Shape-based instance detection under arbitrary viewpoint

Edward Hsiao and Martial Hebert

Abstract Shape-based instance detection under arbitrary viewpoint is a very challenging problem. Current approaches for handling viewpoint variation can be divided into two main categories: *invariant* and *non-invariant*. Invariant approaches explicitly represent the structural relationships of high-level, view-invariant shape primitives. Non-invariant approaches, on the other hand, create a template for each viewpoint of the object, and can operate directly on low-level features. We summarize the main advantages and disadvantages of invariant and non-invariant approaches, and conclude that non-invariant approaches are well-suited for capturing fine-grained details needed for specific object recognition while also being computationally efficient. Finally, we discuss approaches that are needed to address ambiguities introduced by recognizing shape under arbitrary viewpoint.

1 Introduction

Object instance detection under arbitrary viewpoint is a fundamental problem in Computer Vision and has many applications ranging from robotics to image search and augmented reality. Given an image, the goal is to detect a specific object in a cluttered scene from an unknown viewpoint. Without prior information, an object can appear under an infinite number of viewpoints, giving rise to an infinite number of image projections. While the use of discriminative point-based features, such as SIFT [21, 28], has been shown to work well for recognizing texture-rich objects across many views, these methods fail when presented with objects that have little to no texture.

Edward Hsiao
Robotics Institute, Carnegie Mellon University, e-mail: ehsiao@cs.cmu.edu

Martial Hebert
Robotics Institute, Carnegie Mellon University e-mail: hebert@cs.cmu.edu

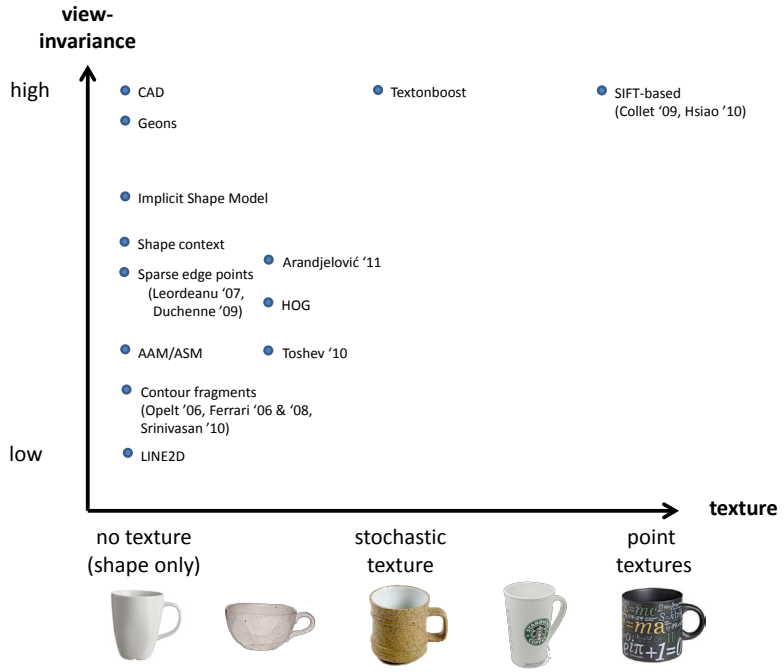


Fig. 1: View-invariance vs. texture for current state-of-the-art methods.

Objects range from being completely uniform in color, to having stochastic textures from materials, to repeatable point textures found on man-made items (i.e., soup cans). In the following, texture-rich objects refer to those where discriminative, point-based features (e.g., SIFT) can be extracted repeatably. Weakly-textured objects, on the other hand, refer to those that contain stochastic textures and/or small amounts of point textures, but which are insufficient for recognizing the object by themselves. Examples of objects of different texture types can be seen in Figure 1.

Weakly-textured objects are primarily defined by their contour structure and approaches for recognizing them largely focus on matching their shape [12, 18, 24, 41]. Many object shapes, however, are very simple, comprised of only a small number of curves and junctions. Even when considering a single viewpoint, these curves and junctions are often locally ambiguous as they can be observed on many different objects. The collection of curves and junctions in a global configuration defines the shape and is what makes it more discriminative.

Introducing viewpoint further compounds shape ambiguity as the additional curve variations can match more background clutter. Much research has gone into representing shape variation across viewpoint. Figure 1 shows a rough layout of current state-of-the-art methods with respect to the type of texture they are designed to recognize versus how much view-invariance they can handle. Current models can roughly be divided into two main paradigms: *invariant* and *non-invariant*.

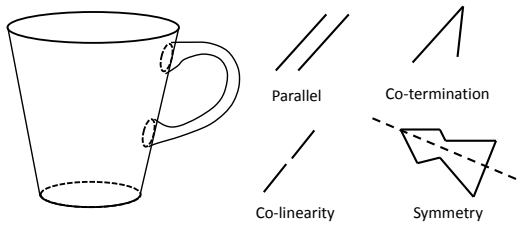


Fig. 2: Invariant methods consider properties of shape primitives that are invariant across viewpoint. Common invariant properties that are used are parallelism, co-termination, co-linearity and symmetry.

Invariant models create a unified object representation across viewpoint by explicitly modeling the structural relationships of high level shape primitives (e.g., curves and lines). *Non-invariant* models, on the other hand, use view-based templates and capture viewpoint variations by sampling the view space and matching each template independently. In this article, we discuss the advantages and disadvantages of invariant and non-invariant methods. We conclude that non-invariant approaches are well-suited for capturing viewpoint variation for specific object recognition since they preserve the fine-grained details. We follow with a discussion on additional techniques that are necessary for addressing shape ambiguities under arbitrary viewpoint.

2 Invariant methods

Invariant methods are based on representing structural relationships between view-invariant shape primitives [4, 17]. Typically, these methods represent an object in 3D and reduce the problem of object detection to generating correspondences between a 2D image and a 3D model. To facilitate generating these correspondences, significant work has gone into designing shape primitives [3] that can be differentiated and detected solely from their perceptual properties in 2D while being relatively independent of viewing direction. Research in perceptual organization [29] and non-accidental properties (NAPs) [44] have demonstrated that certain properties of edges in 2D are invariant across viewpoint and unlikely to be produced by accidental alignments of viewpoint and image features. These properties provide a way to group edges into shape primitives and are used to distinguish them from each other and from the background. Example of such properties are collinearity, symmetry, parallelism and co-termination as illustrated in Figure 2. After generating candidate correspondences between 2D image and 3D model using these properties, the position and pose of the object can then be simultaneously computed.

In earlier approaches, 3D CAD models [13, 24, 46] were extensively studied for view-invariant object recognition. For simple, polyhedral objects, CAD models

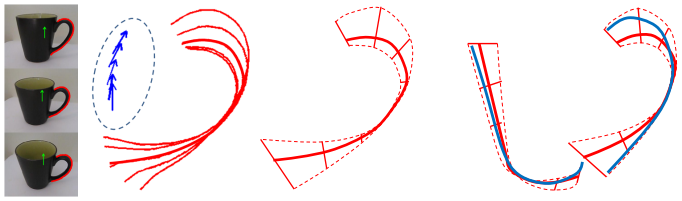
consist of lines. However for complex, non-polyhedral objects, curves, surfaces and volumetric models [25] are used. In general, obtaining a compact representation of arbitrary 3D surfaces for recognition is very challenging. Biederman's Recognition-by-Components (RBC) [3] method decomposes objects into simple geometric primitives (e.g., blocks and cylinders) called *geons*. By using geons, structural relationships based on NAPs can be formulated for view-invariant detection.

Given geometric constraints from NAPs and an object model, the recognition problem reduces to determining if there exists a valid object transformation that aligns the model features with the image features. This correspondence problem is classically formulated as search, and approaches such as interpretation trees [16, 17], Generalized Hough Transforms [17] and alignment [6, 23] are used.

Interpretation trees [16, 17] consider correspondences as nodes in a tree and sequentially identify nodes such that the feature correspondences are consistent with the geometric constraints. If a node does not satisfy all the geometric constraints, the subtree below that node is abandoned. Generalized Hough Transforms (GHT) [17], on the other hand, cluster evidence using a discretized pose space. Each pair of model and image feature votes for all possible transformations that would align them together. Geometric constraints are combined with the voting scheme to restrict the search of feasible transformations. Finally, alignment-based techniques [6, 23] start with just enough correspondences to estimate a *hypothesis* transformation. *Verification* is then used to search for additional model features satisfying the geometric constraints. The hypothesis with the most consistent interpretation is chosen.

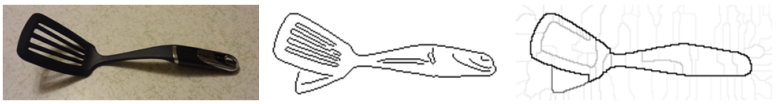
While CAD models and geons have been shown to work well in a number of scenarios, automatically learning 3D models is a considerable challenge [5, 16]. In addition, geons are unable to approximate many complex objects. To address these issues, recent approaches [26, 33] try to learn view-invariant features and non-accidental properties directly from 2D data. A common paradigm is to align and cluster primitives that have similar appearance across viewpoint. For example, the Implicit Shape Model (ISM) [26] considers images patches as primitives and uses Hough voting for recognition. To determine view-invariant features, images patches from all viewpoints of the object are clustered. Each cluster corresponds to a locally view-invariant patch and is associated with a probabilistic set of object centers. A match to a cluster casts a probabilistic vote for its corresponding object positions.

While patches are simple to extract, those on the object boundary contain background clutter and can result in incorrect matches. A more direct approach to modeling shape is to use contours. In the following, we use an approach we developed to illustrate the challenges of learning and using view-invariant curves for object detection. We follow the ISM approach and learn view-invariant curves by grouping curves with similar appearance together. Unlike patches which have a fixed number of pixels, the length of curves varies across viewpoint. We maintain the same number of points by using Coherent Point Drift [32] to generate point-to-point correspondences between curves of nearby views. Given a sequence of object images, we start with the curves of a single view and track the curve deformations by linking the pairwise correspondences. As each frame is processed, a new track is initialized if a curve fragment does not correspond to one that is already being tracked. Tracks



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

Fig. 3: Modeling the deformation of curves across viewpoint. (a) Curve in red tracked across viewpoint. The green arrow specifies the center and upright pose of the object. (b) Aligned curves with their associated centers and pose in blue. (c) Mean curve with deformations computed from aligned curves. (d) Global consistency is difficult to enforce without storing the viewpoint information.



(a) Image (b) Canny edges (c) gPb

Fig. 4: Edge extraction. Current state-of-the-art methods in boundary detection (gPb) are unable to stably extract interior contours which are essential for recognizing specific objects. Canny, on the other hand, can detect these edges, but will also fire on spurious texture edges.

are stopped if the number of unique points remaining is less than 50% of the original curve. Each tracked curve is then represented by its mean and deformations, and is associated with a probabilistic set of object centers as shown in Figure 3.

At recognition time, a modified Iterative Closest Point (ICP) [37] matches image curves with model curves, accounting for the local deformations. If an image curve matches a significant portion of the model curve, it casts a vote for all corresponding poses. The critical issue with allowing local deformations is that it is difficult to enforce global consistency of deformations without storing the constraints for each viewpoint individually. Figure 3d shows an example where the local deformations are valid but the global configuration is not consistent. If the constraints are defined individually for each viewpoint, however, the view-invariance is lost and the approach is equivalent to matching each view independently (i.e., non-invariant).

Another common issue with invariant approaches is that they rely on stable extraction of shape primitives. This is a significant limitation since reliable curve extraction and grouping [29] still proves to be a considerable challenge. While there has been significant development in object boundary detection [1, 8], no single boundary detector is able to extract all relevant curves. The Global Probability of Boundary (gPb) detector, which is designed to ignore stochastic textures, often con-

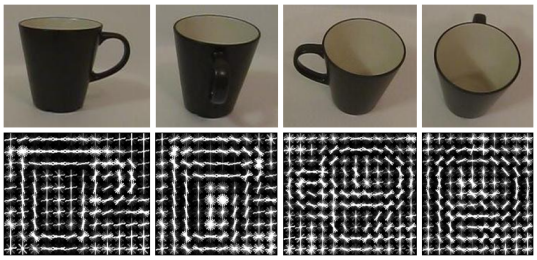


Fig. 5: Non-invariant methods create a template for each viewpoint of the object.

fuses interior contours with stochastic texture as seen in Figure 4. These interior edges provide distinctiveness that is necessary for recognizing specific objects.

Due to the challenges of creating 3D models, extracting shape primitives and learning geometric constraints from data, many recent approaches have moved away from using invariant shape primitives. In the next section, we discuss how non-invariant, view-based methods are able to address the above issues and why they are more effective for specific object recognition under arbitrary viewpoint.

3 Non-invariant (view-based) methods

Non-invariant methods represent an object under multiple viewpoints by creating a "view-based" template [35] for each object view. Each template captures a specific viewpoint, only allowing slight deformation from noise and minor pose variation. Unlike invariant methods which define geometric constraints between pairs or sets of shape primitives, non-invariant methods directly fix both the local and global shape configurations. To combine the output of view-based templates, the scores from each view are normalized [30, 38] and non-maximal suppression is applied.

Non-invariant methods have a number of benefits over invariant ones. First, using view-based templates bypasses the 3D model generation and allows the algorithm to directly observe the exact projection of the object to be recognized. This has the benefit of not approximating the shape with volumetric primitives (e.g., geons), which can lose fine-grained details needed for recognizing specific objects. Secondly, template matching approaches can operate directly on low-level features and do not require extraction of high-level shape primitives. Finally, many non-invariant approaches achieve recognition performances on par or better than invariant ones, while being relatively simple and efficient to implement. Recent results show that they can be successfully applied to tasks such as robotic manipulation.

A number of methods exist for representing object shape from a single view. These range from using curves and lines [11, 12, 39] to sparse edge features [18, 27] and gradient histograms [7]. Methods which use curves and lines often employ 2D view-invariant techniques, similar to the approaches described in Section 2, to re-

duce the number of view samples needed. Interpretation trees [17], Generalized Hough Transforms [17] and alignment techniques [6] which are used for 3D view-invariance are similarly applied to 2D geometric constraints. However, this representation suffers from the same limitations of using high-level shape primitives.

While some approaches use 2D view-invariance, others simply brute force match all the possible viewpoints using low-level features. The Dominant Orientation Template (DOT) method [19] considers the dominant orientation in each cell of a grid. Starting with a single template of an arbitrary viewpoint, new templates are added if the detection score using the previous templates becomes too low. By carefully designing the algorithm for efficient memory access and computation, the approach is able to recognize thousands of templates in near real-time. More recently, the LINE2D [18] approach has demonstrated superior performance to DOT while maintaining similar computation speeds. LINE2D represents an object by a set of sparse edge points, each associated with a quantized gradient orientation. The similarity measure between a template and image location is the sum of cosine orientation differences for each point within a local neighborhood. While LINE2D works well when objects are largely visible, Hsiao *et al.* [22] showed that considering only the points which match the quantized orientations exactly is a much more robust metric when there are occlusions. Finally, the popular Histogram of Oriented Gradients (HOG) [7, 10] approach represents objects by a grid of gradient histograms.

While using low-level features avoids edge extraction, a drawback is the loss of edge connectivity and structure. For example, the HOG descriptor is unable to differentiate between a single line and many lines of the same orientation because their descriptors would be similar. The LINE2D method matches each point individually, resulting in high scoring false positives where neighboring edge points are not connected. These drawbacks, however, are often outweighed by the benefit of operating on low-level features and observing the exact projection of the object in the image.

An additional criticism of non-invariant methods is that they require a large number of templates to sample the view space. For example, LINE2D requires 2000 templates per object. While this many templates may have resulted in prohibitive computation times in the past, advances in algorithms [18, 19] and processing power have demonstrated that template matching can be done very efficiently (e.g., LINE2D and DOT are able to match objects at 10 frames per second). To increase the scalability, template clustering and branch-and-bound [19] methods are commonly used. In addition, templates are easily scanned in parallel and many can be implemented efficiently on Graphics Processing Units (GPUs) [36].

4 Ambiguities

Regardless of whether invariant or non-invariant methods are used, shape recognition under arbitrary viewpoint has many inherent ambiguities. Allowing corners and smooth contours to deform results in a wide range of contours that can match the background, especially for simple shapes. Without additional information, introduc-

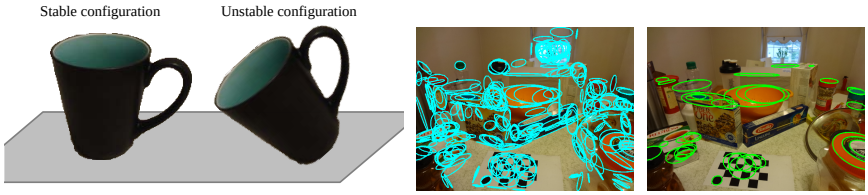


Fig. 6: Stability. (a) In static environments, most objects rest on surfaces. Objects detected in unstable poses can be down-weighted or filtered. We illustrate the usefulness of knowing the support surface orientation with an example of finding image ellipses that correspond to circles parallel to the ground in 3D. These circles are commonly found on household objects. (b) Raw ellipse detections. (c) Ellipse detections remaining after filtering with the ground normal.

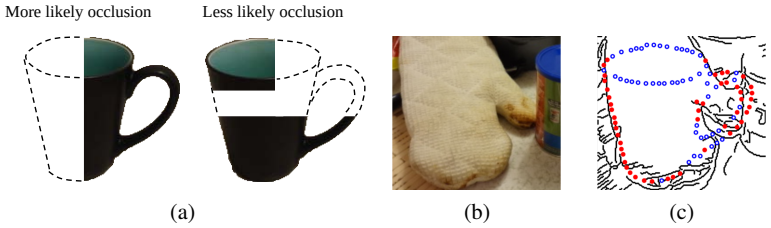


Fig. 7: Occlusion reasoning. Objects in natural scenes are often occluded by objects resting on the same surface. This information can be used to rank the occlusion configurations of an object. (a) The left cup has a more likely occlusion configuration than the right cup. (b) Example false detection window of a cup without occlusion reasoning. (c) Model points that match well to the edgemap are shown in red and those that match poorly are shown in blue. The occlusion configuration is unlikely.

ing view-invariance in shape recognition produces many improbable false positives that align very well to the image.

Objects in real world environments, however, do not appear in arbitrary configurations. Especially when recognizing multiple objects simultaneously, the relationships between object poses are constrained. An approach used in many scenarios is to determine the supporting plane [20] of the objects, such as road in outdoor scenes or table for indoor scenes. Given the supporting surface, the possible stable configurations (Figure 6) of objects on the surface are drastically reduced. Object hypotheses that are in unstable configurations can be filtered or down-weighted. Other approaches, along these lines, reason about scene layout and object recognition together. Given a set of object hypotheses, the approach of [2] determines the best object poses and scene layout to explain the image.

Most shape-based recognition approaches focus solely on finding locations with good matches to the object boundary. However, not all object hypotheses with the

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

Fig. 8: Region information is necessary for robust shape recognition. The false positive shown aligns well to the edgemap but the interior matches poorly. (a) Model object. (b) False positive detection. (c) Zoomed in view of the false positive. (d) Edge points matched on top of the edgemap (red is matched, blue is not matched). (e) Activation scores of individual HOG cells [43]. Hotter color equals better match.

same match percentage are equally plausible (Figure 7). For example in natural scenes, the bottom of an object is more likely to be occluded than the top [9]. Methods for occlusion reasoning [17, 34, 40] range from enforcing local coherency [14] of regions that are inconsistent with object statistics [15, 31, 43] to using relative depth ordering [42, 45] of object hypotheses. Most of these approaches, however, require learning the occlusion structure for each view independently. Recently, our results have shown that explicitly reasoning about 3D interactions of objects [22] can be used to analytically represent occlusions under arbitrary viewpoint and significantly improve shape-based recognition performance.

Finally, while regions with uniform texture are often ignored for recognizing weakly-textured objects, our recent experiments show that they are actually very informative. In Figure 8, the object shape aligns very well to the background, but the texture-less object interior matches poorly. By combining both region and boundary information, many high scoring false positives in cluttered scenes can be filtered.

5 Conclusion

Shape-based instance detection under arbitrary viewpoint is a challenging problem and has many applications from robotics to augmented reality. Current approaches for modeling viewpoint variation can roughly be divided into two main categories: *invariant* and *non-invariant* models. Invariant models explicitly represent the deformations of view-invariant shape primitives, while non-invariant models create a non-invariant, view-based template for each view. While invariant models provide a unified representation of objects across viewpoint, they require generation of 3D models and extraction of high level features which are challenges in themselves. Non-invariant methods are able to bypass these issues by directly operating on low-level features in 2D. They are also able to directly observe the 2D projection without needing to approximate the 3D shape. Recent advances in algorithms and processing power have demonstrated efficient template matching approaches which simultane-

ously detect thousands of templates in near real-time. Since shape recognition under arbitrary viewpoint introduces ambiguities that result in a large number of false positives, additional information such as surface layout estimation, occlusion reasoning and region information are needed for robust recognition.

Acknowledgements This work was supported in part by the National Science Foundation under ERC Grant No. EEEC-0540865.

References

[1] Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2009)
[2] Bao, S., Sun, M., Savarese, S.: Toward coherent object detection and scene layout understanding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2010)
[3] Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological Review **94**(2), 115 (1987)
[4] Biederman, I.: Recognizing depth-rotated objects: A review of recent research and theory. Spatial Vision, 13 **2**(3), 241–253 (2000)
[5] Bilodeau, G., Bergevin, R.: Generic modeling of 3d objects from single 2d images. In: Proceedings of International Conference on Pattern Recognition (2000)
[6] Cass, T.: Robust affine structure matching for 3d object recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(11), 1265–1274 (1998)
[7] Dalal, N.: Finding people in images and videos. Ph.D. thesis, Institut National Polytechnique de Grenoble / INRIA Grenoble (2006)
[8] Dollar, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2006)
[9] Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2009)
[10] Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2008)
[11] Ferrari, V., Fevrier, L., Jurie, F., Schmid, C.: Groups of adjacent contour segments for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2008)
[12] Ferrari, V., Tuytelaars, T., Van Gool, L.: Object detection by contour segment networks. Proceedings of European Conference on Computer Vision (2006)

[13] Flynn, P., Jain, A.: Cad-based computer vision: from cad models to relational graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence **13**(2), 114–132 (1991)
[14] Fransens, R., Strecha, C., Van Gool, L.: A mean field em-algorithm for coherent occlusion handling in map-estimation prob. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2006)
[15] Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Object detection with grammar models. In: Proceedings of Neural Information Processing Systems (2011)
[16] Grimson, W., Lozano-Perez, T.: Localizing overlapping parts by searching the interpretation tree. IEEE Transactions on Pattern Analysis and Machine Intelligence (4), 469–482 (1987)
[17] Grimson, W., Lozano-Pérez, T., Huttenlocher, D.: Object recognition by computer. MIT Press (1990)
[18] Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of texture-less objects. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011)
[19] Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., Navab, N.: Dominant orientation templates for real-time detection of texture-less objects. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2010)
[20] Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. International Journal of Computer Vision (2008)
[21] Hsiao, E., Collet, A., Hebert, M.: Making specific features less discriminative to improve point-based 3d object recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2010)
[22] Hsiao, E., Hebert, M.: Occlusion reasoning for object detection under arbitrary viewpoint. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012)
[23] Huttenlocher, D., Ullman, S.: Recognizing solid objects by alignment with an image. International Journal of Computer Vision **5**(2) (1990)
[24] Ikeuchi, K.: Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks. International Journal of Computer Vision **1**(2), 145–165 (1987)
[25] Koenderink, J.: Solid shape. MIT Press (1990)
[26] Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: Workshop on Statistical Learning in Computer Vision, Proceedings of European Conference on Computer Vision (2004)
[27] Leordeanu, M., Hebert, M., Sukthankar, R.: Beyond Local Appearance: Category Recognition from Pairwise Interactions of Simple Features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2007)
[28] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
[29] Lowe, D.G.: Perceptual organization and visual recognition. Ph.D. thesis, Stanford University (1984)

[30] Malisiewicz, T., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. Proceedings of IEEE International Conference on Computer Vision (2011)
[31] Meger, D., Wojek, C., Schiele, B., Little, J.J.: Explicit occlusion reasoning for 3d object detection. In: Proceedings of British Machine Vision Conference (2011)
[32] Myronenko, A., Song, X.: Point Set Registration: Coherent Point Drift. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(12) (2010)
[33] Payet, N., Todorovic, S.: From contours to 3d object detection and pose estimation. In: Proceedings of IEEE International Conference on Computer Vision (2011)
[34] Plantinga, H., Dyer, C.: Visibility, occlusion, and the aspect graph. International Journal of Computer Vision (1990)
[35] Poggio, T., Edelman, S.: A network that learns to recognize 3d objects. Nature **343**(6255), 263–266 (1990)
[36] Prisacariu, V., Reid, I.: fasthog-a real-time gpu implementation of hog. Department of Engineering Science, Oxford University, Technical Report (2009)
[37] Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3DIM (2001)
[38] Scheirer, W.J., Kumar, N., Belhumeur, P.N., Boult, T.E.: Multi-attribute spaces: Calibration for attribute fusion and similarity search. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2012)
[39] Srinivasan, P., Zhu, Q., Shi, J.: Many-to-one contour matching for describing and discriminating object shape. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2010)
[40] Stevens, M., Beveridge, J.: Integrating Graphics and Vision for Object Recognition. Kluwer Academic Publishers (2000)
[41] Toshev, A., Taskar, B., Daniilidis, K.: Object detection via boundary structure segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (2010)
[42] Wang, N., Ai, H.: Who blocks who: Simultaneous clothing segmentation for grouping images. In: Proceedings of IEEE International Conference on Computer Vision (2011)
[43] Wang, X., Han, T., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: Proceedings of IEEE International Conference on Computer Vision (2009)
[44] Witkin, A., Tenenbaum, J.: On the role of structure in vision. Human and Machine Vision **1**, 481–543 (1983)
[45] Wu, B., Nevatia, R.: Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: Proceedings of IEEE International Conference on Computer Vision (2005)
[46] Zerroug, M., Nevatia, R.: Part-based 3d descriptions of complex objects from a single image. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(9), 835–848 (1999)