Fast Detection of Multiple Textureless 3-D Objects

Hongping Cai, Tomáš Werner, and Jiří Matas

Center for Machine Perception, Czech Technical University, Prague

Abstract. We propose a fast edge-based approach for detection and approximate pose estimation of multiple textureless objects in a single image. The objects are trained from a set of edge maps, each showing one object in one pose. To each scanning window in the input image, the nearest neighbor is found among these training templates by a two-level cascade. The first cascade level, based on a novel edge-based sparse image descriptor and fast search by index table, prunes the majority of background windows. The second level verifies the surviving detection hypotheses by oriented chamfer matching, improved by selecting discriminative edges and by compensating a bias towards simple objects. The method outperforms the state-of-the-art approach by Damen et al. (2012). The processing is near real-time, ranging from 2 to 4 frames per second for the training set size $\sim 10^4$.

1 Introduction

We address scalable near real-time localization and detection of multiple rigid textureless 3-D objects with complex shapes. The objects may be presented in an arbitrary pose and the algorithm should provide an approximate estimate of the pose. Problems of this type arise, for instance, in robotics, where one needs to recognize and localize objects to facilitate manipulation.

The problem is challenging. Impressive results have been achieved in recognition of textured objects using affine-covariant detectors [15] and descriptors attached to them [13,3], but these methods do not apply to textureless objects.

Scanning window methods have shown significant progress in two-class object detection [19,3,12,5]. These methods are robust but not easily extendable to a large number of objects and poses. Moreover, the two-class detectors often need many training samples per object-pose which is unrealistic to assume.

The most informative local features on textureless objects are edges, caused mainly by discontinuities in depth or curvature and thus carrying information about shape. Our representation is thus edge-based, requiring a single training image per object-pose, acquired by an uncalibrated camera. No other information than edges (such as color or intensity) is used.

We propose a new two-stage cascaded detection method, combining a scanning window approach with an efficient voting procedure and a verification stage. At each position of the scanning window, novel edge-based features, computed in constant time, vote for each object-pose. This first stage prunes a vast majority

M. Chen, B. Leibe, and B. Neumann (Eds.): ICVS 2013, LNCS 7963, pp. 103–112, 2013.

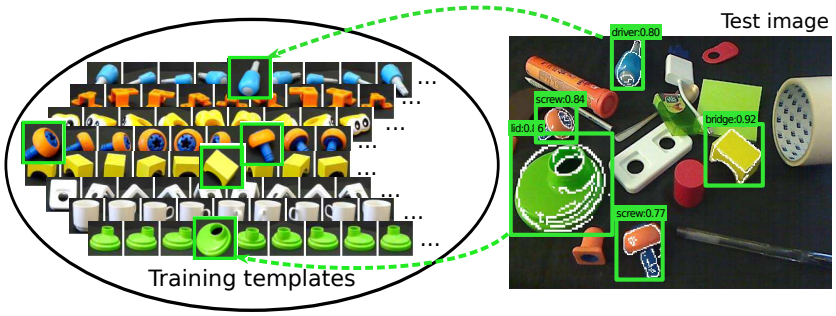


Fig. 1. A part of the training set and a detection result. Of the training images, only edges are used for detection. Color is shown only for illustration.

of windows as plausible hypotheses for location of any object. The second stage of the cascade, verification, is more time-consuming but limited only to a small fraction of windows. It is implemented by oriented chamfer score [18], improved by selection of discriminative edges based on their stability and orientation frequency, and by a compensation of the bias of the chamfer matching towards simpler objects. Fig. 1 shows an example training set and detection result.

Related Work. The approaches to textureless object detection and/or pose estimation divide into two broad categories, model-based and image-based. Model-based approaches have used CAD 3-D models [9,10] which is common in industrial applications, or depth information [8,11].

Image-based viewpoint classification has been addressed in [17,7]. In these works, the number of viewpoints is limited and the task is solved by viewpoint classifier. Unfortunately, this approach does not scale to a larger number of viewpoints or objects.

There are not many works on image-based textureless 3-D detection that would be scalable to many objects and poses. Due to only one training image for one specific viewpoint, this problem is usually tackled by nearest neighbor search on a large training set [2,4,8]. In [2], an early research on 3-D object textureless object detection was done, both model-based and image-based. A shape is represented here by a set of grouped straight lines and efficiently searched for the nearest neighbor using the k-D tree. The recent work [4] uses a similar idea, achieving real-time detection. A shape is represented by a rotation- and scale-invariant descriptor, which records the relations among each edgelet constellations. Unlike [2], which considers only a limited set of relations among lines such as parallelism and co-termination, the relations in [4] are much richer. The work [8] focused on achieving real-time performance in detection of multiple objects and thousands of templates from gradient orientations and 3-D surface normal orientations, using highly optimized implementation.

Speed is a key challenge in such works due to a large set of templates. They utilize fast techniques such as k-D tree [2], hash table [4], hierarchical search [20], look-up table and parallel techniques [8]. In contrast, we achieve high speed using a cascaded approach with fast index table search.

2 Fast Pruning of Object-Pose Hypotheses

The first stage of the cascade efficiently prunes object-pose hypotheses with low similarity to the scanning window. This is done by attaching to each window a sparse descriptor and then finding nearest neighbors in the feature space.

Sparse Edge-Based Image Descriptor. In the scanning window and each training template, we define m *reference points* $p_1, \ldots, p_m$ placed on a regular grid, excluding the margin near the image border (Fig. 2). A window I is assigned the feature vector $(d_I(p_1), \ldots, d_I(p_m), \phi_I(p_1), \ldots, \phi_I(p_m))$, where $d_I(p)$ denotes the distance of point p to the nearest edge in I and $\phi_I(p)$ is the orientation of this edge. This is computed efficiently using the distance transform.

The similarity of a scanning window I and a training template T is defined as the number of matched reference points, where a reference point is matched if both its features are similar up to some tolerances:

$$c(T, I) = \left| \left\{ i \in \{1, \ldots, m\} \;\middle|\; |d_T(p_i) - d_I(p_i)| \leq \theta_d,\ |\phi_T(p_i) - \phi_I(p_i)|_\pi \leq \theta_\phi \right\} \right|. \tag{1}$$

The distance of two angles is measured modulo π, which is denoted $|\cdot|_\pi$.

Fast Voting with Quantized Features. To obtain detection hypotheses, we need to find training templates that are similar, in the sense of (1), to the scanning window. Doing this exhaustively is infeasible. As the function $-c(T, I)$ is not a metric, algorithms like k-D tree cannot be used. Instead, we solve this task approximately using an index table with quantized features. The distance features $d_T(p_i)$ and the orientation features $\phi_T(p_i)$ are quantized into n_d and n_ϕ bins, respectively. For the scanning window I and a template T, a reference point p_i is matched if the distances $d_T(p_i)$, $d_I(p_i)$ and the orientations $\phi_T(p_i)$, $\phi_I(p_i)$ have the same quantized values.

In the training phase, an index table of size $n_d \times n_\phi \times m$ is built. A cell (q_d, q_ϕ, i) of this table contains the list of indices j of all training templates T_j in which the quantized value $d_{T_j}(p_i)$ is q_d and the quantized value of $\phi_{T_j}(p_i)$ is q_ϕ. Thus, the index of each training template occurs in the table m times. To find nearest training templates to a scanning window, each template collects the votes from the cells corresponding to the quantized features of I. The templates with at least θ_v votes are accepted as hypotheses. In order to decrease the risk of discarding true positives, we find the nearest neighbor for each object instead of the single nearest neighbor for the whole training set.

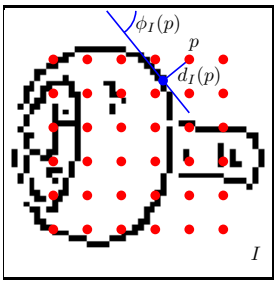


Fig. 2. The novel sparse image descriptor designed for textureless objects, used at the first cascade level. The m reference points are shown in red. Each reference point p in image I is assigned two features, the distance $d_I(p)$ to the nearest edge and the orientation $\phi_I(p)$ of this edge.

The average number of index visits per query is $mn/(n_d n_\phi)$. This number can be large. It can be decreased by grouping several reference points, at the expense of a larger table. In particular, we group triplets of points. A triplet is matched if all its three points match. The feature vector is obtained by randomly picking a triplet of reference points L times, yielding the table size $(n_d n_\phi)^3 L$ and the average number of index visits $Ln/(n_d n_\phi)^3$. In experiments, we refer to this modification as *Table-3pt*, while the single-point search is *Table-1pt*.

3 Verification by Improved Oriented Chamfer Matching

Detection hypotheses generated at the first stage of the cascade are verified at the second stage by a more precise but more expensive method, based on *oriented chamfer matching (OCM)* [18].

3.1 Compensating the Bias Towards Simples Shapes

In [18], the oriented chamfer distance between images I and T is defined as the weighted average $\sum_{e\in T}[\alpha d_I(e)+(1-\alpha)|\phi_I(e)-\phi_T(e)|_\pi]/|T|$ of the distance and orientation components. Here, $\phi_T(e)$ is the orientation of edge e in T and $|T|$ is the number of edges in T. We observed that when $|T|$ has large variance over the training set, this distance is biased towards simpler objects. To compensate this bias, we use the oriented chamfer score in a different form as

$$s_\lambda(T,I) = \frac{\left|\left\{\, e\in T \,\middle|\, d_I(e)\le\theta_d,\ |\phi_T(e)-\phi_I(e)|_\pi \le \theta_\phi \,\right\}\right|}{\lambda|T| + (1-\lambda)\overline{|T|}}, \tag{2}$$

where $\overline{|T|} = \frac{1}{n}\sum_{i=1}^{n}|T_i|$ is the average number of edges over all training templates and $\lambda\in[0,1]$ is a parameter. The numerator of (2) is the number of edges from T that have a match in I (this yielded slightly better results than the weighted average of distances). For $\lambda = 1$, the score corresponds to the distance used in [18]. Setting $0<\lambda<1$ decreases the score for templates with fewer edges than average (Fig. 3 shows an example). As shown in the experiments, this has a significant positive impact on detection performance.

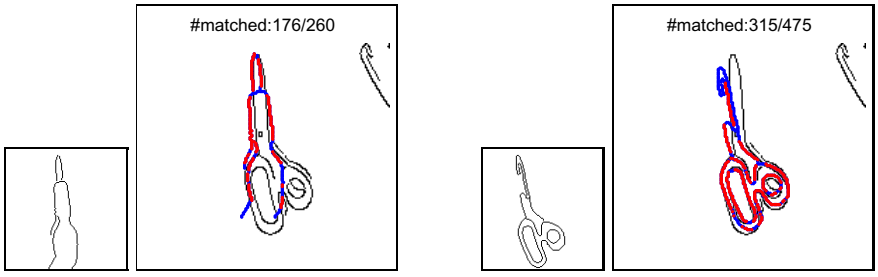


Fig. 3. Compensation of the bias towards simpler shapes. The matched template points are shown in red, the unmatched points in blue. (a) the score s_λ for the driver template: $s_1 = 0.68$, $s_{\frac{1}{2}} = 0.47$. (b) the score for the scissors template: $s_1 = 0.64$, $s_{\frac{1}{2}} = 0.64$. Before resp. after the compensation, the test image is classified as driver resp. scissors.

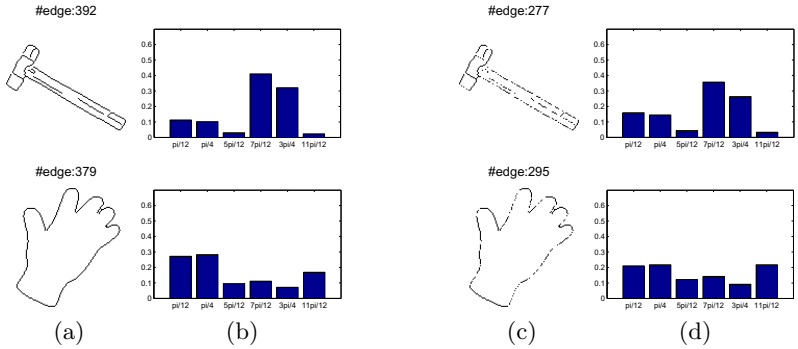


Fig. 4. Selection by edge orientation: $\alpha = 40\%$ of edges corresponding to the two highest bins are randomly removed. (a) original edges, (b) their orientation histogram, (c) selected edges, (d) their orientation histogram.

3.2 Selecting Discriminative Edges

For efficiency, [18] computed OCM on only a subset of randomly selected edge points in the template without drop of performance. In contrast, we want to use only edges that are discriminative for object detection. It has been an important topic in shape matching and detection to learn discriminative edges and discard unstable ones [14,16]. We use two criteria to select edges: stability to viewpoint and frequency of edge orientations.

Selection by Stability to Viewpoint. We define an edge in a training template to be *stable* if it matches, via oriented chamfer matching, the corresponding edge in any image obtained by a slight change of viewpoint (possibly, also of illumination, edge detector parameters, etc.). Stable edges are approximately selected as follows. As our training set does not contain explicit information which training templates are 'similar', we substitute this information. For each template T, we define $\mathcal{N}(T)$ to be the set of k templates that are most similar to T in the sense of the oriented chamfer score (2). We assume that $\mathcal{N}(T)$ will mostly contain templates differing by a small change of viewpoint. For every edge point $e \in T$ we define the score

$$n_T(e) = \left|\left\{\, T' \in \mathcal{N}(T) \;\middle|\; |d_T(e) - d_{T'}(e)| \le \theta_d,\ |\phi_T(e) - \phi_{T'}(e)|_\pi \le \theta_\phi \,\right\}\right|. \quad (3)$$

We keep only the edges from T that have the score greater than a threshold, $n_T(e) \ge \tau k$ where $0 < \tau < 1$.

Selection by Edge Orientations. The similarity score (2) tacitly assumes that the positions and orientations of all edges in the template are independent. Sometimes this is far from true. In particular, if the template contains long straight lines, the edges forming these lines are highly dependent and therefore carry less information than edges originating from small and irregular parts of the object. Take a hammer as an example, as shown in Fig. 4. The handle makes up for about 70% of all the edge points. However, these long lines do not discriminate a hammer from a screw driver or from just a few parallel lines.

We propose the following simple heuristic to account for this effect. First, the histogram of edge orientations in the template is computed. Then, a part, α, of edges in the two highest bins is removed. This is justified by the fact that the edge orientation histogram with long straight lines is likely to have a dominant peak. This method can be understood as a partial equalization of the orientation histogram. Note in Fig. 4 that 70% edges of the hammer fall into the highest two bins. After the selection, most edges of the hammer head are kept, while the handle edges have become notably sparser.

Non-maxima Suppression. After the verification, we obtain a set of detection candidates for scanning windows at various locations and scales. This set of hypotheses is finally filtered by a version of non-maxima suppression, which repeats the following step till no hypotheses are left: find the hypothesis with the highest score (2) and remove all windows that have a large overlap with it.

4 Experiments

4.1 The CMP-8objs Dataset

Due to the lack of suitable public datasets, we created a new *CMP-8objs* dataset of 8 objects with no or little texture. Each object was placed on a turntable with a black background and 180 views were captured by an uncalibrated hand-held video camera, covering approximately a hemisphere of views. The training templates were obtained from these images by cropping and scaling to the common size 48×48 pixels. To achieve partial invariance to image-plane rotation, the templates were synthetically rotated in range $[-40, 40]$ degrees in 9 steps. This resulted in 12,960 training templates. Some of them are shown in Fig. 5(a).

For testing, we captured 60 images of the size 640×480. Some examples are shown in Fig. 8 (top). Each image contains multiple objects in arbitrary poses with partial occlusion. Some of the objects are not in the training set and serve as distractors. The first 30 images have black background while the last 30 images were captured on a desktop with a light wood texture. We manually labeled the ground truth (354 objects in total) with bounding boxes.

We used the following parameters: $\theta_\phi = \pi/9$, $\theta_d = 3.1$, $N_d = 4$, $N_\phi = 6$, $m = 36$, $\theta_v = 12$ in Table-1pt and $\theta_v = 3$, $L = 50$ in Table-3pt. We ran the

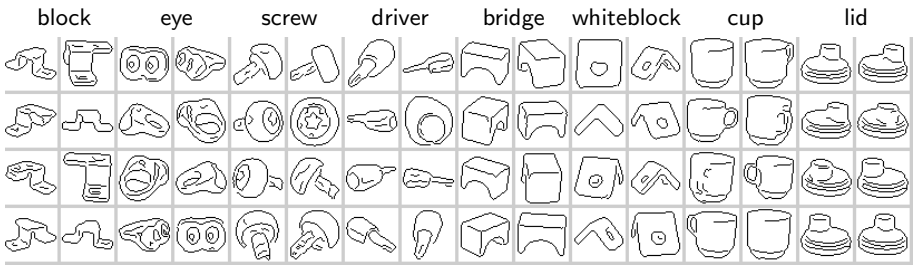
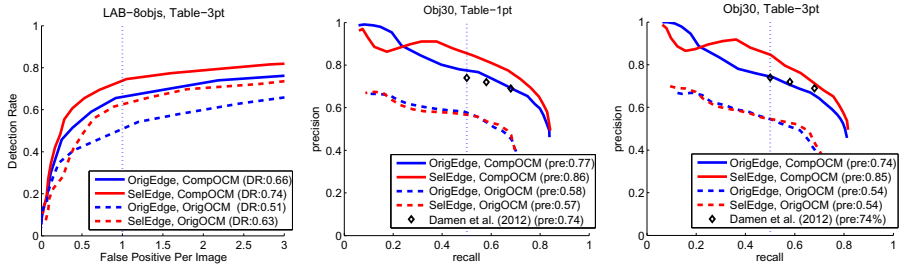


Fig. 5. Examples of training templates for the *CMP-8objs* dataset



(a) (b) (c)

Fig. 6. (a) DR/FPPI curves of Table-3pt on the CMP-8objs dataset. (b)(c) Precision/recall curves on the Obj30 dataset with Table-1pt and Table-3pt. In both methods, applying edge selection (§3.2) and OCM compensation (§3.1) lead to significant increase of performance. The results of Damen et al. were copied from [4, Figure 7].

detector at every position of the scanning window with 3 pixel steps, and at 10 scales with scale factor 1.2. On average, the first cascade stage decreased $\sim 10^9$ training-test pairs per image to $\sim 10^4$.

The performance is quantified by DR/FPPI (detection rate *vs.* the number of false-positives per image) curves, obtained by varying the similarity threshold. This score has been commonly used in shape-based detection [6,12]. The detection is considered correct if the detected object label is the same as the ground-truth label and the detected rectangle and the ground-truth rectangle overlap by more than 50% of the area.

We ran the detectors with four settings obtained by applying or not applying OCM compensation (§3.1) and edge selection (§3.2). We only show the performance for Table-3pt, as Table-1pt yields very similar results. As shown in Fig. 6(a), both techniques improve the performance significantly. With both techniques applied, the detection rate is 74% at FPPI=1, which outperforms the standard OCM without edge selection by 24%.

On average, Table-3pt needed 0.63 s per image, compared to 1.77 s for Table-1pt (on Intel Core i7-3770 at 3.40 GHz). Fig. 8 (top) shows example detections.

4.2 The Obj30 Dataset

We further evaluated our method on the *Obj30* dataset from [4], which contains 1433 training images of 30 textureless 3-D objects. As our detector is not natively invariant to rotation, we expanded the training set to 7056 templates by synthesizing rotated images. The test set has 1300 frames. Unlike CMP-8objs, each test image contains at most two objects on a clear background, as shown in Fig. 8 (bottom). The main challenge of this dataset is in more complex objects and in larger variance of shape complexity.

Because objects on average occupy relatively larger image area in Obj30 than in CMP-8objs, we used larger training templates (120×120) and fewer scales (8). Since the first cascade stage is independent on the template size, this had little effect on the detection time. All the other parameters were the same.

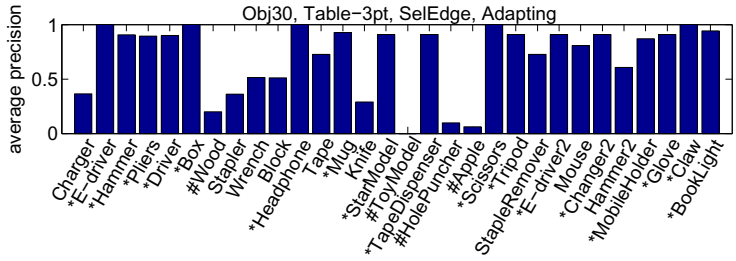


Fig. 7. The average precision (AP) for each object in the Obj30 dataset with the Table-3pt indexing method. Note that 17 objects (*) achieve AP above 85%, and 4 objects (#) are difficult to detect with AP below 20%.

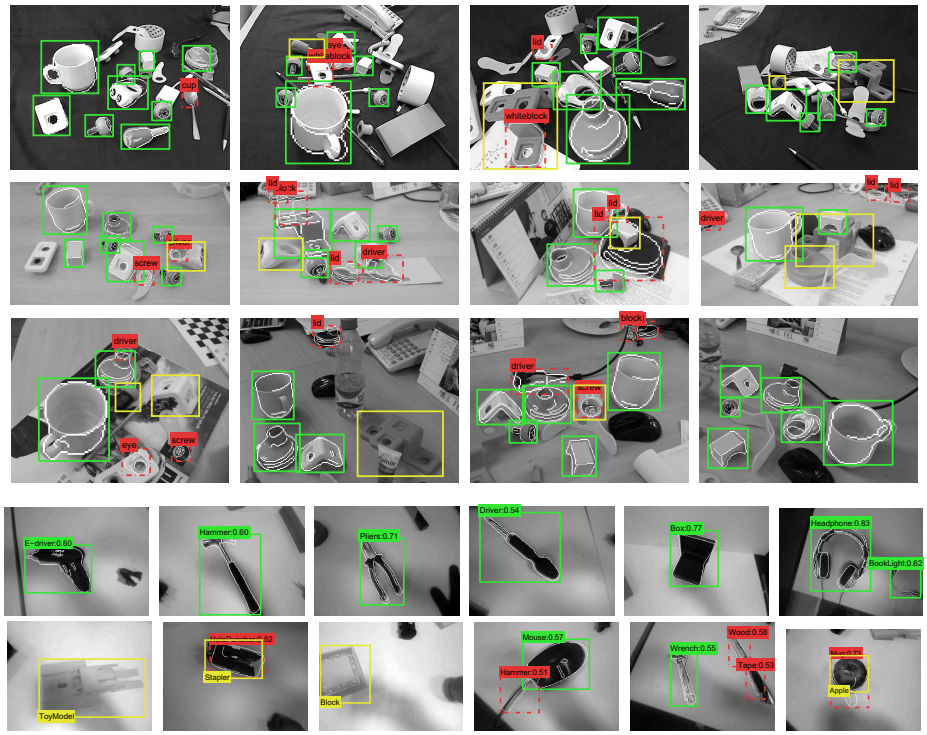


Fig. 8. Example detections for the CMP-8objs (top) and Obj30 (bottom) dataset. True positives, false positives and false negatives are shown in green, red and yellow, respectively. The edges of the found training templates are superimposed.

For evaluation, we used precision/recall curves obtained by varying the matching score threshold. The results are in Fig. 6(b)(c). The figures also show the performance of the algorithm [4].

We again evaluated the effect of OCM compensation (§3.1) and edge selection (§3.2). The edge selection was done only by edge orientation, since this

dataset does not contain enough training images for each objects to select by stability. The OCM compensation improves the performance significantly; not using it results in many false positives with small number of edges. Edge selection without OCM compensation has only a small positive effect. However, the effect of simultaneous OCM compensation and edge selection is very significant.

For the recall fixed to 50%, the precision is 86%/85% with Table-1pt/Table-3pt. This is significantly more than 74% achieved in [4]. This improvement is mainly due to OCM compensation and edge selection. The average running time per frame is 0.37 s for Table-1pt and 0.26 s for Table-3pt, which is less than for CMP-8obj due to fewer training-test pairs, clearer scenes, and fewer scales. This is to be compared to 0.14 s reported in [4].

In Fig. 7, we further report the average precision (APs) for each object, which is commonly used to evaluate visual detection and retrieval. For 17 objects, AP is greater than 85%. Similarly as in [4], objects with more distinctive shapes are more easily detected and less confused, such as E-driver, box, headphone, scissors, and claw. In contrast, false positives tend to be caused by elongated objects, such as knife, wood, hammer and tape (on the side view), though this effect is largely reduced by OCM compensation and edge selection. Example detections are shown in Fig. 8 (bottom).

5 Conclusion

We have proposed a new method for near real-time detection of textureless objects. Each object is represented by a set of training templates with different object poses. Of the training images, only edge information is used. Since the method finds the object-pose template nearest to the scanning window, it provides for free also a rough estimation of object pose.

The detector, applied to all scanning windows at various locations and scales, is a two-level cascade. The first level efficiently prunes the vast majority of background windows. It is based on a novel sparse image descriptor inspired by oriented chamfer matching. The second level verifies the surviving scanning windows by improved oriented chamfer matching. The improvements consist in compensating a bias towards simpler objects and in selecting discriminative edges.

The method outperforms the state-of-the-art approach [4] by 11% on the Obj30 dataset, publicly available with [4]. Good results have been achieved also on the CMP-8obj dataset, which we created newly for this paper. The CMP-8obj dataset with the ground truth is publicly available [1]. The processing is near real-time, on average 4 fps on the Obj30 dataset (with 7,000 training templates) and 1.5 fps on the CMP-8objs dataset (with 13,000 training templates).

We have deliberately used no other information than edges. However, the found detections could be easily filtered based on other cues, such as color, to further improve the performance. This verification could afford to be time-consuming thanks to only a small number of hypotheses.

Acknowledgement. The authors have been supported by EC project FP7-ICT-270138, the Technology Agency of the Czech Republic project TE01020415, and EPSRC project EP/K015966/1.

References

1. http://cmp.felk.cvut.cz/data/textureless
2. Beis, J.S., Lowe, D.G.: Indexing without invariants in 3D object recognition. PAMI 21(10), 1000–1015 (1999)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 2, pp. 886–893 (2005)
4. Damen, D., Bunnun, P., Calway, A., Mayol-Cuevas, W.: Real-time learning and detection of 3D texture-less objects: a scalable approach. In: BMVC (2012)
5. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Multiscale categorical object recognition using contour fragments. PAMI 30(9), 1627–1645 (2008)
6. Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. PAMI 30(1), 36–51 (2008)
7. Gu, C., Ren, X.: Discriminative mixture-of-templates for viewpoint classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 408–421. Springer, Heidelberg (2010)
8. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of texture-less objects. PAMI 34(5), 876–888 (2012)
9. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013)
10. Liebelt, J., Schmid, C., Schertler, K.: Viewpoint-independent object class detection using 3D feature maps. In: CVPR (2008)
11. Liu, M., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T.K., Chellappa, R.: Fast object localization and pose estimation in heavy clutter for robotic bin-picking. International Journal on Robotic Research 31(8) (2012)
12. Liu, M.Y., Tuzel, O., Veeraraghavan, A., Chellappa, R.: Fast directional chamfer matching. In: CVPR (2010)
13. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Intl. Journal of Computer Vision 60(2), 91–110 (2004)
14. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR, pp. 1038–1045 (2009)
15. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. Intl. Journal of Computer Vision 65(1), 43–72 (2005)
16. Opelt, A., Pinz, A., Zisserman, A.: A boundary-fragment-model for object detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006)
17. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: ICCV (2007)
18. Shotton, J., Blake, A., Cipolla, R.: Contour-based learning for object detection. In: ICCV (2005)
19. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR (2001)
20. Wiedemann, C., Ulrich, M., Steger, C.: Recognition and tracking of 3D objects. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 132–141. Springer, Heidelberg (2008)