

Feature and Viewpoint Selection for Industrial Car Assembly

Dirk Stöbel¹, Marc Hanheide¹, Gerhard Sagerer¹, Lars Krüger², and Marc Ellenrieder²

¹ Faculty of Technology, Bielefeld University
P.O. Box 100131, 33501 Bielefeld, Germany

{dstoesse, mhanheid, sagerer}@techfak.uni-bielefeld.de

² Research and Technology, DaimlerChrysler AG
Wilhelm-Runge-Str. 11, D-89081 Ulm, Germany

{lars.krueger, marc.ellenrieder}@daimlerchrysler.com

Abstract. Quality assurance programs of today’s car manufacturers show increasing demand for automated visual inspection tasks. A typical example is just-in-time checking of assemblies along production lines. Since high throughput must be achieved, object recognition and pose estimation heavily rely on offline preprocessing stages of available CAD data. In this paper, we propose a complete, universal framework for CAD model feature extraction and entropy index based viewpoint selection that is developed in cooperation with a major german car manufacturer.

1 Introduction

Quality assurance and final inspection are fundamental steps in production work flow. Automated visual inspection of assemblies is therefore in the focus of recent research (cf. [8], [6], [9] and [5]). Because CAD data of the assembled parts must be available for construction processes, model-based object recognition and pose estimation are eligible methods to allow automated visual inspection.

Real-time production processes dictate the need for fast and accurate online algorithms. The framework we propose hence transfers as much of the algorithmic effort as possible to an *offline* preprocessing stage, yielding very fast and accurate *online* visual inspection. Our framework is based on a new generalized definition of features that supports the incorporation of different feature types under a common layer of abstraction.

Besides the efficient online application, the selection of appropriate camera viewpoints is fundamental to robust visual inspection of assemblies. Our framework therefore also predicts viewpoints which optimally separate different expected assembly configurations of valid and invalid mounting scenarios.

The article is structured as follows: In Section 2, we propose a generalized definition of features for model-based object recognition and pose estimation. It will be shown how the framework models rigid objects and flexible collections of objects. In Section 3, we will discuss how to accurately predict occlusions by applying a mixture of rule-based lookups and bounding volumes intersection tests. Section 4 then addresses the calculation of optimal camera viewpoints using 3D to 2D projection pursuit with collective entropy index. Finally, Section 5

details the framework’s performance in feature extraction, occlusion prediction and object recognition.

2 Characteristic Localized Features

The framework proposed in this paper is a preprocessing stage suited for model-driven 3D/2D object recognition and pose estimation algorithms like the ones introduced by Lowe [7] and Araújo et al. [1]. In general, they use an initial object pose estimate to project features of a given 3D model on the camera view plane. Afterwards, they iteratively obtain improved estimates by matching the projected features with features extracted from real world images.

Object recognition algorithms generally require features that are highly *characteristic*. For pose estimation, features have to be *localized* (must have a spatial position) in the model and image domain. Thus, our framework must automatically extract *Characteristic Localized Features (CLFs)*. In order to be suitable for any 3D/2D object recognition scheme, each CLF must at least meet the following set of requirements:

1. **Projection:** CLFs are spatially represented in 3D. To allow for 2D comparison, CLFs must be projected on a camera view plane, given a camera model and an estimated pose. An appropriate projection prescript has to be defined for every type of CLF.
2. **Visibility determination:** Since CLFs can become occluded under 2D projections, their visibility has to be determinable for any given view. CLFs that are visible are called *active*.
3. **Visual Appearance:** Projected CLFs are compared to image features. Therefore, 2D projection must imply some visual outcome recognizable in real world images. E.g., in case of edges, the visual appearance would typically be a strong local image gradient perpendicular to the edge direction.

These requirements form a unique layer of abstraction that enables the proposed framework to perform all tasks without incorporating any further knowledge about feature types.

Good CLFs are reliably trackable features in image sequences, as presented by Shi and Tomasi [11] or Schmid et al. [10]. Since they have been empirically shown to be appropriate, edges are commonly used (cf. [6]). We chose *contour edges*, i.e. edges that potentially form the object’s outline, to explain our approach in the following. Additionally, the framework incorporates functionality to deal with localized color and texture features.

Edges which possibly form the contour of an object are interesting CLF candidates because the object’s silhouette is always formed by a subset of contour edges. The silhouette will usually appear in real world images as intensity gradients. What is more, Kettner and Welzl [4] provided empirical evidence that the number of contour edges in a 3D model is usually much smaller than the total number of edges. The framework determines the set E_c of a model’s potential contour edges by analyzing the angle between all its adjacent triangles:

$$E_c = \{E | isconvex(E) \wedge \alpha_E = \angle(\mathbf{N}_E^1, \mathbf{N}_E^2) > 0\} \quad (1)$$

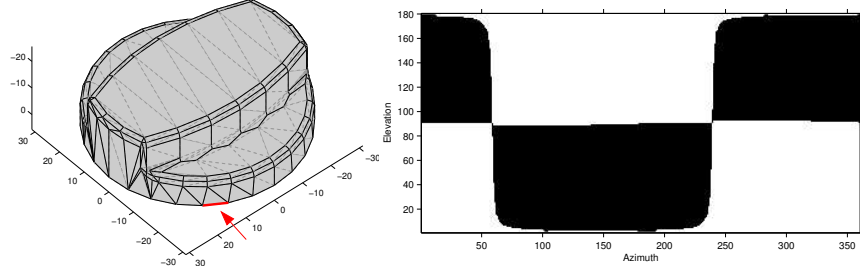


Fig. 1. *Left:* Automatically extracted CLFs (axis units in mm). Model edges are displayed as thin dashed lines, extracted CLFs as thick black ones. *Right:* Visibility map of the CLF highlighted on the left side. Black denotes view angles under which the CLF is active. Axis units denote the view angle measured in degrees.

where N_E^1, N_E^2 represent the normals of two adjacent triangles and E the edge shared by the triangles. The angle α_E allows to assign a score to each element of E_c , because a more acute angle yields a more frequent appearance of the edge under different projections.

All elements of E_c with a certain minimum score are new contour edge CLFs. To meet requirement 2., the visibility of the edge elements is pre-calculated relative to all possible discrete view-angles and stored in separate run-length-encoded *visibility maps*. An example of automatically generated contour edge CLFs and a particular visibility map is displayed in Fig. 1.

Based on the specification of CLFs, a (*basic*) *model* can be defined as a set of CLFs referring to the same rigid object and object coordinate system. Furthermore, an *aggregation* can be described as a tree in which the root node represents the aggregation’s pose with respect to the world coordinate system. Each sub-node represents a basic model and the model’s *pose* (6DOF) relative to the parent node.

3 Occlusion Prediction

Inferring aggregation poses from real world images by means of 3D/2D object recognition schemes always involves the projection of the aggregation features on 2D camera planes. Regarding our framework, the projection of CLFs belonging to an aggregation might result in inactive (occluded) CLFs. Fig. 2 shows that any CLF might either become occluded by parts of the basic model it is attached to or by other basic models of the aggregation. The former occlusion type will be termed *intramodel occlusion*, the latter *intermodel occlusion*.

Automated inspection in car industry requires fast online occlusion prediction. Intermodel occlusions are correctly predicted by lookup operations in the visibility maps. In the worst case, these maps consume space in the order of $O(c \cdot v)$, with c denoting the number of CLFs and v referring to the number of scanned view angles during map calculation. The lookup operation has efficient constant time complexity per call.

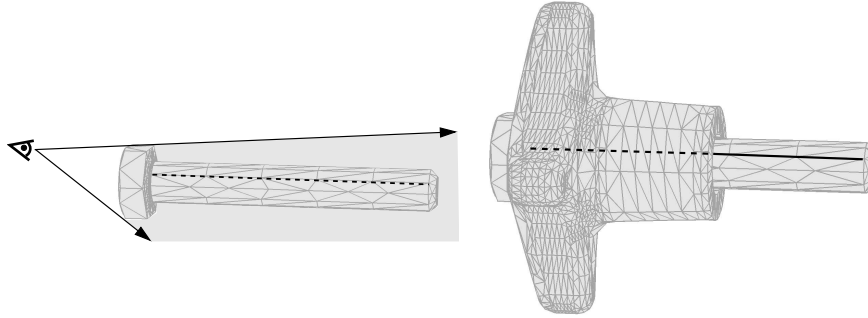


Fig. 2. The two occlusion types occurring with aggregations. *Left:* Intramodel occlusion. A contour edge CLF along the bolt’s thread (dashed black line) is hidden behind the same bolt’s head. *Right:* Intermodel occlusion. The same contour edge CLF, partly occluded (dashed black line) by a knob.

Extending the lookup strategy to aggregations would require to pre-calculate the visibility maps for all CLFs attached to every possible aggregation configuration. This would lead to combinatorial explosion of storage space consumption. Therefore, intermodel occlusion prediction is based on tightly wrapping each aggregated model in a small number of simple geometric bounding volumes such as boxes or spheres. Our framework performs this task offline during aggregation creation. The online part of occlusion prediction first checks the pre-calculated visibility maps. For each visible candidate, view-rays between a virtual camera and points on the candidate CLF are tested for intersection with each bounding volume, thus ruling out features that are (partially) hidden behind parts of the aggregation. The intersection tests have a reasonable worst case time complexity of $O(c_v \cdot b)$, with b denoting the number of bounding volumes and c_v the number of CLFs passing the visibility map test.

4 Viewpoint Selection

In order to support robust recognition, a further task of the framework is to determine those viewpoints from which an assembly might be inspected best. In this context, Vázquez et al. [13] proposed the information theoretic measure *viewpoint entropy*. It expresses the amount of information conveyed in a certain scene that is being watched from a given point.

Measures like viewpoint entropy are often based on the *visual appearance* of a *specific* feature. Though we use an entropy measure, too, the CLF abstraction enables us to estimate the underlying probability distributions from the *location* of a *variety* of features. The entropy measure employed here was recently introduced as a class separability index [12] and is called *collective entropy*. It estimates the quality of a view by measuring how *distinguishable* aggregation configurations will be under projection onto a given camera plane. An example with two configurations is shown in Fig. 3.

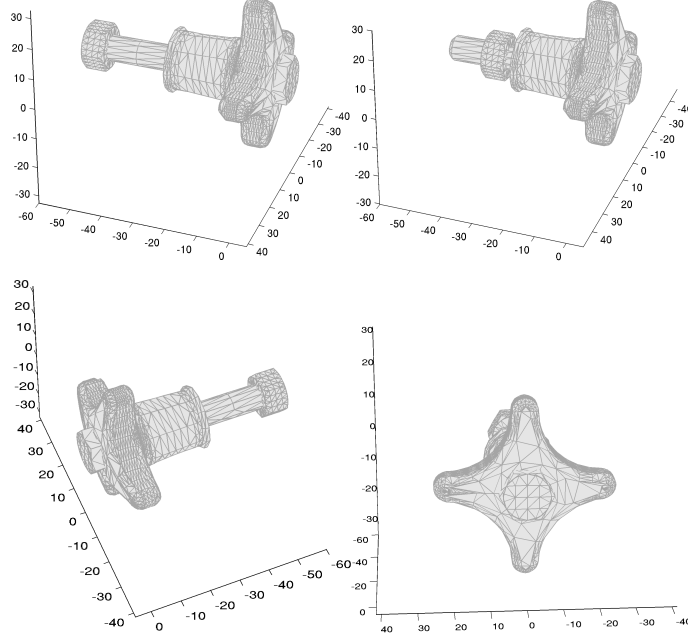


Fig. 3. *Top:* A knob, screw and nut aggregation in configurations typical for invalid (left) and valid (right) mounting. *Bottom:* The good quality view (left) allows good distinction between different nut positions. In the bad quality view (right), the nut position is hard to infer as large parts of it are hidden behind the knob.

Generally, collective entropy describes how well measurements in Cartesian space, each belonging to a distinct class, might be separable from each other with respect to the class labels. It is calculated by partitioning the N measurements m_i into d -dimensional cells with hyper-cuboid topology:

$$m_i = (m_{i_1}, \dots, m_{i_d}) \in \mathbb{R}^d, \quad i = 1, \dots, N \quad (2)$$

$$R_j = [\min m_{i_j}, \max m_{i_j}], \quad 1 \leq j \leq d, \quad i = 1, \dots, N. \quad (3)$$

The faces of the hyper-cuboid cells are constructed by dividing each range of values R_j into B parts of equal length. An initial cell resolution is chosen and the m_i are partitioned accordingly. Afterwards, one obtains the conditional entropy which Cover and Thomas [2] define as

$$H(X|Z) = - \sum_{z \in Z} p(z) \cdot \sum_{x \in X} p(x|z) \cdot \log_2 p(x|z). \quad (4)$$

where each $z \in Z$ is a non empty hyper-cuboid cell and $x \in X$ is the set of measurement class labels. Thus, $H(X|Z)$ indicates how uniformly distributed the measurements are, given a certain partitioning resolution. However, $H(X|Z)$ is not robust against the shifting of cell borders. Singh [12] therefore repeatedly

lowers the cell resolution and recalculates the conditional entropy until a minimum resolution is reached. Collective entropy is then taken as the area under the curve of the conditional entropy values with respect to cell resolution.

Viewpoint selection iteratively places a virtual camera at discrete view angles in an orbit around an aggregation. For each iteration and for each expected configuration, the positions of visible CLFs are projected to the camera plane. Afterwards, the probability distributions in (4) are obtained by Monte Carlo sampling from the CLF location domain. The complete scheme can be regarded as 3D to 2D projection pursuit with collective entropy as projection pursuit index (cf. [3]). To our knowledge, it has not been tried before. The process yields a map that indexes the degree to which any discrete view angle conveys separable information about the observed scene. Some results are shown in Fig. 3.

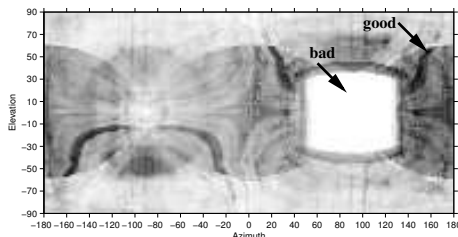


Fig. 4. Complete map of collective entropy indices. Dark areas denote high quality view angles, light areas indicate bad quality (all axis units in degrees). The arrows point to the map positions corresponding to the bottom two views in Fig. 3.

5 Performance

During object recognition, the step inducing the highest computational load is the 3D to 2D projection of features because it involves online occlusion prediction. Therefore, we evaluated the performance of our online algorithm in the following way: First, we chose an evaluation candidate out of a set of aggregations with varying complexity. Single basic models with a total number of less than 1000 CLFs were considered to be of low complexity. In contrast to this, aggregations of more than two basic models with a total number of more than 2000 CLFs were considered to be of high complexity. Each candidate was randomly rotated in 3D and online occlusion prediction carried out in 1000 runs. We then calculated the average execution times which are visualized in Fig. 5). It shows that even for the most complex aggregation the algorithm executes in less than 12ms. The execution time scales in average approximately linear to the total number of CLFs.

To ensure that the results of our automated feature selection are suited for model-based object recognition, we first determined the average amount of active CLFs similar to the above evaluation scheme. The results are listed in Table 1. The average amount of active CLFs is well balanced for the first three objects in Table 1 and rather low for the "oil lid", "knob" and the assembly of "knob", "bolt" and "flat washer", indicating that their CLF sets should be compressed.

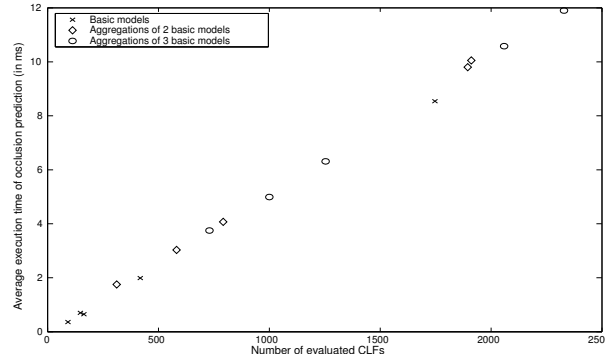


Fig. 5. Performance of occlusion prediction on a Pentium 4 PC (2GHz, 512MByte).

Object recognition and pose estimation performance was evaluated with an industrial system (cf. [6]). A standard camera with 320x240 resolution was moved around a mounted oil lid at constant speed and a distance of approx. 70mm, recording 420 images. The object pose was calculated for each image. The average and standard deviation of relative (i.e. image-to-image) and absolute accuracy are given in Table 2. Note that the average error of parameter estimation relative to the distance of the camera is always smaller than 1%. Thus, model-based pose estimation meets the strong accuracy requirements of car industry.

Table 1. Average number of active CLFs compared to their total number.

object	total no. CLFs	avg. act. CLFs
nut	92 (100%)	13.6 (14.8%)
bolt	148 (100%)	28.0 (18.9%)
flat washer	164 (100%)	37.2 (22.7%)
oil lid	418 (100%)	32.7 (7.8%)
knob	1747 (100%)	74.0 (4.2%)
assembly	2059 (100%)	112.4 (5.5%)

Table 2. Average and standard deviation of relative and absolute pose estimation accuracy.

DOF	$\mu_{relative}$	$\sigma_{relative}$	$\mu_{absolute}$	$\sigma_{absolute}$
x [mm]	-0.15	0.53	-0.31	0.53
y [mm]	-0.002	0.4	-0.81	0.73
z [mm]	-0.007	4.76	-2.6	3.9
roll [°]	0.006	1.27	0.578	1.4
pitch [°]	0.097	0.9	-0.513	1.57
yaw [°]	0.068	0.84	-0.45	1.8

6 Conclusion

We presented a complete, universal framework for automated selection of features and viewpoints for model-based visual inspection that was developed in cooperation with the DaimlerChrysler AG. Given CAD data of real world objects, the

framework extracts characteristic features and prepares them for fast and robust occlusion prediction. It further determines high quality viewpoints to inspect an assembly from.

Our feature extraction approach has been demonstrated for contour edge features. Performance results for the offline model preparation were given accordingly. For online occlusion prediction, execution time in the average case scaled approximately linear to the amount of processed features. The tests have been carried out on CAD models of car production assemblies and standard industrial fixation elements.

The underlying concepts for occlusion prediction and viewpoint selection are not restricted to contour edges, but can also be used for a wide selection of other kinds of 3D localized features which meet the CLF requirements. The proposed framework is thus based on a novel layer of abstraction for features in general. It was successfully tested with an industrial object recognition system.

References

1. H. Araújo, H. L. Carceroni, and C. M. Brown. A fully projective formulation to improve the accuracy of Lowe's pose-estimation algorithm. *Computer Vision and Image Understanding*, 70(2):227–238, 1998.
2. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley series in telecommunications. Wiley-Interscience, 1991.
3. J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. on Computers*, C-23(9):881–889, 1974.
4. L. Kettner and E. Welzl. Contour edge analysis for polyhedron projection. In *Geometric Modelling: Theory and Practice*, pages 379–394. Springer, 1997.
5. K. Khawaja, A. Maciejewski, D. Tretter, and C. Bouman. A Multiscale Assembly Inspection Algorithm. *IEEE Robotics & Automation Magazine*, 3(2):15–22, 1996.
6. Thorsten Kölzow. *System zur Klassifikation und Lokalisation von 3D-Objekten durch Anpassung vereinheitlichter Merkmale in Bildfolgen*. PhD thesis, Bielefeld University, 2002. in german.
7. D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.
8. J. Miura and Katsushi Ikeuchi. Task planning of assembly of flexible objects and vision-based verification. *Robotica*, 16:297–307, 1998.
9. J. Noble. From inspection to process understanding and monitoring: a view on computer vision in manufacturing. *Image and Vision Computing*, 13(3):197–214, 1995.
10. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.
11. Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 593–600, 1994.
12. S. Singh. PRISM - a novel framework for pattern recognition. *Pattern Analysis and Applications*, 6(2):134–149, 2003.
13. P. P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H. P. Seidel, editors, *Vision, Modeling, and Visualization 2001*, pages 273–280, 2001.