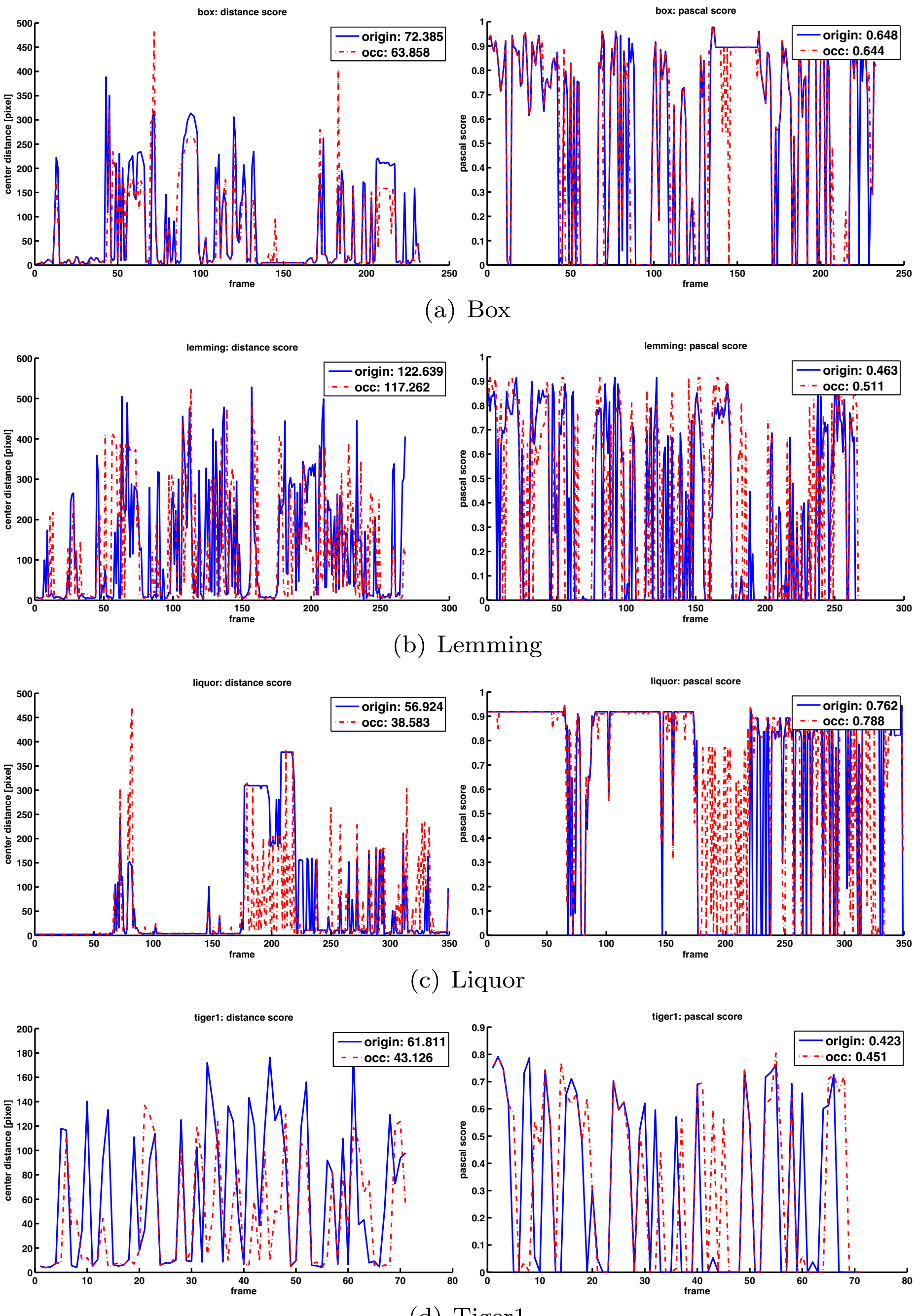


Fig. 5 The comparison of computational time between 1-orientation templates and 7-orientation templates



(a) Box

(b) Lemming

(c) Liquor

(d) Tiger1

Fig. 6 The performance comparisons of with/without the proposed partial occlusion handling approach. The result with occlusion handling is labeled as "occ" and the result without occlusion handling is labeled as "origin"

the performance is. Beside these two metrics, we introduce detected rate to compare the performance of different methods. Detected rate is defined as:

$$d_{rate} = \frac{frame_{detected}}{frame_{total}} \tag{8}$$

5.2 Experiments on compact DOT

Before using C-DOT instead of DOT, we should know whether the performance is significantly reduced. As shown in Fig. 3, we find that the presented C-DOT is effective through comparing the performance using the above sequences. It can be seen that the performance of 1-orientation templates is quite close to the result of 7-orientation templates at most of the cases. In addition, the performance of 1-orientation templates is even better than 7-orientation templates in some cases. Take the Box sequence as an example, the performance of 1-orientation templates is better than 7-orientation templates between the 60th frame and the 80th frame. For the Tiger1 sequence, 1-orientation templates outperform 7-orientation templates. In the cases that C-DOT outperforms DOT, the targets are with complex textures. In these cases, false matches are likely to occur with DOT. Therefore, we can adopt the DOT representation with only the dominant orientations of regions.

Furthermore, we investigate the advantage of C-DOT by comparing the memory requirement and computational time. In this experiment, we employ both 8-bit unit and 16-bit unit to compact DOT templates. 8-bit unit wastes the most bits among the choices in Section 3 while it has the smallest look-up table. 16-bit unit wastes fewer bits than 8-bit unit. The look-up tables of 32-bit units and 64-bit units are too large to be used. Using the Dollar sequence, we demonstrate the results of memory requirement and computational time varying with the number of templates, which are shown in Figs. 4 and 5 respectively. It can be observed that the memory consumption is reduced with C-DOT representation. We also notice that the time consumption of C-DOT is higher than DOT, which is coincided with our previous discussion in Section 3. This phenomenon happens to all the sequences.

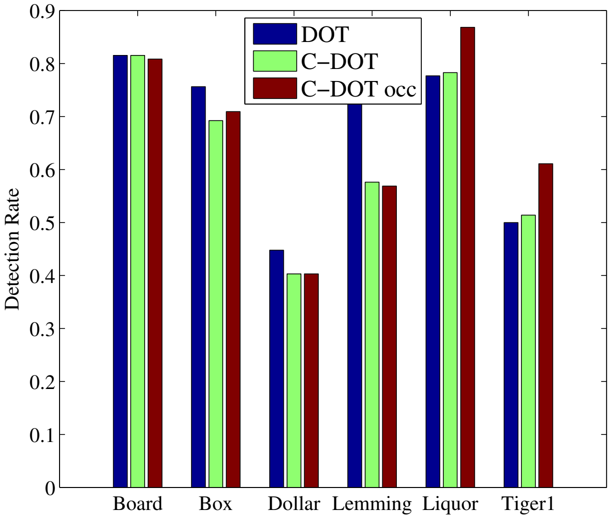


Fig. 7 The comparison of detected rates for all video sequences

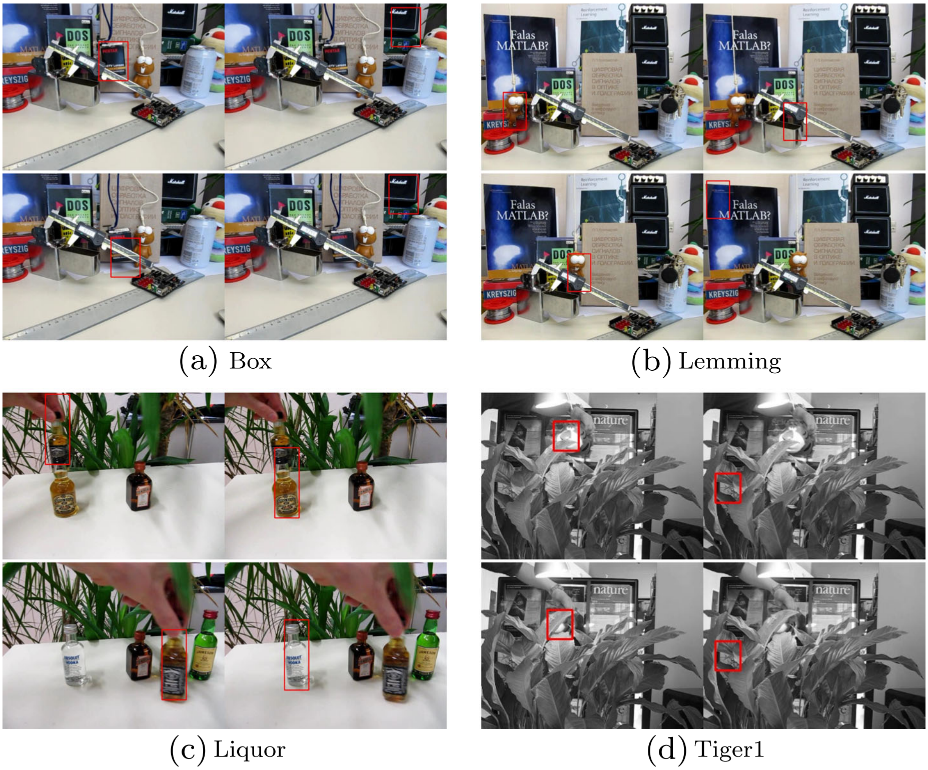
5.3 Experiments on partial occlusions

In this subsection, we evaluate the proposed occlusion handling approach. In our experiments, we employ the presented C-DOT representation, and the parameters mentioned in Section 4 are listed as follows:

$$\begin{cases} T_w = 0.6 \\ T_a = 0.3 \\ T_v = 0.8 \end{cases} \tag{9}$$

It means that the visible area of the target should not be less than 30 % and the noisy part in this visible area should be less than 20 %. For the median filter, we set the radius $w = r = 3$.

First of all, we study the effectiveness by comparing the object matching performance. The comparison results are plotted in Figs. 6 and 7. Among these videos, Liquor


(a) Box

(b) Lemming

(c) Liquor

(d) Tiger1

Fig. 8 Examples of the detection results. Four groups from four sequences are shown. In each group, the images in left columns show the detecting results with the proposed partial occlusion handling approach while the images in the right columns show the detecting result with original DOT and without partial occlusion handling. With the proposed occlusion handling approach, we could find the occluded targets successfully. If no occlusion handling scheme is applied, false regions are found. In the Box and Liquor sequences, some false regions which look similar to the target are located. In the Lemming sequence, the original DOT only finds some randomly false regions

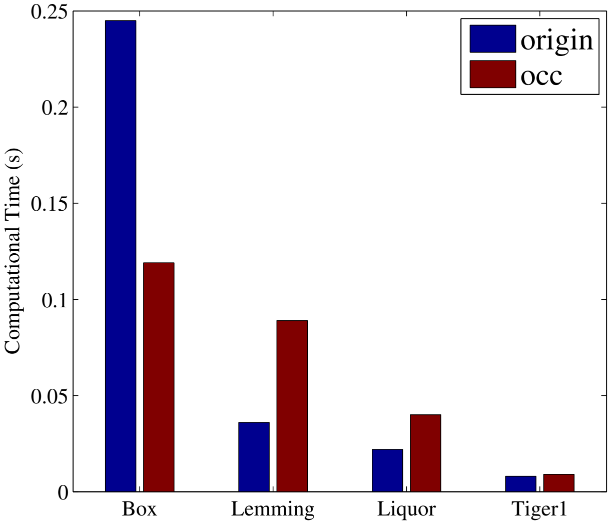


Fig. 9 The comparison of computational time with and without the proposed partial occlusion handling approach. The result with occlusion handling is labeled as “occ” and the result without occlusion handling is labeled as “origin”

and Tiger1 contain more occlusion cases than the Box sequence, and the improvements on both scores and detected rates are more significant. Box contained less occlusion cases and the improvement is marginal. Although the detected rate for Lemming is not significantly improved with occlusion handling, we could find the locations of the target more exactly. Therefore, we get higher distance scores and PASCAL scores. Due to the wide performance gap between DOT and C-DOT for Lemming, the performance of C-DOT is still below DOT even with occlusion handling. Although there are few occlusions in Board and Dollar, the performance is still almost the same as the original DOT implementation. Additionally, some occluded regions that are successfully detected, as shown in Fig. 8.

Also, we investigate the efficiency of the proposed occlusion handling approach, as shown in Fig. 9. The computational time with occlusion handling is usually higher than that without occlusion handling. Since occlusion handling gets rid of some false regions before the sorting operation in the final decision stage, it reduces the computational time on sorting. Therefore, the influence on the processing time is not very significant. For the Box sequence, occlusion handling scheme even improves the speed. It is worthy of mentioning that the speed of DOT is greatly dependent on the value of T_w. Significant speedup could be achieved by choosing a proper value for it. The experimental results demonstrate that DOT can achieve realtime performance even with the proposed occlusion handling approach.

6 Conclusion

In this paper, we have presented a novel method to match objects with low computational cost and high robustness. It offers several distinct advantages over the state-of-the-art approach. The presented method reduces the memory requirement for DOT feature vectors using only the dominant orientation of the highest gradient in the grids and compact it with less bits. Moreover, we tackled the partial occlusions problem by calculating and analyzing the similarity maps to obtain the connected visible parts of the query object in sliding windows. The encouraging experimental results showed that our method performs better than the original approach, especially on the occluded cases.

In the future, there will be plenty of work to do. As DOT is likely to stuck with the small edges, it indicates that the last bits will always be set to one in templates for images with the complicated textures. This may lead to lots of false positive detections. Moreover, the processing time increases significantly when the total number of templates grows. Therefore, we will explore the efficient indexing scheme.

Acknowledgments This work is supported by the Natural Science Foundation of China (61472110, 61202145, 61100104 and 61065007), the Program for New Century Excellent Talents in University (No. NECT-12-0323) and the Natural Science Foundation of Fujian Province of China (2012J01287 and 2014J01256).

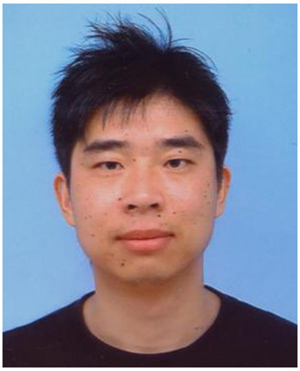
References

1. Babenko B, Yang MH, Belongie S (2009) Visual tracking with online multiple instance learning. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 983–990
2. Bay H, Ess A, Tuytelaars T, Gool LV (2008) Surf: Speeded up robust features. Comp Vision Image Underst 110:346–359
3. Bregonzio M, Gong S, Xiang T (2009) Recognising action as clouds of space-time interest points. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1948–1955
4. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, vol 1. IEEE, pp 886–893
5. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: Ideas, influences, and trends of the new age. ACM Comput Surv 40(2):1–60
6. Gavrila D, Philomin V (1999) Real-time object detection for ”smart” vehicles. In:IEEE international conference on computer vision, vol 1. IEEE, pp 87–93
7. Guan N, Tao D, Luo Z, Yuan B (2012) Nenmf: An optimal gradient method for nonnegative matrix factorization. IEEE Trans Signal Proc 60(6):2882–2898
8. Guan N, Tao D, Luo Z, Yuan B (2012) Online nonnegative matrix factorization with robust stochastic approximation. IEEE Trans Neural Networks Learn Syst 23(7):1087–1099
9. Hajdu A, Pitas I (2007) Optimal approach for fast object-template matching. IEEE Trans Image Process 16(8):2048–2057
10. Hinterstoisser S, Lepetit V, Ilic S, Fua P, Navab N (2010) Dominant orientation templates for real-time detection of texture-less objects. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 2257–2264
11. Hong Z, Mei X, Prokhorov D, Tao D (2013) Tracking via robust multi-task multi-view joint sparse representation. In: IEEE international conference on computer vision. IEEE, pp 1–8
12. Hong Z, Mei X, Tao D (2012) Dual-force metric learning for robust distracter-resistant tracker. In: European conference on computer vision. Springer, pp 513–527
13. Ke Y, Sukthankar R (2004) Pca-sift: A more distinctive representation for local image descriptors. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 506–513
14. Kim HY (2010) Rotation-discriminating template matching based on fourier coefficients of radial projections with robustness to scaling and partial occlusion. Pattern Recog 43:105–119
15. Klaser A, Marszalek M, Schmid C (2008) A spatio-temporal descriptor based on 3d-gradients. In: British machine vision conference, pp 995–1004. PASCAL EPrints
16. Klimovitski A (2001) Using sse and sse2 : Misconceptions and reality. Intel developer update magazine:1–8
17. Lampert C (2010) An efficient divide-and-conquer cascade for nonlinear object detection. In:IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1022–1029
18. Lampert CH (2009) Detecting objects in large image collections and videos by efficient subimage retrieval.In: IEEE international conference on computer vision. IEEE

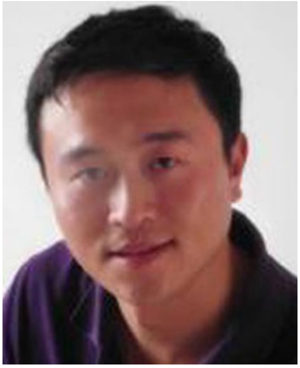
19. Lampert CH, Blaschko MB, Hofmann T (2009) Efficient subwindow search: A branch and bound framework for object localization. IEEE Trans Pattern Anal Mach Intell 31:2129–2142
20. Li H, Tang J, Wu S, Zhang Y, Lin S (2010) Automatic detection and analysis of player action in moving background sports video sequences. IEEE Trans Circ Syst Video Technol 20(3):351–364
21. Li H, Wang X, Tang J, Zhao C (2013) Combining global and local matching of multiple features for precise retrieval of item images. Multimedia Systems 19(1):37–49
22. Li P, Wang M, Cheng J, Xu C, Lu H (2013) Spectral hashing with semantically consistent graph for image indexing. IEEE Trans Multimedia 15(1):141–152
23. Li G, Wang M, Lu Z, Hong R, Cha T (2012) In-video product annotation with web information mining. ACM Trans Multimed Comput Commun Appl 8(4):1–55
24. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
25. McFee B, Galleguillos C, Lanckriet G (2010) Contextual object localization with multiple kernel nearest neighbor. IEEE Trans Image Process 20(2):570
26. Mezaris V, Kompatsiaris I, Boulgouris N, Strintzis M (2004) Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval. IEEE Trans Circ Systems Video Technol 14(5):606–621
27. Olson CF, Huttenlocher DP (1997) Automatic target recognition by matching oriented edge pixels. IEEE Trans Image Process 6(1):103–113
28. Ouyang W, Zhang R (2010) W.K.C.: Fast pattern matching using orthogonal haar transform. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 3050–3057
29. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1–8
30. Rosten E, Drummond T (2005) Fusing points and lines for high performance tracking. In:IEEE international conference on computer vision, vol 2. IEEE, pp 1508–C1511
31. Rosten E, Porter R, Drummond T (2010) Faster and better: A machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell 32:105–119
32. Santner J, Leistner C, Saffari A, Pock T, Bischof H (2010) Prost: Parallel robust online simple tracking. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 723–730
33. Sivic J, Zisserman A (2008) Efficient visual search of videos cast as text retrieval. IEEE Trans Pattern Anal Mach Intell 31:591–606
34. Steger C (2002) Occlusion, clutter, and illumination invariant object recognition. In: International archives of photogrammetry and remote sensing
35. Takacs G, Chandrasekhar V, Tsai S, Chen D, Grzeszczuk R, Girod B (2010) Unified real-time tracking and recognition with rotation-invariant fast features. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 934–941
36. Taylor S, Rosten E, Drummond T (2009) Robust feature matching in 2.3μs. In: IEEE computer society conference on computer vision and pattern recognition workshops. IEEE, pp 15–20
37. Wang X, Han TX, Yan S (2009) An hog-lbp human detector with partial occlusion handling. In: IEEE international conference on computer vision. IEEE, pp 32–39
38. Wang M, Gao Y, Lu K, Rui Y (2013) View-based discriminative probabilistic modeling for 3D object retrieval and recognition. IEEE Trans Image Process 22(4):1395–1407
39. Wang M, Li H, Tao D, Lu K, Wu X (2012) Multimodal graph-based reranking for web image search. IEEE Trans Image Process 21(4):4649–4661
40. Wang M, Ni B, Hua X, Chua T (2012) Assistive tagging: a survey of multimedia tagging with human-computer joint exploration. ACM Comput Surv 4(4). Article 25
41. Wang M, Hua X, Hong R, Tang J, Qi G, Song Y (2009) Unified video annotation via multigraph learning. IEEE Trans Circ Syst Video Technol 19(5):733–746
42. Wei Y, Tao L (2010) Efficient histogram-based sliding window. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 3003–3010
43. Willems G, Tuytelaars T, Gool LV (2008) An efficient dense and scale-invariant spatio-temporal interest point detector. In: European conference on computer vision, pp 650–663. LNCS
44. Willems G, Tuytelaars T, Gool LV (2008) Spatio-temporal features for robust content-based video copy detection. In: ACM international conference on multimedia and information retrieval. ACM, pp 283–290
45. Wu C (2007) Siftgpu:A gpu implementation of scale invariant feature transform. http://cs.unc.edu/ccwu/siftgpu
46. Wu Z, Ke Q, Isard M, Sun J (2009) Bundling features for large scale partial-duplicate web image search. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 25–32

47. Zha ZJ, Wang M, Zheng YT, Yang Y, Hong R, Chua TS (2012) Interactive video indexing with statistical active learning. IEEE Trans Multimedia 14(1):17–27
48. Zha ZJ, Yang L, Mei T, Wang M, Wang Z (2009) Visual query suggestion. In: the 17th ACM international conference on Multimedia. ACM, pp 15–24
49. Zha ZJ, Yang L, Mei T, Wang M, Wang Z, Chua TS, Hua XS (2010) Visual query suggestion: Towards capturing user intent in internet image search. ACM Transactions on Multimedia Computing. Commun Appl 6(3):1–19
50. Zha ZJ, Zhang H, Wang M, Luan H, Chua TS (2013) Detecting group activities with multi-camera context. IEEE Trans Circ Syst Video Technol 23(5):856–869
51. Zhang Z, Cao Y, Salvi D, Oliver K, Waggoner J, Wang S (2010) Free-shape subwindow search for object localization. In: IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 1086–1093
52. Zhou T, Tao D (2013) Shifted subspaces tracking on sparse outlier for motion segmentation. In: International joint conference on artificial intelligence. ACM, pp 1946–1952

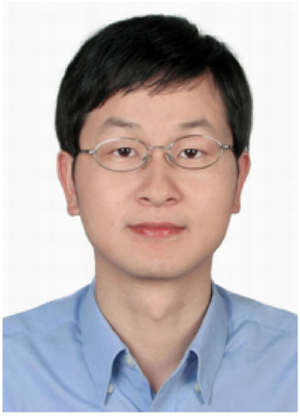
Chaoqun Hong is currently a lecturer in the department of Computer Science, Xiamen University of Technology, PR China. He received the Ph.D. degree in 2011 from Zhejiang University, PR China. His research interests include video codec, image processing, computer vision and patter recognition.



Jianke Zhu is currently an associate professor in Zhejiang University. He received his Ph.D. degree in Computer Science and Engineering from Chinese University of Hong Kong. He was a postdoc in BIWI computer vision lab of ETH Zurich. Dr. Zhu's research interests include computer vision and multimedia information retrieval. He is a member of the IEEE.



Dr. Jun Yu received the B.Eng. and Ph.D. degrees from Zhejiang University, Zhejiang, China. He is currently a Professor with the School of Computer Science and Technology, Hangzhou Dianzi University. From 2009 to 2011, he was with Singapore Nanyang Technological University. From 2012 to 2013, he was a Visiting Researcher with Microsoft Research Asia. Over the past years, his research interests include multimedia analysis, machine learning, and image processing. He has authored and co-authored more than 50 scientific articles. He has (Co-)Chaired for several special sessions, invited sessions, and workshops. He served as a Program Committee Member or reviewer of top conferences and prestigious journals. He is a Professional Member of the IEEE, ACM, and CCF.



Jun Cheng received the B.Eng., B.Fin., and M.Eng. degrees from the University of Science and Technology of China, China, in 1999 and 2002, respectively, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2006. He is currently with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, as a Professor and the Director of the Laboratory for Human Machine Control. His current research interests include computer visions, robotics, machine intelligences, and control.

Xuhui Chen is currently the deputy secretary of the department of Computer Science, Xiamen University of Technology, PR China. He received the Ph.D. degree in 2004 from Xi'an Jiaotong University. He was a postdoc in Arizona State University of U.S.A. His research interests include wireless sensor network and business intelligence.