Recognition and Tracking of 3D Objects

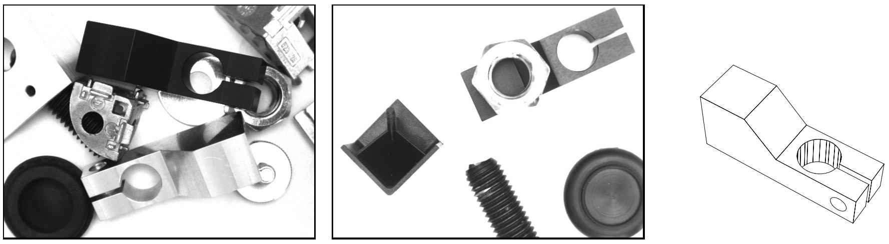
Christian Wiedemann, Markus Ulrich, and Carsten Steger

MVTec Software GmbH
Neherstr. 1, 81675 München, Germany
{wiedemann,ulrich,steger}@mvtec.com

Abstract. This paper describes a method for recognizing and tracking 3D objects in a single camera image and for determining their 3D poses. A model is trained solely based on the geometry information of a 3D CAD model of the object. We do not rely on texture or reflectance information of the object's surface, making this approach useful for a wide range of object types and complementary to descriptor-based approaches.
An exhaustive search, which ensures that the globally best matches are always found, is combined with an efficient hierarchical search, a high percentage of which can be computed offline, making our method suitable even for time-critical applications. The method is especially suited for, but not limited to, the recognition and tracking of untextured objects like metal parts, which are often used in industrial environments.

1 Introduction

In industrial or robot applications, often untextured objects like the metallic clamps shown in Figures 1(a) and (b) must be recognized and tracked in monocular images. Obviously, the automation level of many industrial processes could be improved significantly if the pose of such objects could be determined reliably. However, we are not aware of any published technique that is able to robustly recognize and track an untextured 3D object in a monocular image in a reasonable amount of time.



(a) (b) (c)

Fig. 1. (a) Image of two differently colored metallic clamps in a cluttered environment. (b) Image of a partially hidden clamp. Our approach is able to detect the objects in (a) and (b) reliably. (c) CAD model of the clamp shown in (a) and (b) that serves as input for the model generation.

2 Related work

Approaches for recognizing 3D objects in monocular images have been extensively studied. One challenge is the very large six dimensional search space if the object is imaged from an unknown viewpoint.

For a time, view-based approaches for 3D object recognition were very popular. The search image was compared with precomputed 2D views of the object to determine the object pose ([5],[4],[7]). These approaches tried to deal with the full geometric search space by clustering the views. None of them became accepted in practice. Some view-based approaches use previous knowledge to reduce the search space, e.g., [21], which assumes that the object is lying on a conveyor belt and therefore appears in a known distance in front of the camera.

Other approaches circumvent the large geometric search space. Feature-based approaches ([6], [8]) use features like gray value edges or more complex features that result from grouping extracted primitives. The 3D pose of the object is calculated directly from corresponding 2D-3D features. These approaches circumvent the large geometric search space but they have to deal with the search space that results from establishing the correspondences, which can be very large as well. More complex features can be used to reduce this search space, but then the feature extraction becomes less robust, especially if parts of the object are occluded. This makes these approaches unsuitable for real applications.

Descriptor-based methods ([2], [15], [3], [18], [1], [11]) create artificial views of the object in which feature points are determined together with discriminative descriptors. Based on these descriptors, a classifier is trained. In the search phase, the correspondence between the model and the search image is established by classifying the descriptors derived from the search image. The big advantage of descriptor-based approaches is that their run-time is independent of the size of the geometric search space. They show outstanding performance in several scenarios but they are restricted to the recognition of textured objects because only then meaningful descriptors can be determined.

There exist tracking approaches ([12], [16], [10], [13], [14]) that are able to determine the pose of 3D objects based on their geometry or texture. They do not have to deal with the above mentioned search space because the initial object pose is required to be known. If the object appears too far away from the initial pose, these approaches will fail. In practice, this is a critical problem because in most applications it cannot be avoided that the object is occluded and reappears at an unpredictable position in the image.

3 Detailed description of the approach

3.1 3D model generation

The input of the model generation step is the triangulated surface of the object model. Figure 1(c) shows a CAD model of the clamps shown in Figures 1(a) and (b). The object mainly consists of planar surfaces as well as of a cylinder, which is approximated by several planar faces.

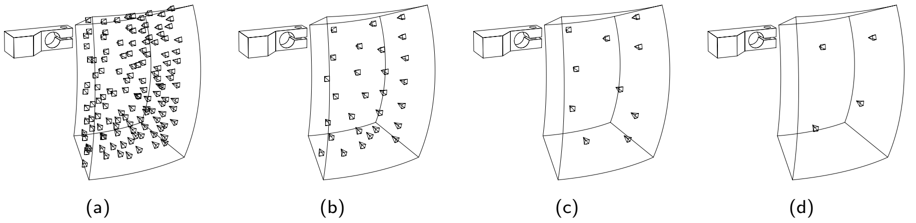


Fig. 2. Resulting model views inside a certain section of a spherical shell on pyramid level 1 (a), level 2 (b), level 3 (c), and level 4 (d). The views are visualized by small square pyramids representing the cameras.

Hierarchical view generation We propose a hierarchical view-based approach that combines a pyramid search with a hierarchy of object views. With this, on the top pyramid level only few views must be investigated. This allows to apply an exhaustive search, which ensures that the globally best matches are always found, but would be too expensive without the hierarchical approach.

For the training of the 3D model, different views of the object are generated automatically by placing virtual cameras around the 3D object and by projecting the object into the image plane of each virtual camera using the camera parameters of the real camera to be used for the recognition. The object is assumed to be at the center of a sphere that defines a spherical coordinate system. The virtual cameras, which are used to create the views, are arranged around the object in such a way that they all point to the center of the sphere. The range of poses covered by the virtual cameras can be restricted to a certain section of a spherical shell (see Figure 2) by specifying intervals for the spherical parameters λ (longitude), φ (latitude), and d (distance).

The sampling of the views within the pose range is automatically determined during the training process to maximize robustness and speed of the recognition. To further increase the speed of the recognition, the model is created on multiple pyramid levels. Because higher pyramid levels allow a coarser sampling of the views, the computation is performed for each pyramid level separately.

In Figures 2(a)–(d) the necessary views on different pyramid levels are shown. Here, it is sufficient to distinguish only four different views on the fourth pyramid level. At each view a reference to all child views is stored. The child views are those views on the next lower pyramid level that represent the same range of poses as the view on the current pyramid level. The references are stored in a tree structure. This information is used in the online-phase to query for a given view on a higher pyramid level the views on the next lower pyramid level to refine the matches.

Model image generation After the tree has been completely generated, for each pyramid level and each view on this level, a 2D model is created with the approach presented in [19]. The 2D model consists of a plurality of edge points with a corresponding gradient direction vector. The similarity measure used during the search is the mean absolute value of the dot products of the corresponding normalized edge gradient directions in the model and in the search image. It is robust to occlusions, clutter, and non-linear contrast changes.

It is essential that the scale-space effects that are introduced by the image pyramid in the later search image are taken into account also during the model generation. Therefore, it is not possible to directly use the projected 3D model edges scaled to the corresponding pyramid level to create the 2D model. Instead, the 3D model is projected into an image of the original resolution using an appropriate hidden-line-algorithm, e.g. [17]. From this image, an image pyramid is derived. This pyramid shows similar scale-space effects than the pyramid of the search image. Therefore, the model edge points can now be extracted from the respective pyramid level.

The projection is done in such a way that a 3-channel color image is obtained, where the three channels represent the three elements of the normal vector of the faces of the 3D object. This has the advantage that the edge amplitude that can be measured in this color image is directly related to the angle in 3D space between the normal vectors $\mathbf{n}_1 = (R_1, G_1, B_1)$ and $\mathbf{n}_2 = (R_2, G_2, B_2)$ of two neighboring faces of the 3D object. Let us assume without loss of generality (because of the isotropy of the color tensor, see below) that the two projected faces cause a vertical edge in the image. Then, the first derivatives in row direction are $gr_R = gr_G = gr_B = 0$ and in column direction are $gc_R = R_2 - R_1$, $gc_G = G_2 - G_1$, and $gc_B = B_2 - B_1$.

The edge amplitude in a color image can be obtained by computing the eigenvalues of the color tensor $\mathbf{C}$ [9]:

$$\mathbf{C} = \begin{pmatrix} grr & grc \\ grc & gcc \end{pmatrix} \tag{1}$$

where in the case of a 3-channel image

$$\begin{aligned} grr &= {gr_R}^2 + {gr_G}^2 + {gr_B}^2 \\ grc &= gr_R gc_R + gr_G gc_G + gr_B gc_B \\ gcc &= {gc_R}^2 + {gc_G}^2 + {gc_B}^2 \end{aligned} \tag{2}$$

The edge amplitude A is the square root of the largest eigenvalue of $\mathbf{C}$.

$$A = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \tag{3}$$

Thus, the edge amplitude computed in the image corresponds to the length of the difference vector of the two normal vectors. The angle between both normal vectors is derived from the edge amplitude by using the following formula:

$$\delta = 2 \arcsin(A/2) \tag{4}$$

The obtained color image of the projected model serves as the model image and is passed to the model generation step of the approach presented in [19], extended by color edge extraction. First, the edge amplitude in the model image is computed [9]. Only pixels that exceed a certain threshold are included

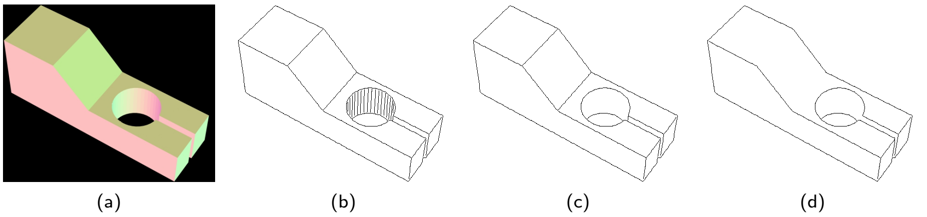


Fig. 3. Generated model image (a) and edges after applying a threshold on the color edge amplitude corresponding to a minimum face angle of 5° (b), 15° (c), and 50° (d).

in the model. Often, the 3D description of the model contains many edges that are invisible in a true image of the object. For example, such edges result from triangulation methods of the CAD software that are used to approximate curved surfaces by a sufficient number of planar faces. Consequently, these edges must not be included in the 2D model. For example, the edges of the planar faces that approximate the cylindrical hole in Figure 1(c) must be suppressed. Because of the relation described above, one can suppress such edges by passing an appropriate threshold for the minimum face angle δ_{min}, which is very intuitive. Then the minimum angle can easily be transformed to a threshold value A_{min} that can be applied to the edge amplitude by solving (4) for A.

Figure 3(a) shows the resulting model image of one sample view. In Figure 3(b) the edges that result when setting $\delta_{min} = 5°$ ($A_{min} = 0.087$) are visualized. Because the planar faces that approximate the cylinder occur in 8° steps, the vertical edges are still visible. The edges that are obtained when setting $\delta_{min} = 15°$ ($A_{min} = 0.261$) are shown in Figure 3(c). The edges of the cylinder are successfully suppressed. Thus, the 3-channel model images enables the use of existing 2D edge-based matching approaches by simply passing a threshold for the edge amplitude to eliminate edges that are invisible in a real image. It also has the advantage that the problem of generalizing a 3D model when using a pyramid approach becomes unnecessary because the scale-space effects are taken implicitly into account by computing the image pyramid of the model image.

Finally, the 2D model is generated from the 3-channel image on the associated image pyramid level (see [19] and [9] for details). The 2D model created on the current pyramid level is automatically rejected if it does not show enough distinct characteristics that are necessary to distinguish the model from clutter in the image (see [20] for details).

The 3D model consists of a plurality of 2D models on several pyramid levels. For each 2D model, the corresponding 3D pose is stored. Additionally, 2D models on neighboring pyramid levels are connected in form of the tree described above.

3.2 3D object recognition

In the online-phase the created 3D model is used for recognizing the 3D object in a single camera image and for determining the 3D pose of the object with respect to the camera coordinate system. First, an image pyramid is built from the input image. The recognition starts at the highest pyramid level on which at least one valid 2D model is available. All 2D models on this pyramid level are searched by computing a similarity measure [19] that is robust to occlusions, clutter, and non-linear contrast changes between the 2D models of the views and the current image pyramid level. For this, the 2D models are rotated and scaled in the necessary range and the similarity measure is computed at each position of the scaled and rotated 2D models in the image. The 2D poses (position, rotation, scaling) of matches that exceed a certain similarity threshold are stored in the list of match candidates. On the next lower pyramid levels all 2D models that do not have a parent node in the tree are searched in the same way as the views on the highest pyramid level. Additionally, the match candidates that have been found on the previous pyramid level are refined. The refinement is performed by selecting all child views in the tree and computing the similarity measure between the 2D models of the child views and the current image pyramid level. However, the range of investigated positions, rotations, and scalings can be limited to a close neighborhood of the parent match. This process is repeated until all match candidates are tracked down to the lowest pyramid level. The combination of a pyramid approach with hierarchical model views that are arranged in a tree structure is essential for time-critical applications and has not been applied in previous recognition approaches.

The 2D models are created during the training by assuming a camera that is directed to the object center. Note that by this the original 6 degrees of freedom are not covered as often believed, because the object may appear at an arbitrary position in the search image. The 2D transition between the 2D model and the imaged object corresponds to a 3D rotation of the camera around its optical center. Therefore, the 2D model and the imaged object are related by a 2D homography. To cope with this, we transform the 2D model by applying the homography before performing the matching. This is an absolutely essential step that to our best knowledge has not been applied in previous view-based recognition approaches. The parameters of the homography are computed based on the position of the object in the image that is approximately known from the next higher pyramid level.

On the top pyramid level, an exhaustive search must be performed because no previous knowledge is available. Thus, the matching is performed at all image positions. However, projectively transforming the models depending on the current image position would be too expensive. Fortunately, on the highest level in general the projective distortions are very small because of the subsampling that comes with the image pyramid. To further reduce the distortions on the top pyramid level, the planar 2D model as well as the image is mapped to the surface of a sphere before applying the matching on the highest pyramid level. Then, the projection does not change when rotating the camera around its optical center. Unfortunately, there is no mapping from the sphere into the plane without introducing distortions. However, in general these distortions are smaller than the projective distortions. The tracing through the pyramid is performed in the original (non-spherical) image as described above.

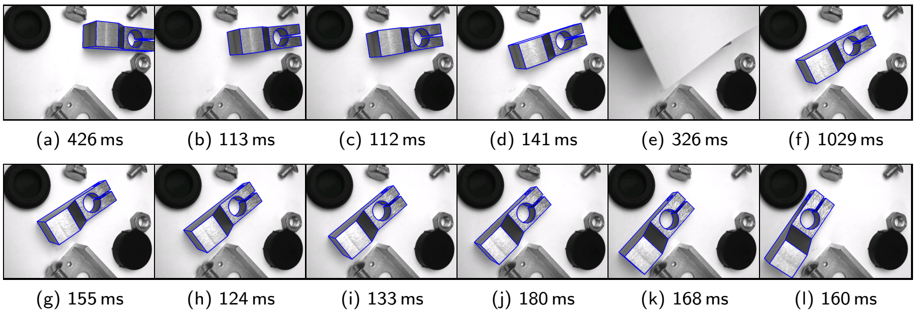
As result of the matching one obtains the 2D poses (position, rotation, scaling) of the 2D matches that exceed a certain similarity measure. For each match, the corresponding 3D object pose is computed based on the 2D matching pose and the 3D pose of the model view that is associated with the match.

3.3 Pose refinement

The accuracy of the obtained 3D pose is limited to the sampling of the views and the sampling of the 2D poses during the 2D matching. For practical applications, this is insufficient. Therefore, a refinement of the 3D pose is performed by using a least-squares adjustment. For this, the 3D object is projected into the search image by using the current pose. Lines that are invisible or that represent object edges at which the angle between the two adjacent object faces is below the specified minimum face angle (see Section 3.1) are suppressed. The visible projected model edges are sampled to discrete points. For each sampled edge point a corresponding subpixel-precise image edge point is searched in the neighborhood of the projected model edge. Finally, the refined 3D pose is obtained through a robust iterative non-linear optimization using the Levenberg-Marquardt algorithm. During the optimization the squared distances of the image edge points to their corresponding projected model edge are minimized directly over the 6 pose parameters. After the minimization, the refined pose parameters are available. Because from the refined pose parameters new correspondences can arise, the optimization algorithm is integrated within an outer iteration loop. The hidden-line algorithm is only applied in the first iteration. In further iterations, not the complete 3D model edge but only the part that was visible in the first iteration is projected. This speeds up the pose refinement significantly. In most cases the error that is introduced by this simplification only marginally degrades the obtained accuracy because the initial pose is already close enough to the optimum to prevent significant changes in the perspective.

3.4 Object tracking

To be able to track 3D objects, first the object must be detected and its pose must be determined without prior knowledge of its position. This start pose of the object can be determined with the above described approach. Then, to track the object, the above described recognition approach is used in a slightly modified form: The search range on the highest pyramid level is restricted to a small range of poses around the pose of the object determined in the previous image. This reduces the number of hypotheses on the highest pyramid level drastically and leads to a faster and even more robust recognition of the object. If the object cannot be recognized in the image, the search range is increased. With this, it is possible to robustly track objects that are occluded in some frames and that reappear at an unpredictable position. In Figure 4, a sequence of 12 images is shown together with the corresponding recognition time. This sequence depicts the above described tracking approach.



(a) 426 ms (b) 113 ms (c) 112 ms (d) 141 ms (e) 326 ms (f) 1029 ms

(g) 155 ms (h) 124 ms (i) 133 ms (j) 180 ms (k) 168 ms (l) 160 ms

Fig. 4. Results of a tracking sequence: In (a), the complete pose range must be investigated because no initial pose is available. This results in a comparatively slow recognition. In (b)-(d), the pose range is restricted to a small range around the pose of the object in the previous image, leading to a faster recognition of the object. In (e), the object is occluded and consequently not recognized. In (f), no initial pose is available because the object was not recognized in the previous image, which leads to a slower recognition. Finally, in (g)-(l), the object is tracked using the restricted pose range.

4 Evaluation

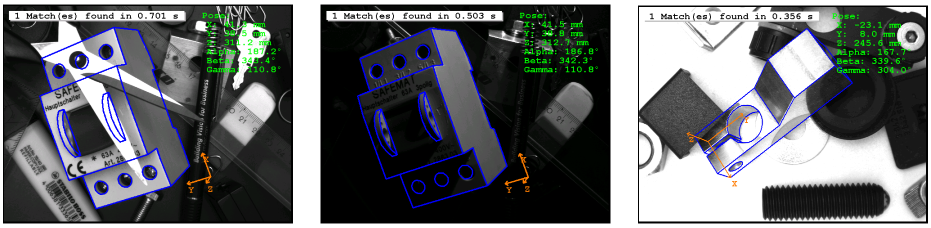
As a first evaluation of the recognition of 3D objects, we simulated various objects. If the search range overlapped the simulated range of poses, the objects were detected without exception. This is the benefit of the applied exhaustive search, which always returns the global optimum.

To evaluate the accuracy of the detection, we compared the object pose determined in a real image with a reference pose, derived using a calibration target. Two different objects (clamp and fuse) were used for the evaluation. We acquired 8 bit gray scale images of size 640×480 with a focal length of 8.5 mm, where the objects where placed in a distance range between 150–350 mm in front of the camera. Table 1 shows the results derived from 50 test images of each object. All tests were performed on a 2.33 GHz Intel Xeon E5345. The models were created within the pose range $\lambda = \varphi = [-50; +50]^\circ$ and $d = [150; 200]$ mm for the clamp model and $d = [250; 350]$ mm for the fuse model.

To illustrate the robustness to occlusions, clutter, and contrast changes, in Figure 5 typical images are shown in which the objects could be found correctly.

Object	σ_{pos} [mm]	σ_{pos} [%]	σ_{rot} [°]	time [s]
Clamp	0.39	0.20	0.48	0.3
Fuse	0.74	0.21	0.60	0.9

Table 1. Result of the accuracy evaluation. The first two columns contain the standard deviations of the position in mm and as percentage with respect to the absolute distance. The third column contains the standard deviations of the (Rodriguez) rotation angle in degrees. The fourth column contains the mean recognition times.



(a) (b) (c)

Fig. 5. The approach is robust to occlusions and clutter (a), to non-linear contrast changes (b), and to reflections on the object's surface (c) to a high degree. For illustration purposes, the pose of the found object is visualized by overlaid edges.

5 Conclusions

A robust and fast method for recognizing and tracking a 3D object in a single camera image and for determining its 3D pose was presented. Only geometry information is used for recognition, and hence no texture information is needed. The novel combination of a pyramid approach with hierarchical model views that are arranged in a tree structure is essential for time-critical applications. The generation of 3-channel model images enables the use of existing 2D edge-based matching approaches by simply passing a threshold for the edge amplitude to eliminate object edges that are not visible in a real image. The projective transformation of 2D models during the tracing is essential for a high robustness of the recognition approach. Finally, a high accuracy is obtained by applying a subsequent 3D pose refinement without performing an expensive projection in each iteration. Furthermore, optional methods are provided that efficiently map the model and the image to a spherical projection to eliminate projective distortions on the highest pyramid level that in some cases otherwise would reduce the robustness of the 2D matching. Computation times well below 1 second allow even time-critical applications to benefit from our approach.

The presented tracking approach is based on the above described recognition approach. It has the advantages that no initialization of the tracking process is necessary and that the search range can be specified arbitrarily large. With this, it is possible to recover the object, even if it was occluded and reappears at an unpredictable position.

References

1. H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *9th European Conference on Computer Vision*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417, Berlin, 2006. Springer-Verlag.
2. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
3. A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Computer Vision and Pattern Recognition*, pages 26–33, 2005.

4. H. Borotschnig, L. Paletta, M. Prantl, and A. Prinz. Appearance based active object recognition. *Image and Vision Computing*, 18(9):715–727, June 2000.
5. J. H. M. Byne and J. A. D. W. Anderson. A CAD based computer vision system. *Image and Vision Computing*, 16(8):533–539, June 1998.
6. M. S. Costa and L. G. Shapiro. 3D object recognition and pose with relational indexing. *Computer Vision and Image Understanding*, 79(3):364–407, September 2000.
7. C. M. Cyr and B. B. Kimia. 3D object recognition using shape similarity-based aspect graph. In *8th International Conference on Computer Vision*, volume I, pages 254–261, 2001.
8. P. David and D. DeMenthon. Object recognition in high clutter images using line features. In *10th International Conference on Computer Vision*, pages 1581–1588, 2005.
9. S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33:116–125, 1986.
10. T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.
11. S. Hinterstoisser, S. Benhimane, and N. Navab. N3M: Natural 3D markers for real-time object detection and pose estimation. In *11th International Conference on Computer Vision*, 2007.
12. H. Kollnig and H. H. Nagel. 3d pose estimation by directly matching polyhedral models to gray value gradients. *International Journal of Computer Vision*, 23(3):283–302, 1997.
13. A. Ladikos, S. Benhimane, and N. Navab. A real-time tracking system combining template-based and feature-based approaches. In *International Conference on Computer Vision Theory and Applications*, 2007.
14. J. Liebelt and K. Schertler. Precise registration of 3d models to images by swarming particles. In *Computer Vision and Pattern Recognition*, pages 1–8, 2007.
15. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
16. É. Marchand, P. Bouthemy, and F. Chaumette. A 2d-3d model-based approach to real-time visual tracking. *Image Vision Comput.*, 19(13):941–955, 2001.
17. M. S. Paterson and F. F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete & Computational Geometry*, 5(1):485–503, 1990.
18. J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Computer Vision and Pattern Recognition*, pages 822–828, 2005.
19. C. Steger. Occlusion, clutter, and illumination invariant object recognition. In *International Archives of Photogrammetry and Remote Sensing*, volume XXXIV, part 3A, pages 345–350, 2002.
20. M. Ulrich. *Hierarchical Real-Time Recognition of Compound Objects in Images*. PhD thesis, Bauingenieur- und Vermessungswesen der Technischen Universität München, 2003.
21. C. Von Bank, D. M. Gavrila, and C. Wöhler. A visual quality inspection system based on a hierarchical 3d pose estimation algorithm. In *Deutsche Arbeitsgemeinschaft für Mustererkennung*, volume 2781 of *Lecture Notes in Computer Science*, pages 179–186. Springer-Verlag, 2003.