Bin-Picking based on Harmonic Shape Contexts and Graph-Based Matching

J. Kirkegaard
Image House A/S
Copenhagen, Denmark

T.B. Moeslund
Lab. of Computer Vision and Media Technology
Aalborg University, Denmark

Abstract

In this work we address the general bin-picking problem where 3D data is available. We apply Harmonic Shape Contexts (HSC) features since these are invariant to translation, scale, and 3D rotation. Each object is divided into a number of sub-models each represented by a number of HSC features. These are compared with HSC features extracted in the current data using a graph-based scheme. Results show that the approach is somewhat sensitive to noise, but works in presence of occlusion.

1. Introduction

Bin-picking refers to the problem of having a robot automatically pick an object from a bin. Many industrial applications are successfully being handled every day using computer vision and robots. However, for these systems the objects either have a simple (2D) shape and/or are organized in a structured manner. The general bin-picking problem where the objects have a 3D shape and are randomly organized, see figure 1, have not yet been solved in a general manner primary due to severe occlusion problems.

One approach to the general problem is to apply a CAD model of the objects in the bin. E.g., using the appearance [1] or circular features [7]. Alternatively 3D data can be found using [10, 9] and compared with the CAD model.

A common problem is that approaches tend to use global features, which result in a high sensitivity due to occlusions. In this work we find the pose using a constellation of local features in the form of 3D points. We represent the features in such a way that they become invariant to translation, scale, and rotation. To handle the occlusion problem we use a matching strategy based on graph theory allowing us to make a pose estimation even when a number of the features are occluded or corrupted by noise.

The focus of the paper is on the invariant features and the matching of these features from the 3D data and the CAD model. We therefore assume that 3D data of the bin is available. In section 2 we describe how the 3D bin-data can be segmented into interesting and uninteresting points. In section 3 we describe the invariant features and in section 4 we describe how to use these features for matching and pose estimation. In section 5 and 6 the results are presented and discussed, respectively.

Figure 1: Randomly organized objects in a bin. An object is approximately the size of a coconut.

2. Segmentation of Surface Features

Using 3D points as features results in a vast amount of different features. However, some 3D points are more discriminative than others. For example, features on a large smooth surface might not be the best choice since these by nature will result in ambiguities in the matching process. Therefore we do a segmentation in order to find positions where the ambiguity is low.

The general idea is to find those positions $\mathbf{p}$ where the curvature of the data changes since these will indicate edges or other transitions.

The changes in the curvature are found by first approximating the data by a mesh [6]. We then divide the mesh into regions each having a similar shape and then defining the mesh vertexes, $\mathbf{p}$, where changes occur as points where a transition between different shape types occur.

From differential geometry we know that the curvature of a surface at a given point, $\mathbf{p}$, can be defined using the principal curvatures κ_1 and κ_2. By combining the principal curvatures we get the *Gaussian (H)* and *mean (K)* curvatures defines as [8]:

$$H(\mathbf{p}) = (\kappa_1(\mathbf{p}) + \kappa_2(\mathbf{p}))/2 \quad (1)$$

$$K(\mathbf{p}) = \kappa_1(\mathbf{p}) \cdot \kappa_2(\mathbf{p}) \quad (2)$$

Looking merely at the sign of the *Gaussian* and *mean* curvatures we can classify each point on the mesh into one of six categories as shown in table 1.

signum(K)	signum(H)	Shape class
0	0	Planar
0	+	Concave cylindrical
0	−	Convex cylindrical
+	+	Concave elliptical
+	−	Convex elliptical
−	any	Hyperbolic

Table 1: Surface classification scheme based on the sign of the *mean* and *Gaussian* curvatures. [11]

3. Harmonic Shape Contexts

Having pre-segmented the 3D data we now have to find some way of extracting features for the remaining (or a subset of these) points that can be used in the matching process. These features should be invariant to both translation, scale, and rotation in order to reduce the search space to a realistic size. In this work we use the Harmonic Shape Contexts (HSC) feature representation since these have exactly the invariant characteristics we are after [5].

The HSC is a regional feature which characterizes the surface of the object in a small finite region around each point of interest. A regional feature is a compromise between global and local surface features combining the noise robustness of the former with the occlusion robustness of the latter.

The HSC is a generalization of the Shape Contexts features, which use a 3D histogram to describe the surface. The cells in the histogram are defined by a sphere centered at the point of interest with the sphere's north pole vector aligned with the normal vector of the data around this point [3]. The sphere is divided linearly in the azimuthal (east-west) and in the colatitudinal (north-south) directions of the sphere, while logarithmical in the radial dimension.

A given cell accumulates a weighted count for each neighborhood point whose spherical coordinates fall within the ranges of the cell. The actual contribution (i.e., the weighing) to the cell count is balanced by local point density and the volume of the cell [6, 3].

Any given spherical function, i.e., a function $f(\theta, \phi)$ defined on the surface of a sphere parameterized by the colatitudinal and azimuthal variables θ and ϕ, can be decomposed into a weighted sum of spherical harmonics as given by equation 3 [5].

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} A_l^m \, Y_l^m(\theta, \phi) \quad (3)$$

The terms A_l^m are the weighing coefficients of *degree* m and *order* l, while the complex functions $Y_l^m(\cdot)$ are the actual spherical harmonic functions of *degree* m and *order* l. The complex function $Y_l^m(\cdot)$ is given by equation 4, where $j = \sqrt{-1}$.

$$Y_l^m(\theta, \phi) = K_l^m \, P_l^{|m|}(\cos\theta) \, e^{jm\phi} \quad (4)$$

The term K_l^m is a normalization constant, while the function $P_l^{|m|}(\cdot)$ is the *Associated Legendre Polynomial*. The key feature to note from equation 4 is the encoding of the azimuthal variable ϕ. The azimuthal variable solely inflects the *phase* of the spherical harmonic function and has no effect on the *magnitude*. This means that $||A_l^m||$, i.e., the norm of the decomposition coefficients of equation 3, is invariant to parameterization in the variable ϕ.

The rotationally invariant property of the spherical harmonic transformation makes it suitable for use in encoding the shape contexts representation enabling a translation, scale, and rotational regional feature, see [6] for details.

Representing the Shape Contexts in terms of a weighted sum of spherical harmonics yields an infinite number of coefficients. However, in praxis the information will be band-limited and therefore only coefficients up to a certain bandwidth are to be stored. Concatenating these coefficients yields a feature vector that represent one particular 3D point.

4. Matching

The primary purpose of extracting HSC features from the scene and model meshes is to perform a matching between features found in the scene and features found in the CAD model.

Due to (self)occlusion not all features will be visible in the CAD model and the data at the same time, i.e., only partial views of the objects are possible. Therefore, we follow a multi-view matching approach where the model is represented as a number of sub-models only containing data observable form a certain viewpoint. We hereby gain an easier (and in fact also more correct) matching process, however at the price of a more time consuming iteration through the model base performing a matching of each sub-model.

4.1. Finding Correspondences

The actual matching of a HSC feature from the data and one from the CAD model is based on three factors: i) the *normalized correlation coefficient* between the two feature vectors of the HSC, ii) the absolute difference between the *mean curvature* and iii) the absolute difference between the *Gaussian curvatures* of the two particular points. The curvature comparison serves as a rough initial classification of the quality of the match. Only if the quality is good the correlation is performed.

Matching only two points at a time is obviously not enough and therefore more matches are considered at a time. More precisely, a match with a very high correlation factor (close to unity) may be a very poor choice if it is incompatible with all other matches, while a match with a medium correlation factor may be a good match if it has a large number of other compatible matches.

The consensus approach is based around the term *compatibility* between matches, which relates to whether several matches are concurrently possible from a geometrical point of view. The only way more matches can be compatible is if the distance between the respective points in the first data set is equal (approximately) to the distance between the corresponding points in the CAD model. This rigidity constraint can be used to determine compatibility between matches, i.e., a match is not only judged by the correlation factor but also by its number of compatible matches.

This consensus based approach for finding a large number of compatible matches can be formulated as a graph search problem, as shown by figure 2. A *node* in the graph represents a *match* between two points, while an *edge* between two nodes in the graph indicates that the two matches are *compatible*.

Each node in the graph is attached a weight, which indicates how good a match it represents, i.e., the computed correlation factor for the actual match is used. In figure 2 the node weights are indicated by a shade of gray, where darker nodes represent better matches (higher weights). The actual compatibility between the nodes is based on the rigidity constraint.

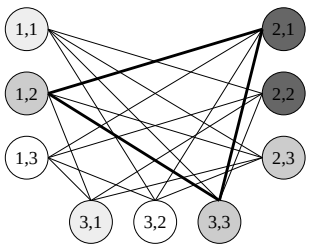


Figure 2: A graph representing the problem of matching a set of points between two meshes. A node represents a match while an edge represents compatibility between matches.

The problem of finding the best match between the points in each of the meshes, is then reduced to the problem of finding the maximally weighted set of mutually compatible nodes. For the simple example shown in figure 2 this set is shown as the nodes connected by the bold edges.

The particular problem of finding the maximally weighted set of interconnected nodes in a graph is known as the *maximum clique* problem. The problem is NP complete meaning that no algorithm with polynomial time complexity is known to exist. A solution to the problem can either be obtained by brute force methods (i.e., try all subsets of interconnected nodes and find the maximally weighted) or approximated by different heuristic methods. [2]

The problem of finding the *maximum clique* can be formulated as an optimization of the function f as shown by the equation 5 where n is the number of nodes in the graph.

$$f(\mathbf{x}) = \sum_{i=0}^{n-1} w_i x_i - \lambda \sum_{i=0}^{n-1} w_i c_i \tag{5}$$

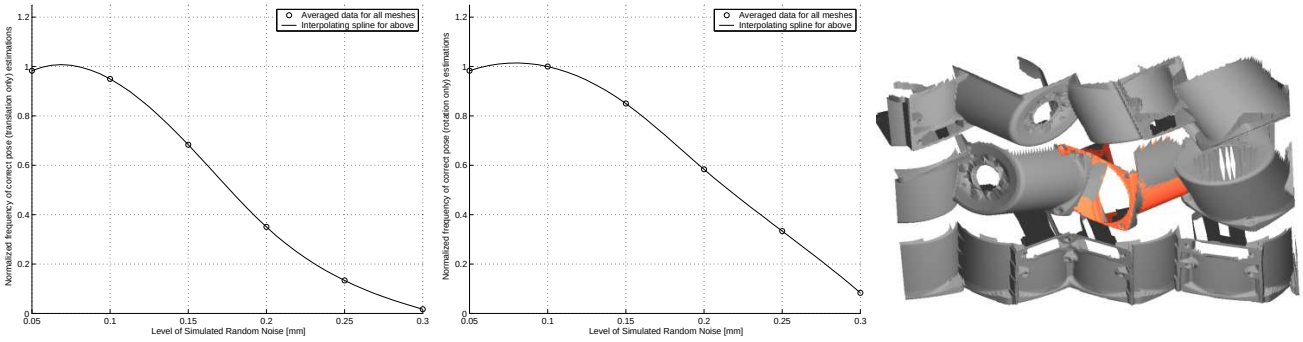
The function f accumulates the total weight of the nodes in the clique, based on a membership vector $\mathbf{x}$ and the individual node weights w_i. The elements in the membership vector are binary values indicating if the given node is part of the clique or not. The variables c_i are also binary and indicate whether the given node i can be part of the clique defined by the membership vector. This effectively subtracts the weights of the falsely included nodes and yields an output containing an accumulated weight only for the nodes which can be part of a clique. The state of c_i is determined based on the inverse graph, i.e., for a set of nodes to constitute a clique in a given graph, no two clique nodes must be connected in the inverse graph. The weighting factor λ is introduced to control the balance between the gain and the penalty, however it is set to unity in the current implementation.

Due to the nature of the problem the optimization of equation 5 is done using simulated annealing [6].

4.2. Pose Estimation

After the matching process we are left with two equally sized ordered sets containing matching 3D vectors. The pose estimation problem is then reduced to finding the rigid transformation aligning the two point sets.

As the point matches are obtained from real data contaminated by noise it is highly unlikely that a single rigid transformation can explain all the matching points. This effectively means, that the rotation matrix and translation vector has to be determined in an optimal way favoring as many point matches as possible. As the problem of determining the rotation and translation is separable, the approach taken is based around initially determining the rotation matrix. The problem of finding the optimal rotation aligning



(a) (b) (c)

Figure 3: The normalized recognition of the correctly pose estimated objects as a function of the added noise level. a) Translation. b) Rotation. c) An example of a correctly recognized object in an occluded scene (result overlaid in red).

two data sets is also known as the *orthogonal Procrustes problem* and is solved by performing a singular value decomposition of the mean translated point sets. [4]

5. Results

To evaluate the suggested approach we use the object type shown in figure 1. For this particular object type we use 64 (four viewpoints for each of the three axes: 4^3) sub-models. Each HSC is constructed by 16 colatitudinal divisions, 32 azimuthal divisions, and 10 radial divisions. Combing this with a chosen bandwidth [6] we end up with 1360 coefficients which constitute a feature vector [6].

We investigate how well the pose estimating can be carried out in the presence of noise. The test is performed by generating 120 different configurations of the model for each noise level and then counting the number of correctly pose estimated instances of the object. A correct pose estimation is defined to be when the $L2$ norm between the simulated and estimated rigid transformations is below 1.5 for the rotation matrices and the translation vectors, respectively. In figure 3 the results are shown.

The occlusion-handling capabilities of the suggested approach are assessed by analyzing scenes with randomly organized objects. For example in the scene in figure 3.c two objects are correctly pose estimated[1].

6. Discussion

Not all objects in the scene can be pose estimated correctly, but we have found that always at least one object can be pose estimated correctly, which is the success criterion in bin-picking since the scene changes each time an object has been removed. Tests also show that the approach is sensitive to noise. This will be the focus of future work by including smoothing of the mesh data, using methods based on curvature consistency with respect to equation 1 and 2.

[1] In addition six objects are pose estimated correct with respect to 5DoF. This is a typical situation when dealing with self-symmetric objects.

References

[1] I. Balslev and R. D. Eriksen. From belt picking to bin picking. *Proceedings of SPIE - The International Society for Optical Engineering*, 4902, 2002.

[2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw Hill Book Company, first edition, 1990.

[3] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. *ECCV*, 2004.

[4] G. H. Golup. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[5] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2003.

[6] J. Kirkegaard. Pose estimation of randomly organised stator housings using structured light and harmonic shape contexts. Master's thesis, Lab. of Computer Vision and Media Technology, Aalborg University, 2005.

[7] T. B. Moeslund and J. Kirkegaard. Pose estimation of randomly organised stator housings with circular features. In *Lecture Notes in Computer Science 3540*, 2005.

[8] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 34(2), 2002.

[9] J. Salvi, J. Pags, and J. Battle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4), 2004.

[10] R. D. Schraft and T. Ledermann. Intelligent picking of chaotically stored objects. *Assembly Automation*, 23(1), 2003.

[11] E. Trucco and A. Verri. *Introductory Techniques for 3D Computer Vision*. Prentice Hall, first edition, 1998.