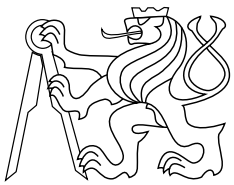
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD Thesis Proposal

ISSN 1213-2365

Texture-less Object Detection

Tomáš Hodaň

hodantom@cmp.felk.cvut.cz

CTU–CMP–2015–05

August 31, 2015

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/hodan/Hodan-TR-2015-05.pdf

Supervisor: prof. Jiří Matas

This work was supported by the EC FP7 programme under grant no. 270138 DARWIN, 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).

Research Reports of CMP, Czech Technical University in Prague, No. 5, 2015

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Abstract

Learning, detecting and accurately localizing texture-less objects is a common requirement for applications in both personal and industrial robotics. Despite their ubiquitous presence, texture-less objects present significant challenges to contemporary methods for visual object detection and localization.

In our work we aim at simultaneous detection of multiple texture-less objects with sub-linear complexity in the number of known objects, real time performance, robustness to occlusion and clutter, low false detection rate, and accurate object localization. So far, we have proposed two methods. One method works with both color and depth features. It adopts the sliding window paradigm with an efficient cascade-style evaluation of each window location. The method can run in real-time, achieves the state of the art performance, and its practical relevance was demonstrated in a real robotic application. In the other proposed method, which works only with image edges, we focused on efficient generation of detection hypotheses based on constellations of short edge segments. Experimental evaluation proved the method to be faster and more robust than the method of Damen et al. (2012), on which our method is based. The method was also shown to be suitable to support an augmented reality application for assembly guidance. Besides, we have created a new RGB-D dataset which will be used for a challenge at ICCV 2015 workshop.

This document describes the topic in more detail, reviews related work, presents our contributions and introduces open issues and goals of the thesis.

Contents

Chapter 1

Introduction

Texture-less, smooth and uniformly colored objects occur frequently in domestic and work environments (Figure 1.1). Learning, detecting and accurately localizing such objects is a common requirement for applications in both personal and industrial robotics — knowing object location and orientation allows the robot end effector to act upon the object (Figure 1.2).

(a) (b) (c) (d)

Figure 1.1: Texture-less objects around us. They are common not only in domestic and work environments — kitchen equipment (a), hand tools (b), toys (c), but also in industry (d). The images are samples from publicly available datasets [30, 42, 7, 27].

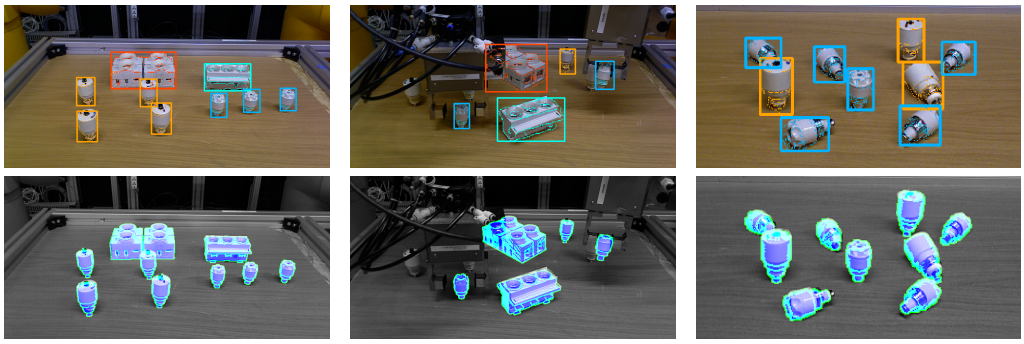


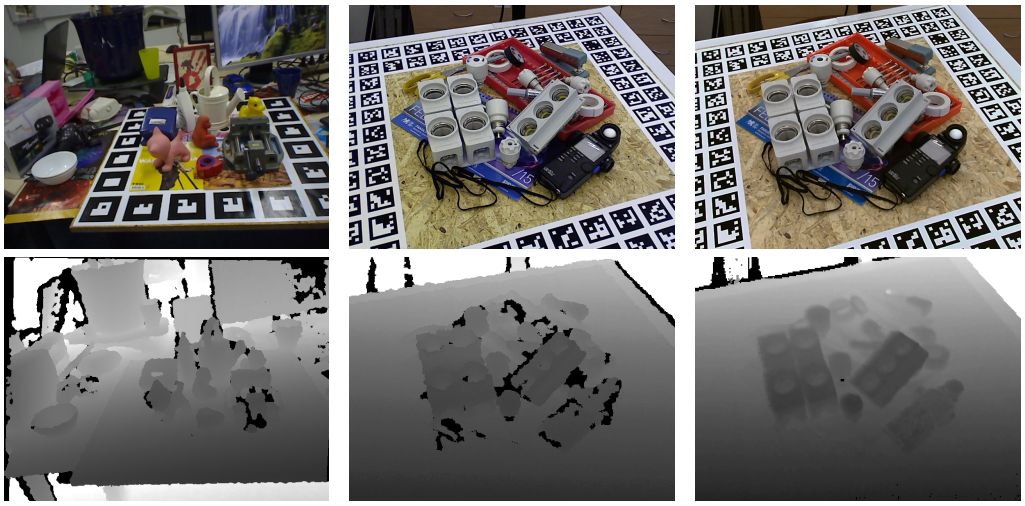
Figure 1.2: An example robotic assembly scenario relying on texture-less object detection — an arm with a gripper is assigned the task of picking up electrical fuses at arbitrary locations and inserting them into the corresponding fuse boxes [2, 28]. To enable object manipulation, the objects need to be detected (top) and their 6D poses, *i.e.* 3D location and 3D orientation, estimated (bottom).

Despite their ubiquitous presence, texture-less objects present significant challenges to contemporary visual object detection and localization methods. This is mainly because common local appearance descriptors are not discriminative enough to provide reliable correspondences. Appearance of a texture-less object is dominated by its global shape, its material properties and by the configuration of light sources. Unless these are known in advance and precisely controlled, the recognition method needs to be robust to changes in these factors.

In 2D images (either color or greyscale images), the most sensible solution to describe the texture-less objects is to use a representation amenable to the image edges, *i.e.* the points where image characteristics, typically image intensity, change sharply. In the case of texture-less objects, these edges correspond mostly to objects' outline.

The detection task can be simplified when depth images are used as additional input data. The RGB-D images (color + depth) can be obtained using Kinect-like sensors which produce aligned color and depth images that concurrently capture both the appearance and geometry of a scene (Figure 1.3). The extra information about 3D shape allows for a more discriminative description which is expected to reduce hallucinations, *i.e.* false positive detections. Moreover, the depth information can make the search process more efficient by considering only image regions where the measured depth or 3D shape is in accordance with the objects to be detected.

RGB-D sensors have been available for years, but consumer-level sensors



(a) (b) (c)

Figure 1.3: RGB-D images captured by the structured-light sensors Microsoft Kinect v1 (a) and Primesense Carmine 1.09 (b), and by the time-of-flight sensor Microsoft Kinect v2 (c). The images in (b) and (c) capture the same scene in the same lighting conditions. The color images (top) are aligned with the depth images (bottom) which for each pixel contain information about distance to the scene. The more distant the scene, the brighter the visualized pixel. Pixels with unknown depth information are black. The images in (a) are from the dataset of Hinterstoisser et al. [24] and images in (b,c) from the T-LESS dataset [27].

became widely available only recently, with the launch of Microsoft Kinect in November 2010. The technology is based on the structured-light principle. A light speckle pattern is projected onto the scene using a near-infrared laser emitter and the light reflected back to a standard off-the-shelf infrared camera is analyzed to estimate the depth of the scene surfaces [31, 32]. Despite the exact details not being publicly available as the technology is patented, it is evident that the sensor relies upon established computer vision techniques such as depth from focus and depth from stereo. In 2014, the second generation of Microsoft Kinect was released. This completely new sensor is based on the time-of-flight principle in which the depth of a scene is measured by the absolute time needed by a light wave to travel into the scene and, after reflection, back to the sensor.

One of the main shortcomings of the RGB-D sensors which limits their general utilization is the restricted sensing range going from tens of centimeters to several meters. The depth measurements are also affected by noise due

to multiple reasons such as reflections, transparent objects and light scattering. Small surface details are either completely unperceivable or significantly distorted by the sensor noise. Furthermore, since the color and depth images are captured from different viewpoints, the depth images aligned to the color images suffer from imprecise or missing depth information around occlusion boundaries. As can be seen in Figure 1.3(c), the noise is significantly reduced in the depth images produced by the Kinect sensor of the second generation. However, the sensing range is still restricted.

Because of the limitations of the RGB-D sensors and because of the large numbers of 2D images already available online, the task of texture-less object detection using only 2D images is undoubtedly important as well.

In our work we aim at simultaneous detection of multiple texture-less objects with sub-linear complexity in the number of known objects, real time performance, robustness to occlusion and clutter, low false detection rate, and accurate object localization. So far, we have proposed two methods. One method works with both color and depth features. It adopts the sliding window paradigm with an efficient cascade-style evaluation of each window location. The method can run in real-time, achieves the state of the art performance, and its practical relevance was demonstrated in a real robotic application presented in Figure 1.2. In the other proposed method, which works only with image edges, we focused on efficient generation of detection hypotheses based on constellations of short edge segments. Experimental evaluation proved the method to be faster and more robust than the method of Damen et al. [12], on which our method was based. The method was also shown to be suitable to support an augmented reality application for assembly guidance.

Besides, we have created a new RGB-D dataset for detection and pose estimation of texture-less objects [27]. The dataset will be used for a challenge at the *1st International Workshop on Recovering 6D Object Pose*, which is organized in conjuction with ICCV 2015 in Santiago, Chile [1].

The rest of the document is organized as follows. After reviewing related work in Chapter 2, we present our work in Chapter 3 and the future work in Chapter 4. Our publications can be found in appendices.

Chapter 2

Related Work

While object recognition is a long-standing and widely studied problem, most attention until recently has been paid to the recognition of textured objects, for which discriminative local appearance features, invariant to changes in viewpoint and illumination, can be readily extracted [35, 43, 11]. These objects are often assumed to have piece-wise planar surfaces. Their appearance variations can be therefore modeled by a simple geometric transformation (*e.g.* similarity), which can be reliably determined from the rich textural information. Candidate object locations in the scene are typically determined by identifying so-called interest points or interest regions, a strategy which drastically reduces the overall computational cost compared to exhaustive image search. However, when applied to texture-less objects, common interest point detectors, such as SIFT [36], typically fail to identify corresponding image regions and common local appearance descriptors are no longer discriminative enough to provide reliable correspondences [42].

The following sections review methods which are suitable for detection of texture-less objects, either in RGB or RGB-D images. The methods are divided into four categories: template matching methods, shape matching methods, methods based on dense features, and deep learning methods.

2.1 Template Matching Methods

The template matching methods sweep sliding windows of several discrete sizes over the entire image with a small pixel step, searching for a match against the stored object templates. They can be seen as operating top-down — each location and scale of the sliding window can be considered as an hypothesized rough object segmentation, which is then verified by matching local features of the image region against the stored templates.

Template matching is one of the earliest techniques applied to object detection in images. An object is represented by a set of templates that capture possible global object appearances exhaustively. Each template (either RGB or RGB-D) captures the object in one 3D pose. There can be also templates featuring different lighting conditions, background clutter and various occlusion levels. Traditional approaches use only a few stored templates per object. The similarity of a window and a template is expressed by correlation coefficients. A sufficiently high correlation score indicates a successful match. Correlation can employ color, intensity, image gradients, edges, depth or even 3D shape. Invariance is achieved only w.r.t. translation, with little tolerance to misalignments. The methods also scale poorly to large numbers of objects and special attention has to be paid to implementation details, otherwise they are too slow for real-time operation. The interested reader is referred to [6] for a survey.

Due to their low generalization and limited applicability, template matching techniques were for some time out of the mainstream research agenda. Instead, research in object recognition concentrated on approaches based on the viewpoint-invariant local features obtained from objects rich in texture. Such approaches require only a small number of training images per object and generalize well to a wide range of possible appearances. As computers became faster and equipped with more memory, template-based methods grew in popularity again. Today it is not unusual to maintain thousands of templates per object, thus capturing appearance variations exhaustively.

An efficient template matching technique was introduced by Hinterstoisser et al. [25, 22]. Instead of a raw image, object templates are represented by a set of carefully selected feature points in different modalities (specifically orientation of intensity gradients and orientation of 3D surface normals). Measurements at feature points are quantized and represented as bit vectors, allowing for fast matching with binary operations. Tolerance to misalignments is achieved by comparing the binarized representation with pixels in a small local neighbourhood. The pose retrieved from the best matching template is used as a starting point for subsequent refinement with the Iterative Closest Point (ICP) algorithm [4, 39]. With data structures optimized for fast memory access and a highly vectorized implementation using special SSE hardware instructions, the method is capable of real-time matching of several thousands of templates. However, the performance is expected to degrade noticeably for large object databases, since the time complexity is linear in the number of loaded templates (around 3000 templates are used per object).

The work was later extended by Rios-Cabrera et al. [38] who introduced machine learning methods. Negative (*i.e.* background) examples are added

to the template learning process and the templates are locally weighted to emphasize discriminative areas. The templates are then clustered and an AdaBoost-like cascade is built to distinguish among templates within a cluster. This results in faster detection times, smaller number of templates required, and better discriminativity for large number of objects.

Cai et al. [7] proposed a template matching approach which achieves a sub-linear complexity in the number of trained objects by hashing edge measurements (distances and orientations of the nearest edges from points on a fixed regular grid), generating a small set of template candidates for each sliding window location. The candidates are verified by the oriented chamfer matching.

Current top-performing object detectors on generic datasets, such as PASCAL [17] and ImageNet [14], employ detection proposals to guide the search for objects, thus avoiding the exhaustive sliding window search over the whole image. The proposal methods [29], which can be either hand-crafted or trained, build on the assumption that all objects of interest share common visual properties that distinguish them from the background. Given an image, the aim of the proposal methods is to output a set of proposal regions that are likely to contain objects. If high object recall can be reached with considerably fewer windows when compared to the sliding window detectors, significant speed-ups can be achieved, enabling the use of more sophisticated classifiers. The template matching methods for texture-less object detection could benefit from the detection proposals as well. In particular, the edge-based proposal method by Zitnick and Dollár [47] seems to be suitable for texture-less objects.

2.2 Shape Matching Methods

The aim of the shape matching methods is to represent the object shape by relative relationships between 2D or 3D shape features, either within local neighbourhoods or globally over the whole image. Each object is usually represented by a set of feature descriptors. Detection hypotheses are generated by finding correspondences between the training descriptors and the descriptors extracted from a test image. The correspondences can be validated through a Hough-style voting. This category of methods can be seen as operating bottom-up and is in principle similar to the traditional detection methods based on local appearance features [35].

For example, Carmichael and Hebert [8] employ weak classifiers using neighbourhood features at various radii for detecting wiry objects like chairs and ladders. This results in a time consuming process. Chia et al. [9] en-

able lines and ellipses to vote for the object's centre using their position and scale, similar to Hough transform voting. Similarly, Opelt et al. [37] use the standard boosting to classify contour fragments which then vote for the object's centre. Danielsson et al. [13] 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. [34].

Most of the above shape-based methods target object category recognition, while others aim at instance detection of rigid objects. An early approach by Beis and Lowe [3] 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. Ferrari et al. [18] 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 real-time 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. Examples include the works by Lamdan and Wolfson [46] and Grimson [20]. More recently, Damen et al. [12] proposed a scalable method based on a tractable extraction of constellations of edgelets (*i.e.* short line segments) with library lookup using descriptors which are invariant to translation, rotation and scale changes. The approach learns object views in real-time, and is generative — enabling more objects to be learnt without the need for re-training. During testing, a random sample of edgelet constellations is tested for the presence of known objects. A similar method was presented by Tombari et al. [42]. Instead of tracing constellations under predefined angles, as is done in [12], they group neighboring line segments aggregated over limited spatial supports, which is supposed to increase robustness to background clutter and object occlusion.

Many of the shape-based methods rely on edges 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 [23] 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 [15]. 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 texture-less object detection.

Recognition capability of the above shape matching methods using only 2D shape features is inherently lower compared to methods using depth information. Consequently, the 2D methods tend to produce more false positive detections for a given recall.

A 3D shape-based method was presented by Drost et al. [16]. During training, all possible pairs of points on a 3D object model are described and recorded in a hash table. During detection, sampled pairs of points from the test scene are described and used to vote for corresponding object pose hypotheses. The most voted pose clusters can be then refined with ICP. Choi and Christensen [10] further augmented the point pair feature with color information. The efficiency and performance of these methods depend directly on complexity of the 3D scene, which might limit their applicability to real-time applications.

2.3 Methods Based on Dense Features

Another category of bottom-up methods is based on dense features, where every pixel is involved in prediction about the detection output. A descriptor of local patch surrounding the pixel or simple measurements in the local pixel's neighborhood are used for this purpose. As was discussed before, local 2D features are not discriminative enough in the case of texture-less objects. Hence, these methods can be successfully used only if depth information is available, which allows for a richer description of the local neighborhood.

Sun et al. [40] and Gall et al. [19] used a generalized Hough voting scheme, where all pixels cast a vote in a quantized prediction space parametrized by 2D object center and scale. Their methods were shown able to predict coarse

object poses. Brachmann et al. [5] demonstrated that a similar approach can be applied to texture-less objects.

Methods of this type are inherently robust to occlusion, which was demonstrated by Tejani et al. [41]. In their method, they adapt the state-of-the-art template matching feature by Hinterstoisser et al. [25] into a scale-invariant local patch descriptor. Each test patch independently votes for an object and its 3D pose through randomized Hough trees. The resulting, most voted for object detection is accompanied by a detailed occlusion map.

The methods of Brachmann et al. [5] and Tejani et al. [41] are currently one of the top-performing methods on the dataset of Hinterstoisser et al. [24], which is a commonly used dataset for texture-less object detection. A possible limitation of these methods is their linear complexity in the number of trained objects.

2.4 Deep Learning Methods

Following the recent impressive performance boost in many computer vision fields brought by the convolutional neural networks (CNN), it is tempting to experiment with them also in the task of texture-less object detection.

So far, Wohlhart et al. [45] used CNN to obtain descriptors of object views that efficiently capture both the object identity and 3D pose. The CNN was trained by enforcing simple similarity and dissimilarity constraints between the descriptors. The similarity between the resulting descriptors is evaluated by the Euclidean distance and therefore scalable nearest neighbor search methods can be used to efficiently handle a large number of objects under a large range of poses. The learnt descriptor was shown to generalize to unknown objects. The method can work with either RGB or RGB-D images and was shown to outperform state-of-the-art methods on the dataset of Hinterstoisser et al. [24]. However, instead of the whole test images, only regions containing the objects to be detected were used as input of the method.

Held et al. [21] show that CNN outperforms state-of-the-art methods for recognizing textured and texture-less objects from novel viewpoints, even when trained from just a single image per object.

Very recently, Krull et al. [33] used CNN in the hypothesis verification stage to learn how to compare rendered and observed images, while being robust to occlusion and complicated sensor noise. It was empirically observed that the CNN does not specialize to the geometry or appearance of specific objects, and it can be used with objects of vastly different shapes and appearances, and in different backgrounds.

Chapter 3

Our Contributions

As detailed in the following sections, we have so far proposed two methods for texture-less object detection and created a new RGB-D dataset which is supposed to boost further progress in the field.

3.1 Cascade-style Evaluation of Sliding Window Locations

In [28] (Appendix A), we proposed a practical method for detection and accurate 3D localization of multiple texture-less and rigid objects depicted in RGB-D images. The detection procedure adopts the sliding window paradigm, with an efficient cascade-style evaluation of each window location. A simple pre-filtering is performed first, rapidly rejecting most locations. For each remaining location, a set of candidate templates (*i.e.* trained object views) is identified with a voting procedure based on hashing, which makes the method's computational complexity largely unaffected by the total number of known objects. The candidate templates are then verified by matching feature points in different modalities. Finally, the approximate object pose associated with each detected template is used as a starting point for a stochastic optimization procedure that estimates accurate 3D pose. Experimental evaluation shows that the proposed method yields a recognition rate comparable to the state of the art, while its complexity is sub-linear in the number of templates. The method was successfully applied in an industrial robotic scenario within the Darwin project [2] (Figure 1.2).

3.2 Efficient Hypothesis Generation by Edgelet Constellations

In [26] (Appendix B), we proposed a purely edge-based method based on the approach of Damen et al. [12]. The method exploits the recent structured edge detector by Dollár and Zitnick [15], 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 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.

3.3 T-LESS Dataset

We created a new RGB-D dataset for detection and pose estimation of texture-less objects [27]. We call it T-LESS.

The dataset contains training and test RGB-D images for 30 texture-less electrical parts (Figure 3.1) which exhibit properties commonly found in industrial applications, *i.e.* they have no discriminative color, no texture and are often mutually very similar in their shape. Sample test images are shown in Figure 3.2.

The dataset enables evaluation of texture-less object detection methods, texture-less object pose estimation methods, or combinations of the two using quantitative criteria based on the supplied ground truth.

Other RGB-D datasets for texture-less objects are currently available, the most relevant of which being the dataset of Hinterstoisser et al. [24]. On this dataset, the state-of-the-art methods achieve over 95% recognition rate for most objects. Motivated by the need to present new challenges and thus boost further progress in the field, the dataset was designed with the following distinguishing features:

1. Relatively small objects which are in many cases mutually very similar (Hinterstoisser's dataset contains larger objects with very low mutual similarity in both appearance and shape).

2. Accurate ground truth for all known objects in each test image allowing the evaluation of multiple object detection and pose estimation (Hin-

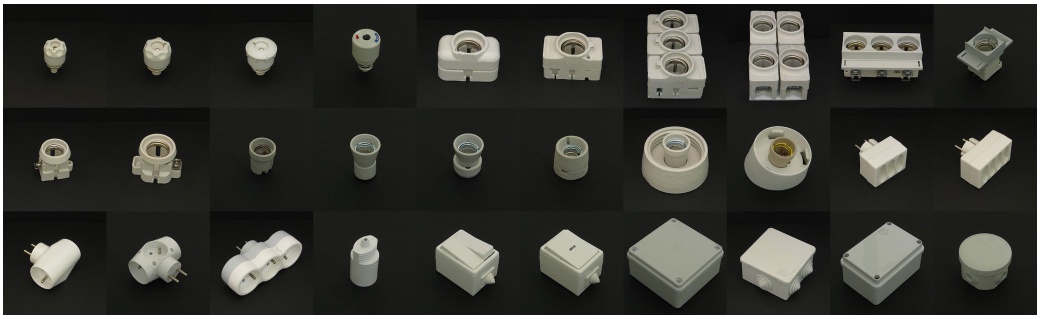


Figure 3.1: 30 texture-less objects contained in the T-LESS dataset. For every object, training templates uniformly covering full view sphere are provided. Each template is annotated with a 3D pose of the object.

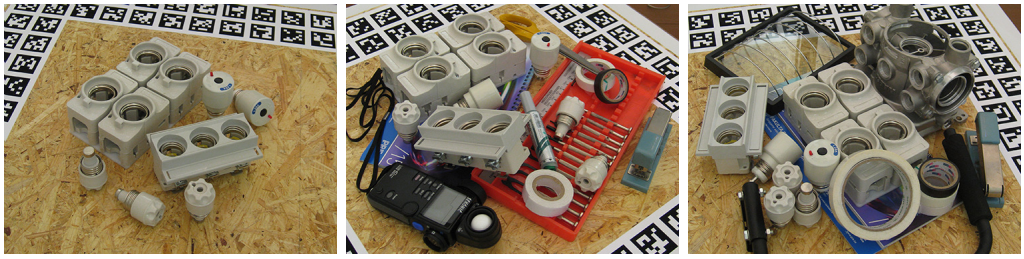


Figure 3.2: Sample test images from the T-LESS dataset. Ground truth in the form of 3D object poses is available for each test image.

terstoisser's dataset provides the ground truth only for a single object per image).

3. Test images present significant distractions in the form of clutter and occlusions (Hinterstoisser's dataset features heavy clutter, but only mild occlusions).

4. Calibrated data from a structured-light sensor, a time-of-flight sensor, and a high-resolution camera (Hinterstoisser's dataset contains only RGB-D 640x480 px images from a structured-light sensor).

The T-LESS dataset will be used for a challenge at the *1st International Workshop on Recovering 6D Object Pose*, which is organized in conjuction with ICCV 2015 in Santiago, Chile [1].

Chapter 4

Goals of the Thesis

The aim of our work is to devise a method for simultaneous detection of multiple texture-less objects (in either RGB or RGB-D images) with the following properties:

1. **Sub-linear complexity in the number of known objects**
2. **Real time performance**
3. **Robustness to occlusion and clutter**
4. **Low false detection rate**
5. **Accurate object localization**

Although the proposed methods [28, 26] address most of these points, there are still open issues. The method proposed in [28] is robust to only mild occlusion. Robustness of its hypothesis generation stage could be increased by taking into account an occlusion model when selecting measurements to be hashed. Furthermore, the verification stage could be made occlusion-aware by *e.g.* taking into account compactness of the matched features. Matched features of a true positive detection of a partially occluded object are likely to be more compact, *i.e.* concentrated on the visible object part, than in the case of a false positive detection, where the features tend to be scattered around the image region. The verification stage could be also improved by reflecting ideas from the field of shape matching [44]. Another possible improvement of [28] is automatic learning of measurement sets for hashing, avoiding the currently used hand-crafted sets. This would allow the method not only to use input data more effectively, but also to adapt to any type of input data.

The method proposed in [26] is relatively sensitive to occlusion and its performance degrades also in the presence of background clutter. When

tracing the edgelet constellations, the method bounces on a randomly selected edgelet from a set of edgelets lying in the tracing direction. As was shown in [26], the robustness to clutter could be increased if the tracing process favours closer edgelets. It could also benefit from edge probability maps [15], which could be used instead of the binary edge maps. The edge probability is supposed to correlate with edge repeatability. If this assumption is confirmed, the edge probability would be a good indicator for selection of the edgelet to be bounced on. The ability to extract a smaller set of more repeatable edgelet constellations would reduce the overall computational cost. An improvement of the verification stage is a topic common to both proposed methods.

Besides improving the proposed methods, we want to keep exploring alternative approaches. Application of convolutional neural networks is one interesting direction. Window proposals for texture-less objects, avoiding the exhaustive sliding window search, is another topic worth exploring. Although mainly focused on detection of texture-less objects, we are interested also in approaches applicable to wiry objects, and eventually in universal approaches applicable to all objects, including textured objects.

Bibliography

[1] 1st international workshop on recovering 6D object pose, in conjuction with ICCV 2015, Santiago, Chile. `http://www.iis.ee.ic.ac.uk/ComputerVision/3DPose-2015.html`, Aug 2015.

[2] DARWIN: Dexterous assembler robot working with embodied intelligence. `http://darwin-project.eu/`, Aug 2015.

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

[4] P.J. Besl and N.D. McKay. A Method for Registration of 3-D Shapes. *PAMI*, 14(2):239–256, 1992.

[5] E. Brachmann, A. Krull, F. Michel, S. Gumhold, and J. Shotton. Learning 6d object pose estimation using 3d object coordinates. In *ECCV*, 2014.

[6] Roberto Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009.

[7] Hongping Cai, Tomáš Werner, and Jiří Matas. Fast detection of multiple textureless 3-D objects. In *ICVS*, volume 7963 of *LNCS*, pages 103–112. 2013.

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

[9] 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.

[10] C. Choi and H.I. Christensen. 3D Pose Estimation of Daily Objects Using an RGB-D Camera. In *IROS*, pages 3342–3349, 2012.

[11] A. Collet, M. Martinez, and S. Srinivasa. The MOPED framework: Object Recognition and Pose Estimation for Manipulation. *I. J. Robotic Res.*, 30(10):1284–1306, 2011.

[12] D. Damen, P. Bunnun, A. Calway, and W. Mayol-Cuevas. Real-time Learning and Detection of 3D Texture-less Objects: A Scalable Approach. In *BMVC*, pages 1–12, Sep 2012.

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

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

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

[16] Bertram Drost, Markus Ulrich, Nassir Navab, and Slobodan Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*, pages 998–1005, 2010.

[17] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2014.

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

[19] Juergen Gall, Angela Yao, Negin Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(11):2188–2202, 2011.

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

[21] David Held, Sebastian Thrun, and Silvio Savarese. Deep learning for single-view instance recognition. *arXiv preprint arXiv:1507.08286*, 2015.

[22] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of texture-less objects. *IEEE PAMI*, 2012.

[23] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *CVPR*, 2010.

[24] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In *ACCV*, 2012.

[25] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011.

[26] Tomáš Hodaň, Dima Damen, Walterio Mayol-Cuevas, and Jiří Matas. Efficient texture-less object detection for augmented reality guidance. In *2015 IEEE International Symposium on Mixed and Augmented Reality Workshops (ISMARW)*.

[27] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset and evaluation protocol for detection and pose estimation of texture-less objects. `http://cmp.felk.cvut.cz/t-less/`, Aug 2015.

[28] Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3D pose estimation of texture-less objects in RGB-D images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[29] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *arXiv preprint arXiv:1502.05082*, 2015.

[30] Edward Hsiao and Martial Hebert. Occlusion reasoning for object detectionunder arbitrary viewpoint. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(9):1803–1815, 2014.

[31] Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[32] Jeff Kramer, Nicolas Burrus, Florian Echtler, Herrera C Daniel, and Matt Parker. *Hacking the Kinect*, volume 268. Springer, 2012.

[33] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. *arXiv preprint arXiv:1508.04546*, 2015.

[34] 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.

[35] D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157, 1999.

[36] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004.

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

[38] Reyes Rios-Cabrera and Tinne Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *ICCV*, pages 2048–2055, 2013.

[39] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[40] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese. Depth-encoded Hough Voting for Joint Object Detection and Shape Recovery. In *Proc. ECCV'10*, pages 658–671, 2010.

[41] Alykhan Tejani, Danhang Tang, Rigas Kouskouridas, and Tae-Kyun Kim. Latent-class hough forests for 3d object detection and pose estimation. In *ECCV*, 2014.

[42] Federico Tombari, Alessandro Franchi, and Luigi Di. BOLD features to detect texture-less objects. In *ICCV*, pages 1265–1272, 2013.

[43] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[44] Remco C Veltkamp. Shape matching: Similarity measures and algorithms. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 188–197. IEEE, 2001.

[45] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. *arXiv preprint arXiv:1502.05908*, 2015.

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

[47] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.

Appendix A

Publication at IROS 2015

An article accepted at 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2015).

Detection and Fine 3D Pose Estimation of Texture-less Objects in RGB-D Images

Tomáš Hodaň[♯], **Xenophon Zabulis**[♭], **Manolis Lourakis**[♭], **Štěpán Obdržálek**[♯], **Jiří Matas**[♯]
[♯]*Center for Machine Perception, Czech Technical University in Prague, Czech Republic*
hodantom|xobdrzal|matas@cmp.felk.cvut.cz
[♭]*Institute of Computer Science, Foundation for Research and Technology - Hellas, Heraklion, Greece*
zabulis|lourakis@ics.forth.gr

***Abstract*— Despite their ubiquitous presence, texture-less objects present significant challenges to contemporary visual object detection and localization algorithms. This paper proposes a practical method for the detection and accurate 3D localization of multiple texture-less and rigid objects depicted in RGB-D images. The detection procedure adopts the sliding window paradigm, with an efficient cascade-style evaluation of each window location. A simple pre-filtering is performed first, rapidly rejecting most locations. For each remaining location, a set of candidate templates (*i.e.* trained object views) is identified with a voting procedure based on hashing, which makes the method's computational complexity largely unaffected by the total number of known objects. The candidate templates are then verified by matching feature points in different modalities. Finally, the approximate object pose associated with each detected template is used as a starting point for a stochastic optimization procedure that estimates accurate 3D pose. Experimental evaluation shows that the proposed method yields a recognition rate comparable to the state of the art, while its complexity is sub-linear in the number of templates.**

I. Introduction

Texture-less, smooth and uniformly colored objects occur frequently in robotic applications that range from personal robotics to intelligent manipulation and assembly. Common to such applications is the requirement of identifying and accurately localizing known objects so that they can be acted upon by a robot end effector. Fig. 1 depicts an example of a robotic assembly scenario involving several texture-less objects. An arm with a gripper is assigned the task of picking up electrical fuses, at arbitrary locations in its workspace, and inserting them into the sockets of corresponding fuse boxes.

The method detailed in this paper aims at the reliable simultaneous detection of multiple texture-less objects with low false detection rate, real time performance, and sub-centimeter accuracy in object localization. The input to the method consists of RGB-D images provided by a consumer-grade depth sensor such as Kinect. Such sensors provide aligned color and depth images that concurrently capture both the appearance and geometry of a scene.

This work was supported in part by the EC FP7 programme under grant no. 270138 DARWIN, 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.

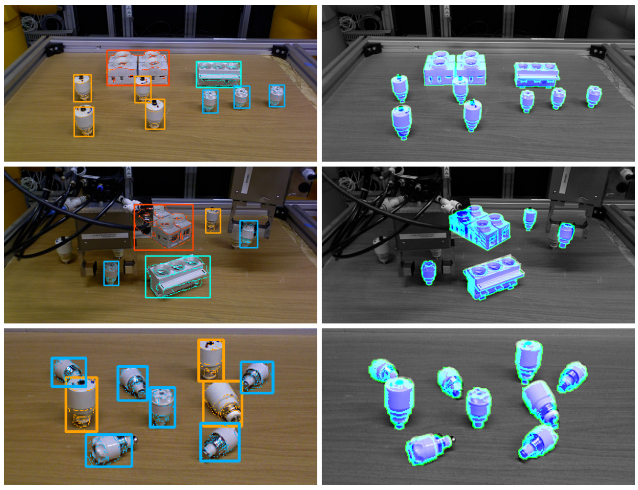


Fig. 1. *Left:* Detection of multiple instances of multiple texture-less objects (fuses and fuse boxes of different types) in a robotic assembly task. *Right:* Superimposed renderings of object models at the estimated 3D poses.

While object recognition is a long-standing and widely studied problem, most attention until recently has been paid to the recognition of textured objects, for which discriminative appearance features, invariant to changes in pose and illumination, can be readily extracted [1]. These objects are often assumed to have piece-wise planar surfaces. Their appearance variations can be therefore modeled by a simple geometric transformation (*e.g.* similarity), which can be reliably determined from the rich textural information. Candidate object locations in the scene are typically determined by identifying so-called interest points or interest regions [2], a strategy which drastically reduces the overall computational cost compared to exhaustive image search. However, when applied to texture-less objects, interest point detectors typically fail to identify corresponding image regions and common local appearance descriptors are no longer discriminative enough to provide reliable correspondences [3].

Recognition and localization of texture-less objects is challenging in several respects. An object's appearance is dominated by its shape, its material properties and by the configuration of light sources. Unless these are known in

Fig. 2. Evaluation cascade of the proposed method. Note the typical numbers of detection candidates advancing through individual stages (for a template size $108 \times 108\,px$, a VGA input image and a scale space with four larger and four smaller scales with a scaling factor 1.2). A detection candidate is a triplet consisting of a template identifier (object and its orientation), a sliding window scale and a sliding window location.

advance and precisely controlled, it is easier to capture possible object appearances exhaustively rather than attempting to describe them with covariant features. In other words, each object can be represented by hundreds or even thousands of images, called templates, which depict it from multiple viewing angles. Being equivalent to matching an image region to one of the templates or asserting there is no suitable such template (*i.e.* the region corresponds to background), the detection task avoids the need of generalization to unknown transformations.

Existing approaches to the detection of texture-less objects usually proceed by sweeping sliding windows of several discrete sizes over the entire image with a small pixel step, searching for a match against all stored object templates. These methods scale poorly to large numbers of objects and special attention has to be paid to implementation details, otherwise they are too slow for real-time operation.

The proposed method addresses the excessive computational complexity of sliding window approaches and achieves high efficiency by employing a cascade-style evaluation of window locations. Fast filtering is performed first, rejecting quickly most of the locations by a simple salience check. For each remaining location, candidate templates are obtained by an efficient voting procedure based on hashing measurements sampled on a regular grid. This makes the complexity of the method sub-linear in the total number of known objects. The candidate templates are then verified by matching feature points in different modalities. Each template is associated with a training-time pose, *i.e.* a 3D rotation and distance to the camera reference frame origin. Therefore, a successful match against a template provides a rough estimate of the object's 3D location and orientation. As a final step, a stochastic, population-based optimization scheme is applied to refine the pose by fitting a 3D model of the detected object to the input depth map. The pipeline of our method is illustrated in Fig. 2 together with the typical numbers of detection candidates advancing through individual stages.

After reviewing relevant related work in Sec. II, the proposed method is detailed in Sec. III and IV. Sec. V presents experimental results and Sec. VI concludes the paper.

II. Related Work

A. Texture-less Object Detection

Template matching is one of the earliest techniques applied to object detection in images. Traditional approaches typically use only a few stored templates per object, perform a sequential scan of the input image and compute correlation coefficients between each window and the stored templates. A sufficiently high correlation score indicates a successful match. Correlation employs intensity images, image gradients or edges. Invariance is achieved only w.r.t. translation, with little tolerance to misalignments. The interested reader is referred to [4] for a survey.

Due to their low generalization and limited applicability, template-based techniques were for some time out of the mainstream research agenda. Instead, research in object recognition concentrated on approaches based on viewpoint-invariant local features obtained from objects rich in texture [5]. Such approaches require only a small number of training images per object and generalize well to a wide range of possible appearances. As computers became faster and equipped with more memory, template-based methods grew in popularity again. Today it is not unusual to maintain thousands of templates per object, thus capturing varying visual aspects exhaustively.

In recent work, Hinterstoisser et al. [6], [7] have introduced an efficient template matching technique. Instead of a raw image, object templates are represented by a set of carefully selected feature points in different modalities (specifically orientation of intensity gradients and orientation of 3D surface normals). Measurements at feature points are quantized and represented as bit vectors, allowing for fast matching with binary operations. Tolerance to misalignments is achieved by comparing the binarized representation with pixels in a small local neighbourhood. The pose retrieved from the best matching template is used as a starting point for subsequent refinement with the Iterative Closest Point (ICP) algorithm [8]. With data structures optimized for fast memory access and a highly vectorized implementation using special SSE hardware instructions, the method is capable of real-time matching of several thousands of templates. However, its performance is expected to degrade noticeably for large object databases, since its time complexity is linear in the number of loaded templates (around 3000 templates are employed per object). The matching procedure of [6] inspired the verification stage of our proposed method.

An alternative approach to 3D object detection that requires only 3D object models was presented by Drost et al. [9]. During training, all possible pairs of 3D points on a model are described and recorded in a hash table. During detection, sampled pairs of 3D points from the test scene are described and used to vote for corresponding object pose hypotheses. The most voted pose clusters can be then refined with ICP. Choi and Christensen [10] further

augmented the point pair feature with color information. The efficiency and performance of these methods depend directly on the complexity of the 3D scene, which might limit their applicability to real-time applications.

Another class of methods relies solely on intensity edges, *e.g.* [11], [12], [3]. Albeit such methods can operate very fast, their recognition capability is inherently lower compared to methods also taking into account depth information. Cai et al. [11] employed a sliding window approach with hypothesis generation based on hashing distances and orientations of the nearest edges from points on a fixed regular grid. We use a similar hashing scheme in the proposed method.

B. 3D Pose Estimation

A common aspect of the approaches mentioned in Sec. II-A is that the pose associated with the detected object is approximate. This is due to the limited resolution of the pose sampling process employed in training or possible mismatches, and necessitates the refinement of the retrieved pose with a geometric optimization step. The ICP algorithm [8] is the most common choice for this purpose. ICP represents the gold standard method for geometrically aligning two sets of points whose relative pose is approximately known. However, when the two point sets are relatively far apart or have a small overlap, ICP's strategy of matching closest points generates large numbers of incorrect correspondences. The situation is aggravated by the inevitable presence of noise and outliers. As a result, ICP can easily get stuck in local minima and its performance largely depends on the quality of initialization. To counter this, numerous enhancements to the basic ICP have been proposed that aim to improve the speed of convergence or increase robustness to local minima, outlying points and noise [13]. These enhancements often require considerable trial and error for tuning their parameters to a particular application. Here we take a different approach and refine 3D pose with an optimization scheme based on Particle Swarm Optimization (PSO) [14]. PSO has proven to be an effective framework for dealing with other flavors of pose estimation, *e.g.* [15], [16].

III. Detection of Texture-less Objects

Detection of objects in an input RGB-D image is based on a sliding window approach, operating on a scale pyramid built from the image. Let $\mathcal{L}$ denote the set of all tested locations. The number of locations $|\mathcal{L}|$ is a function of image resolution, spatial image sampling by the sliding window (*e.g.* every 5 pixels), scale range (*e.g.* two or four octaves), and scale space discretisation. The known objects are represented with a set $\mathcal{T}$ of template images – RGB-D images of a fixed size. There are several thousands of templates per object, capturing its appearance from all possible viewing angles, but from a fixed distance. The training distance, which then affects the depth channel of an RGB-D template, is object-specific but fixed for all templates of a certain object. This distance is chosen so that the object would optimally fill the template image if observed with a camera with identical intrinsic parameters (focal length and resolution) as the camera observing later the test scene. Each template is associated with the object ID, the training distance Z_t and the object orientation $\mathbf{R}_0$ it represents.

In general, every window $w_\mathbf{l},\ \mathbf{l} = (x, y, s),\ \mathbf{l} \in \mathcal{L}$, needs to be tested against every template, which makes the asymptotic complexity $\mathcal{O}(|\mathcal{L}||\mathcal{T}|)$. This is computationally very demanding even for moderate numbers of known objects. We therefore propose a cascaded evaluation, where the set of candidate locations $\mathcal{L}$ is quickly reduced (Sec. III-A) and the candidate templates $\mathcal{T}$ are pruned (Sec. III-B) before the template matching itself (Sec. III-C) is performed.

A. Pre-filtering of Window Locations

To reduce the number of image locations, an image window is first assessed with a simple *objectness* measure [17], [18], *i.e.* its likelihood that it contains any of the objects. This corresponds to a two-class classifier distinguishing between background and object classes, with the object class encompassing all the templates in $\mathcal{T}$.

Our objectness measure is based on the number of depth-discontinuity edgels within the window, and is computed with the aid of an integral image for efficiency. Depth-discontinuity edgels arise at pixels where the response of the Sobel operator, computed over the depth image, is above a threshold θ_e, which is set to 30% of the physical diameter of the smallest object in the database. The window is classified as containing an object if its number of depth edgels is at least 30% of the number of depth edgels in the template containing the least amount of them. This setting is tolerant to partial occlusions but still strong enough to prune most of the window locations – roughly 90% to 99% of them in images of our robot workspace (Fig. 1), depending on the scene clutter. Only image windows that pass the objectness test are processed further.

B. Hypothesis Generation

In this phase, a small subset of candidate templates is quickly identified for each image window that passed the objectness test. Up to N templates with the highest probabilities $p_t(t|w_\mathbf{l}),\ t \in \mathcal{T}$ are retrieved. This can be seen as a multi-class classification problem where there is one class for each training template, but none for the background.

The procedure retrieves candidate templates from multiple (for robustness) trained hash tables, which is a constant complexity $\mathcal{O}(1)$ operation in the number of stored templates. Each hash table $h \in \mathcal{H}$ is indexed by a trained set $\mathcal{M}_h$ of measurements taken on the window $w_\mathbf{l}$ or template t, discretized into a hash key. $\mathcal{M}_h$ is different for each table. The table cells contain lists of templates with the same key, the lists are then used to vote for the templates. A template can receive up to $|\mathcal{H}|$ votes, in which case all the template's measurement sets (for all the tables) would be discretised to the same hash keys as measurements on the window $w_\mathbf{l}$. Up to N templates with the highest number of votes, and with at least v votes, are passed onward to the next step of the detection cascade. The voting is still an $\mathcal{O}(|\mathcal{T}|)$ operation for each $w_\mathbf{l}$.

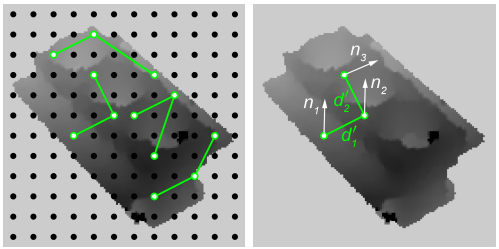


Fig. 3. Templates and test windows are hashed using measurements from trained triplets of grid points. *Left:* Sample triplets which are valid for the shown template, *i.e.* their points lie in the object mask. *Right:* A triplet is described by depth differences $\{d_1', d_2'\}$ and normal vectors $\{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\}$.

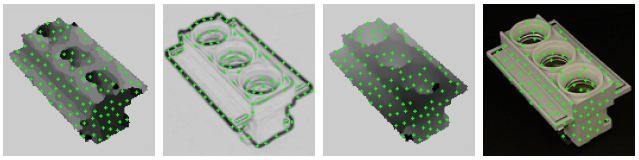


Fig. 4. Feature points in different modalities whose consistency is evaluated in the hypothesis verification. The points are trained independently for each template. *Left to right:* surface normals, image gradients, depth, color.

Measurement sets $\mathcal{M}_h$ and their quantization. The hashing/voting procedure was inspired by the work of Cai et al. [11]. A regular grid of 12×12 reference points is placed over the training template or sliding window. This yields 144 locations from which k-tuples are sampled. We use triplets in our setup, *i.e.* $k = 3$ (Fig. 3). Each location is assigned with a depth d and a surface normal $\mathbf{n}$. A measurement set $\mathcal{M}_h$ is a vector consisting of $k-1$ relative depth values and k normals, $\mathcal{M}_h = (d_{2_h} - d_{1_h}, d_{3_h} - d_{1_h}, \ldots, d_{k_h} - d_{1_h}, \mathbf{n}_{1_h}, \ldots, \mathbf{n}_{k_h})$. The relative depths $d_{i_h} - d_{1_h}$ are quantized into 5 bins each, with the quantization boundaries learned from all the training templates to provide equal frequency binning, *i.e.* each bin contains the same number of templates. To quantize surface normals we use the approach proposed in [7], where the normals are quantized to 8 discrete values based on their orientation. For triplets of reference points we have two relative depths and three normals, leading to a hash table size of $5^2 8^3 = 12800$ bins.

Training-time selection of measurement sets. To provide robustness to occlusion and noise, multiple hash tables are built, which differ in the selection of the k-tuples drawn from the 144 reference points. The k-tuples can be chosen randomly. Alternatively, they can be optimally selected to (a) cover maximally independent measurements (for robustness), and (b) to fill the tables as uniformly as possible (for stable detection time). The optimal selection is unfortunately NP-complete, therefore we employ a hybrid heuristic strategy. We first randomly generate a set of m, $m \gg |\mathcal{H}|$, k-tuples and then retain the subset with the largest joint entropy of the quantized measurements.

For the results reported below, $|\mathcal{H}| = 100$ hash tables were employed, chosen from $m = 5000$. The minimal number of votes per template was v = 3 and the maximum number of candidates passed to the verification stage was $N = 100$.

C. Hypothesis Verification

The verification stage corresponds to the traditional template matching. Thanks to the template selection in the previous step, only up to N templates are considered for each image window w_1 that passed the initial objectness test. This makes the complexity of this stage constant in the number of stored templates. Since the templates were already identified in the previous step, the verification can be seen as a set of up to N separate two-class classification problems discriminating between the object represented by a template and the background class, *i.e.* $p(\text{obj}|t_i, w_1) \lesseqgtr p(\text{bkg}|t_i, w_1)$.

The verification proceeds in a sequence of tests evaluating the following: **I** object size in relation to distance, **II** sampled surface normals, **III** sampled image gradients, **IV** sampled depth map, and **V** sampled color. The tests are ordered according to increasing computational cost. Any failed test classifies the window as non-object – corresponding to either the background, or an object not represented by template t_i – and subsequent tests are not evaluated.

Test **I** verifies that the observed object size (*i.e.* the level of the scale pyramid) corresponds to its distance measured in the depth map. The object is expected at distance Z_e calculated as $Z_e = Z_t s$, where s is the scale factor of the pyramid level and Z_t is the template's training distance. If the measured depth Z_w is within the interval $|Z_e/\sqrt{f}, Z_e \cdot \sqrt{f}|$, where f is the discretization factor of the scale space, the depth Z_w is considered to be feasible, otherwise the test fails.

Tests **II** and **III** verify orientation of surface normals and intensity gradients at several feature points. Following [19], the point locations are greedily extracted during the training stage, independently for each template (Fig. 4). The feature points for the surface normal orientation test are extracted at locations with locally stable orientation of normals (*i.e.* further away from depth discontinuities). For the intensity gradient orientation test, the feature points are extracted at locations with large gradient magnitude (*i.e.* typically on the object contour). We extract 100 points in both cases. The orientations are quantized and compared template-against-sliding window, which can be done very fast by bitwise operations using response maps described in [6].

The depth map test **IV** and the color test **V** reuse the locations of feature points extracted for the surface normal test **II**. In the depth test, difference d between the depth in the template and the depth in the window is calculated for each feature point. A feature point is matched if $|d - d_m| < kD$, where d_m is the median value of ds over all feature points, D is the physical object diameter, and k is a coefficient (set to 0.05 in our experiments). Finally, pixel colors in test **V** are compared in the HSV space, as done in [19].

A template passes the tests **II** to **V** if at least θ_c of the feature points have a matching value within a small neighbourhood. In our experiments $\theta_c = 60\%$ to tolerate partial occlusions, and the extent of the local neighborhood is 5×5 pixels to compensate for the sliding window step, and for the discretization of orientations during training. A verified template that passes all the tests is assigned a final

score computed as $m = \sum_{i \in \{\mathbf{II} \ldots \mathbf{V}\}} c_i$, where c_i is the fraction of matching feature points in tests **II** to **V**.

D. Non-maxima Suppression

The verified templates are accumulated from all different locations and scales. Since different views of one object are often alike, and since multiple objects may by rather similar, unique detections are identified by repeatedly retaining the candidate with the highest score r, and removing all detections that have a large overlap with it. The score is calculated as $r = m(a/s)$, where m is the verification score defined above, s is the detection scale, and a is the area of the object in the considered template. Weighting the score by the object area favours detections which explain more of the scene (*e.g.* when a cup is seen from a side, with the handle visible, we prefer a template depicting the handle over other templates where the handle is occluded, but which would otherwise yield the same matching score). The retained detections are passed to the 3D pose estimation stage, together with the approximate 3D poses which have been associated with the training templates.

IV. Fine 3D Pose Estimation

Fine 3D pose estimation refers to the accurate computation of translation and rotation parameters that define an object's position and orientation in space, assuming that approximate initial values for these parameters are provided. This process receives as inputs a mesh model of the object, an initial object pose $\{\mathbf{R}_0,\ \mathbf{t}_0\}$, a depth image and the sensor's intrinsics and outputs a refined pose $\{\mathbf{R},\ \mathbf{t}\}$. Objects are represented with arbitrary 3D mesh models, which can originate from CAD drawings or from digital scans. A mesh $\mathcal{M}$ is comprised of an ordered set of 3D vertex points V and an ordered set G of triplet indices upon V that define the mesh triangles. The 3D oriented bounding box B of each model is precomputed using the eigenvectors of V's covariance matrix.

Candidate poses $\{\mathbf{R}_i, \mathbf{t}_i\}$ are generated and then evaluated by using them to synthesize renderings of $\mathcal{M}$, producing depth images S_i (see Sec. IV-B). A scoring function yields score $o(i)$, which quantifies the similarity between each image S_i and the input using depth, edge and orientation cues (Sec. IV-C). PSO is used to optimize the scoring function and find the pose whose rendered depth image is the most similar to the input one (Sec. IV-D). An overview of the approach is provided in Fig. 5 whereas its components are briefly described in the following subsections. A detailed presentation and evaluation of our pose estimation pipeline can be found in [20].

A. Initialization

The initial pose used to bootstrap pose estimation can be quite crude. The projection on the sensor of the model at the initial pose determines a 2D, axis-aligned bounding box b. This box is inflated proportionally to the distance of the initial pose to mitigate any pose inaccuracies and is assumed to enclose most of the projection of the target object on the input image.

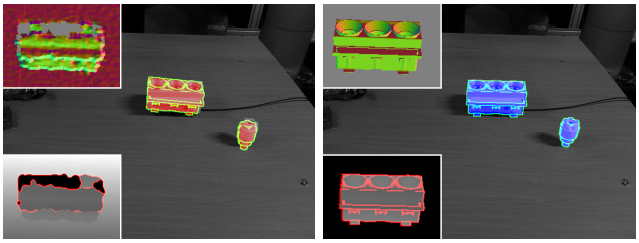


Fig. 5. Pose estimation for an electrical fuse and fuse box. *Left:* Initial poses superimposed on a captured image (their misalignment can be seen by zooming in). Thumbnails show in magnification captured depths and detected edges (bottom), along with color-coded surface normals (top) in the region of the fuse box. *Right:* Refined poses. Thumbnails show corresponding fuse box depths and surface normals obtained by rendering.

To suppress sensor noise, the acquired depth image is median filtered with a 5×5 kernel and the result is retained as image D. Depth shadows and other shortcomings of consumer depth sensors manifest themselves as invalid pixels in D, not contributing with 3D points. Surface normals for valid depth pixels are estimated by local least-squares plane fitting and stored in N [21]. Binary image E is computed from D, by thresholding the output of the Sobel operator applied to D. The distance transform T of E is computed for later use. The above computations are parallelized on a GPU at the pixel level, while the computation of T uses the parallel formulation of [22]. To evaluate an input frame, only the depth image D is uploaded to the GPU which then uses it to compute N and T. No other input data exchange with the GPU occurs during pose estimation.

B. Pose Hypotheses Rendering

A rendering process simulates depth images of the target object at a hypothesized pose against a blank background. Pose rendering is formulated as follows. Transform $\{\mathbf{R}_0, \mathbf{t}_0\}$ brings the model in an approximate location and orientation, in the depth sensor's reference frame. Candidate poses are parametrized relative to this initial pose, using a relative translation $\mathbf{t}_i$ and an "in place" rotation $\mathbf{R}_i$ about the centroid $\mathbf{c}$ of points in V. Specifically, the model is first translated by $-\mathbf{c}$, rotated by $\mathbf{R}_i$, and translated back to place by $\mathbf{c}$. Rotation $\mathbf{R}_i$ is the product of primitive rotations about the 3 axes: $\mathbf{R}_i = \mathbf{R}_x(\theta_i) \cdot \mathbf{R}_y(\phi_i) \cdot \mathbf{R}_z(\omega_i)$. The transformation model point $\mathbf{x}$ undergoes is thus $\mathbf{R}_i \cdot (\mathbf{x} - \mathbf{c}) + \mathbf{c} + \mathbf{t}_i$. To avoid repeated transformations, the initial and candidate poses are combined into the following:

$$\mathbf{R}_i \cdot \mathbf{R}_0 \cdot \mathbf{x} + \mathbf{R}_i \cdot (\mathbf{t}_0 - \mathbf{c}) + \mathbf{c} + \mathbf{t}_i. \qquad (1)$$

The rotational component of candidate poses is parameterized using Euler angles whereas their translation is parameterized with Euclidean coordinates. The model transformed according to Eq. (1), is rendered in depth image S_i. Depth edges and surface normals of S_i are computed and stored in binary image E_i and data structure N_i, respectively.

Computation and storage of S_i, E_i, and N_i is delegated to the GPU. The process employs Z-buffering to respect visibility and realistically render self-occlusions. Parallelization

is performed at two levels of granularity. At a fine level, rendering is parallelized upon the triangles of the rendered mesh. At a coarser level, multiple hypotheses are rendered simultaneously, with a composite image gathering all renderings. In this manner, multiple hypotheses are evaluated in a single batch, resulting in better utilization of GPU resources and reduced communication. Edge detection is applied once, directly upon the composite image.

C. Pose Hypotheses Evaluation

A candidate pose is evaluated with respect to the extent to which it explains the input depth image. Objective function $o(\cdot)$ avails a score $o(i)$ and considers the similarity of depth values, surface normals, as well as depth edges between D and S_i. Two range images are corresponded in terms of their coordinates and are compared as follows.

Depth values are directly compared between D and S_i for pairs of pixels. For n pixel pairs, depth differences δ_k, are computed and the cumulative depth cost term is defined as:

$$d_i = \sum_{k=1}^{n} 1/(|\delta_k| + 1), \tag{2}$$

where $|\delta_k|$ is set to ∞ if greater than threshold d_T ($20\,mm$ in our implementation) to avoid comparing with background surfaces. For the same n pairs of pixels, the cost due to surface normal differences is quantified as:

$$u_i = \sum_{k=1}^{n} 1/(|\gamma_k| + 1), \tag{3}$$

where γ_k is the angle between the two surface normals, provided by their dot product. Edge differences are aggregated in an edge cost using E and E_i. Let m be the number of edgels of E_i within b. For each such edgel j, let ϵ_j denote the distance from its closest edgel in D which is looked up from T. The corresponding edge cost term is then:

$$e_i = \sum_{j=1}^{m} 1/(\epsilon_j + 1). \tag{4}$$

Each of the cost terms in Eqs. (2), (3) and (4) involves two ordered pixel sets, one from each image D and S_i, that contain the pixel locations to be compared. As d_i, u_i, and e_i have different numeric ranges, the combined cost is defined by their product $o(i) = -d_i \cdot e_i \cdot u_i$, where the minus sign is used to ensure that optimal values correspond to minima, since d_i, e_i and u_i are non-negative. Summing the reciprocals of partial differences $|\delta_k|$, $|\gamma_k|$ and ϵ_j rewards poses that maximize the support (*i.e.* spatial overlap) between the compared regions of D and S_i. The objective function improves when more pixels in the rendered depth map closely overlap with the imaged surfaces in the input image.

As no segmentation is employed, inaccurate pose hypotheses might cause the rendered object to be compared against pixels imaging background or occluding surfaces. To counter this, only pixels located within b are considered. Hypotheses that correspond to renderings partially outside b obtain a poor similarity score and the solution does not drift towards an irrelevant surface. Also, during the evaluation of each hypothesis, the oriented bounding box B_i that corresponds to hypothesis i is computed by transforming B according to Eq. (1). By so doing, depth pixels from D that correspond to 3D points outside B_i are not considered in the comparison, as they are irrelevant to the hypothesis being evaluated.

D. Pose Estimation

The search space for the pose estimation is constrained in a 6D neighborhood of the initial pose estimate. Each dimension of the pose search space is bounded, defining a search hyperrectangle centered on the initial pose estimate. As the cost of an exhaustive search in $\mathcal{R}^6$ is prohibitive, a numerical optimization approach is adopted to minimize objective function $o(\cdot)$. This minimization is performed with PSO, which stochastically evolves a population of candidate solutions dubbed particles, that explore the parameter space in runs called generations. PSO does not require knowledge of the derivatives of the objective function, depends on very few parameters and requires a relatively small number of objective function evaluations until convergence. Compared to gradient-based optimization methods, PSO has a wider basin of convergence, exhibiting better robustness to local minima. Furthermore, as particles evolve independently at each generation, it is amenable to an efficient parallel implementation [20].

V. Experiments and Evaluation

A. Object Localization

The presented method was evaluated quantitatively with the aid of the publicly available dataset by Hinterstoisser et al. [19]. This dataset includes 15 texture-less objects and provides for each a 3D mesh model and a test sequence consisting of approximately 1200 RGB-D frames in VGA resolution. The test sequences feature heavy 2D and 3D clutter, mild occlusions and large viewpoint variations and are accompanied by the ground truth object pose for each frame. The task is to localize the given object in each frame, *i.e.* to detect it and estimate its 3D pose.

Training templates were rendered from the provided 3D models so that they uniformly covered the upper view hemisphere (with a step of 10° in both azimuth and elevation). To achieve invariance to rotation around the optical axis, an in-plane rotation to each template (from -40° to 40° with a step of 10°) was also applied. In total, each object was represented by 2916 templates of size $108 \times 108\,px$. Each test image was scanned at 9 scales (4 larger and 4 smaller scales with scaling factor 1.2) with a scanning step of $5\,px$.

We compare our method to the LINEMOD [6] and LINEMOD++ [7] methods of Hinterstoisser et al. and the method of Drost et al. [9]. These methods were already briefly described in Sec. II-A; here we provide more details regarding their relation to our method. LINEMOD follows an exhaustive template matching approach. LINEMOD++ extends it by two post-processing verification steps. Specifically, a color check and a depth check (by a rough but fast ICP) are performed in order to prune hypotheses. A finer ICP is then carried out for the best of the remaining hypotheses. The essential difference of our method is the addition of the

Sequence	Our method	LINEMOD++	LINEMOD	Drost et al.
1. Ape	93.9	**95.8**	69.4	86.5
2. Benchvise	**99.8**	98.7	94.0	70.7
3. Bowl	98.8	**99.9**	99.5	95.7
4. Box	**100.0**	99.8	99.1	97.0
5. Cam	95.5	**97.5**	79.5	78.6
6. Can	**95.9**	95.4	79.5	80.2
7. Cat	98.2	**99.3**	88.2	85.4
8. Cup	**99.5**	97.1	80.7	68.4
9. Driller	**94.1**	93.6	81.3	87.3
10. Duck	94.3	**95.9**	75.9	46.0
11. Glue	**98.0**	91.8	64.3	57.2
12. Hole punch	88.0	**95.9**	78.4	77.4
13. Iron	97.0	**97.5**	88.8	84.9
14. Lamp	88.8	**97.7**	89.8	93.3
15. Phone	89.4	**93.3**	77.8	80.7
Average	95.4	**96.6**	83.0	79.3

TABLE I

RECOGNITION RATES [%] FOR THE DATASET OF [19] AND $k_m = 0.1$ (*i.e.* THE PERCENTAGE OF OBJECTS LOCALIZED WITH AN ERROR SMALLER THAN 10% OF THEIR DIAMETER).

pre-filtering and the hypothesis generation stage, avoiding the exhaustive search.

We used the same quantification of pose error as in [19]. That is, for the ground truth pose $\{\mathbf{R}_g, \mathbf{t}_g\}$ and the estimated pose $\{\mathbf{R}_e, \mathbf{t}_e\}$, the error is $e = 1/\nu \sum_i |\mathbf{g}_i - \mathbf{e}_i|$, where $\mathbf{g}_i = \mathbf{R}_g x_i + \mathbf{t}_g$, $\mathbf{e}_i = \mathbf{R}_e x_i + \mathbf{t}_e$, and i enumerates the ν vertices of V. For objects with ambiguous pose due to their symmetry (namely "Cup", "Bowl", "Box" and "Glue"), the error is computed as $e = 1/\nu \sum_i min_j |\mathbf{g}_i - \mathbf{e}_j|$. An object is considered to be correctly localized if $e \leq k_m d$, where k_m is a fixed coefficient and d is the diameter of the model, *i.e.* the maximum distance between any of its vertices.

As in the methods being compared, the best detection of the object of interest was selected and evaluated in each frame. For the detected template with the highest matching score, the corresponding initial 3D pose was refined by our pose estimation method and the error of the resulting pose was calculated as explained above. Table I compares the recognition rates (for $k_m = 0.1$) of our method with the rates of the other methods which were published in [19]. Our method achieved an average recognition rate of 95.4% (*i.e.* the percentage of correctly localized objects) and outperformed LINEMOD and the method of Drost et al. The average recognition rate achieved by LINEMOD++ is better by 1.2%. Recognition rates of our method with respect to different values of k_m can be found in Fig. 6 (top). The benefit of the pose estimation stage can be seen in Fig. 6 (middle), where the average pose errors of the initial and the refined poses are compared. Pose refinement employed PSO with parallelized hypotheses rendering (cf. Sec. IV-B). With 100 particles and 100 generations it required around $0.19\,s$ per frame to estimate the pose of a single model with $\approx 7K$ triangles. For comparison, when hypotheses were evaluated sequentially on the GPU, PSO required $4.8\,s$. In [20], we show that our PSO-based pose estimation delivers superior results compared to the commonly used ICP. Visualizations

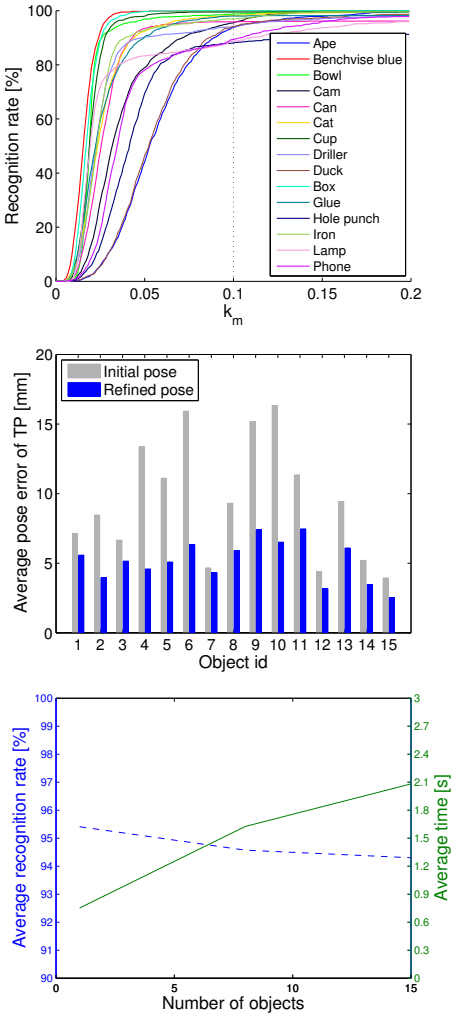


Fig. 6. *Top:* Recognition rates for various k_m. *Middle:* Average errors of initial and refined poses of true positives for $k_m = 0.1$ (object names corresponding to the numbers can be found in Table I). *Bottom:* Average recognition rate (dashed line) and average recognition time w.r.t. the number of loaded object templates (there were 2916 templates per object).

of sample results are in Fig. 7.

To evaluate scalability, we also run our method when templates of 8 and 15 objects were loaded for detection. As can be seen in Fig. 6 (bottom), the complexity of our method was proven sub-linear in the number of templates ($0.75\,s$ vs. $2.08\,s$ when templates of 1 and 15 objects were loaded) while the recognition rate drops only slightly (by only 1% when the 43740 templates of all 15 objects were loaded w.r.t. the case when only 2916 templates of a single object were loaded). The sub-linearity is achieved by the hypothesis generation stage which allows the comparison of only a small set of templates for each window location.

In terms of running time, the whole method needed on average $0.75\,s$ per VGA frame. This time was achieved by our parallelized C++ implementation on a modern desktop PC equipped with a 16 core CPU and an NVIDIA GTX 780 GPU (the GPU was only employed during the pose

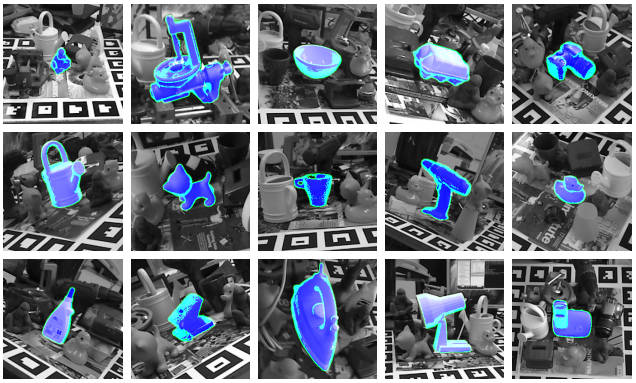


Fig. 7. Sample 3D pose estimations on the dataset of [19] (cropped).

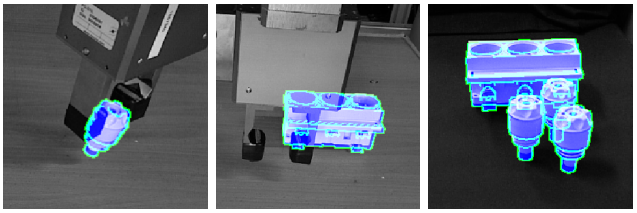


Fig. 8. Cropped close-ups of 3D pose estimations in a manipulation task.

estimation). As reported in [19], the method of Drost needed on average $6.3\,s$ and LINEMOD++ only $0.12\,s$. The latter time was achieved with a highly optimized implementation using heavy SSE parallelization and a limited scale space (for each object, only a limited set of scales were considered). With a similar level of optimization, we expect our method to run even faster since it evaluates only a small set of templates for each window. In the case of multiple object detection and localization that arises often in robotic applications, our method is expected to outperform LINEMOD++ in terms of running time.

B. Robotic Application

The intended application of the proposed method relates to robotic manipulation and assembly of objects. Figs. 1 and 8 demonstrate its suitability for a manipulation task with electrical parts along with its tolerance to mild occlusions. When a robotic gripper was present in the scene, its posture was accurately provided by motor encoders and used to mask out the corresponding pixels in the image, preventing them from contaminating pose estimation. As shown in Fig. 6 (middle), the poses estimated with the presented method achieve a sub-centimeter average accuracy in the estimated pose (for the recognition of 95.4%). This meets the requirements imposed by the compliant grippers of the industrial robotic arms used in our application (Stäubli RX130 and RX90).

VI. Conclusion

An approach for texture-less object detection and 3D pose estimation in RGB-D images has been presented. It is based on a sliding window approach with a cascade-style evaluation of each window location. Sub-linearity in the number of training templates is achieved by an efficient voting procedure based on hashing which generates a small set of candidate templates for each window location. The templates are verified in several modalities and approximate object poses associated with the matched templates are refined by a stochastic optimization scheme. Experiments on a public dataset have demonstrated that the proposed method detects objects with a recognition rate comparable to the state of the art and achieves sub-centimeter accuracy in localization. The method is therefore well-suited to multiple object detection, a commonly required task in robotic applications.

References

[1] A. Collet, M. Martinez, and S. Srinivasa, "The MOPED framework: Object Recognition and Pose Estimation for Manipulation," *I. J. Robotic Res.*, vol. 30, no. 10, pp. 1284–1306, 2011.

[2] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: A survey," *Found. Trends. Comput. Graph. Vis.*, vol. 3, no. 3, pp. 177–280, July 2008.

[3] F. Tombari, A. Franchi, and L. Di, "BOLD features to detect texture-less objects," in *ICCV*, 2013, pp. 1265–1272.

[4] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009.

[5] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2, 1999, pp. 1150–1157.

[6] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *ICCV*, 2011.

[7] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of texture-less objects," *IEEE PAMI*, 2012.

[8] P. Besl and N. McKay, "A Method for Registration of 3-D Shapes," *PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[9] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *CVPR*, 2010, pp. 998–1005.

[10] C. Choi and H. Christensen, "3D Pose Estimation of Daily Objects Using an RGB-D Camera," in *IROS*, 2012, pp. 3342–3349.

[11] H. Cai, T. Werner, and J. Matas, "Fast detection of multiple textureless 3-D objects," in *ICVS*, ser. LNCS, 2013, vol. 7963, pp. 103–112.

[12] D. Damen, P. Bunnun, A. Calway, and W. Mayol-Cuevas, "Real-time Learning and Detection of 3D Texture-less Objects: A Scalable Approach," in *BMVC*, Sep 2012, pp. 1–12.

[13] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *3DIM*, 2001, pp. 145–152.

[14] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[15] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect," in *BMVC*, 2011, pp. 1–11.

[16] S. Ivekovič, E. Trucco, and Y. Petillot, "Human Body Pose Estimation with Particle Swarm Optimisation," *Evolutionary Computation*, vol. 16, no. 4, pp. 509–528, 2008.

[17] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2189–2202, Nov 2012.

[18] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014, pp. 3286–3293.

[19] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *ACCV*, 2012.

[20] X. Zabulis, M. Lourakis, and P. Koutlemanis, "3D object pose refinement in range images," in *ICVS*, ser. LNCS, 2015, vol. 9163, pp. 263–274.

[21] H. Badino, D. Huber, Y. Park, and T. Kanade, "Fast and accurate computation of surface normals from range images," in *ICRA*, 2011, pp. 3084–3091.

[22] T.-T. Cao, K. Tang, A. Mohamed, and T.-S. Tan, "Parallel Banding Algorithm to Compute Exact Distance Transform with the GPU," in *I3D*, 2010, pp. 83–90.

Appendix B

Publication at ISMARW 2015

An article accepted at 2015 IEEE International Symposium on Mixed and Augmented Reality Workshops (ISMARW 2015).

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.

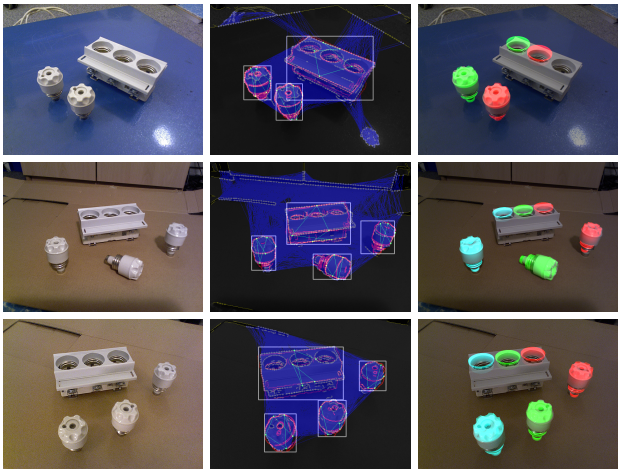
1 Introduction

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-

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.

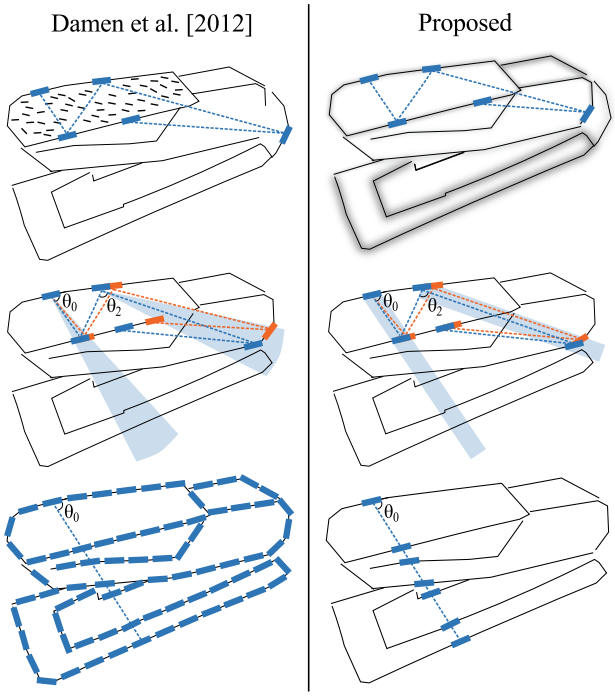


Figure 2: Proposed modifications to [5]. *Top:* The supervised edge detector achieves a higher edge repeatability and has a lower sensitivity to fine textures. *Middle:* Tracing in wedges is replaced by tracing in stripes, yielding less geometrically deviated constellations. *Bottom:* A Hough-based representation 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.

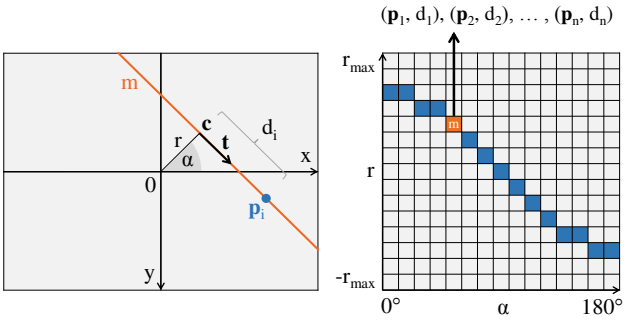


Figure 3: A Hough-based representation for efficient directional search of edgelets. The midpoints of the edgelets from the image space *(left)* correspond to a sinusoid in the Hough space *(right)*. Each bin in the Hough space represents a line m in the image space and stores a list of edgelets whose 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.

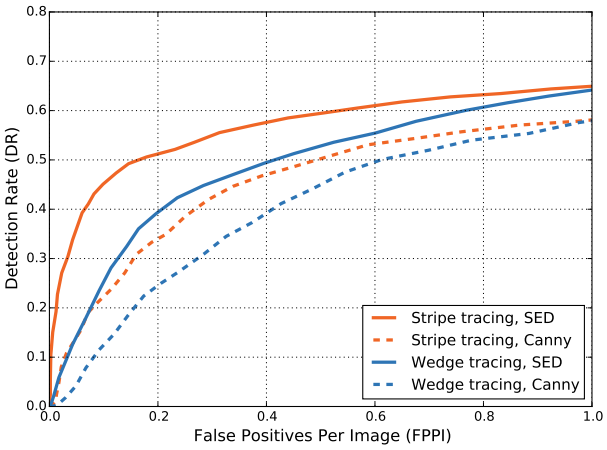


Figure 4: DR/FPPI for different edge detectors (Canny vs. SED) and different tracing sections (wedge vs. stripe). The curves were generated by changing the detection cost threshold.

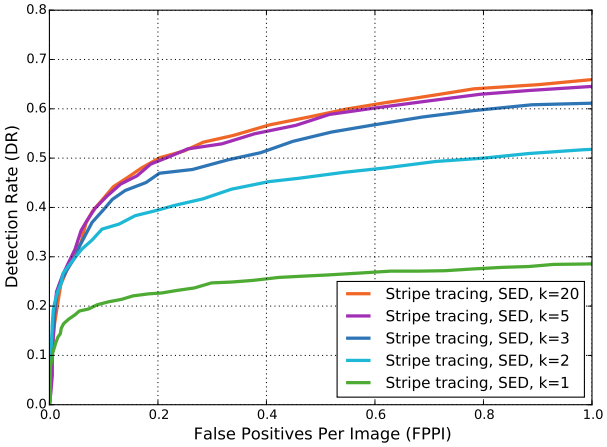


Figure 5: DR/FPPI evaluation of different values of k (the number of closest edgelets considered when tracing a constellation in the detection stage – one of these edgelets was randomly picked and bounced on).

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

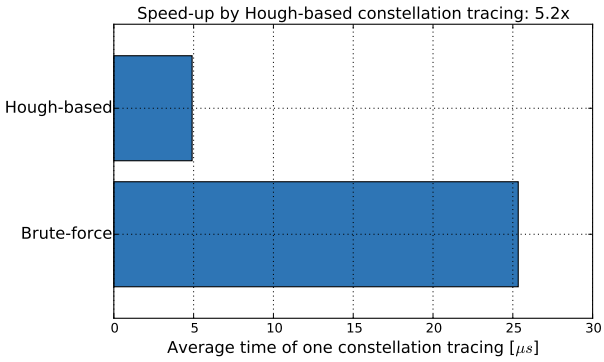


Figure 6: Average time of the proposed Hough-based constellation tracing vs. the brute-force approach used in the original method.

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.

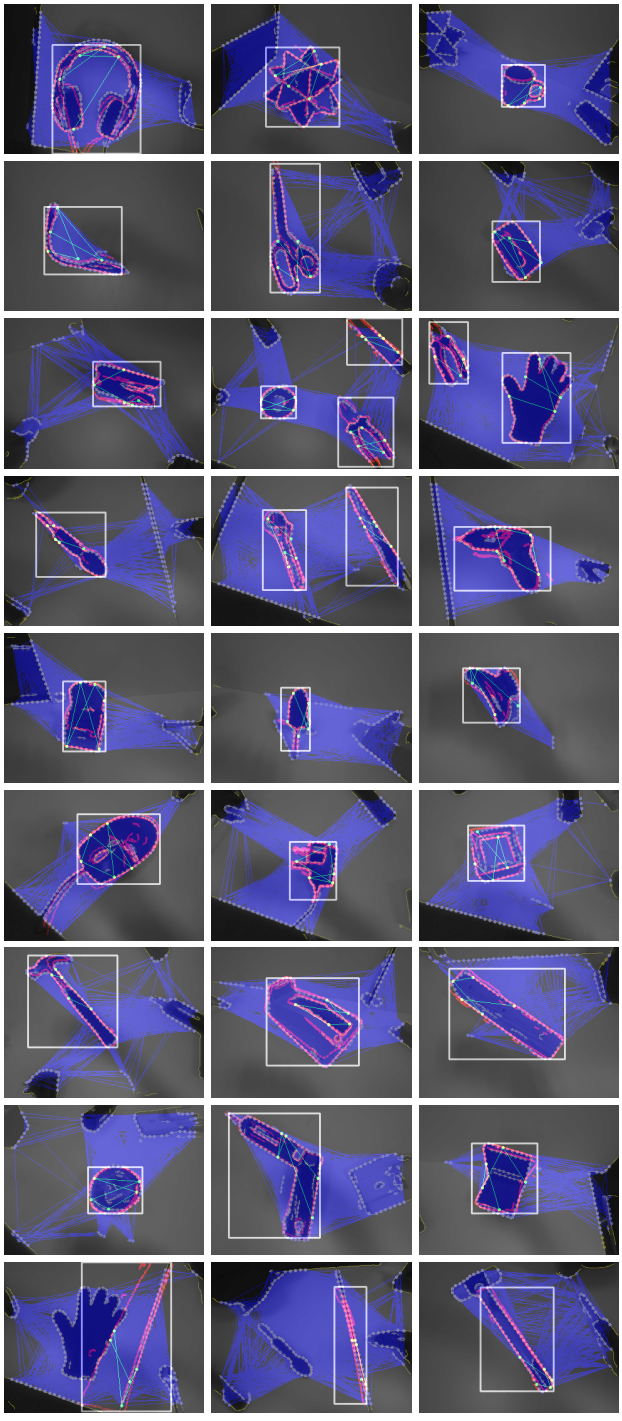


Figure 7: Example detection results in test images from the Bristol Tools dataset. The last row shows typical failure cases caused by the presence of several thin objects in the dataset which tend to match with any parallel lines. Centers of the detected edgelets are visualized by dots, connections of 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.