A Visual Quality Inspection System Based on a Hierarchical 3D Pose Estimation Algorithm

Clemens von Bank Dariu M. Gavrila Christian Wöhler

DaimlerChrysler Research and Technology, P. O. Box 2360, D-89013 Ulm, Germany

Abstract

This paper presents a quality inspection system based on an efficient model and view based algorithm for locating objects in images and estimating their pose. Off-line, edge templates are generated from a 3D model. On-line, a hierarchical edge template matching technique generates matching solutions from which the pose of the object is derived. This approach tackles the difficult typical trade-off between tesselation size and efficiency. The proposed method works for arbitrarily shaped 3D objects. The accuracy of pose estimation exceeds that of state-of-the-art algorithms even if the objects are viewed on a cluttered background. Since no high-level feature extraction is required, the algorithm is robust against changing ambient conditions such as illumination. The inspection system is successfully tested on two real-world inspection scenarios in the engine production.

1 Introduction

The aim of the visual inspection system described in this paper is to identify an object and to estimate its pose in order to determine whether it has been correctly assembled in the production process. The inputs to the system are the 3D model of the object and an image of the scene. Numerous approaches have been proposed in the literature for solving such a recognition task. These approaches can be categorized in two ways: a) according to the representation of the data, and b) according to the method for matching.

Data representation schemes can be divided into viewer-centered and object-centered representations (for surveys see [3], [6], [9], [13]). In an object-centered representation all features of an object are described with respect to a coordinate system fixed relative to the object. The advantage of such a representation is that only one model is required to fully describe the object. However, the visibility of object features in images is viewpoint dependent due to self occlusion. Viewer-centered representations implicitly account for self occlusion by representing an object as a set of views taken by a (virtual) camera. The main drawback of a viewer-centered representation is the large number of views required to describe an object.

Once the data is represented in a certain scheme, an appropriate matching strategy has to be chosen. Tree searching [11], graph matching [17], and indexing techniques [8] are well known methods for symbolic matching. They have been

successfully used to recognize objects in arbitrary poses based on pose-invariant features (e.g. angle between two planar surfaces). However, symbolic matching has two major drawbacks: a) there is no systematic way to automatically determine robust, pose-invariant features from a 3D model, and b) high-level feature extraction in image data is difficult.

In addition to symbolic matching, there are other techniques for matching low-level features. Geometric Hashing [16] provides an efficient method for setting up a correspondence between a set of characteristic model points and a set of scene points by multiply encoding the model with respect to various normalizations. However, finding points which describe an object sufficiently well and which can be reliably extracted from range images is difficult. Another method for matching two sets of 3D points is the Iterative Closest Point (ICP) algorithm [4] which is often applied to registration. The ICP algorithm requires a good initialization in order to avoid getting stuck in local minima of the error function.

Concerning applications of such methods, template matching techniques are employed for medical purposes for pose estimation of artificial implants [12]. In the field of industrial quality inspection, robot systems are used for optical measurement (e. g. photogrammetry) purposes based on 2D images and 3D range data of industrial parts [5]. CAD data is used to generate the corresponding 3D models. The described applications, however, primarily involve localization and gauging of the inspected objects rather than pose estimation. In [1] a vision-based automatic assembly unit is presented in which a pose estimation of industrial parts is performed by combining an adapted eigenspace method to obtain an initial estimation and a model-based technique for refinement. This algorithm, however, requires that the objects are put on a uniform background.

In this paper a viewer-centered representation of the image data is chosen. The views are generated automatically from a 3D object model with a virtual camera; edge templates are computed for each view. The difficult trade-off between tesselation constant and accuracy of pose estimation is alleviated by a technique for hierarchical template matching [10]. The described method is applied to two real-world industrial quality inspection tasks.

2 The hierarchical Chamfer matching algorithm

The input image first undergoes an edge detection procedure. A Distance Transform (DT) then converts the segmented binary edge image into a so-called distance image. The distance image encodes the distance in the image plane of each image point to its nearest edge point. If we denote the set of all points in the image as $A = \{a_1, \cdots, a_N\}$ and the set of all edge points as $B = \{b_1, \cdots, b_M\}$ with $B \subseteq A$ then the distance $d(a_n, B)$ for point a_n is given by

$$d(a_n, B) = \min(\|a_n - b_m\|, \forall\, m = 1, \cdots, M) \tag{1}$$

where $\|\cdot\|$ is a norm on the points of A and B (e.g. the Euclidean norm). For numerical simplicity we use the so called chamfer-2-3 metric [2] to approximate the Euclidean metric.

The chamfer distance $D_C(T, B)$ between an edge template consisting of a set of edge points $T = \{t_1, \cdots, t_Q\}$ with $T \subseteq A$ and the input edge image is given by:

$$D_C(T, B) = \frac{1}{Q} \sum_{n=1}^{Q} d(t_n, B) \tag{2}$$

In applications, a template is considered matched at locations where the distance measure ("dissimilarity") $D(T, I)$ is below a user-supplied threshold θ. To reduce false detections, the distance measure was extended to include oriented edges [10].

In order to recognize an object with unknown rotation and translation, a set of transformed templates must be correlated with the distance image. Each template is derived from a certain rotation of the 3D object. In previous work, a uniform tesselation often involved the difficult choice for the value of the tesselation constant. If one chooses a relatively large value, the views that lie "in between" grid points on the viewing sphere will not be properly represented in the regions where the aspect graph is undergoing rapid changes. This will decrease the accuracy of the measured pose angles. On the other hand, if one chooses a relatively small value for the tesselation constant, this will result in a large number of templates to be matched online; matching all these templates sequentially will be computationally intensive and prohibitive to any real-time performance.

Here, the difficult trade-off regarding tesselation constant is alleviated by a technique for hierarchical template matching, introduced in [10]. That technique, designed for DT-based matching, aims to derive a representation off-line which exploits any structure in a particular template distribution, so that, on-line, matching can proceed optimized. This is done by grouping similar templates together and representing them by two entities: a "prototype" template and a distance parameter. When applied recursively, this grouping leads to a template hierarchy. It is built bottom-up, level by level using a partitional clustering algorithm based on simulated annealing.

Online, matching involves traversing the tree structure of templates. Each node corresponds to matching a (prototype) template $\mathbf{p}$ with the image at some particular locations. For the locations where the distance measure between template and image is below a user-supplied threshold θ_p, one computes new interest locations for the children nodes (generated by sampling the local neighborhood with a finer grid) and adds the children nodes to the list of nodes to be processed. For locations where the distance measure is above the threshold, search does not propagate to the sub-tree; it is this pruning capability that brings large efficiency gains. Further details and applications of this algorithm to object detection in outdoor scenes are described in [10].

In our system, we do not need to estimate scale – the distance to the object is known at an accuracy of better than 3 percent due to the fact that the parts are transported on a conveyor belt. Template matching does not have to search all scales explicitly. Hence, the original pose estimation problem of determining

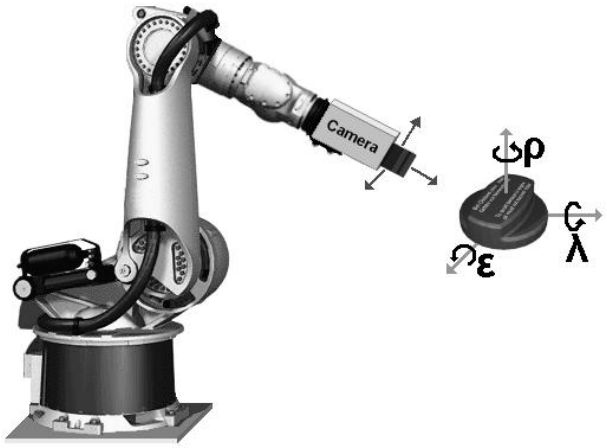

Fig. 1. Sketch of the robot-based inspection system with a definition of the pose angles ϵ (elevation), ρ (rotation), and λ (latitude) in the camera coordinate system.

no.	ρ range	$\Delta\rho$	ϵ range	$\Delta\epsilon$	λ range	$\Delta\lambda$
1	$0° \ldots 180°$	$2°$	$18° \ldots 72°$	$6°$	$-12° \ldots +12°$	$6°$
2	$0° \ldots 20°$	$2°$	$30° \ldots 50°$	$2°$	$-10° \ldots +10°$	$2°$
3	same as 2, but without writing modelled					

Table 1. Properties of the three oil cap template hierarchies (ranges of pose angles ρ, ϵ, λ and tesselations constants in degrees). Hierarchy 1 consists of 4550 templates, hierarchies 2 and 3 of 1331 templates, respectively.

6 degrees of freedom (DOF) can be reduced to a 5 DOF (3 pose angles and 2 image position coordinates) problem.

For pose fine tuning, the pose angles are interpolated between the n_b "best" template matching solutions, with $n_b = 30$ in our system. This is justified because in our pose estimation scenario the dissimilarity values of the 30 best solutions usually do not differ by more than about 20% and thus all these solutions contain a significant amount of information about the pose.

In many applications, templates are generated from real-world image data [7]. For inspection tasks, however, one can assume that a CAD model of the object to be inspected is available. We therefore generate realistic 2D templates from CAD data using the public domain software POVRAY [15], simulating the properties of the surface material and the illumination conditions by employing raytracing techniques. The pose of the object is defined by the three angles ρ (rotation), ϵ (elevation), and λ (latitude), as shown in Fig. 1.

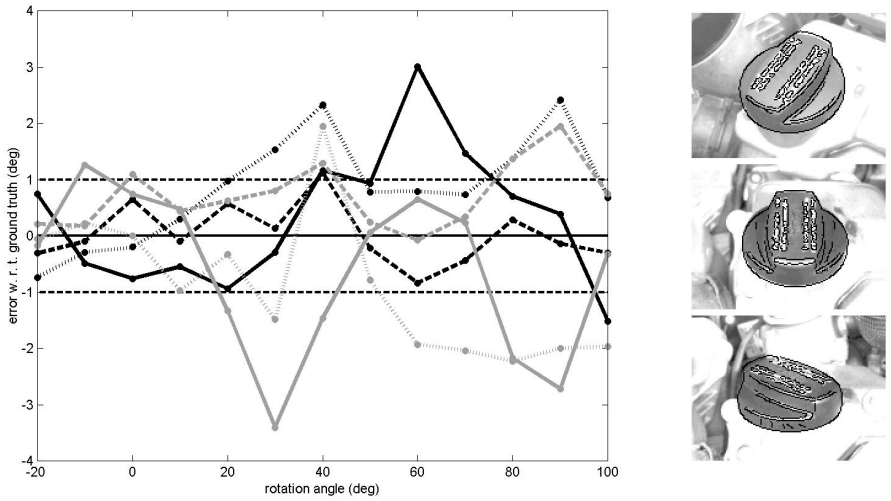

Fig. 2. Left: Deviations of ϵ (solid lines), ρ (dashed lines), and λ (dotted lines) from their ground truth values. Illumination is with cylindric lamp only (black lines) and with both cylindic and halogene lamp (gray lines). The true elevation angle is constantly set to $\epsilon = 70°$. Right: Matching results (best solution) for several poses of the oil cap.

3 Applications

In the oil cap application scenario, we make use of a calibrated robot system. The accuracy of calibration with respect to the world coordinate system is about 0.1° with respect to camera orientation and 0.1 mm with respect to camera position. As the engine itself is not part of the robot system, the relation between world coordinate system and engine coordinate system has to be established separately, which reduces the accuracies stated above by about an order of magnitude.

First, the difference between the measured and the true pose of the correctly assembled oil cap is determined depending on the camera viewpoint and the illumination conditions. The scene is illuminated by a cylindric lamp around the camera lens and a halogene spot. The background of the scene may be rather cluttered. For this examination we use template hierarchy 1 (cf. Table 1). For camera viewpoints with $-10° \leq \rho \leq 10°$ and $50° \leq \epsilon \leq 60°$, the measured pose lies within the calibration accuracy interval of 1° for all three angles. Fig. 2 shows that for $\epsilon = 70°$, this is even true for $-20° \leq \rho \leq 20°$. This implies that from a correspondingly chosen viewpoint, the algorithm is highly sensitive with respect to deviations from the reference pose. Hence, it is possible to determine the pose of the oil cap to an accuracy of about 1°. For comparison, the state-of-the-art technique for pose estimation of industrial parts presented in [1] yields pose errors of about 3° even when the object is put on a uniform background. Significantly changing the illumination conditions by switching off the halogene

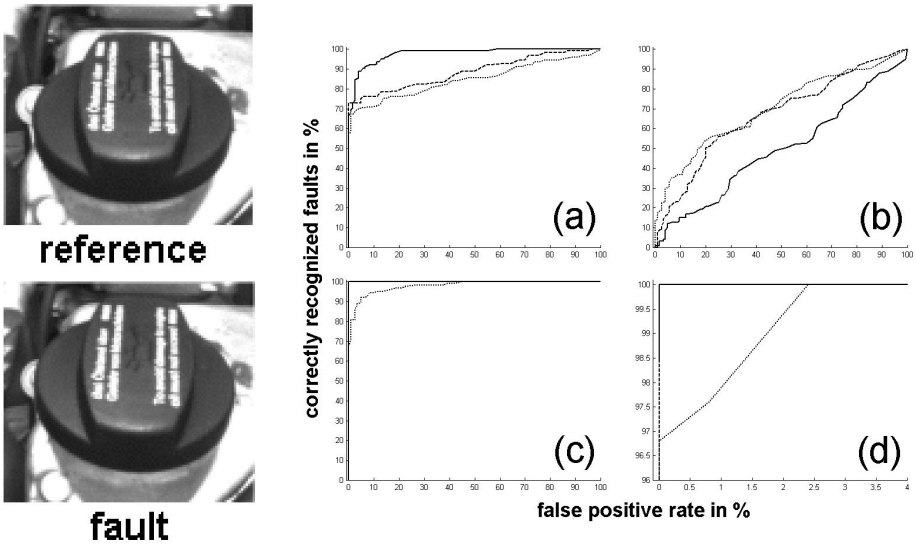

Fig. 3. Left: Reference pose of the oil cap along with the examined typical fault situation. Right: ROC curves of the inspection system, based on measured values of (a) elevation, (b) rotation, (c) latitude, (d) all three pose angles. Solid, dashed, and dotted lines denote three different camera viewpoints. Note that the axis scaling in (d) is different from that in (a)-(c).

lamp does not affect the pose estimation results. The computation time of the system amounts to about 200 ms on a Pentium IV 2.4 GHz processor.

As the described system aims at distinguishing incorrect from correct poses, i. e. performing a corresponding classification of the inspected object, the rate of correctly recognized faults (the rate of incorrectly assembled oil caps which are recognized as such by the inspection system) is determined versus the rate of correctly assembled objects erroneously classified as incorrectly assembled (false positive rate). This representation of the system behaviour is called ROC (receiver operating characteristics) curve. We determine the recognition behaviour of the system for three different camera viewpoints. Here, we will concentrate on a typical fault situation showing angle differences $\Delta\rho = 0°$, $\Delta\epsilon = 2.5°$, $\Delta\lambda = -3.5°$ with respect to the reference pose. In the production environment, the engine and thus the attached oil cap is positioned with a tolerance of about 1 cm with respect to the camera. This random positional inaccuracy was simulated by acquiring 125 different frames of each examined fault situation from 125 camera positions inside a cube of 1 cm size which are equally spaced at 2.5 mm in each coordinate direction. This offset is taken into account appropriately in the pose estimation based on the measured position of the oil cap in the image. As a first step, a fault is assigned based on each of the three angles separately if the corresponding angle deviates from the reference value by more than a given threshold. By varying this threshold, a ROC curve is generated for each angle

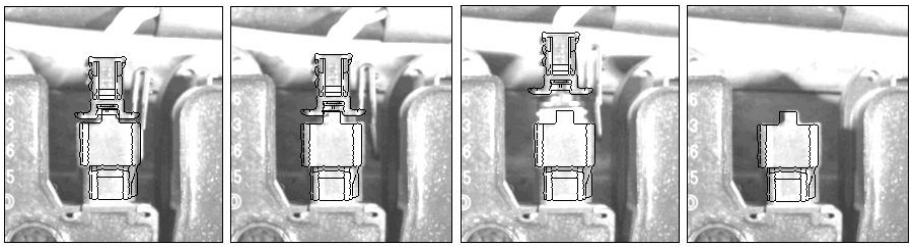

Fig. 4. Ignition plug inspection: Reference configuration (left) and three fault configurations with the corresponding matching results, using two templates. Image scale is 0.2 mm per pixel.

separately as shown in Fig. 3 right, (a)-(c). We then generate a combined ROC curve by assuming that the oil cap is assembled incorrectly if the deviation of at least one of the pose angles is larger than the corresponding threshold. These thresholds are then adjusted such that the area under the ROC curve becomes maximum. This generally yields a ROC curve showing very few misclassifications on the acquired test set, as illustrated in Fig. 3 right, (d). Both with template hierarchy 1 that covers a wide range of pose angles with a large tesselation constant, and with hierarchy 2 that covers with a small tesselation constant only a region on the viewing sphere close to the reference view (cf. Table 1), very high recognition rates close to 100 percent can be achieved. With hierarchy 3, which is identical to hierarchy 2 except that the writing on top of the oil cap has been omitted, the performance decreases, but not significantly: At a false positive rate of 0%, still a rate of correctly recognized faults of 98.4% is achieved.

For inspection of the ignition plug, we regard in addition to the reference configuration three fault configurations: The clip is not fixed, the plug is loose, and the plug is missing (Fig. 4). The connector and the plug are modelled as two separate objects such that the offset of the plug in y direction can be used to distinguish fault configurations from the reference configuration. The matching results in Fig. 4 show that the y position of the plug relative to the connector can be determined an accuracy of about 0.5 mm, which is sufficient to faithfully distinguish correctly from incorrectly assembled ignition plugs.

4 Conclusion

This paper presented a system for industrial quality inspection based on an approach to object recognition and pose estimation by two-dimensional edge-based template matching, with templates generated from CAD data. The behaviour of the system was robust, without sacrificing for efficiency. The usual high cost of template matching was dramatically reduced by a hierarchical edge matching scheme based on distance transforms, allowing close to real-time performance. Compared to similar techniques used for template matching in intenstiy images, this method has the main advantage that even significant changes in the illumination need not be considered in the matching process as long as the object

contours are perceivable. In contrast to most state-of-the-art techniques for pose estimation, no initialization of the algorithm with an approximate pose known a-priori is necessary.

We have regarded two real-world industrial inspection scenarios in the context of engine production. In the first scenario, the system recognizes if an oil cap has been correctly assembled to the engine. The second scenario deals with the inspection of an ignition plug. Our results are encouraging and suggest that the system is suitable for a wide variety of further inspection tasks.

References

1. G. Bachler, M. Berger, R. Röhrer, S. Scherer, A. Pinz. A Vision Driven Automatic Assembly Unit. *Proc. International Conference on Computer Analysis of Images and Patterns*, pages 375-382, Ljubljana, Slovenia, 1999.
2. H. Barrow. Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proc. International Joint Conference on Artificial Intelligence*, pages 659–663, 1977.
3. P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, 1985.
4. P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
5. C. Brenner, J. Böhm, J. Gühring. An experimental measurement system for industrial inspection of 3D parts. *Photonics Fast, Intelligent Systems and Advanced Manufacturing (ISAM)*, vol. 3521, *SPIE*, Boston, 1998.
6. R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. *Computing Surveys*, 18(1):67–108, 1986.
7. C. Demant. Industrial Image Processing. Springer-Verlag, Berlin, 1999.
8. P. J. Flynn and A. K. Jain. 3d object recognition using invariant feature indexing of interpretation tables. *CVGIP: Image Understanding*, 55(2):119–129, 1992.
9. P. J. Flynn and A. K. Jain. Three-dimensional object recognition. In T. Y. Young, editor, *Handbook of Pattern Recognition and Image Processing: Computer Vision*, pages 497–541. Academic Press, 1994.
10. D. Gavrila and V. Philomin. Real-time Object Detection for "Smart" Vehicles. In *Proc. International Conference on Computer Vision*, pages 87–93, Kerkyra, 1999.
11. W. E. L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artificial Intelligence*, 44(1–2):121–165, 1990.
12. W. A. Hoff, R. D. Komistek, D. A. Dennis, S. Walker, E. Northcut, K. Spargo. Pose Estimation of Artificial Knee Implants in Fluoroscopy Images Using a Template Matching Technique. *Proc. of the 3rd IEEE Workshop on Applications of Computer Vision*, pages 181–186, Sarasota, Florida, 1996.
13. X. Jiang and H. Bunke. *Dreidimensionales Computersehen.* Springer-Verlag, Berlin, 1997.
14. J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.
15. POV-Ray: The Persistence of Vision Raytracer, http://www.povray.org
16. H. J. Wolfson and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.
17. E. K. Wong. Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3):287–303, 1992.