Pose Estimation of Known Objects by Efficient Silhouette Matching

Christian Reinbacher, Matthias Rüther and Horst Bischof
Institute for Computer Vision and Graphics
Graz University of Technology
Graz, Austria
{*reinbacher,ruether,bischof*}*@icg.tugraz.at*

***Abstract*—Pose estimation is essential for automated handling of objects. In many computer vision applications only the object silhouettes can be acquired reliably, because untextured or slightly transparent objects do not allow for other features. We propose a pose estimation method for known objects, based on hierarchical silhouette matching and unsupervised clustering. The search hierarchy is created by an unsupervised clustering scheme, which makes the method less sensitive to parametrization, and still exploits spatial neighborhood for efficient hierarchy generation. Our evaluation shows a decrease in matching time of 80% compared to an exhaustive matching and scalability to large models.**

***Keywords*-component; formatting; style; styling;**

I. Introduction

The knowledge of exact position and orientation of objects is essential for many applications like sorting and packaging of objects or robotic pick and place. Pose estimation is a hard task for various reasons. The pose of an object with respect to a camera is specified by six degrees of freedom (DOF) which yields a very large search space. Additional difficulties arise from partial occlusion, symmetric objects, shiny surfaces and untextured or slightly transparent objects.

The problem of estimating the pose of a known object from a single image has been extensively studied. View-based approaches [1], [2], [3] have been popular for a time. They determine the object pose by comparing the query image with precomputed 2D reference views of a known 3D model. The views have to cover the entire search space, so these methods are rather slow and unsuitable for applications. Generally, the 6 DOF are reduced to 3 DOF by normalizing the object position, which eliminates translation. The reference views are then created by placing a virtual camera on a sphere around the object. The resulting DOF introduced by the possible rotations still span a large search space, so the accuracy of view-based approaches is directly related to the sampling density of the pose range.

Local, iterative approaches form the second large group of pose estimation methods. Here, pose estimation is considered as an optimization problem where some model has to be fitted to the query image. Rosenhahn et al. [4] compared several iterative shape matching approaches for their applicability to pose estimation. Subsequent work of this research group covered pose estimation of free-form objects as well as an extension to pose tracking [5]. Though these models work very well for a variety of tasks, they are likely to get stuck in local minima without a good initialization.

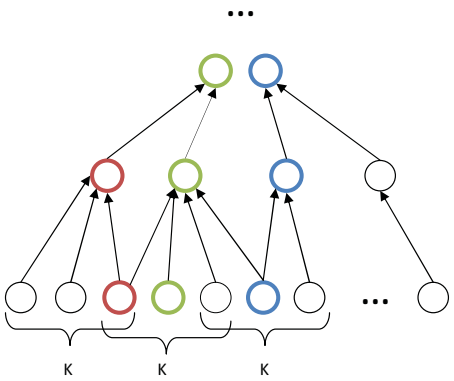

Figure 1. Organization of the 2D reference views in a hierarchical structure.

The only option to avoid such local minima is to perform an exhaustive search over all precomputed views. This is not feasible for a real-time application. To circumvent this problem, Gavrila [6] introduced a hierarchical shape matching scheme and applied it to the task of pedestrian detection and pose estimation. However, he requires a pairwise similarity for all elements in the set of reference views. The approach shows quadratic behavior in terms of runtime and memory consumption. Ulrich et al. [7] reduce the quadratic part of this problem by building the hierarchy from bottom up, merging neighboring views first (the neighborhood is defined in spherical coordinates).

A. Contribution

The main contribution of this work is a pose estimation method based on matching of object silhouettes, accelerated by a hierarchical clustering scheme. Like in [7], we do not require pairwise similarities between all views. Our hierarchical clustering approach uses an unsupervised clustering method and a pairwise similarity measure. The set of reference views is automatically created from a 3D model. The method relies solely on the object silhouettes which can be robustly extracted for a majority of objects even when

they are smooth, untextured and slightly transparent, and can be efficiently pregenerated even for complex meshes, using state of the art graphics hardware.

The approach of Ulrich et al. [7] is closest related to the approach proposed in this work. In order to speed up the matching process, they organize the reference views in a hierarchical manner. They start by merging neighboring views until the similarity exceeds a threshold. Then they repeat this process for all views of the current hierarchy. The identified clusters build the next hierarchy level. Once a level contains only a few views, the clustering algorithm is stopped. In contrast to them, we allow to take larger neighborhoods into account for creating the hierarchy levels. A larger neighbourhood yields more information which can be used to identify meaningful cluster centers. Recent works [8], [9] show that the exploitation of inter class variabilities is an improvement compared to simple pairwise similarity measures. We soften the influence of the spatial neighborhood by relaxing assumptions on spatial proximity in higher hierarchy levels.

II. Method

The proposed method consists of two phases. In the offline phase a set of reference views is pregenerated based on the 3D model of the object. These views are organized in a hierarchical structure. In the online phase the contour of the object is matched against the reference views by efficiently traversing the hierarchy created in the offline phase.

A. Creation of Reference Views

To speed up the matching process, we seek to reduce the number of reference views which have to be compared to the query image. This is done by introducing a hierarchy which conflates similar views and disperses dissimilar ones. In the first step we create the complete set of reference views by placing virtual cameras at uniformly spaced viewpoints on a sphere. We use a method for regular sampling of spheres and other rotation groups proposed by Mitchell [10]. The object is located in the center of this sphere. The cameras are simulated using OpenGL®, the CAD model is represented by a triangle mesh. One reference view consists of a set S_i of outer object contour points $S_i = \{\mathbf{x}_{i,1} \dots \mathbf{x}_{i,n_i}\}$ and the pose parameters in spherical coordinates (α, β), $V_i = \{S_i, \alpha_i, \beta_i\}$.

When observing real-world objects one notices that they look rather similar when a small viewpoint change occurs (see Figures 2 (a)-(d)). To make use of this property we look at a $k \times k$ neighborhood in spherical space, and try to reduce it to a few representative reference views. The neighbors are hereby compared to each other using a suitable similarity metric, which results in an affinity matrix $A_{k^2 \times k^2}$. As the pairwise similarity measure, we use the *Normalized Cross Correlation* to compare two contours. The contour points are stored in polar coordinates around the barycenter. The contour is sampled at uniformly spaced angles which results in sets of contour points of equal length.

An unsupervised clustering scheme then identifies prototypes $P_{m,i}$ with $1 \leq m \leq k^2$ based on A. Each view is assigned one prototype which we further call parent. Any scheme which is able to cluster entities based on their affinity can be used for this process. We decided to use *Affinity Propagation* (AP) proposed by Frey and Dueck [8] because it has shown state of the art performance for a variety of unsupervised clustering tasks [11] and needs no parametrization. Ideally k would be chosen to include the whole set of reference views. This would allow AP to exploit all inter class variabilities and choose an optimal set of cluster prototypes. For a typical number of 5000 to 10000 views this would not be feasible in terms of memory requirements and computation time.

Even if we make k small, the application of the above algorithm to every $k \times k$ neighborhood would be infeasible because every view would have to be considered k^4 times. To speed up this process we only allow a slight overlap between consecutive neighborhoods. The overlap o can be chosen in the range $[0, k-1]$ where 0 means no overlap at all and $k-1$ results in every entry being used as neighborhood center. After processing all reference views, we arrive at a set of prototypes P. They define the second level of our hierarchy. At higher levels the neighborhood can no longer be defined by spatial proximity of the prototypes themselves. Instead, two prototypes are neighbors if the neighbors of their children share the same parents $\mathrm{Neighbors}(i) = \mathrm{Parents}(\mathrm{Neighbors}(\mathrm{Children}(i)))$. Applying the same clustering method again, new hierarchy levels are created until all remaining prototypes are neighbors of each other. Note that the hierarchy is not necessarily a tree. Due to the partial overlap of neighborhoods, each node may have several parents. Figure 1 outlines the hierarchical clustering method. The colored nodes are propagated as prototypes to the next hierarchy level.

B. Querying Reference Views

The 3D pose of the object with respect to the camera is obtained by the pose of the closest match in the set of reference views. In the first step the object contour $S = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ is compared to all elements of the hierarchy top level. We applied two different methods of traversing through the hierarchy, namely best-first search and beam search. The apparent drawback of best-first search is that the search space is pruned too early, discarding possible good matches. It can not be guaranteed that the children of one node have a higher similarity to the query image than the node itself. To partly compensate for that issue we propose to use beam search. We start with a low threshold and increase it for every level of the hierarchy. One has to potentially look at more views but this also increases the probability to find the correct match.

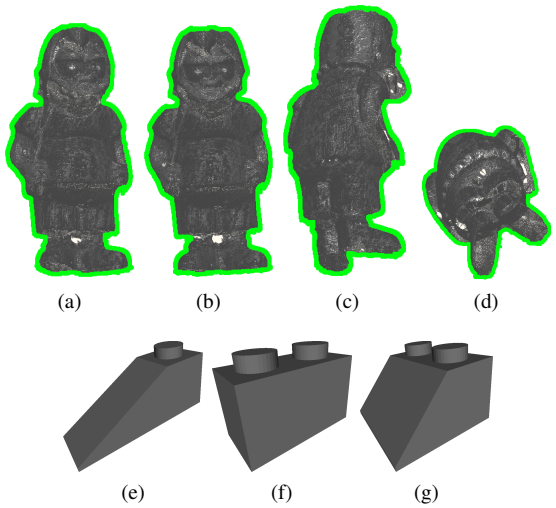

Figure 2. Objects we used for the experimental evaluation. These renderings were created from the CAD model of the objects.

Table I
RESULTS FOR FINDING AN OPTIMAL PARAMETER SET FOR 5000 REFERENCE VIEWS.

k	o	Comparisons	Correct
3	0	2486	75%
3	1	2256	86%
5	0	925	67%
5	**1**	**1126**	**76%**
7	0	730	52%
7	1	853	64%

Table II
RESULTS FOR THE SYNTHETIC EXPERIMENTS WITH A FIXED PARAMETER SET OF $k = 5$ AND $o = 1$ AND A SET OF 5000 REFERENCE VIEWS.

Model	Comparisons	Difference to Global Optimum
Block 1	925	1.4%
Block 2	613	5.6%
Block 3	1266	0.4%
Nelson	1148	3.67%

III. EXPERIMENTS

In order to evaluate our pose estimation method we use four different objects: a toy figure with rather high mesh complexity of 180.000 triangles, and three building block models, resembling typical CAD-constructed objects. Renderings are shown in Figure 2. The toy figure features a smooth surface with few edges whereas the building blocks consist only of a few faces with rather distinct edges. The toy figure was scanned, the models for the building blocks were handcrafted.

A. Synthetic Experiments

In order to determine the best parameters for our algorithm we first set up an experiment with synthetic data, for which reliable ground-truth is available. For each model described in the previous section, a set of 5000 reference views is generated, covering the entire view sphere: $0 \leq \alpha < 2\pi$, $0 \leq \beta < 2\pi$. Another set of 100 query views is created at random, evenly distributed locations on the view sphere. The translation and scale of the object are kept fix so only the rotation parameters have to be determined. The camera model is assumed to be represented by an orthogonal projection. The goal of this experiment is to find a tradeoff between the number of database entries compared to the query image and the number of correct matches found by adjusting the parameters of the offline and online phase. A good parameter setting greatly speeds up the online database search.

Table I shows the results of this experiment. A match is considered correct if a linear search over all views yields the same database entry. For a view database size of 5000 we found an optimal parameter set $k = 5$ and $o = 1$, which resulted in approximately 1000 comparisons per matching process. Regarding the complexity of our matching hierarchy, the simple building blocks produced 4 levels with 60 views in the top hierarchy level, and the Nelson model produced 5 levels with 30 views in the top level.

In our next experiment we evaluate the pose estimation accuracy and matching time for different objects. For each query view we determine the best match by a linear search, always giving the global optimum, and the hierarchical search method. Results are given in Table II for the number of pairwise comparisons necessary to get the result. The relative difference between the similarities is defined as

$$e = \frac{|e_{best} - e_{match}|}{e_{best}} \quad , \tag{1}$$

where e_{best} is the similarity to the overall best match determined by an exhaustive search over all reference views, and e_{match} is the similarity to the hierarchy based match result.

B. Pose Estimation

Experiments on real images are conducted using the same set of reference images as before. A total of 24 query images were acquired and matched to the reference views. Images of the objects were taken using a metrically calibrated camera setup. The range of possible poses was set to cover the whole viewing sphere. The views are organized in a hierarchical structure following the algorithm presented in Section II-A. The results of these experiments are given in Table III. It shows the number of reference views which had to be compared to the input image in the online phase and the difference to the globally optimal match given by (1).

We further compared our method to the approach of Ulrich et al. which is commercially available in the software

Table III
RESULTS FOR THE EXPERIMENTS WITH REAL IMAGES WITH A FIXED PARAMETER SET OF $k = 5$ AND $o = 1$ AND A SET OF 5000 REFERENCE VIEWS.

Model	Comparisons	Difference to Global Optimum
Block 1	1554	2.48%
Block 2	532	6.1%
Block 3	1285	0.28%
Nelson	720	3.12%

Table IV
RESULTS OF THE TESTS WITH REAL-WORLD IMAGES. DUE TO THE LACK OF GROUND TRUTH INFORMATION THE POSE ESTIMATIONS WERE INSPECTED VISUALLY.

	Our Method		Ulrich et al. [7]		
Model	**Corr.**	**Miss**	**Corr.**	**Miss**	**Fail**
Block 1	5	1	6	0	0
Block 2	3	1	3	1	0
Block 3	6	0	5	1	0
Nelson	7	1	0	0	8
Sum	**21**	**3**	**14**	**2**	**8**

package Halcon. The lack of a ground truth for the established poses only allow a visual inspection as verification. Table IV gives an overview of the results. We distinguish the three cases *correct*, *missed* and *failed* where *missed* means that the pose estimation was not correct and *failed*, that the object was not detected. Our method assumes that an object is present in the image which means that the state *fail* can not occur. Considering only the outer contour many objects have ambiguities i.e the building blocks look similar when viewed from above or below. With the limited data available we count the flipped estimates as correct matches. Besides the number of correct estimates we collected the number of views which had to be compared to the input image to obtain the pose. On average 20% of the database entries were compared to the image during the database search.

IV. CONCLUSION

We presented a hierarchical, view-based pose estimation method based on affinity propagation for efficient clustering and hierarchical organization of the search space. Computational complexity in creating the hierarchy is reduced by exploiting spatial vicinity and can be used with virtually any pairwise similarity measure. The 3D models can be of arbitrary complexity, as long as they can be handled by state of the art graphics hardware.

Experimental results show a reduction in matching time between 70% and 88%, compared to a naive linear search. Accuracy of the pose estimation method is harder to quantify, because there always exist symmetry axes. Even a slight mismatch in the shape similarity produces a large rotational offset in object space. Regarding shape similarity, we were always able to get reasonable matching results, differing by less than 6% form the best match. This way, using more than one view for pose estimation might give considerably better results in terms of rotational accuracy in object space.

ACKNOWLEDGEMENT

This work was supported by the Austrian Research Promotion Agency (FFG) under the project SILHOUETTE (825843).

REFERENCES

[1] J. Byne and J. Anderson, "A CAD-based computer vision system," *Image and Vision Computing*, vol. 16, no. 8, pp. 533 – 539, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/B6V09-3TFBT6R-4/2/c9964fc48556831725ffe7b406a742ed

[2] M. S. Costa and L. G. Shapiro, "3D object recognition and pose with relational indexing," *Computer Vision and Image Understanding*, vol. 79, no. 3, pp. 364 – 407, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6WCX-45FBSGV-P/2/2d806591960316e95d05915b059d010f

[3] C. Cyr and B. Kimia, "3D object recognition using shape similiarity-based aspect graph," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 2001, pp. 254–261.

[4] B. Rosenhahn, T. Brox, D. Cremers, and H. peter Seidel, "A comparison of shape matching methods for contour based pose estimation," in *In Combinatorial Image Analysis, LNCS 4040*, 2006, pp. 263–276.

[5] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *Int. J. Comput. Vision*, vol. 73, pp. 243–262, 2007.

[6] D. M. Gavrila, "A bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1408–1421, 2007.

[7] M. Ulrich, C. Wiedemann, and C. Steger, "CAD-based recognition of 3D objects in monocular images," in *Proc. IEEE International Conference on Robotics and Automation ICRA '09*, 2009, pp. 1191–1198.

[8] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[9] P. Kontschieder, M. Donoser, and H. Bischof, "Beyond pairwise shape similarity analysis," in *Proc. of Asian Conference on Computer Vision (ACCV)*, 2009.

[10] J. C. Mitchell, "Discrete uniform sampling of rotation groups using orthogonal images," *SIAM Journal of Scientific Computing*, vol. 30, pp. 525–547, 2007.

[11] X. Li, H. Su, and J. Chu, "Multiple model soft sensor based on affinity propagation, gaussian process and bayesian committee machine," *Chinese Journal of Chemical Engineering*, vol. 17, no. 1, pp. 95 – 99, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B82XJ-4VS6137-H/2/e0a73d238da74674395a7817570c9051