Object Pose Detection in Industrial Environment

Dipl.-Ing. Matthias Palzkill, Univ.-Prof. Dr.-Ing. Dr. h. c. Alexander Verl Fraunhofer IPA,
Nobelstr. 12, 70569 Stuttgart, Germany, { mzp, verl }@ipa.fhg.de

Summary

This paper shows a method for object pose detection that is successfully applied to industrial applications running a three-shift system. The industrial applications are fully automated feeding systems, commonly known as bin-picking. The proposed method is a generic approach to detect 6 degrees of freedom of any solid objects with arbitrary geometry. The proposed method is using 3D range data and is based on a hypothesize-and-test approach .In the first step, object poses are hypothesized by means of pose clustering. In the second step, the verification of estimated object poses is realised by an appearance-based template-matching approach. In addition, the method's interfaces are designed to ensure compatibility to 3D sensor systems and handling systems [5]. During long-term operations in different use-cases, the method showed its usability regarding the crucial requirements, such as robustness, accuracy, portability and speed.

1 Introduction

Object pose detection is of major importance for automation in industrial environment, such as fully automated feeding systems. Typical use-cases are bin-picking and conveyor-picking. These applications are characterized by chaotically stored objects, which need to be picked and placed by handling systems.
Starting point for these tasks are any kind of charge carriers, like lattice boxes. They can be seen as a standard for manufacturing internal material flow systems. They can be filled easily with different objects and transported comfortably with forklifts or lift trucks. In addition, their stackability ensures ideal stocking. Also, when objects need to cover comparatively short distances for processing they are placed on conveyors. In both cases, the objects lose their state of order. States of the art for restoring objects' state of order are mechanical or manual solutions. These current solutions are expensive and inflexible (cf. [1], [2]). From an automation point of view, this task is absolutely challenging making it an on-going subject for decades in automation, particularly in robotics. The considered handling tasks are one of the last not yet automated gaps in material flow chains. A strong market growth could be observed making robot supported intralogistics a key industry for modern processes in economics and production [4].

1.1 Related Work

There exist various numbers of approaches to solve object pose detection and their categorisation is not consistent. Among others, literatures classify between model- and view-based approaches ([12], [13]), feature- and appearance-based approaches or introduces several classes (cf. [11]). The amount of approaches especially designed for industrial purpose is comparably small. A specific characteristic is that almost exclusively dealt with rigid objects of well-known geometry.
In this paper, top-down and bottom-up classifications are applied. Top-down approaches are based on deduction, meaning to verify a theory based on empirical data. Commonly used methods are template-matching approaches. Templates are matched on sensory input in order to accept or decline a theory (i.e. object pose). Due to their lack of a generally acceptable threshold, the comparative techniques can be rather used to find the relative best result. As a matter of fact, these approaches result in a huge number of comparisons, very likely to cause difficulties when it comes to fast evaluation. In contrary, bottom-up approaches create theories based on empirical data. Usually, extracted features and their correlations are used to draw conclusions about an object's pose. In order to gain reliable object poses, a lot of features have to be correspondent to each other. Especially, cluttered or noisy scenes can highly influence the approach's reliability.

1.2 Requirements for Object Pose Detection

Based on enquiries of industry, following requirements and boundary conditions of a vision system for industrial use can be stated:
Process reliability – On the one hand the vision system must always detect a pose in order to keep the system going, on the other hand it must be robust and reliable to guarantee the object will be picked correctly. Whenever the vision system fails the automation stops and needs to be restarted manually. This is harming the contracted availability, which is supposed to be fairly above 95% in general.

Duration of detection – The actual time, that the vision system needs to detect object poses, is significantly influencing the overall cycle time of the automation. The majority of considered applications is required to pick an object between 6 and 15 seconds. Even though the detection can at least be partially performed parallel to the robot's movement, it becomes obvious that the desired duration needs to be less than a couple of seconds. Moreover, due to interference contours and possible collisions, not every detected object can be actually grasped. This is why it is essential that as much objects as possible are located within a scene.

Accuracy of detection – The required precision for a detected object pose is defined by the gripping system and the placement area. Though, it is possible to correct a certain amount of inaccuracy (cf. [6]) with mechanical alignment, the gripping accuracy is supposed to be within a few millimetres and degrees. Also, it has to be considered that handling systems like 6-axis robots have absolute position accuracy between 1 mm and 4 mm [7], adding a default value to resulting chain of error. This means that in order to ensure accurate gripping, the object detection must be as precise as possible.

Portability – The spectrum of object geometries and colours varies strongly. It is absolutely common that on a single automation, a dozen of different objects has to be handled. This urges the need for the vision system to be adaptable on practically any given geometry.

1.3 Setup

A bin-picking application is schematically shown in Figure 1 and usually contains the following components:

The **3D-sensor** system captures the scene and provides the range. There were different 3D line-scanners used. In order to obtain an area scan of the scene, the device has to be moved while scanning. For this purpose the device is mounted at the robot, which performs the movement, or at a separate kinematics, such as a pivot (see Figure 2).

The **vision software** includes the Object Pose Detection, which is described in chapter 2.

The **handling kinematics** including a gripping system have been 6-axis industrial robots by the manufacturers Fanuc, Kawasaki or Universal Robots. 6-axis robots are commonly used, due to their flexibility, usability and competitive pricing (cf. [3]).

The overall **cell control** is typically realised by a programmable logic controller (PLC). In the considered applications SIMATIC S7 by Siemens or Allan-Bradley by Rockwell Automation were used.

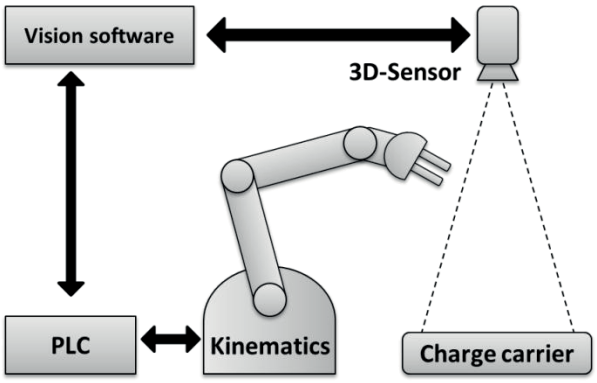

Figure 1 Schematic layout of bin-picking cell. The charge carrier is captured by a 3D-sensor. Obtained range data is analysed by a vision system, which provides object poses. Overall sequence is controlled by a separate PLC, which also forwards movement commands to kinematics (i.e. industrial robots)

Figure 2 Utilized 3D-sensor systems and pivot. From left to right: line scanner Sick LMS 400[8]; line scanner Leuze LPS 36[9]; pivot Schunk PR 70[10]

2 Method

The introduced method is a two-staged algorithm based on hypothesize-and-test approaches. First step is a fast pose estimation, which creates object pose hypotheses. The hypotheses represent the most likely object poses based on extracted features. The pose estimation is done by pose clustering. Being on its own the hypotheses are vague and do not allow reliable object pose detection, but they are capable of reducing the number of possible solutions. Consequently, in the end reliable object pose verification must be evaluated in order to accept only correct solutions.

2.1 Preprocess

Before the estimation and verification can be executed, following steps of preprocess are required:

Obtaining point clouds – Two commercial available sensor systems were used: SickLMS 400 is suitable for boxes with a volume between 1 m^3 and 4 m^3 and objects larger than circa 10 cm. Leuze LPS36 is qualified for boxes and objects smaller than mentioned above. Both devices measure along a spatial line, for which reason they have to be moved around in order to obtain a point

cloud, which captures the whole scene. By mounting the sensor system at the robot, it could be used as the required kinematic. The robot can move either linear above or pan over the bin of interest. On the one hand this means a reduction of possible cycle time; on the other hand the robot can access several boxes in its workspace. To finally obtain a point cloud, each measurement is registered with the robot's pose at that time.

Obtaining depth images – Depth images are obtained by performing a parallel projection on a received point cloud. The point cloud is initially referring to the sensor coordinate system (sensor frame). It is transformed to the charge carrier coordinate system (supply frame) and mapped to a depth image. Another way to obtain depth images is to synthetically render the object in a given orientation. There are several rendering methods available, such as VTK or OpenGL. It can be shown that implementing self-made rendering algorithms on CUDA are significantly faster in runtime.

2.2 Signal flow

The signal flow of the proposed algorithm contains training and detection mode. During training a knowledge base for the estimation is generated. It stores all relations between features and poses that must to be recognized. During the actual object detection, these relations are used to vote for hypotheses. The verification receives these hypotheses. For each hypothesis, a template is generated which is used for matching with the sensory input. The process is shown in Figure 3.

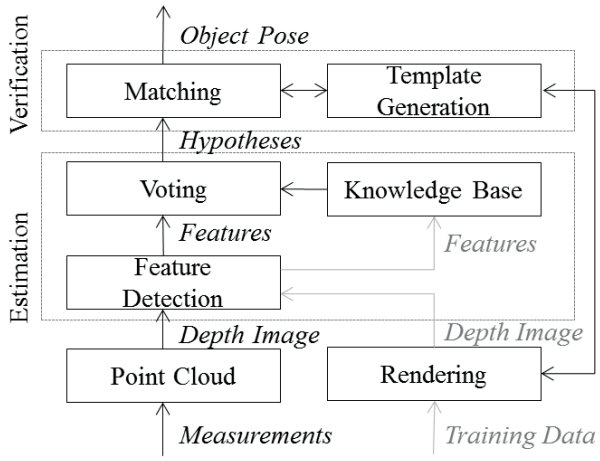

Figure 3 Signal flow of the proposed method. Black arrows show detection mode. The point cloud is mapped on a depth image, on which features are detected. Each feature votes for an object pose. The votes lead to hypotheses, which are verified eventually by template matching. Grey arrows show training mode. The object is rendered in different poses. All extracted features are stored in a knowledge base, which connects each feature to a set of poses, used to generate hypotheses in detection mode.

2.3 Object Pose Verification

The verification is done by template-matching. The pose to be verified is rendered and matched with the measured depth image. It is assumed that the measurement's error is normal distributed, such that the probability of presence can be computed by taking the differences of the object's visible surface. In addition, the object's curvature is examined to confirm the probability.

3 Object Pose Estimation

The estimation is done by pose clustering. "Pose clustering is also called hypothesis accumulation and generalized Hough transform and is characterized by a 'parallel' accumulation of low level evidence followed by a maximum or clustering step which selects pose hypotheses with strong support from the set of evidence."[14]. While pose clustering is common in 2D (e.g. [15]), it is rarely used in 3D range data. The general idea is that each occurrence of a feature changes the likelihood of possible object poses. The strongest supported votes result in hypotheses, which are passed to the verification. Each hypothesis represents a pose consisting of all its estimated degrees of freedom.

3.1 Feature extraction

In general, the shown method is suitable for any kinds of features, which comply with the requirements such as noise-robustness. In the following, a local feature is introduced that is similar to gradients. On each pixel of the depth image, the corresponding feature is calculated by its local neighbourhood. The difference of z-values between the considered pixel and each neighbour is taken into account. By definition, the difference can be positive, negative or equal. The result of the comparison with all neighbours finally leads to a hash value, which represents the feature (see Figure 4). The size of the neighbourhood was chosen with 12. This leads to a computational manageable amount of about 0.5 million different features values.

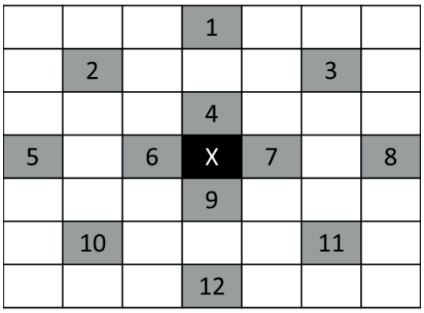

Figure 4 Exemplary calculation of feature value for pixel in depth image. Grey pixels are the considered neighbourhood. Each neighbour has one of three different conditions. The default is 0, if both z-values are equal. If the neighbour's z-value is less, its condition becomes 1 and if larger it is 2. The sequence of all conditions creates a specific value for the feature.

3.2 Knowledge base generation

A knowledge base is required, to be able to store the relationship between features and object poses. The proceeding is to generate an object's view and to perform feature extraction on the resulting depth image. The detected features are then associated to the respective object pose. The result is a set of associations for each found feature, called votes. This means that each feature can vote for several poses relative to the feature's location. Any feature will be declined as soon as its number of votes exceeds a reasonable count. This is done to keep the feature meaningful. The idea is that the added-value for a feature that is found on every view is lower than a feature occurring only once. Moreover, it is necessary to limit the size of the knowledge base due to finite computer memory. Since poses are voted, a vote contains a translatory and a rotatory part. Usually, translatory positions describe the object's centre, within this approach the translatory part points at the highest point of the respective object pose. For a given rotation the highest point can be converted to the more suitable object's centre. The assumption is that objects can be usually gripped, if their highest point is visible. During pose clustering it becomes obvious, that the highest point representation is highly reasonable. It should be noted that the complete method will work as well when the second highest point is also considered.

Discretisation of training and voting space – There are three major influences to determine the chosen rotatory resolutions: Required gripping accuracy, sensors' measurement data quality and available computer memory. The training space represents all poses that are considered for the knowledge base generation, whereas the voting space contains only that poses that can actually be voted. Since the voting space is supposed to be of less or equal size, it can be considered as a pre-clustering step. In typical bin-picking applications all rotations are likely to occur. When it comes to conveyor-picking, the objects are usually lying on the ground. They are isolated and tend to be in a physically stable state. Exemplary rotatory solutions space for a bin-picking application can be seen in Figure a). Figure b) shows the exemplary rotatory solution space for conveyor-picking. The representation is in axis angles. The length of each vector is its rotation around the respective axis angle.

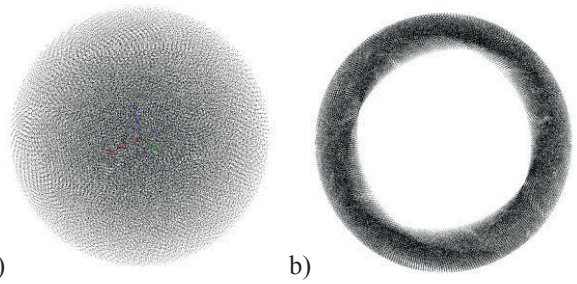

Figure 5 Discretisation of training-space and voting-space in three-dimensional representation. Each illustrated dot represents a vector, which states a rotation in axis-angle convention. The vector's coordinates are the axis and its length is the axis-angle.

3.3 Pose clustering

The proceeding for pose clustering is that each extracted feature is compared with the knowledge base. If the feature has been already considered during knowledge base generation, it is used for voting. Figure b) shows all votes for a depth image with contact bridges (see Figure a).

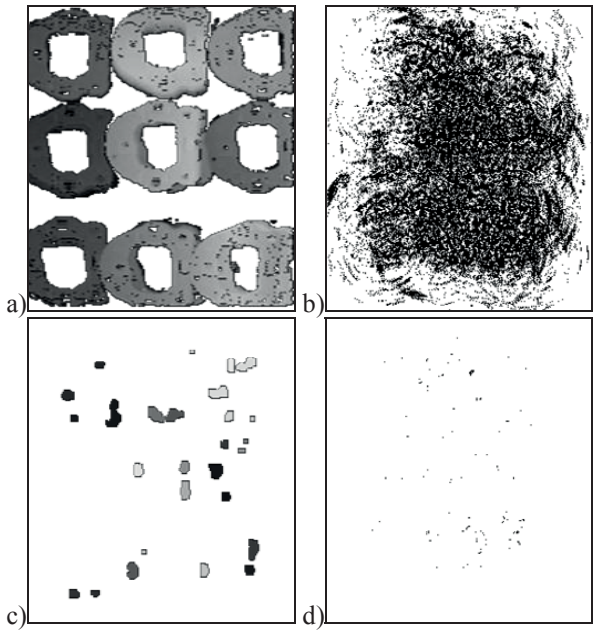

Figure 6 Pose clustering results. a) Depth image of contact bridges b) Distribution of all votes c) Local maxima in depth image d) Valid votes that are used for hypothesis

The pose clustering maps from feature space into pose space. Finding the strongest supported votes results in a considerable computational complexity. The pose space can be significantly reduced by accepting only plausible votes at the first place. In order to install an efficient method those votes will be suppressed, that are leading to objects being predominantly occluded. A general way to detect these votes is to test if they actually point on a potential highest point. Once the position of an object is referred to its highest point and not directly to its centre, all votes that do not point at local maxima in depth image can be ignored.

Consequently, before the hypothesis generation can take place, all local maxima in the considered depth image are to be located. In a typical range image (see Figure a) less than 100 local maxima with given parameter are located (see Figure c). After declining all votes that do not point at local maxima only few valid votes are left (see Figure d).

4 Results

This method has been successfully tested with gear-shafts, track links, contact bridges and steering wheels within industrial bin-picking applications. All results were computed with customary desktop-computers having a Quad-Core Processor and 8 GB of RAM.
Figure shows the results for contact bridges. On average, it took less than 350 ms to detect an object.
Although there are still ways to improve the performance, the acceptance critera for this bin-picking application were fulfilled, enabling a production in three-shift system.

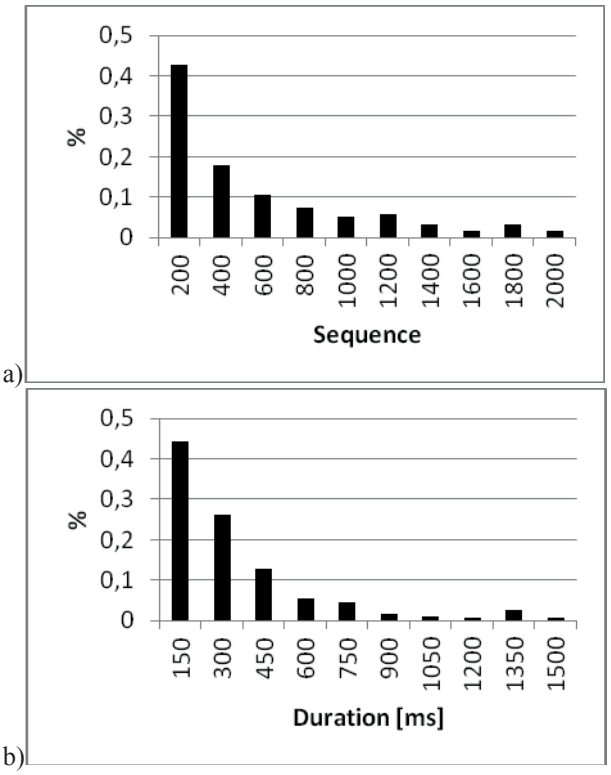

Figure 7 Results for about 3000 cycles of bin-picking. a) Shows required number of tested hypotheses to actually find a valid solution in percent (e.g. in above forty percent it takes less than 200 hypothesis to find a valid object pose). b) Shows time it took to find a valid solution in percent (e.g. in above forty percent it takes less than 150 ms to find a valid object pose).

5 Literature

[1] Leonard, S.; Croft, E.; Chan A.; Little James, J.: Robust motion generation for vision-guided robot bin-picking. In: Proceedings of the ASME International Mechanical Engineering Congress and Exposition, IMECE, 2007.

[2] Balslev Ivar; Eriksen, Ren D.: From belt picking to bin packing. In: Proc. SPIE 4902, 2002, p. 616.

[3] Hofmann, Holger: Presentation Market Report Bin Picking. Automation – Market – Competence AMC. Heppenheim: Hofmann, 2011.

[4] Günther, P.: Aufgaben der Zukunftsbranche Intralogistik. In: Intralogistik aus Baden-Württemberg 2006/2007, Stuttgart, Germany, 2006, pp. 4-5.

[5] Palzkill, Matthias: Automatisierter Griff in die Kiste. In: SPS-MAGAZIN, 08/2011.

[6] Hesse, Stefan: Greifertechnik – Effektoren für Roboter und Automaten, München: Carl Hanser Verlag, 2011.

[7] Czichos, Horst: Mechatronik – Grundlagen und Anwendungen technischer Systeme. Wiesbaden: Friedr. Vieweg & Sohn Verlag, 2006, p. 141.

[8] Sick AG; www.sick.com

[9] Leuze electronic; www.leuze.com

[10] Schunk GmbH & Co. KG; www.schunk.com

[11] Jain, Anil K.; Chira, Dorai: 3D object recognition: Representation and matching. In: Statistics and Computing (2000) 10, pp. 167-182.

[12] Bennamoun, M.; Mamic, G. J.: Object recognition: fundamentals and case studies. London: Springer, 2002.

[13] Bicego, M.; Castellani, U.; Murino, V.: A hidden Makov model approach for appearance-based 3D object recognition. In: Pattern Recogn. Lett. 26, Nr. 16, 2005, pp. 2588-2599.

[14] Stockman, G.: Object recognition and localization via pose clustering. In: Computer Vision, Graphics, and Image Processing, Volume 40, Issue 3, December 1987, pp. 361-387.

[15] Strzodka, R.; Ihrke, I.; Magnor, M.: A graphics hardware implementation of the Generalized Hough Transform for fast object recognition, scale, and 3D pose detection. In: Proceedings of IEEE International Conference on Image Analysis and Processing (ICIAP'03), 2003, pp. 188-193.