BOLD features to detect texture-less objects

Federico Tombari
DISI, University of Bologna
federico.tombari@unibo.it

Alessandro Franchi
Datalogic Automation
alessandro.franchi@datalogic.com

Luigi Di Stefano
DISI, University of Bologna
luigi.distefano@unibo.it

Abstract

Object detection in images withstanding significant clutter and occlusion is still a challenging task whenever the object surface is characterized by poor informative content. We propose to tackle this problem by a compact and distinctive representation of groups of neighboring line segments aggregated over limited spatial supports and invariant to rotation, translation and scale changes. Peculiarly, our proposal allows for leveraging on the inherent strengths of descriptor-based approaches, i.e. robustness to occlusion and clutter and scalability with respect to the size of the model library, also when dealing with scarcely textured objects.

1. Introduction

Object detection is among the most widely studied topics in computer vision. Currently, the established paradigm to accomplish detection of textured objects relies on matching *descriptors*, i.e. compact representations of local features such as blobs, corners as well as other types of salient regions extracted from images. The most popular, and arguably most effective, approach within this paradigm is SIFT [17], although also a number of more recent proposals, such as e.g. SURF [2] and ORB [23], provide good performance.

One fundamental requirement for the above techniques to behave effectively is the presence of enough information onto the object surface to anchor feature detection and description. As illustrated in Fig. 1, whenever such information is lacking due to the application requiring detection of *texture-less* objects, state-of-the-art local invariant features exhibit a dramatic performance degradation. However, texture-less objects are ubiquitous, and occur in particular in many vision tasks related to advanced manufacturing, such as e.g. visual inspection for process or quality control and robot guidance. Another emerging scenario wherein the objects of interest are not guaranteed to feature rich textures deals with visual perception for service robotics, where personal robots having to interact with typical household materials are being envisioned and prototyped. Hence, texture-less object detection is relevant to foster deployment of computer vision in both established as well as emerging scenarios.

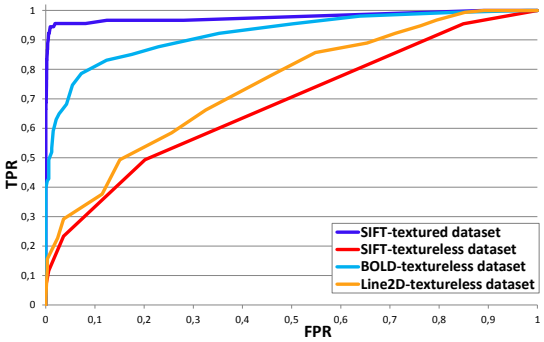


Figure 1: Textured vs. texture-less object detection. SIFT behaves nicely on textured objects but performance drops dramatically when the objects sought for lack enough texture details onto their surface. Our proposal (referred to as BOLD) can advance the state-of-the-art in texture-less object detection (compare BOLD to LINE-2D [11]).

Given the aforementioned limitations of descriptor-based object detectors, state-of-the-art proposals tackle the texture-less object detection problem by means of edge-based template matching [11,12,24,25]. One major merit of edge-based template matching is the ability to detect seamlessly both textured as well as texture-less objects. It suffers from other limitations though, in particular related to the ability to withstand significant occlusion and clutter as well as to the scalability with respect to the size of the model library. As for the former, it is inherent to the approach that to tolerate a high degree of occlusion just a small fraction of matching edges has to be accepted to trigger a detection, which however in cluttered scenes often does not result in a cue enough peculiar to avoid a large number of false detections. Concerning the latter, although efficient search schemes as well as careful hardware-related optimization have been devised to help speeding-up the process [11, 12, 24], it is, alike, somehow inherent to the ap-

proach that a set of views (as large as mandated by the required degree of pose invariance) of each sought object needs to be matched to the current image. Hence, search time grows linearly with the size of the model library. This means that, especially when a relatively large pose space has to be explored, often just a few models can be handled in practice through edge-based template matching.

The above weak points are dealt with effectively by descriptor-based object detectors. Indeed, a large model library is searched efficiently by storing all descriptors belonging to sought objects within a fast indexing structure (e.g. a k-d tree or randomized forest [18]) which enables fast lookup in logarithmic rather than linear time. Descriptor-based methods are also very robust to occlusions because, due to their high distinctiveness, just a few matching features can provide enough evidence to judge reliably upon the presence of an object even in heavily cluttered scenes.

Based on the above considerations, we have investigated on whether and how the inherent benefits of descriptor-based methods may be leveraged to detect also texture-less objects. Accordingly, in this paper we propose novel features that can be injected seamlessly into a standard SIFT-like object detection pipeline so as to provide notable performance improvements with respect to state-of-the-art edge-based template matching (see again Fig. 1). Purposely, we exploit groups of neighboring line segments to build up a representation of object parts which we term Bunch Of Lines Descriptor (BOLD). The cues deployed in our descriptor are peculiarly encoded into a compact two-dimensional histogram and include relative orientations and displacements between pairs of segments as well as contrast polarity.

2. Related work

The state-of-the-art in edge-based template matching for texture-less object detection is likely represented by LINE [11], which has been proposed both for 2D (LINE-2D) as well as RGB-D images (LINE-MOD). The former relies on image gradients only, the latter deploys surface normals too. Key to the method is a robust encoding of gradient information together with a careful hardware-aware optimization which delivers fast matching time. Thus, 3D object detection can be achieved by matching in real-time thousands of templates gathered during the training stage by looking at the object from different vantage points and distances. As demonstrated in [9], though, the method can be harmed by partial occlusions. Another recent relevant template matching approach for texture-less object detection is proposed in [25], which however, unlike BOLD, requires full-3D object models to carry out the training stage.

As for previous works related to description and matching of edges and contour information, we report here a brief overview of those more closely related to our proposal. One of the first methods to describe object contours is the "cubist" approach by Nelson and Selinger [19], whereby the object representation is simplified by means of a loosely structured combination of local context regions keyed by distinctive boundary fragments called *Key Curves*. Unlike BOLD, these fragments are described by simple features such as compactness and curvature in order to efficiently index the model database. Then, Belongie et al. [4] proposed *Shape Context*, a log-polar histogram of the relative coordinates of uniformly sampled Canny edges. Being a global descriptor, this method is not designed to withstand occlusion and clutter. Similarly to Shape Context, Carmicheal and Hebert [5] propose to describe edge densities computed on a 2D image grid, these descriptors being then used to train a cascade of classifiers.

Ferrari et al. [10] introduced a new family of scale-invariant local shape features aimed at object categorization which are based on chains of k-connected, roughly straight contour segments called k-Adjacent Segment (kAS). Each kAS is described as a signature including distances between segment pairs, segment absolute orientations and lengths. Kim et al. [15] proposed to learn feature correspondences by training a classifier on descriptors that include a high number of geometric and color traits between pairs of edge lines such as length, absolute orientation and intensity/color values along the line. Damen et al. [6] match sequences of short line segments called constellation of edgelets, i.e. a sequence of angles that defines the direction of the tracing vectors that connect a subset of object edges. Constellation descriptors encode the relative orientations and distances between consecutive edgelets. As it will be illustrated in next Section, [6, 10, 15] deploy different geometric features with respect to those encoded by BOLD.

David and DeMenthon [7] generate a pose hypothesis for each model-scene pair of extracted line segments. These poses are then ranked by the average distance between the 10-Nearest Neighbor segments on the model transformed according to the current pose hypothesis and the respective scene segments. This method does not include any feature descriptor proposal as it relies on geometric verification only. Finally, related approaches that address the feature detection stage only are [1, 14], which then rely on, respectively, Shape Context-like [14] and SIFT-like [1] descriptors.

3. BOLD features

As discussed in previous Section, several approaches aimed at texture-less object detection or recognition rely on edges and segments, mainly extracted from objects' contours, as the basic trait underpinning the semantic perception process. Edges and segments are also the starting point of our method. In particular, we propose a descriptor for

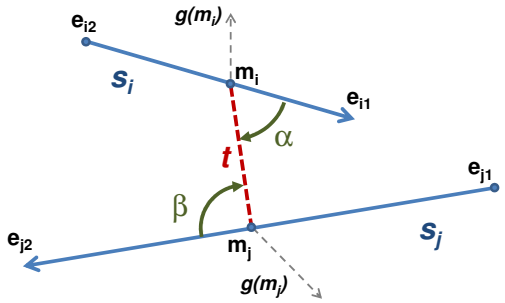


Figure 2: The geometrical primitives deployed by BOLD are the relative orientations (represented by angles α and β in figure) between pairs of oriented line segments.

line segments, which can be extracted by means of a variety of approaches such as either polygonal approximation of the output of an edge detector [8, 22] or a specific line detection algorithm [16, 21, 26]. Additionally, further pruning may be enforced to improve repeatability of extracted line segments, e.g. to discard short segments possibly due to noise. Assuming a set of repeatable line segments, S, has been extracted from the image, for each segment we compute a BOLD descriptor, which aggregates together geometrical cues related to neighboring segments.

3.1. Geometric primitives

The BOLD descriptor aggregates together geometric primitives computed over pairs of neighboring segments. These primitives should yield invariance to rotation, translation and scale, and at the same time be robust to noise and efficient to compute. As also depicted in Figure 2, let us denote vectors in boldface and consider a segment pair $\mathbf{s_i}$, $\mathbf{s_j} \in S$, with $\mathbf{m_i}$, $\mathbf{m_j}$ representing their respective midpoints. Likewise, we denote as $\mathbf{e_{i1}}$,$\mathbf{e_{i2}}$ the two endpoints of $\mathbf{s_i}$, and as $\mathbf{e_{j1}}$,$\mathbf{e_{j2}}$ those of $\mathbf{s_j}$. We then refer to the segment connecting $\mathbf{m_i}$ and $\mathbf{m_j}$ as to $\mathbf{t}$, the *midpoint segment*. In particular, we define two different midpoint segments according to the two possible signs:

$$\mathbf{t_{ij}} = \mathbf{m_j} - \mathbf{m_i} \tag{1}$$

$$\mathbf{t_{ji}} = \mathbf{m_i} - \mathbf{m_j} \tag{2}$$

We carefully investigated and tested a number of pairwise geometric primitives, including those proposed in previous literature, such as relative segment length, distance between segments, absolute and relative segment orientations [6, 10, 15]. Based on this analysis, we sifted out the primitive that provides the best trade-off between descriptiveness and robustness, as outlined in the following.

First of all, to define our primitive each line segment has to be associated with a canonical orientation. Given the direction of the segment, we propose to leverage on the intensity gradient at the midpoint, $\mathbf{g}(\mathbf{m_i})$, to determine the sign. Specifically, we define a canonically oriented line segment $\mathbf{s_i}$ as follows:

$$sign\left(\mathbf{s_i}\right) = \frac{\left(\mathbf{e_{i2}} - \mathbf{e_{i1}}\right) \times \mathbf{g}\left(\mathbf{m_i}\right)}{\parallel \left(\mathbf{e_{i2}} - \mathbf{e_{i1}}\right) \times \mathbf{g}\left(\mathbf{m_i}\right) \parallel} \bullet \mathbf{n} \tag{3}$$

$$\mathbf{s_i} = sign\left(\mathbf{s_i}\right) \cdot \left(\mathbf{e_{i2}} - \mathbf{e_{i1}}\right) \tag{4}$$

where $\times$ is the cross product, $\bullet$ the dot product and $\mathbf{n}$ the unit vector normal to the image plane pointing towards the observer. Hence, (3) yields ± 1 depending on the cross product between $\mathbf{e_{i2}} - \mathbf{e_{i1}}$ and the gradient at the midpoint having or not the same sign as the normal pointing outward from the image plane, which then determines whether the endpoints must be actually swapped or not to get $\mathbf{s_i}$. It is worth noting here that, as segments extracted from the image typically lay close to intensity contours, the gradient magnitude at the midpoint is usually as high as to guarantee a repeatable and robust contrast polarity, which indeed renders the canonical orientation assigned to segments through (3) and (4) likewise stable and robust.

Based on the previous definition, the proposed geometric primitive consists in the two angles shown in Figure 2, which can be uniquely associated to a pair of oriented segments: α measures the clockwise rotation which aligns $\mathbf{s_i}$ to $\mathbf{t_{ij}}$, β the clockwise rotation to align $\mathbf{s_j}$ to $\mathbf{t_{ji}}$. To obtain such angles, we start from the computation of the smaller angle between two vectors:

$$\alpha^* = arccos\left(\frac{\mathbf{s_i} \bullet \mathbf{t_{ij}}}{\parallel \mathbf{s_i} \parallel \cdot \parallel \mathbf{t_{ij}} \parallel}\right) \tag{5}$$

$$\beta^* = arccos\left(\frac{\mathbf{s_j} \bullet \mathbf{t_{ji}}}{\parallel \mathbf{s_j} \parallel \cdot \parallel \mathbf{t_{ji}} \parallel}\right) \tag{6}$$

which yields measurements within the range $[0, \pi]$. Then, we apply a further disambiguation step to pick either the smaller or larger angle between the vector pair

$$\alpha = \begin{cases} \alpha^*, & \frac{\mathbf{s_i} \times \mathbf{t_{ij}}}{\|\mathbf{s_i} \times \mathbf{t_{ij}}\|} \bullet \mathbf{n} = 1 \\ 2\pi - \alpha^* & otherwise \end{cases} \tag{7}$$

$$\beta = \begin{cases} \beta^*, & \frac{\mathbf{s_j} \times \mathbf{t_{ji}}}{\|\mathbf{s_j} \times \mathbf{t_{ji}}\|} \bullet \mathbf{n} = 1 \\ 2\pi - \beta^* & otherwise \end{cases} \tag{8}$$

and hence provides measurements within the entire $[0, 2\pi]$ angular range.

The disambiguation step given by equations (7),(8) allows for distinguishing among local configuration that otherwise would have been considered as equivalent, e.g. as in the example in Fig. 4 which illustrates how the disambiguated angles can detect unlikely transformations such as simultaneous mirroring and contrast polarity inversion. Usually, higher distinctiveness comes to a price in terms of robustness: we will show later in this Section that the chosen angles (α, β) are consistently more effective than (α^*,

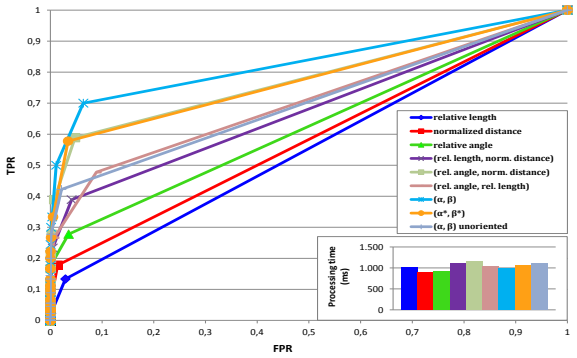


Figure 3: Comparison of pairwise geometric primitives. Recognition times include evaluation over 80 models.

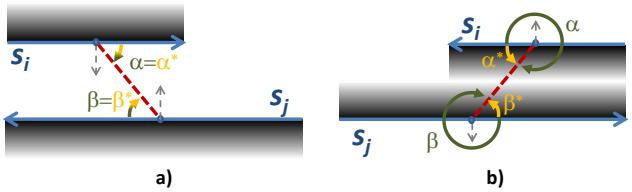


Figure 4: The disambiguated angles defined according to (7),(8) highlight potentially invalid transformations such as simultaneous mirroring and contrast polarity inversion: in (b) α^*, β^* take the same values as in (a), whilst α and β take different values.

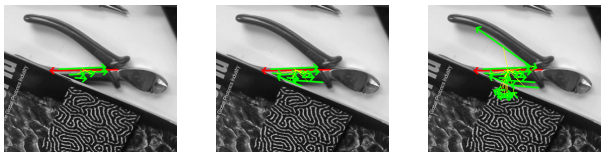
β^*). It is also important to point out that (α, β) depend not only on the relative orientation between the two segments but also on their relative spatial displacement. Overall, they thus represent a compact geometric primitive encoding relative orientation and position as well as, due to segments being oriented, contrast polarity. To the best of our knowledge, the proposed geometric primitive has not been deployed by any previous work. The most similar approach can be found in [15], which, among other features, defines a relative segment orientation based on the midpoint segment but without relying on establishment of a canonical orientation for each segment, which we found hindering notably the repeatability of angle measurements.

As already mentioned, we carried out an in-depth experimental analysis to help devise the most effective geometrical primitives to be deployed within BOLD. An excerpt from the results is shown in Figure 3, where we compare (α, β) with other commonly deployed primitives [6, 10, 15] such as relative orientation between segments, normalized length and normalized midpoint distance. In this experiment, all primitives are accumulated into histograms, which is the way pairwise geometrical primitives are aggregated in BOLD (see Section 3.2). As (α, β) yield 2D histograms while the other considered primitives 1D histograms, we also compare our proposal with 2D histograms built by using jointly multiple primitives. As anticipated, we also evaluated using the smaller angles between vectors (α^*, β^*), as well as measurement of such angles without canonically orienting segments, which results in taking always the smallest possible angle between vectors (referred to here as *(α, β) unoriented*). By building histograms out of different primitives we attain different descriptors that can be plugged seamlessly into the object detection pipeline described in Section 4 and thereby evaluated comparatively as depicted in Figure 3. Results show the overall superiority of angle-based primitives with respect to distances or lengths. We ascribe this mainly to the former turning out more robust with respect to the potential fragility of the segment extraction stage. The Figure also demonstrates the effectiveness of relying on canonically oriented segments (α^*, β^* vs. α, β *unoriented*) as well as the neat performance improvement brought in by the proposed angle disambiguation step (α, β vs. α^*, β^*). As for computational efficiency, all the considered primitives appear approximately equivalent in terms of their impact on overall detection time.

3.2. Aggregation of geometric primitives

For each line segment, s_i, the BOLD descriptor is built by aggregating (α, β) primitives computed for the set of neighboring segments (referred to as *bunch*) given by the k nearest neighbors (kNN) segments of s_i, k being a parameter of the method. The kNN approach represents an effective way to define an adaptive support for the descriptor, thus rendering the approach inherently scale invariant. Moreover, the kNN search over the 2D domain can be carried out efficiently by means of indexing techniques [3]. Purposely, a distance between line segments has to be defined. In our proposal we simply compute the distance between midpoints, although other approaches, such as sampling uniformly along segments and computing the closest distance between sampled points [7], may be deployed.

Successively, for each pair formed by s_i and one of the k segments in its bunch, the geometric primitives (α, β) are computed and aggregated together. We have investigated on two main aggregation approaches. According to the former, a signature of the primitives is computed by ordering the neighboring segments of a bunch based on the distance to the central segment, then building the descriptor as the ordered chain of primitives associated to each segment. As for the latter, the angles are accumulated into a 2D joint histogram, with the domain of both dimensions (i.e. the angular range $[0, 2\pi]$) discretized according to a given quantization step θ (a parameter of the descriptor). The histogram approach turned out to notably outperform the signature approach, due to the higher robustness with regards to clutter and occlusion, as in a signature a single segment missing from the bunch tends to disrupt description. Moreover, thanks to quantization, the histogram-based descriptor in-



(a) k=5 (b) k=10 (c) k=20

Figure 5: Bunches computed for different values of k: a higher value leads to more descriptiveness, but tends to include clutter.

herently provides good robustness to inaccuracies in segment localization. In our experiment we set the number of bins for each histogram dimension $b = \frac{2\pi}{\theta} = 12$. On the other hand, like in most histogram-based descriptors, quantization effects may decrease distinctiveness of BOLDs. To counterattack this, we apply bilinear interpolation by assigning to each entry of the histogram - and to its closest bins - a weight that depends on the distance of the measurement from the center of the bin. Finally, BOLD descriptors are normalized by their L_2 norm, so as to get vectors laying onto the unit sphere. This is beneficial when using matching measures derived from the L_2 norm to obtain upper bounded values of the distance between descriptors.

3.3. Deploying multiple bunches

The number of neighboring segments, k, is a key parameter of the BOLD descriptor. A high number of segments tends to increase distinctiveness of BOLDs, since there are lower ambiguities due to similar bunches arising from non corresponding object parts. On the other hand, a high value of k tends to include, within the same bunch, neighboring segments that may belong to clutter, this leading to somewhat corrupted histograms (see the example in Figure 5). Accumulating primitives over histograms helps increasing the robustness up to a certain extent, i.e. until the number of clutter elements does not exceed that of object elements. Moreover, a good choice for k depends also on the type of objects to be detected: simple shapes made out of a few segments call for a small k, so as not to incorporate clutter, whereas for more complex objects a higher k is usually beneficial. As such, the choice of k is critical.

Instead of trying to tune this parameter based on specific scenarios, we propose to simultaneously deploy multiple k values to describe each line segment s_i. This allows for seamlessly and effectively encoding of both simple shapes and local parts as well as larger scale structures. Indeed, we have found out that this approach not only avoids the user to have to choose a critical parameter, but also improves performance significantly. Figure 6 reports object detection results attained by a single bunch approach with different k as well as by deploying multiple bunches: the best

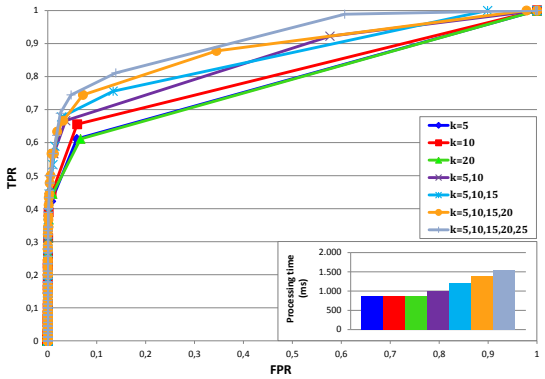


Figure 6: BOLD descriptors employing single vs. multiple bunches with different k values.

single-bunch configuration turns out $k = 10$, but remarkably improved performance can be attained by using multiple bunches altogether, this without slowing down too notably the overall process. Hence, in the experimental evaluation we will use the multi-bunch approach with k set to 5, 10, 15, 20.

4. Object detection pipeline

In this section we describe our object detection approach, which deploys BOLD within a standard SIFT-like pipeline [17] where the detection and description stages are modified to deal with texture-less objects. Object contours can change notably at different scales, and sometimes edges can completely disappear if either the object is blurred or a significant scale variation does occur. For this reason, the first step of our pipeline is represented by multi-scale extraction of line segments. In particular, we build a scale space by rescaling the input image at different resolutions, then extract line segments at each level of the pyramid. The scale of each segment is retained so that, in the next step, the BOLD descriptor for each segment is computed taking into account only the neighbors found at the same scale. This counteracts the issue of missing segments due to large scale variations.

Successively, we rely on the Euclidean distance and the FLANN Randomized Kd-tree Forest [18] to match BOLD descriptors extracted from the input image to those gathered at training time from the objects belonging to the model library. Although we have evaluated matching measures specifically conceived for histogram data, such as the Histogram Intersection, we have found that the Euclidean distance yields good results without sacrificing efficiency. Similarly to [17], feature correspondences are then validated through a Generalized Hough Transform and the final pose is computed through a Least-Square Estimation of the required transformation (e.g. a similarity or homography).

(a) *D-Textureless*: 3 models (left) and 1 scene (right)

(b) *Caltech Covers*: 3 models (left) and 1 scene (right)

(c) *CMU-KO8*: 2 models (top) and 2 scenes

Figure 7: Examples of models and scenes from the datasets used in the experimental evaluation.

5. Experimental evaluation

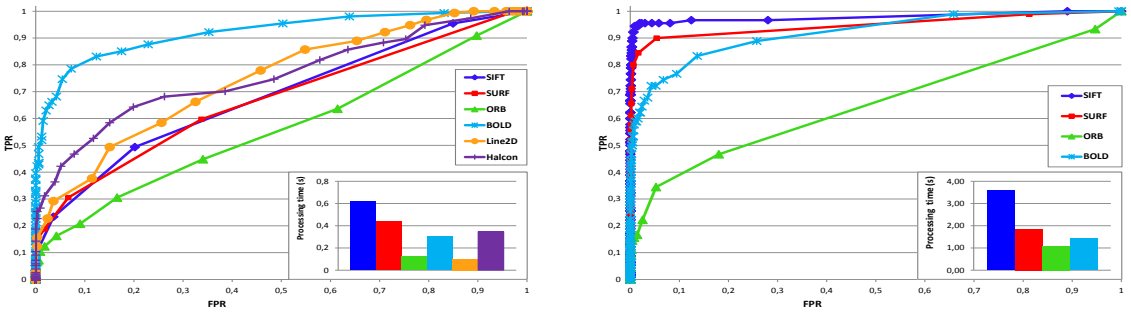
We compare here the BOLD pipeline for texture-less object detection to two prominent edge-based template matching-based approaches, *i.e.* LINE-2D [11] and the shape-based matching tool available in the HALCON library by MVTec[1]. Moreover, we include in our comparison descriptor-based methods for textured object detection such as SIFT [17], SURF [2] and ORB [23]. To extract the line segments needed to compute BOLD we use the LSD algorithm [26]. Although BOLD is in principle independent of a specific line segment detector, we found that LSD provides enough repeatability to enable effective object detection. In particular, we found that performance using LSD turns out significantly superior to polygonal approximation of Canny edges.

The implementations of LINE-2D, SURF and ORB are taken from OpenCV [2], while for SIFT we rely on Rob Hess's implementation[3]. As for SIFT and SURF we simply plug their specific detection/description stages into the reference object detection pipeline described in Section 4, while for the ORB pipeline we employed LSH in the matching stage as suggested in [23]. As for HALCON, we have used the `find_scaled_shape_model` function included in the free demo version of the library. All methods were run with their default parameters, except for HALCON for which we carried out a specific parameter tuning on a similar -but distinct- dataset with respect to that used for testing. Experiments have been executed on an Intel Core2 Quad 2.5 Ghz CPU with 4 GBs of RAM. All algorithms have been compiled on a 64-bit environment. We wish to point out that, unlike HALCON, SURF and LINE-2D, the BOLD implementation used in the experiments is not optimized to take advantage of multi-core architectures or SIMD instructions (e.g. SSE2), though our method may in principle be parallelized easily.

Given the scarceness of public datasets for texture-less object detection withstanding clutter and occlusions, we have acquired our own, referred to as *D-Textureless*. This dataset has been acquired with a webcam, comes with hand-labeled ground-truth and includes 9 texture-less models and 55 scenes with clutter and occlusions. A fairly large pose space has to be explored by the algorithms due to models appearing rotated, translated and scaled in the scenes. All 9 models are searched in each scene, which in turn may include one or more models, but one instance of each at most. To complement our comparison, we evaluate BOLD also on a textured dataset built from publicly available images and referred to as *Caltech Covers*. Specifically, this dataset includes 80 models randomly chosen from the Caltech Game Covers dataset [4] and 50 scenes, which we built synthetically by randomly rotating, translating and scaling a pre-defined number of models (from 1 to 3), together with additional covers not included in the model database so as to create clutter as well as occlusions up to 90%. Again, in each scene we look for all 80 models. *D-Textureless* and *Caltech Covers* are referred to, respectively, as texture-less and textured dataset in Fig. 1, while *Caltech Covers* has been used also in the experiments reported in Figures 3, 6. Examples of models and scenes from the two datasets are shown in Fig. 7a and Fig. 7b.

Fig. 8 reports the ROC curves yielded by the considered algorithms on the two datasets. Focusing on the texture-less objects (Fig. 8a), it can be seen that BOLD neatly outperforms all methods. Moreover, and as expected, template-matching methods such as HALCON and LINE-2D perform much better than existing descriptor-based methods like SIFT, SURF and ORB. Despite the absence of either machine level optimizations or multi-threading, BOLD turns out faster than HALCON, due to the relatively large number of sought objects, although slower than LINE-2D. In the experiments with *Caltech Covers*, we did not include LINE-2D and HALCON. Indeed, the former requires the algorithm to be trained carefully from nearly all the possible vantage points that may occur in the actual scene, which

[1] `www.mvtec.com/halcon`
[2] `www.opencv.org`
[3] `robwhess.github.com/opensift`
[4] `vision.caltech.edu/malaa/datasets/caltech-games`



(a) *D-Textureless* (b) *Caltech Covers*

Figure 8: Comparison on texture-less dataset *D-Textureless* (left) and textured dataset *Caltech Covers* (right).

is not feasible when the model database contains as many as 80 objects; the latter can be trained by a single image per model, but turns out excessively slow with such a large model database (see also Figure 10). As shown in Fig. 8b, SIFT is clearly the best performer when dealing with textured objects, neatly surpassing SURF and then BOLD. As for efficiency, in case of a relatively large model database, BOLD is faster than SIFT and SURF and slower only than ORB, which nevertheless seems not as effective with the *Caltech Covers* dataset.

We also address detection of texture-less 3D objects under arbitrary viewpoint on the challenging *CMU Kitchen Occlusion* dataset (CMU-KO8), recently introduced by Hsiao and Hebert [13] to assess their occlusion reasoning model based on the computation of the statistics of object dimensions in a given environment. The authors incorporate their model into LINE-2D according to three different variants and show improved performance on their dataset, which consists of 8 common household texture-less objects sought in 800 *single view* and 800 *multi view* cluttered scenes with various levels of occlusion (see Fig. 7c). In *single view* experiments the object is seen in the scene from the same vantage point as in the -single- training image, while *multi view* experiments focus on variations of the elevation angle, the training set comprising 25 views of each object. According to [13], in Figure 9 we provide the results attained by the BOLD object detection pipeline in terms of recall (i.e. detection rate) versus false positives per image (fppi) curves averaged across each of the two experiments. As for general methods conceived to operate without any prior knowledge on the working environment, from Figure 9 we can observe that BOLD neatly outperform LINE-2D in both experiments. Then, Figure 9 confirms the benefits brought in by deployment of environment-specific statistics on object sizes, as the variants of LINE-2D proposed in [13] overall compare favorably with respect to a state-of-the-art general purpose approach such as BOLD. Interestingly, in the conservative (*i.e.* low fppi) portion of the curve of the

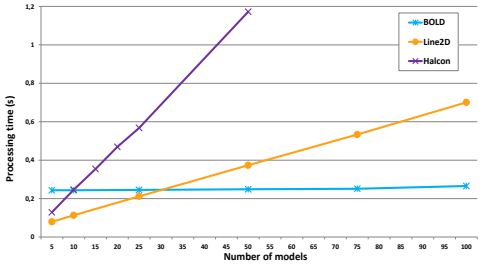


Figure 10: Scalability with respect to the number of models.

single view experiment BOLD delivers a higher detection rate than the methods proposed in [13]. Figure 9 shows also two failure cases and one successful detection enabled by quite impressive matches dealing with segments on the mug handle occluded by a semi-transparent plastic bag.

Finally, to analyze the scalability of the considered algorithms, in Figure 10 we report the measured execution times versus the number of sought models for the *D-Textureless* dataset[5]. As expected, template matching methods scale linearly with the number of models, with HALCON showing a much steeper increase of computation time than LINE-2D. On the other hand, BOLD provides a nearly constant detection time up to as many as 100 models.

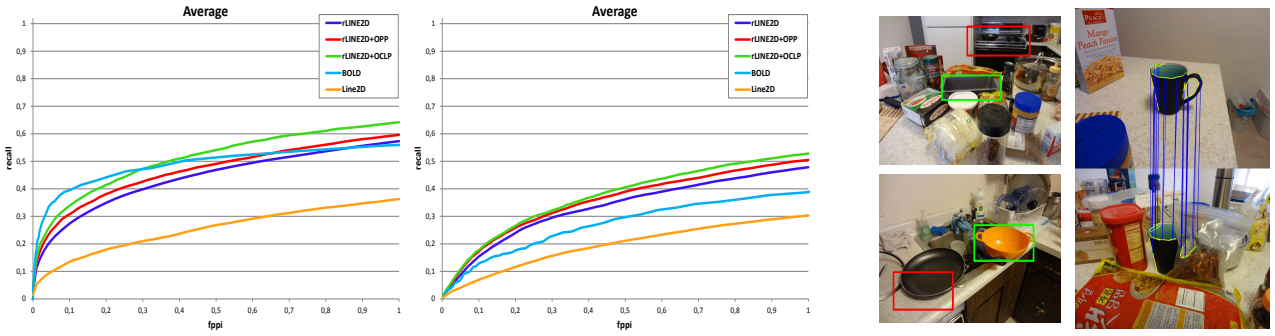
Additional qualitative results related to CMU-KO8 as well as the *D-Textureless* dataset can be found on BOLD's project page [6].

6. Concluding remarks

BOLD features allows for leveraging on a fairly standard descriptor-based pipeline to detect effectively also texture-less objects, thereby achieving state-of-the-art robustness to clutter and occlusion and unprecedented scalability with respect to the size of the model database. The main limitation

[5] As the dataset comprises only 9 different models, we simply replicated them as needed to run this experiment.

[6] http://vision.deis.unibo.it/BOLD



(a) *Single view* (b) *Multi view* (c) *BOLD: 2 failure cases and an example of feature matching*

Figure 9: CMU-KO8: quantitative (BOLD, LINE-2D and the three methods proposed in [13]) and qualitative (BOLD) results

of our proposal deals with detection of highly curvilinear (e.g. round) or simple (i.e. made out of a few lines) objects in scenes with heavy occlusion and clutter. Such objects show just a few repeatable BOLDs: if some get corrupted due to occlusion or clutter, then the object may hardly be detected. To enlarge the set of shapes effectively dealt with by our proposal, we plan to include description of oriented elliptical arcs [20].

References

[1] M. Awais and K. Mikolajczyk. Feature pairs connected by lines for object recognition. In *Proc. ICPR*, 2010.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. ECCV*, pages 404–417, 2006.

[3] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. CVPR*, pages 1000–1006, 1997.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[5] O. Carmichael and M. Hebert. Shape-based recognition of wiry objects. *PAMI*, 26(12):1537–1552, 2004.

[6] D. Damen, A. Gee, A. Calway, and W. Mayol-Cuevas. Detecting and localising multiple 3d objects: A fast and scalable approach. In *IROS ASP-AVS Workshop*, 2011.

[7] P. David and D. DeMenthon. Object recognition in high clutter images using line features. In *Proc. ICCV*, 2005.

[8] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: Int. J. for Geographic Inf. and Geovis.*, 10(2):112–122, 1973.

[9] B. Drost and S. Ilic. 3d object detection and localization using multimodal point pair features. In *3DIMPVT*, 2012.

[10] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV*, 87(3):284–303, 2010.

[11] 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. *PAMI*, 2012.

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

[13] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*, 2012.

[14] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proc. CVPR*, 2004.

[15] G. Kim, M. Hebert, and S.-K. Park. Preliminary development of a line feature-based object recognition system for textureless indoor objects. In *Proc. ICAR*, volume 370, pages 255–268, 2007.

[16] P. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available at: `www.csse.uwa.edu.au/~pk/Research/MatlabFns/`, 2000.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. VISAPP*, pages 331–340, 2009.

[19] R. Nelson and A. Selinger. A cubist approach to object recognition. In *Proc. ICCV*, pages 614–621, 1998.

[20] V. Patraucean, P. Gurdjos, and R. Grompone Von Gioi. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In *Proc. ECCV*, 2012.

[21] A. Pope and D. Lowe. Vista: A software environment for computer vision research. In *Proc. CVPR*, 1994.

[22] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Proc. ICCV*, pages 2564–2571, 2011.

[24] C. Steger. Occlusion, clutter, and illumination invariant object recognition. In *Proc. ISPRS*, 2002.

[25] M. Ulrich, C. Wiedemann, and C. Steger. Combining scale-space and similarity-based aspect graphs for fast 3d object recognition. *PAMI*, 34(10):1902–1914, 2012.

[26] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *PAMI*, 32(4):722–732, 2010.