2015 IEEE International Symposium on Mixed and Augmented Reality Workshops

Efficient Texture-less Object Detection for Augmented Reality Guidance

Tomáš Hodaň[1*] Dima Damen[2†] Walterio Mayol-Cuevas[2‡] Jiří Matas[1§]

[1]Center for Machine Perception, Czech Technical University in Prague, Czech Republic
[2]Department of Computer Science, University of Bristol, United Kingdom

Abstract

Real-time scalable detection of texture-less objects in 2D images is a highly relevant task for augmented reality applications such as assembly guidance. The paper presents a purely edge-based method based on the approach of Damen et al. (2012) [5]. The proposed method exploits the recent structured edge detector by Dollár and Zitnick (2013) [8], which uses supervised examples for improved object outline detection. It was experimentally shown to yield consistently better results than the standard Canny edge detector. The work has identified two other areas of improvement over the original method; proposing a Hough-based tracing, bringing a speed-up of more than 5 times, and a search for edgelets in stripes instead of wedges, achieving improved performance especially at lower rates of false positives per image. Experimental evaluation proves the proposed method to be faster and more robust. The method is also demonstrated to be suitable to support an augmented reality application for assembly guidance.

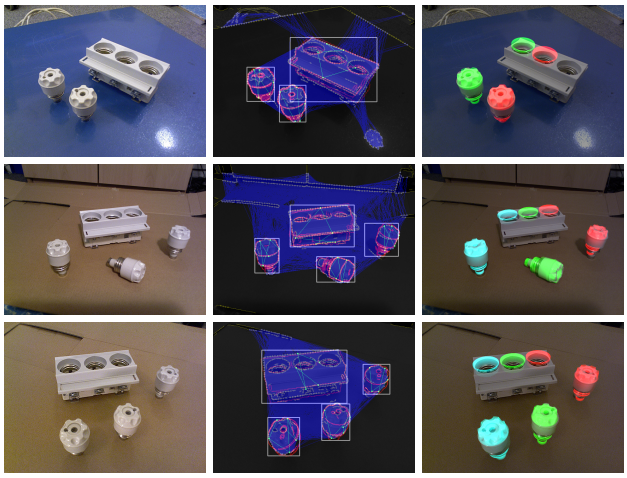
1 Introduction

Object-centric augmented reality (AR) is constrained by limitations of the available methods to describe shapes and detect objects. Current approaches rely mainly on either well textured objects or fiducial markers and thus struggle when having to deal with the many objects that have little texture or no suitable surfaces that allow to attach markers to them. This type of challenging objects does include many useful ones, from hand tools to furniture and machine components, for which the most sensible solution would be to describe them by their unaltered shape, *e.g.* to use a representation amenable to the objects' outline.

Furthermore, in many circumstances, the ability to train objects in-situ just before being able to detect them is not only appealing from the operational point of view, but potentially important so that any such system can work anywhere and instantly after training. This calls for methods that are fast enough to work without the luxury of offline processing.

Working with shape outlines is difficult. The feature representation stage is relatively fragile because it typically relies on edge detection. From the signal processing perspective, edge detection is challenging as determining the end of a shape is often a difficult decision to take under realistic illumination and background conditions. Since this is usually done by binary classification (as in *e.g.* the Canny edge detector), and at one scale, edge detection can become less repeatable than salient regions used to anchor visual descriptors when objects are well textured. This calls for a more careful selection of outline representation.

*hodantom@cmp.felk.cvut.cz
†dima.damen@bristol.ac.uk
‡walterio.mayol-cuevas@bristol.ac.uk
§matas@cmp.felk.cvut.cz



(a) (b) (c)

Figure 1: An application of the proposed object detection method for augmented reality guidance. Input images (a). Object detections (b), color coding as in Figure 7. Each training view of an object defines a 3D pose. For detected objects, the 3D pose is used to render assembly hints (c) by augmenting them via re-coloring.

In this paper, we consider the use of a data driven outline determination method, the structured edge detector [8]. Besides, we enhance the prior work of Damen et al. [5] by a more efficient and more accurate tracing of constellations. The result is a faster and more robust method for detection of texture-less objects in 2D images. We also show how the method can be used to support AR guidance for an assembly task (Figure 1).

The paper is organized as follows. We first review relevant works in Section 2 before presenting the prior work [5] and the proposed improvements in Section 3. Experimental evaluation is presented in Section 4, where the improved performance as a result of the proposed modifications is experimentally demonstrated. We finally explain how the proposed method can be used for AR guidance in Section 5, before concluding the paper in Section 6.

2 Related Work

The superiority of the shape description for detection of texture-less objects over the traditional texture-based description has been explored by Tombari et al. [14].

Many methods represent the shape by relative relationships between edge features, either within local neighbourhoods or globally over the whole image, to create features for classification. However, the most of the methods are not aimed at real-time operation which is a crucial requirement for AR applications.

For example, Carmichael and Hebert [3] employ weak clas-

978-1-4673-8471-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ISMARW.2015.23

sifiers using neighbourhood features at various radii for detecting wiry objects like chairs and ladders. This results in a time consuming process. Chia et al. [4] enable lines and ellipses to vote for the object's centre using their position and scale, similar to Hough transform voting. Similarly, Opelt et al. [13] use the standard boosting to classify contour fragments which then vote for the object's centre. Danielsson et al. [7] learn consistent constellations of edgelet features over categories of objects from training views. The most consistent pair of edgelets in the learnt model is selected as the aligning pair and exhaustively matched against all pairs in the test image. Extension of the pairs of edgelets to multiple edgelets forming a fully connected clique was proposed by Leordeanu et al. [12].

Most of the above approaches target object category recognition, while others aim at instance detection of rigid objects. An early approach by Beis and Lowe [1] detects the object's straight edges and groups them if co-terminating or parallel. For co-terminating lines, for example, the descriptor is made up of the angles between edges and their relative lengths. This reduces the search complexity at the expense of limiting the type of objects that can be handled.

More recent works, like Ferrari et al. [9], use a representation based on a network of contour segments. Recognition is achieved by finding the path in the network which best resembles the model derived from hand drawn contours. Starting from one base edgelet, that matches a corresponding model edgelet, the contour is iteratively extended based on the relative orientations and distances between test edgelets and the model's edgelets. Extending the contour and backtracking are iterated until the contour matching is completed or the path comes to a dead end. When breaks in the edge map cannot be bridged, partial contours are detected and combined in a hypothesis estimation post process. Although these methods demonstrate impressive detection performance, they do not target fast teach-and-use operation and are geared towards single object detection, with complexity scaling linearly when multiple objects need to be detected.

Scalability to multiple objects was considered in earlier works by the use of indexing and geometric hashing, similar in form to the library look-up that we use in our method. Examples include the early works by Lamdan and Wolfson [15] and Grimson [10]. More recently, Cai et al. [2] proposed a template matching approach which achieves a sub-linear complexity in the number of trained objects by hashing edge measurements, generating a small set of template candidates for each sliding window location.

Techniques aimed for fast detection get closer to our aim of in-situ teach-and-use. Hinterstoisser et al. [11] represents patches by histograms of dominant orientations followed by efficient bitwise matching which enables detection of one object within 80 ms, using 1600 reference views per object. However, the representation is not rotation- or scale-invariant (hence the need for a large number of reference views) and the complexity increases with multiple objects, with detection time increasing to 333 ms for 3 objects.

Many of the shape-based methods above do rely on the edge maps which are commonly computed via standard edge detectors such as Canny. This is mainly due to their relatively high speed of computation but also due to the lack of alternatives. Some methods like [11] consider multi-channel edge detection to improve the reliability of detected edges. But it could be argued that the edge maps needed for object detection are those that favour the object's outline and prominent features while eschewing clutter and noise. A fast supervised method for object outline detection has been proposed by Dollár and Zitnick [8]. The result is a cleaner edge map which also has a probabilistic representation of the edge response. Despite the desirable property of better outline detection, the method has been tested only on individual images. An evaluation on a sequence of images or at least multiple viewpoints of the same object captured by a moving camera is required to show its stability and thus suitability for our AR scenario.

3 Proposed Method

3.1 Bristol Multi-Object Detector

In [5], a scalable method for learning and detection of texture-less objects is proposed. The method is shape-based, view-variant, and importantly, can work in a teach-and-use manner and in real-time. It has also been shown to be scalable to multiple views of tens of objects.

Given a binary edge map, the method samples edgelets $E = \{e_1, e_2, \ldots, e_n\}$ of a fixed length. Each edgelet e_i is represented by its midpoint and orientation. The method introduces the notion of *fixed paths* to tractably select and describe constellations of edgelets. A fixed path is a predefined sequence of angles $\Theta = (\theta_0, \cdots \theta_{m-2})$, where the first angle θ_0 is defined relative to the first edgelet orientation. For every fixed path, the method only selects edgelet constellations with relative positions that satisfy the angles of the fixed path.

Each constellation $C = (i_1, i_2, \ldots, i_m)$, where i_j is the index of the j-th edgelet of the constellation, is described by

$$f(C) = (\phi_1, ..., \phi_{m-1}, \delta_1, ..., \delta_{m-2}),$$

which specifies the relative orientations and distances between the consecutive edgelets in the constellation. $\phi_k = \widehat{e_k, e_{k+1}}$ is the relative orientation of consecutive edgelets, and $\delta_k = g(e_{k+1}, e_{k+2})/g(e_k, e_{k+1})$ is the relative distance between edgelets, where $g(e_i, e_j)$ is the distance between midpoints of edgelets e_i and e_j. The descriptor is similarity-invariant, and the matching method is tolerant to a moderate level of occlusion. When descriptors are matched, the detection candidates are verified by using the oriented distance transform to confirm the object's presence and avoid hallucinations. We refer to this method as the Multi-Object Detector (MOD), and build on its latest version 1.2 [6].

We identify three areas of improvement in MOD. First, the method relies on a binary edge map whose quality is crucial. The quality is affected by undesirable edges that result from shadows or fine textures within the object or in its vicinity. Moreover, missing edges that represent the object's shape would reduce the number of constellations for a given fixed path. Second, the method defines a tolerance in the tracing angles, allowing higher displacement for further edges and thus higher geometric deviation. Third, when tracing a constellation, the method searches for the next edgelet through all the edgelets in the image exhaustively. This calls for a more efficient approach. The proposed improvements are described in the following paragraphs and illustrated in Figure 2.

3.2 Object Outline Detection

To address the first problem, we use the state of the art structured edge detector (SED) by Dollár and Zitnick [8]. It is a supervised edge detector trained on manually labeled ground-truth boundaries for naturalistic scenes. This training emphasizes object outlines, avoids shadows and generally achieves better consistency under different background and lighting conditions.

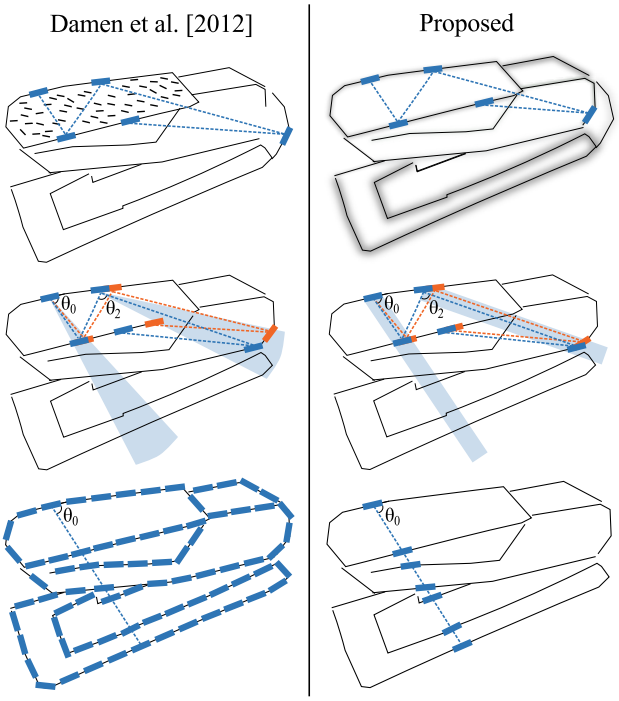


Figure 2: Proposed modifications to [5]. *Top:* The supervised edge detector achieves a higher edge repeatability and has a lower sensitivity to fine textures. *Middle:* Tracing in wedges is replaced by tracing in stripes, yielding less geometrically deviated constellations. *Bottom:* A Hough-based representation is used for fast retrieval of edgelets along a certain direction, avoiding the exhaustive search.

Structured random forests are trained from hand-labeled examples, where a 16×16 image patch is used to classify whether the center pixel is an edge pixel. The ensemble is used to provide a probabilistic estimate for the classification outcome. Results of the supervised edge detector prove its ability to remove noise in the edge map that results from textured clutter or within-object fine texture. The emphasis of the detector on object outlines is highly relevant to textureless objects, where the outline formulates the majority of edges in the object's shape. Though previously trained and tested on images of natural scenes, we evaluate the ability of SED to extract the object's outline in interior and industrial scenes, with input from a moving camera.

3.3 Tracing Section

In the original method, for each angle θ_i, a tolerance of ε radians is allowed when tracing constellations, *i.e.* the edgelets are searched for in a *wedge*. As the tolerance is introduced in the tracing angles, a larger displacement is allowed in edgelets that are further apart (Figure 2 middle). To make the allowed displacement independent of distance, we propose to search for edgelets along a *stripe* of a fixed width. We expect this modification to also remove the preference for further edges in forming constellations. In order to compensate for the sampling error and thus to minimize the miss rate in the search for edgelets, the width of the stripe is set such that it reflects the edgelet length.

3.4 Hough-based Constellation Tracing

In the original method, the relative orientations and distances of all edgelet pairs are calculated in the pre-processing step. The constellations are then traced in a brute-force manner. To search for the next edgelet in a given direction, all pairs starting with the last edgelet are checked, *i.e.* the complexity is $O(n)$, where n is the number of edgelets.

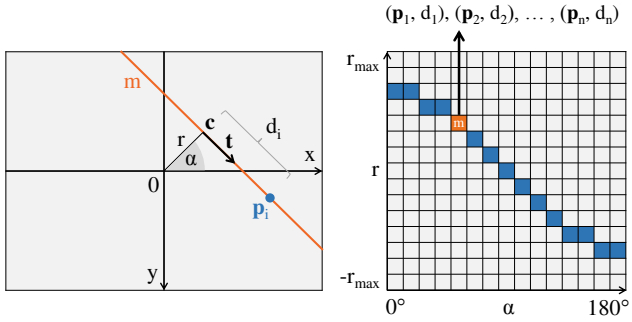


Figure 3: A Hough-based representation for efficient directional search of edgelets. The midpoints of the edgelets from the image space *(left)* correspond to a sinusoid in the Hough space *(right)*. Each bin in the Hough space represents a line m in the image space and stores a list of edgelets whose midpoints $\mathbf{p}_i$ lie on that line. The points in the list are sorted by d_i, a signed distance from $\mathbf{c}$ to $\mathbf{p}_i$ ($d_i = \mathbf{t} \cdot \mathbf{p}_i$, where $\mathbf{t}$ is a unit vector in the defined direction of the line m).

To efficiently search for edgelets in a given direction, we propose a Hough-based representation (Figure 3). The Hough space is parametrized by r, a signed distance from the origin to the point $\mathbf{c}$ which is the closest point on the corresponding line, and α, the angle between the horizontal axis and the line connecting the origin and the point $\mathbf{c}$. Each bin of the quantized Hough space represents a line $m_{\alpha,r}$ in the image space and stores a list $L_{\alpha,r}$ of edgelets whose midpoints lie on this line. The edgelets in the list are sorted by the signed distance d from $\mathbf{c}$ to their midpoints. To retrieve edgelets lying in the given direction from an edgelet with midpoint $\mathbf{p}_i$, one just needs to get the list $L_{\alpha,r}$ from the proper bin in the Hough space and locate the insertion position of d_i in the sorted list to determine edgelets lying on the required half-line. The complexity of the search for edgelets lying on a half-line is thus $O(\log |L_{\alpha,r}|)$, where typically $|L_{\alpha,r}| \ll n$ in natural images.

To retrieve edgelets lying in a stripe of a defined width, we collect edgelets from the half-lines included within the search stripe. The Hough-based representation is constructed such that in every column α_i, each edgelet is recorded only once. A list of unique edgelets within the search stripe can be thus obtained by visiting several neighbouring bins in the column α_i.

The memory requirement of the Hough-based representation is nB_α. The average-case complexity of its construction is $O(nB_\alpha + B_\alpha B_r mk)$, where B_α and B_r are the number of quantization bins of parameters α and r respectively. $O(nB_\alpha)$ is the complexity of recording n edgelets, each in B_α bins. $O(B_\alpha B_r mk)$ is the complexity of sorting the lists $L_{\alpha,r}$ in all bins of the quatized Hough space, where m is the average list length and k is the maximum displacement of an element from its sorted position. When the edgelets are first sorted by y and x coordinate of their midpoints (this order can be achieved at a little cost when taking it into consideration during detection of edgelets), and then mapped into the Hough space, the resulting lists $L_{\alpha,r}$ are automatically sorted by the distance d. Due to the quantization errors of α and r, the order can be sometimes violated. But since the el-

ements are expected to be close to their sorted positions (the maximum displacement k is expected to be small), the lists can be sorted efficiently by *e.g.* the insertion sort algorithm.

4 Experimental Evaluation

The proposed modifications were evaluated quantitatively on the publicly available Bristol Tools dataset [5], which includes 1364 training and 1219 test images (annotated with 2D ground truth bounding boxes) of 30 texture-less objects. All images were rescaled to the resolution of $320 \times 240\,px$, in line with the results presented in [5]. A detection was considered true positive if the overlap (*i.e.* intersection over union) of its bounding box with a ground truth bounding box of the same object was at least 50%. The detection time limit was set to $3\,s$.[1]

First, we evaluate the performance when using different edge detectors (Canny vs. SED) as well as using different tracing sections (wedge vs. stripe). To obtain a binary edge map, we applied a non-maxima suppression to the edge probability map produced by SED, and threshold it by $t = 0.05$ (*i.e.* pixels with the edge probability higher than t were considered as edge points). The thresholds of the Canny edge detector were set to 0.05 and 0.2, as in MOD v1.2 [6]. The length of edgelets was set to $8\,px$, the width of the tracing stripe to $9\,px$ (*i.e.* $4\,px$ on each side of the tracing ray), and the tolerance in the tracing angle defining the span of the wedge was set to $\varepsilon = 0.06\,rad$. The minimum and the maximum distance between two constellation edgelets was required to be 5 and $150\,px$ respectively. To construct the Hough-based representation, the quantization step was set to 0.5° for α and $1\,px$ for r, totaling 360 bins for α and 400 for r ($400\,px$ is the diagonal length of an image with resolution $320 \times 240\,px$). As in MOD v1.2, only one fixed path was used: $\Theta = (-0.807, -2.173, 2.868, 2.737)$, where the angles are in radians.

For detection, the whole codebook including the trained constellations needs to be loaded into RAM. In order to meet the memory limit of the used computer ($4\,GB$ of RAM), we did not trace all possible constellations during training, *i.e.* we did not bounce on all edgelets lying in the tracing section. Instead, we randomly sampled 5 edgelets to be bounced on. This is likely not to be the optimal solution and a better, perhaps a deterministic approach is needed.

As shown in Figure 4, edges detected by SED produced consistently better results than edges detected by Canny. We attribute the increase in performance to the fact that SED is specifically trained to detect object boundaries which are supposed to contain most of the shape information of texture-less objects. Tracing in the stripe section yielded a higher detection rate (DR), especially for a lower false positives per image (FPPI). The DR/FPPI curves were obtained by changing the threshold of the detection cost defined in [5].

In principle, the MOD v1.2 is represented in this evaluation by the method which uses the Canny edge detector and the wedge tracing. However, the Hough-based search of constellation edgelets was used in the evaluated method (edgelets from the half-lines spanning the given wedge were collected), whereas MOD v1.2 performs the exhaustive search. Another difference is that we did not greedily remove the corresponding edgelets in the test image once they were assigned to a verified hypothesis, *i.e.* we did not invalidate them for subsequent detections. Instead, we collected all verified hypotheses and performed a non-maxima suppression in the end.

[1]The evaluation was done on a virtual machine with a limited computational power. We believe that the $3\,s$ corresponds to approximately $1\,s$ when running on a standard computer.

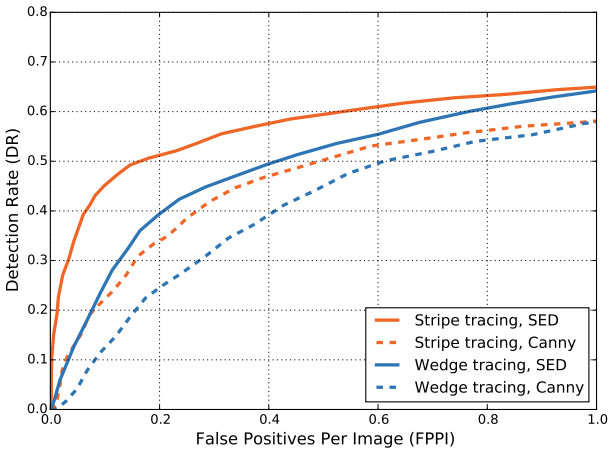


Figure 4: DR/FPPI for different edge detectors (Canny vs. SED) and different tracing sections (wedge vs. stripe). The curves were generated by changing the detection cost threshold.

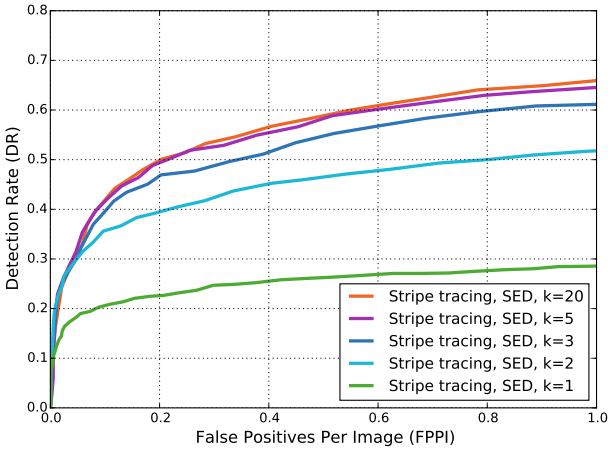


Figure 5: DR/FPPI evaluation of different values of k (the number of closest edgelets considered when tracing a constellation in the detection stage – one of these edgelets was randomly picked and bounced on).

Example detection results in test images from the Bristol Tools dataset can be found in Figure 7. The last row shows typical failure cases caused by the presence of several thin objects in the dataset which tend to match with any parallel lines in the test scene. An improvement of the verification function is necessary to disambiguate these cases.

Next, we investigate the effect of considering only k closest edgelets from the tracing stripe, when one of these edgelets is randomly picked and bounced on in the detection stage. For tracing the training constellations, we bounced on maximum of 50 closest edgelets in this experiment. As shown in Figure 5, there is no big gain when $k > 5$. This is potentially an important finding since considering only 5 closest edgelets is supposed to increase the robustness to clutter

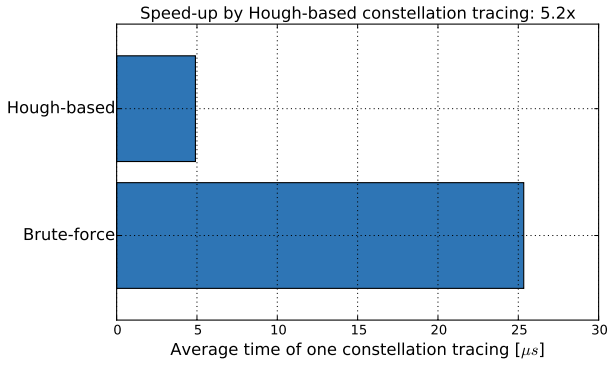


Figure 6: Average time of the proposed Hough-based constellation tracing vs. the brute-force approach used in the original method.

and noticeably reduce the number of possible constellations. More detailed investigation of this observation is a subject of our future work.

The Hough-based directional search of edgelets brought 5.2× speed-up when compared to the exhaustive search over all edgelets in the test image (Figure 6). This speed-up was measured on the Bristol Tools dataset which contains images with only mild clutter. We presume the difference will be even more significant in the case of complex scenes in which a large number of edgelets is present.

As shown in [5], the time complexity of the method is sublinear in the number of learnt objects. With an optimized and parallelized code (construction of the Hough-based representation, tracing of constellations, and also hypothesis verification can be all parallelized efficiently), we will be interested in evaluating the increase in the number of objects that can be handled in real-time. The impact of the scene complexity, especially of the level of background clutter, is another subject of our future study.

5 Augmented Reality - Assembly Guidance

With the ability to detect and locate objects using their shape alone, it is possible to develop various useful augmented reality applications. Detection of previously learnt objects can not only allow recovering information details about these objects, such as their identity and technical specifications, but also, with their localization in the image, it is possible to make spatial connections. This can be useful in *e.g.* assembly tasks.

When 3D models of the objects are available and their individual views are related to corresponding 3D viewpoints, it is possible to do further augmentations such as colour changes to highlight relevant objects or their parts.

Figure 1 presents an example application of texture-less object detection for augmented reality guidance. In this case, the objects have very little texture and are essentially described by their shape's outline. Our method is able to locate the known objects and colour them in a way that provides guidance for assembly — the various objects are coloured in a way that intuitively indicates what goes where.

6 Conclusion

A method for efficient texture-less object detection has been presented and its suitability for augmented reality guidance has been demonstrated. The method builds on the approach of Damen et al. [5] which it improves in several ways. First, it exploits the structured edge detector which is experimentally shown to achieve consistently better results when compared to the standard Canny edge detector. Second, the edgelet constellations are traced in stripes instead of wedges. The resulting constellations are less geometrically deviated, yielding a higher detection rate, especially at lower rates of false positives per image. Last but not least, the proposed method uses a Hough-based representation for efficient directional search of edgelets, achieving more than 5 times speed-up in constellation tracing.

Acknowledgements

This work was supported by CTU student grant SGS15/155/OHK3/2T/13 and by the Technology Agency of the Czech Republic research program TE01020415 (V3C – Visual Computing Competence Center) TE01020415.

References

[1] J. Beis and D. Lowe. Indexing without invariants in 3D object recognition. *IEEE Transactions on Pattern And Machine Intelligence (PAMI)*, 21(10), 1999.

[2] H. Cai, T. Werner, and J. Matas. Fast detection of multiple textureless 3-d objects. In *Proc. of the 9th Intl. Conf. on Computer Vision Systems (ICVS)*, 2013.

[3] O. Carmichael and M. Hebert. Object recognition by a cascade of edge probes. In *British Machine Vision Conference (BMVC)*, 2002.

[4] A. Chia, S. Rahardja, D. Rajan, and M. Leung. Object recognition by discriminative combinations of line segments and ellipses. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[5] D. Damen, P. Bunnun, A. Calway, and W. W. Mayol-Cuevas. Real-time learning and detection of 3d texture-less objects: A scalable approach. In *BMVC*, pages 1–12, 2012.

[6] D. Damen, P. Bunnun, and W. Mayol-Cuevas. MOD: Bristol's multi-object detector v1.2. `http://www.cs.bris.ac.uk/~damen/MultiObjDetector.htm`, Aug 2014.

[7] O. Danielsson, S. Carlsson, and J. Sullivan. Automatic learning and extraction of multi-local features. In *International Conference on Computer Vision (ICCV)*, 2009.

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013.

[9] V. Ferrari, T. Tuytelaars, and L. Gool. Object detection by contour segment networks. In *European Conference on Computer Vision (ECCV)*, 2006.

[10] W. Grimson and D. Huttenlocher. On the sensitivity of geometric hashing. In *International Conference on Computer Vision (ICCV)*, 1990.

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *European Confernece on Computer Vision (ECCV)*, 2006.

[14] F. Tombari, A. Franchi, and L. Di Stefano. Bold features to detect texture-less objects. In *IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[15] Y.Lamdan and H.J.Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *International Conference on Computer Vision (ICCV)*, 1988.

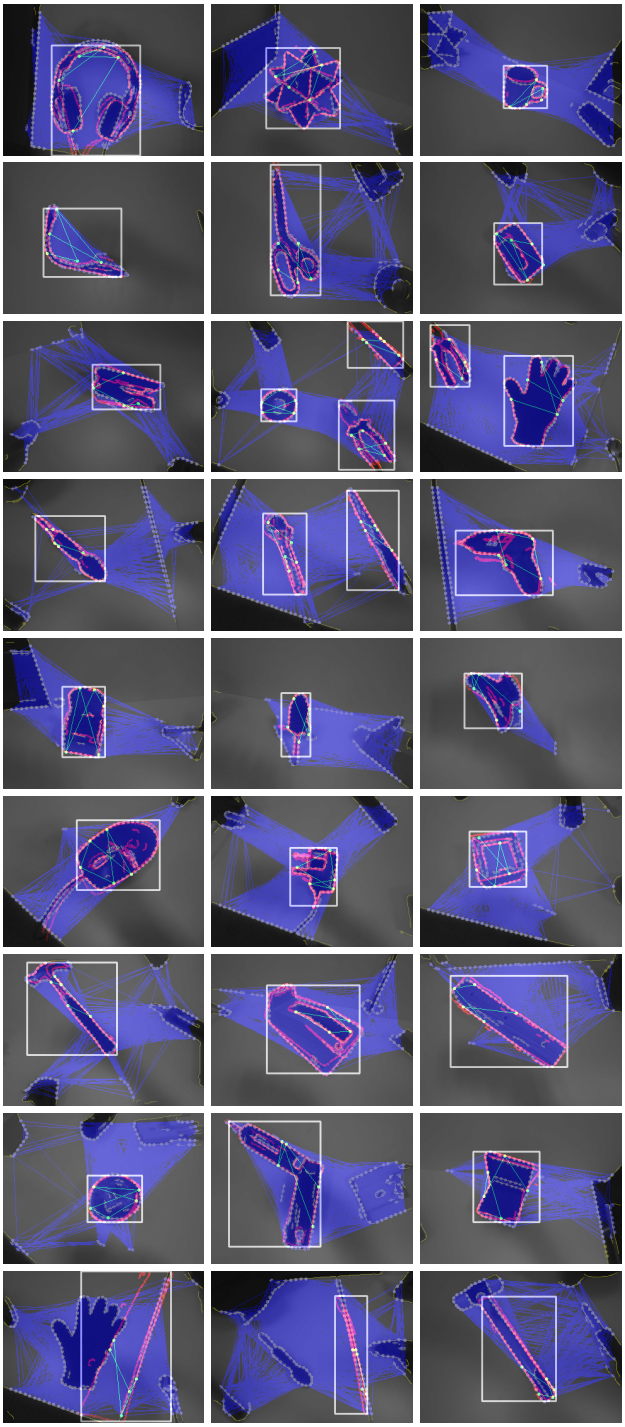


Figure 7: Example detection results in test images from the Bristol Tools dataset. The last row shows typical failure cases caused by the presence of several thin objects in the dataset which tend to match with any parallel lines. Centers of the detected edgelets are visualized by dots, connections of the traced edgelet constellations are drawn in blue, constellations which generated detections are highlighted in green, and edges of the detected object views are drawn in red.