Perspective Planar Shape Matching

Andreas Hofhauser[a,b] and Carsten Steger[b] and Nassir Navab[a]

[a]TU München, Boltzmannstraße 3, 85748 Garching bei München, Germany;
[b]MVTec Software GmbH, Neherstraße 1, 81675 München, Germany

ABSTRACT

This paper discusses an edge-direction-based template matching algorithm that allows to detect industrial objects despite perspective distortion. We construct a deformable template by decomposing a shape model into independent parts, where the deformation is restricted to, e.g., a homography. The deformable template in combination with a coarse-to-fine strategy allows to overcome the speed limitations of an exhaustive template matching of a 3D search range. The relevant size of the model that is used for the search at the highest pyramid level is not reduced. Therefore, we do not suffer the speed limitations that prior methods have. Furthermore, enforcing a consistent polarity in each part, but ignoring different polarities between different parts allows us to efficiently and robustly detect untextured metallic objects that are encountered in typical factory automation scenarios. Finally, we present results of an experimental evaluation with respect to speed, robustness and accuracy.

Keywords: Template Matching, Pose Estimation, 3D Machine Vision, Factory Automation, Deformable Models, 3D Alignment

1. INTRODUCTION

Methods that exhaustively search a template in an image are currently the most prominent machine vision algorithms used to detect objects in an image. If the objects are located in a 2D plane, e.g., a conveyor belt, the currently available algorithms are able to reach highest speed, robustness and accuracy requirements. However, if more degrees of freedom than, e.g., translation, rotation, and anisotropic scale should be allowed, the naive extension rapidly reaches the limit that can currently be processed in real-time. For instance, a perspective distortion would require an eight-dimensional search space for a homography. In this paper, we address these issues and show that, with several contributions, it is possible to benefit from the robustness and accuracy of template matching even when an object is perspectively distorted. Furthermore, we show in a number of experiments that it is possible to achieve an interactive rate of detection. This was only possible with descriptor-based approaches until now. However, descriptor-based methods are limited in their use for machine vision because they assume that the objects have discriminative interest points with texture. For the template matching, we use a similarity metric that compares normalized edge directions between model and search image. In this paper we introduce a metric that combines the normalized edge-direction-based similarity metric with the decomposition of the shape model. On the one hand, the polarity changes are taken into account for each part for discriminating its position robustly. On the other hand, polarity changes between different parts of the object are ignored, i.e., part-based polarity changes are compensated. These part-pased polarity changes are, for instance, observed on metallic objects that often occur in factory automation scenarios, e.g., pick-and-place applications (see Figure 1 and Figure 2 for example images).

1.1 Related Work

We roughly classify algorithms for pose detection into template matching and descriptor-based methods. In the descriptor-based category, the rough scheme is to first determine discriminative "high level" features, extract surrounding discriminative descriptors from these feature points and to establish the correspondence between model and search image by classifying the descriptors. The big advantage of this scheme is that the runtime of the algorithm is independent of the degree of the geometric search space.[1–4] While showing outstanding performance in several scenarios, they fail if the object does not have discriminating texture, e.g., if it contains

Send correspondence to Andreas Hofhauser: E-mail: hofhausa@in.tum.de

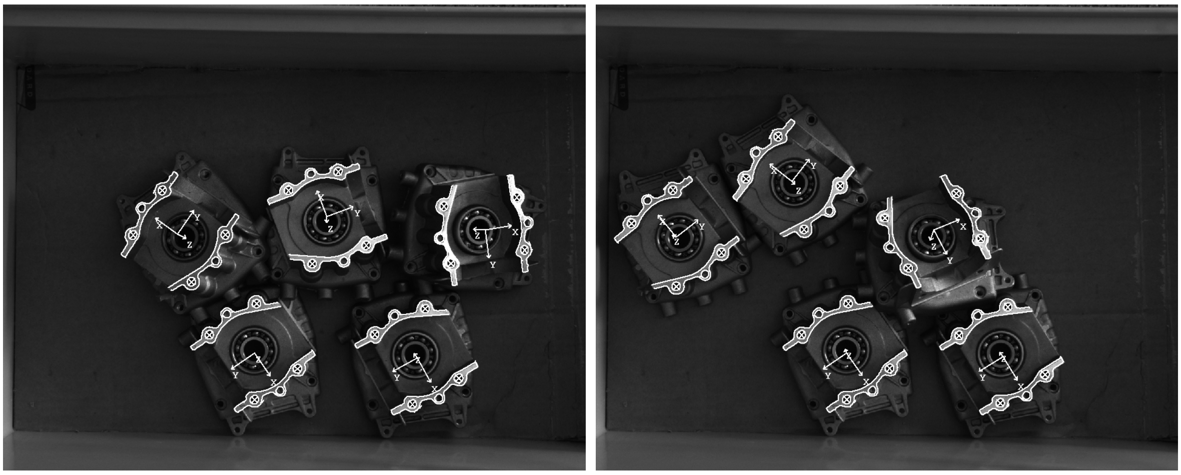

Figure 1. Typical images from a pick-and-place scenario. A planar subpart of a non-planar object is taken as the model and the detection results are depicted.

only sparse edge information in the image. The feature descriptors overlap in the feature space and are not discriminating anymore.

In the template matching category, we subsume algorithms that perform an explicit search. Here, a similarity measure that is either based on intensities (like sum of absolute differences, normalized correlation (NCC), or mutual information) or gradient features is evaluated. However, the evaluation of intensity-based metrics is computationally expensive. Additionally, they are typically not invariant against nonlinear illumination changes, clutter, or occlusion.

For the case of feature-based template matching, only a sparse set of features is compared between the template and the search image. While extremely fast and robust if the object undergoes only rigid transformations, these methods become intractable for a large number of degrees of freedom, e.g., when an object is allowed to deform perspectively. Therefore, we use a match metric that can be evaluated efficiently and allows for local perspective deformations, while preserving robustness to illumination changes, partial occlusion and clutter. While a match metric with normalized directed edge points was used for rigid object detection[5] and also for articulated object detection,[6] its adaptation to 3D object detection is new. A novelty in the proposed approach is that the search method takes all search results for all parts into account at the same time. Despite the fact that the model is decomposed into sub-parts, the relevant size of the model that is used for the search at the highest pyramid level is not reduced. Hence, the presented method does not suffer the speed limitations of a reduced number of pyramid levels that prior art methods have. This is in contrast to, e.g., a component-based detection[6] that could conceptually also be adapted for the perspective object detection. Here, small sub-parts must be detected, leading to a lower number of pyramid levels that can be used to speed up the search. One interesting aspect of the decomposition of the model is that it directly allows to compensate not only non-linear illumination effects but even partial polarity changes.

2. PERSPECTIVE SHAPE-BASED OBJECT DETECTION

In the following, we detail the perspective shape-based model generation and matching algorithm. While the location of the objects is determined with the robustness of a template matching method, we avoid the necessity of expanding the full search space.

2.1 Shape Model Generation

For the generation of our model, we decided to rely on the result of a simple contour edge detection. This allows us to represent objects from template images as long as there is any intensity change. Note that in contrast to corners or interest *point* features, we can model objects that contain only curved *contours* (see the detection results in Figure 3). Furthermore, directly generating a model from an untextured CAD format is possible in principle. Our shape model M is composed of an unordered set of edge points

$$M = \big\{x_i, y_i, d_i^m, c_{jl}, p_j | i = 1 \dots n, j = 1 \dots k, l = 1 \dots size(j)\big\}. \tag{1}$$

Here, x and y are the row and column coordinates of the n model points, while d^m denotes the gradient direction vector at the respective row and column coordinate of the template. We assume that spatially coherent structures stay the same even after a perspective distortion. Therefore, we cluster the model points with expectation-maximization-based k-means such that every model point belongs to one of k clusters. The indicator c_{jl} maps, for each cluster, the indices of the model points that belong to the respective cluster. If a point is not in a cluster, the value 0 is returned. Here, the function $size$ returns the number of elements in cluster j. For the later detection we have to distinguish whether a cluster of a model can be used as a point feature (which gives us two equations for the x and y location) or only as a contour line feature (which suffers from the aperture problem and gives only one equation). This is a label that we save for each cluster in p_j. We detect this by analyzing whether a part contains only one dominant gradient direction or gradient directions with various, e.g., perpendicular, directions. For this, we determine a feature that describes the main gradient directions for each of the j clusters of the model:

$$\mathit{ClusterDirection}_j = \left\| \frac{\sum_{i=1}^{n} \frac{d_i^m}{\|d_i^m\|} c_{ji}}{\sum_{i=1}^{n} c_{ji}} \right\|. \tag{2}$$

If a model part contains only direction vectors in one dominant direction, the length of the resulting *ClusterDirection* vector has the value one or in case of noise slightly below one. In all other cases, e.g., a corner-like shape or directions of opposing gradient directions, the length of *ClusterDirection* is significantly smaller than one. For a straight edge with a contrast change, the sign change of the gradient polarity gives us one constraint and hence prevents movement of that cluster along the edge. Therefore, we label it as a point feature. Because the model generation relies on only one image and the calculation of the clusters is realized efficiently, this step needs, even for models with thousands of points, less than a second. This is an advantage for users of a machine vision system, because an extended offline phase would make the use of a model generation algorithm cumbersome.

2.2 Metric Based on Local Edge Patches

Given the generated model, the task of the perspective shape matching algorithm is to extract instances of that model in new images. We adapted a match metric[5] that is designed to be inherently invariant against nonlinear illumination changes, partial occlusion and clutter. If a location of an object is described by (x, y) (this is 2D translation only, but the formulas can easily be extended for, e.g., 2D affine transformations), then the score function for rigid objects reads as follows:

$$s(x,y) = \frac{1}{n} \sum_{i=1}^{n} \frac{\langle d_i^m, d^s_{(x+x_i, y+y_i)} \rangle}{\|d_i^m\| \cdot \|d^s_{(x+x_i, y+y_i)}\|}, \tag{3}$$

where d^s is the direction vector in the search image, $\langle \cdot \rangle$ is the dot product and $\| \cdot \|$ is the Euclidean norm. The point set of the model is compared to a dense gradient direction field of the search image. Even with significant nonlinear illumination changes that propagate to the gradient amplitude the gradient direction stays the same. Furthermore, a hysteresis threshold or non-maximum suppression is completely avoided in the search image, resulting in true invariance against arbitrary illumination changes.[*] Partial occlusion, noise and clutter results in random gradient directions in the search image. These effects lower the maximum of the score function but do not alter its location. Hence, the semantic meaning of the score value is the ratio of matching model points. It is interesting to note that comparing the cosine between the gradients leads to the same result, but calculating this formula with dot products is several orders of magnitudes faster.

The idea of extending this metric for 3D object detection is that we instantiate globally affine 2D transformations. By allowing successive small movements of the parts of the model, we implicitly evaluate a much

[*] Homogenous regions and regions that are below a minimum contrast change (e.g., less than 3 grey values) can optionally be discarded, because they give random directions that are due to noise. While this is not needed conceptually, it gives a small speedup.

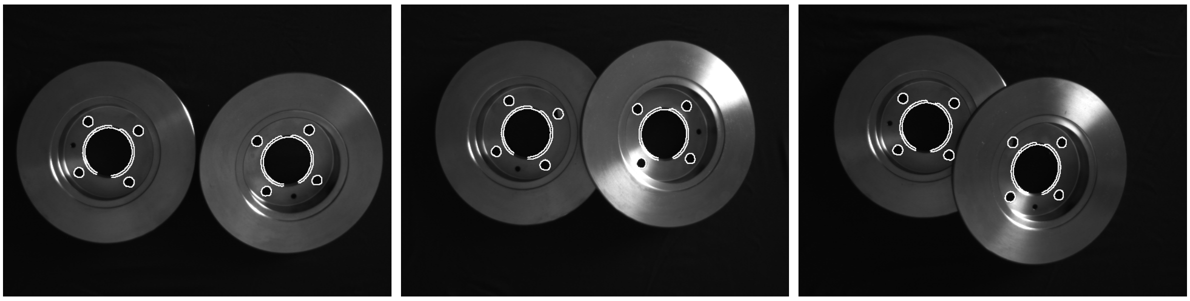

Figure 2. Two brake disks are detected despite partial polarity changes and perspective distortion. The illumination from the side creates specularities on the surface of the metallic object.

higher class of nonlinear transformations, like perspective distortions. Following this argument, we distinguish between an explicit global score function s_{global}, which is evaluated for, e.g., affine 2D transformations[†] and a local implicit score function s_{local} that allows for local deformations. The global score function s_{global} is a sum of the contributions of all the clusters at a certain position. Note that all clusters (sub-parts) of the model contribute to the score function.

$$s_{global}(x, y) = \frac{1}{n} \sum_{j=1}^{k} s_{local}(x, y, j). \tag{4}$$

We assume that even after a perspective distortion the neighborhood of each model point stays the same and is approximated by a local euclidean transformation. Hence, we instantiate a local euclidean transformation T for each cluster and apply it on the model points of that cluster in a small local neighborhood. If the cluster is a point feature we search in a 5×5 pixel window. For a line feature, we search 2 pixels in both directions of $ClusterDirection_j$ of the respective cluster. The local score then is the maximum alignment of gradient directions between the locally transformed model points of each cluster and the search image. Accordingly, the proposed local score function s_{local} is:

$$s_{local}(x, y, j) = \max_{T} \sum_{l=1}^{size(j)} \frac{\langle d^m_{c_{jl}}, d^s_{(x+T(x_{c_{jl}}), y+T(y_{c_{jl}}))} \rangle}{\|d^m_{c_{jl}}\| \cdot \|d^s_{(x+T(x_{c_{jl}}), y+T(y_{c_{jl}}))}\|} \tag{5}$$

For the sake of efficiency, we exploit the mapping that was generated in the offline phase for accessing the points in each cluster (the c_{jl} mapping). Furthermore, we cache $T(x_{c_{ji}})$ and $T(y_{c_{ji}})$ since they are independent of x and y. While the proposed metric is invariant against non-linear illumination changes, it still requires the polarity to stay the same. However, the metric can be modified to become robust against global, partial and local polarity changes. A global polarity change can be coped with by taking the absolute value of the global metric:

$$s_{global\_illumination}(x, y) = \|s_{global}(x, y)\|. \tag{6}$$

This metric returns the same result independent of the global polarity. Hence, the same score is returned if a bright instance of an object is in front of a darker background or a dark object instance is in front of a bright background. Objects with a shiny, metallic surface generate contrast reversals that are hard to predict in a general sense and the polarity changes several times over the object. However, for the local neighborhood the polarity stays the same. This situation can be exploited by ignoring polarity changes between parts, but enforcing the same polarity in each part

$$s_{local\_part}(x, y, j) = \|s_{local}(x, y, j)\|. \tag{7}$$

[†] For sake of clarity we write formulas only for 2D translation. They can easily be extended for, e.g., 2D translation, rotation, scaling, and anisotropic scaling. This is done in our implementation.

2.3 Perspective Template Matching

After defining an efficient score function that tolerates shape changes, we integrated it into a general purpose object detection system. We decided to alter the conventional template matching algorithm such that it copes with perspectively distorted planar 3D objects. Hence, the perspective shape matching algorithm first extracts an image pyramid of incrementally zoomed down versions of the original search image. At the highest pyramid level, we extract the local maxima of the score function s_{global} (4). The local maxima of the score function s_{global} are then tracked through the image pyramid until either the lowest pyramid level is reached or no match candidate is above a certain score value, which determines the amount of occlusion that has to be compensated. While tracking the candidates down the pyramid, a rough alignment was already extracted during the evaluation of the current candidate's parent on a higher pyramid level. Therefore, we first use the alignment originating from the candidate's parent to warp the model. Now, starting from this warped candidate the local transformation T that gives the maximum score determines the local displacement of each cluster to image. Since we are interested in perspective distortions of the whole objects, we fit a homography with the normalized DLT algorithm[7] to the locations of the original cluster centers to the displaced cluster centers given by T. Depending on whether the clusters have been classified as point features or as line features, each correspondence gives two or one equation in the DLT matrix. Then we iteratively warp the whole model with the extracted update and refine the homography on each pyramid level until the update becomes close to the identity or a maximum number of iterations is reached. Up to now, the displacement of each part T is determined up to pixel resolution. However, because the total pose of the object is determined by the displacements of many clusters, we typically obtain a very precise position of the objects. To reach the high accuracy and precision that is a requirement in many machine vision applications, we extract subpixel-precise edge points in the search image and determine correspondences for each model point. Given the subpixel-precise correspondences, the homography is again iteratively updated until convergence. Here, we minimize the distance of the tangent of the model points to the subpixel-precise edge point. Hence, each model edge point gives one equation in the DLT matrix because it is a line feature.

2.4 Pose Estimation Algorithms

Once the correspondences between model and image have been determined, the final 3D pose of the planar object must be determined. For this, we calibrate the cameras and determine the metric size of the model. The metric size of a model can either be given in the CAD data or by measuring the size of the model by placing a calibration plate next to the template object in the image. Given the correspondence described by the homography, there exist two approaches to obtain a pose. One is to take the homography that maps the points of the model to the image and to decompose it.[8,9] The second approach is to directly fit the pose given the point correspondences.[10] Therefore, we evaluated a bundle adjustment solution where we take the result from an analytic solution[9] as starting values for the pose and minimize the residual errors in the image using the Levenberg-Marquardt algorithm. Because all the pose estimation methods receive the same input correspondences, we can directly compare the results. This helps us to make an informed decision about which method is most suitable for our application. It is important to mention that given perfect input data, all methods provide the perfect pose. However, because we expect that even the most precise correspondence extraction is prone to some noise, we are interested in the accuracy and precision that each method can provide. A particularly surprising fact for us was that in preliminary experiments we observed even for subpixel displacements of four corners of a rectangle that define a simulated pose of a cube to result in significant errors in the measured pose of some methods. While this difference is hardly noticed visually by projection of, e.g., an augmentation, it can lead to significant damage in a factory automation or medical navigation scenario.

3. EXPERIMENTS

For the evaluation of the accuracy and robustness of the proposed object detection and pose estimation algorithm, we conducted experiments under synthetic and real world conditions.

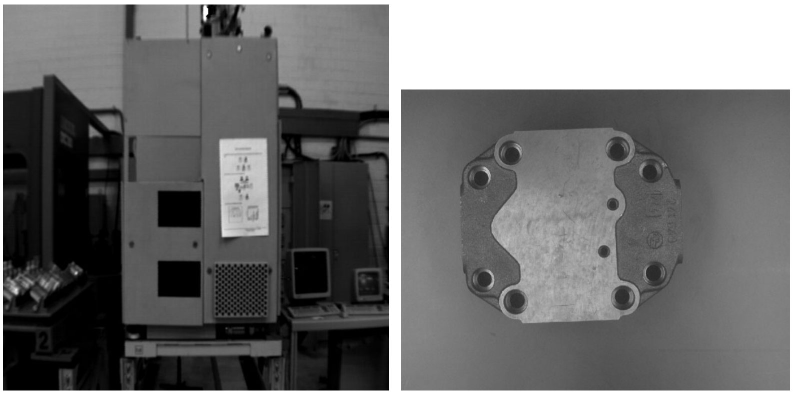

Figure 3. On the left, an image is shown that leads to many interest points. On the right, an industrial object is shown that makes a detection based on corner points difficult.

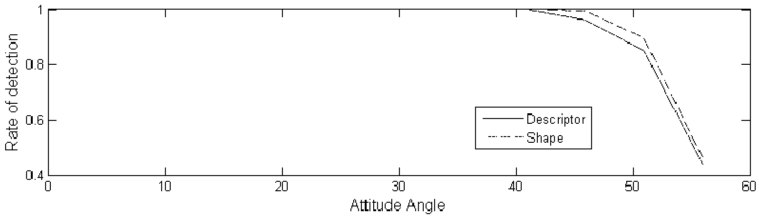

Figure 4. Detection rate plotted against growing latitude angle. Here, the image with significant texture is used.

3.1 Simulation Experiments

The purpose of the simulation experiments is to obtain a comprehensive test and to get the direct comparison with ground truth data that would not be possible in the real world. To make sure that we test the object detection for the relevant range of 3D poses of interest we developed the following test scenario: We generate a uniformly sampled Gaussian sphere with a fixed radius that defines the distance between camera and object. At the points of the sphere we place virtual cameras that look towards the center of the sphere. For the sake of clarity, the pose range can be imagined as one hemisphere of the earth, and we look towards the center of the earth. In the center of the sphere, we project different images that represent the object to be found. The exhaustive test of the complete hemisphere allows us to specify regions in which the object detection works reliably. Given the same images, we conduct a competitive evaluation of our proposed object detection algorithm with respect to robustness of detection with one of the best-performing descriptor-based methods, that were proposed recently.[4] In Figure 3, two test images are depicted. One is significantly textured and leads to many interest *points* that can be extracted. The second image shows an industrial object that mainly consists of curved *edges*. It is interesting to note that for the first image 200 stable interest points can be extracted, but for the second only 30. In Figure 4, we plot the detection rate with respect to the latitude angle. Because we compensate rotation in the template matching, we expect the longitudinal angle to be of little significance. Therefore, we are particularly interested in how much latitude change, which leads to the perspective warping of the image, we are able to tolerate. For the proposed method, it is possible to specity that, we have detected the object if the global score is above 0.85, for example. For the descriptor-based method, it is a non-trivial task the to say that detection was successful, because the RANSAC algorithm will always return at least a random answer consisting of 4 point correspondences. Therefore, we use both methods to detect the object and then used the NCC between the template and the unwarped result to have the same measure for the comparison.

From the results it can be seen that both methods robustly tolerated latitude angle changes of up to 50 degrees on the textured pattern. On the untextured object the descriptor-based method returned many outliers. It is surprising that the descriptor-based method gives no advantage in the amount of attitude angles that can be tolerated. The second interesting aspect of a competitive comparison is to examine the overall accuracy

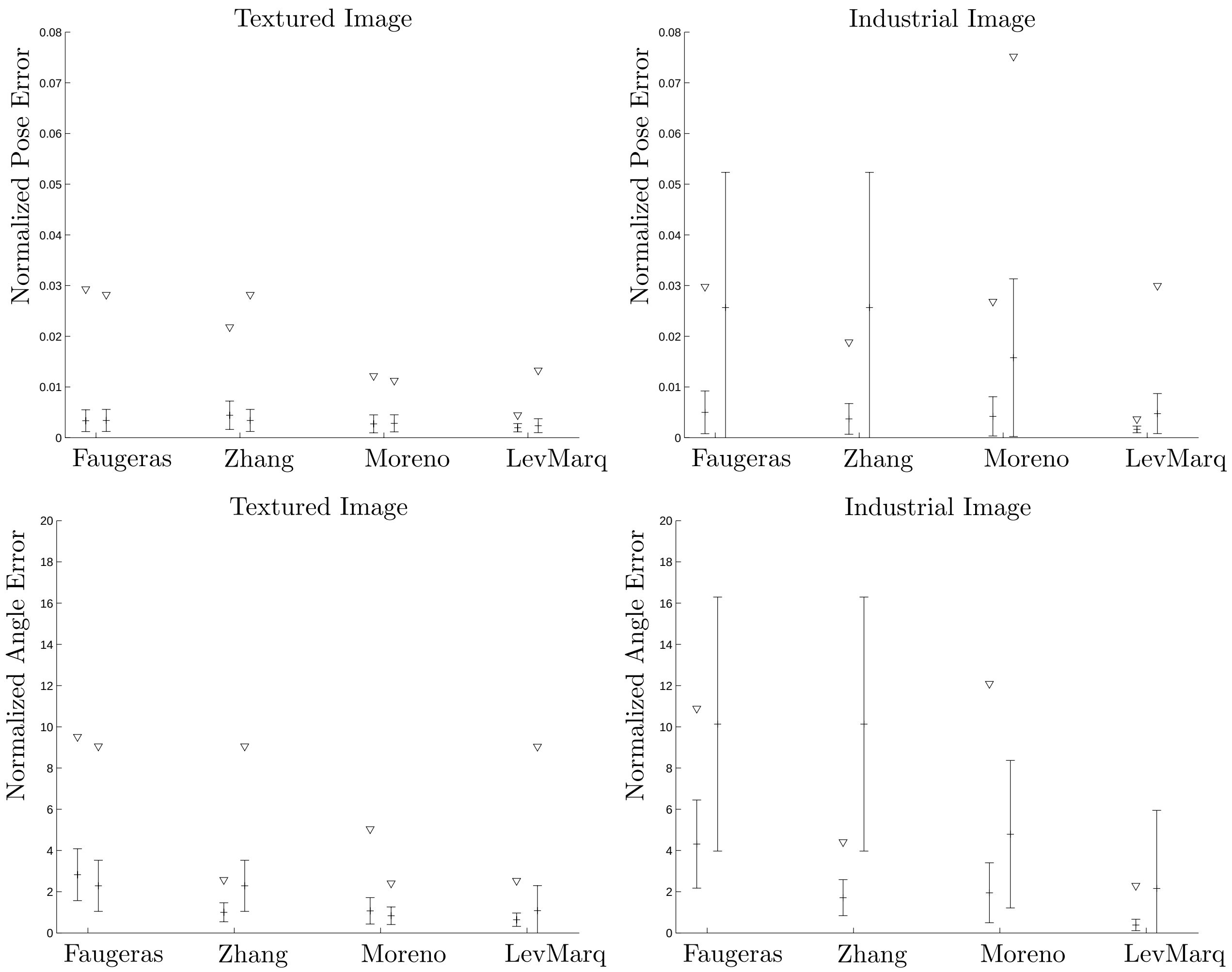

Figure 5. Accuracy, precision and maximum error of normalized position (upper two) and angle (lower two) on the textured image (left two plots) and industrial image (right two plots). For each pose estimation algorithm, the data was provided by the proposed algorithm on the left and on the right with a descriptor-based method. The different pose estimation algorithms show significantly different behavior. The maximum error is displayed with a diamond for the factory automation scenario, where one requirement is to guarantee that certain limits are not exceeded. As can be seen, only the bundle adjustment pose estimation in combination with the proposed edge-based template matching provides satisfactory results independent of image content.

and precision of each method. For determining this, we restricted the latitude angle to a range where both methods worked reliably. Figure 5 shows the mean, standard deviation and maximum error for translational and rotational errors for both images and both detection methods and the different pose estimation methods. The translational error is the euclidean distance between the measured position and the ground truth position normalized by the ground truth position. The rotational error is depicted as the angle that is needed in the axis–angle representation to bring the measured pose to the ground truth pose. Even if this measure does not tell us which direction the rotational axis has, it gives us an intuitive rotational error.

As can be seen in Figure 5, the proposed approach in combination with a bundle adjustment refinement excels in terms of precision, accuracy, and maximum error over all other combinations. This is due to the fact that we exploit all the edge information in the images and not only a few interest points, which makes the result independent of the image content. Furthermore, the algorthm excels in the direct comparison in terms of accuracy, independent of the pose estimation algorithm. In contrast to this, the descriptor-based approach has a huge variance depending on the image texture (see Figure 5 for the results for the methods that are at the

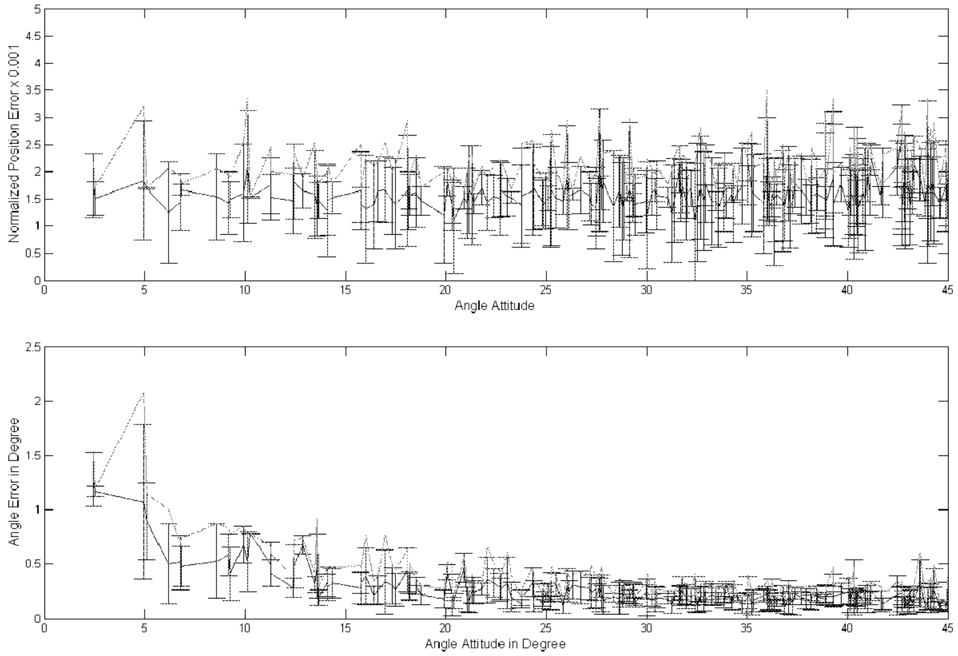

Figure 6. The dependence of the accuracy, precision, and maximum of the translational and rotational error on the latitude angle when the perspective shape matching is used for the industrial test image. A directly perpendicular view onto the object leads to very small pixel changes in case of a small rotation. Hence, the rotatial part of the pose can only be extracted accurately if the object is tilted.

right side of each pair). Furthermore, because of the random nature of descriptor-based methods[4] it is difficult to give guarantees for detection results. In fact, different initializations of the random number generator can lead to changes in the resulting match correspondences. While this is probably acceptable for many scenarios, it must be avoided, e.g., in all factory automation, robotics or medical scenarios. However, it must be mentioned that the online runtime of the proposed method (approximately 60 ms) is slightly slower than a fast descriptor-based approach[4] (approximately 30ms), but faster than, e.g., a descriptor-based method that does not rely on an offline learning phase,[1] which would be a fair comparison[‡]. A further significant difference is in the memory footprint, an important requirement for the growing market of smart vision sensors and embedded platforms. While a descriptor-based method that is based on randomized trees requires an order of 100 Mbyte of memory, a deformable model needs memory of order 10-100 Kbyte and temporary memory of less than 10 Mbyte.

In Figure 6 we evaluated how the accuracy, precision and maximum error depends on the attitude angle. It is encouraging that the normalized pose error is independent of the attitude angle. However, the angle error decreases with increasing latitude angle at an overall small absolute value. The bigger error at an exactly perpendicular camera position is not surprising because here a change of angle has the smallest influence on the image.

3.2 Bin Picking of Car Parts

For evaluation purposes, we simulated a typical situation for a bin picking application. We disassembled several two-strike engines and taught the different parts to a machine vision system (see Figure 1 and Figure 7) that uses the proposed object detection algorithm. To be able to infer the 3D position, we calibrated the camera by a standard algorithm for camera calibration.[11] The metric size of the objects were inferred by placing an object at a fixed position and obtaining a model image. Then, a calibration plate was placed onto the part to infer the 3D

[‡] Laptop PentiumM 1.86GHz 1.00GB RAM.

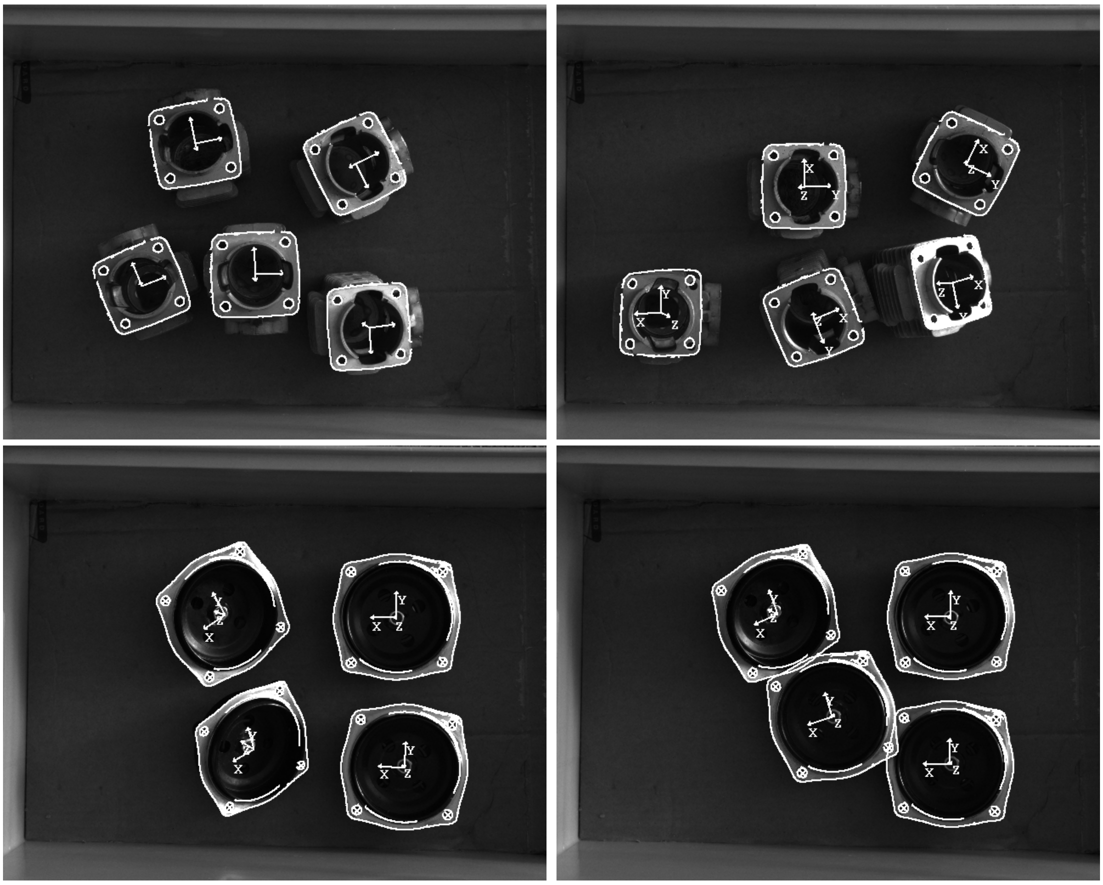

Figure 7. Sample images used in the real world experimental evaluation scenario.

plane in which the object resided. In order to obtain accurate results, it is important to keep the camera-object relation fixed, while an image with the object and an image with the calibration plate are acquired.

Because the thickness of the calibration plate is known, the offset was corrected. This allowed to accurately measure the metric size of the model. With the camera calibration and the measured model, we were able to extract the absolute pose of the part in the camera coordinate frame (see Section 2.4). The different parts were placed randomly inside the boxes and were imaged by a single camera. Because the positions and illumination were not controlled, random perspective distortions and non-linear illumination changes occured. However, the position of the objects were extracted with high robustness and accuracy. The detection of up to 6 parts in a 640 × 480 image with subpixel precision and least-squares pose estimation runs faster than 100ms. The algorithm allows transparent tuning if, for instance, fewer parts or a reduced pose range can be expected. Here, the processing time can be significantly decreased to reach frame-rate detection. Another machine vision example is the challenging task of detecting a complete car door (see Figure 8). Here, the user must manually select a sub-part of the inner side of the car door that is approximately planar.

4. CONCLUSION

In this paper we presented an industrial solution for planar 3D object detection that can be utilized in a wide range of applications, ranging from metrology to pick-and-place scenarios. For this, we extended an already existing edge-polarity-based match metric so that it tolerates local shape changes. Furthermore, the decomposition of the shape model was combined with a match metric that ignores part-based polarity changes. In an evaluation we showed the applicability of the method for an industrial bin-picking scenario.

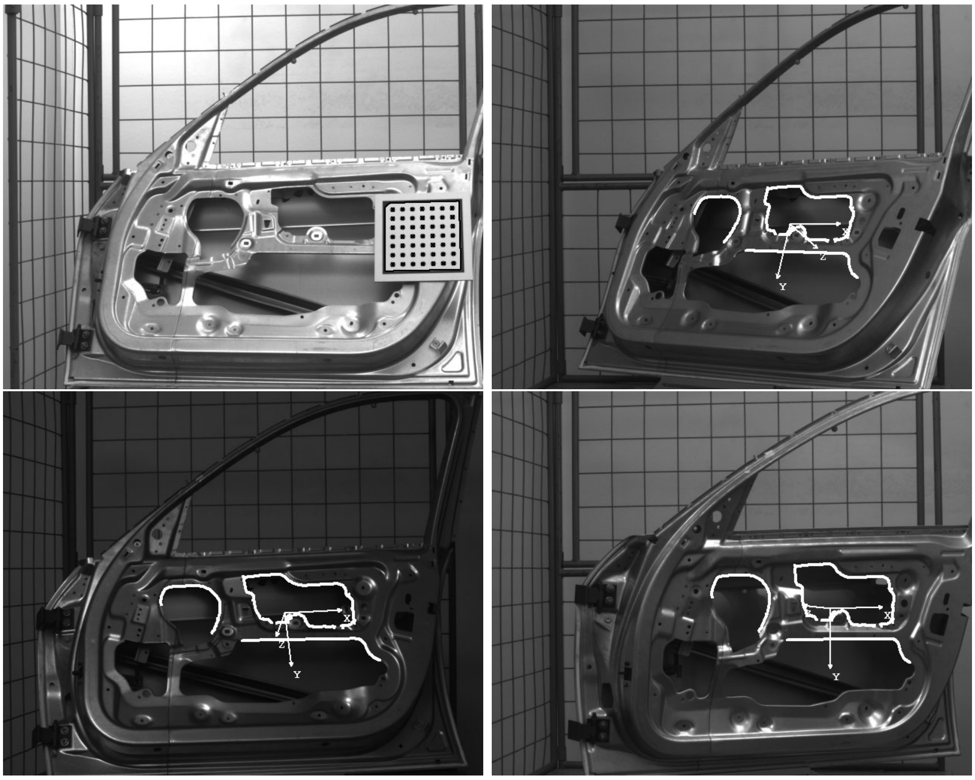

Figure 8. A planar part is used for the detection of the car door. The teaching image shows the calibration plate.

REFERENCES

[1] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision* **60**(2), 91–110 (2004).

[2] Berg, A., Berg, T., and Malik, J., "Shape matching and object recognition using low distortion correspondences," in [*Conference on Computer Vision and Pattern Recognition*], **1**, 26–33 (2005).

[3] Pilet, J., Lepetit, V., and Fua, P., "Real-time non-rigid surface detection," in [*Conference on Computer Vision and Pattern Recognition*], **1**, 822 – 828 (2005).

[4] Özuysal, M., Fua, P., and Lepetit, V., "Fast keypoint recognition in ten lines of code," in [*Conference on Computer Vision and Pattern Recognition*], **1**, 1–8 (2007).

[5] Steger, C., "Occlusion, clutter, and illumination invariant object recognition," in [*International Archives of Photogrammetry and Remote Sensing*], **XXXIV, part 3A**, 345–350 (2002).

[6] Ulrich, M., Baumgartner, A., and Steger, C., "Automatic hierarchical object decomposition for object recognition," in [*International Archives of Photogrammetry and Remote Sensing*], **XXXIV, part 5**, 99–104 (2002).

[7] Hartley, R. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, Cambridge (2000).

[8] Faugeras, O., [*Three-Dimensional Computer Vision: A Geometric Viewpoint*], MIT Press, Cambridge, MA (1993).

[9] Zhang, Z., "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(11), 1330–1334 (2000).

[10] Moreno-Noguer, F., Lepetit, V., and Fua, P., "Accurate non-iterative O(n) solution to the PnP problem," in [*Conference on Computer Vision and Pattern Recognition*], 1–8 (2007).

[11] Steger, C., Ulrich, M., and Wiedemann, C., [*Machine Vision Algorithms and Applications*], Wiley-VCH, Weinheim (2007).