1 CAD-Based pose estimation for random bin-picking
Abstract:
In this paper, we propose a CAD-based 6-DOF pose estimation design for random bin-picking of multiple different objects using a Kinect RGB-D sensor. 3D CAD models of objects are constructed via a virtual camera, which generates a point cloud database for object recognition and pose estimation. A voxel grid filter is suggested to reduce the number of 3D point cloud of objects for reducing computing time of pose estimation. A voting-scheme method was adopted for the 6-DOF pose estimation a swell as object recognition of different type objects in the bin. Furthermore, an outlier filter is designed to filter out bad matching poses and occluded ones, so that the robot arm always picks up the upper object in the bin to increase pick up success rate. A series of experiments on a Kuka 6-axis robot revels that the proposed system works satisfactorily to pick up all random objects in the bin. The average recognition rate of three different type objects is 93.9

2 Eye-in-hand vision-based robotic bin-picking with active laser projection
Abstract:
In this paper, an eye-in-hand vision-based robotic bin-picking system is proposed. The system can identify the pose of a plumbing part from a pile and grip it correctly. A monocular eye-in-hand camera and a laser projector are employed to reconstruct the 3-D point cloud of plumbing parts stacked together. The projection direction of the laser line projector is controlled to change in order to scan thepile of objects while the camera is observing. 3-D points can then be determined by the a priori known geometry between the camera and the laser line projector. To estimate the pose of an object, the iterative closest point (ICP) is employed to match the point clouds of the object and the model. The transformation between the object and the model can thus be determined. A computed closer point (CCP) approach is proposed to estimate the pose of an object since the deviation from the object to the model is initially large in nature. The proposed CCP approach combining with the ICP algorithm can improve the success rate and accuracy of point cloud matching. The proposed system has been validated by experiments with potential applications in production lines.

Introduction:
Another way is matching an object and a model by using 3-D data. Point cloud, a set of points, is often used to determine the pose of an object. The iterative closest point (ICP) algorithm [2] is an illustrious algorithm that can be used to match two point clouds and then compute the transformation matrix between them. The ICP algorithm employs closest point matching method to match points from reconstructed data point cloud to model point cloud. By applying this relation, a transformation between two point clouds can be determined after a number of iterations. Because some closest points are not corresponding points in general, the result of computation may be inaccurate. If the deviation of a data point cloud from a model point cloud is large, the result of matching these two point clouds and the transformation between them cannot be successfully completed in most cases. Some studies have been focused on improving the ICP

algorithm [5, 6, 24].

A typical bin-picking system includes target detection, pose estimation, and gripping manipulation. Rodrigues et al. [18] proposed a bin-picking system that employs a single camera to measure objects in 2-D images, and the poses of objects can be estimated by 2-D data. Some systems employ 3-D cameras [16] or a camera with a laser projector [1] to scan objects for a manipulator to pick up an object from a bin. Based on 3-D data, the pose of a target can be estimated by using 3-D templates [19]. In this research, the proposed bin-picking system employs an active laser projector to reconstruct 3-D point cloud for pose estimation. The 3-D point cloud will be segmented by using a segmentation approach. For the purpose of determining the pose of an object, a computed closer point (CCP) approach is proposed that is based on object geometry. This approach can be used to estimate the pose of an object roughly to reduce the difference between the poses of an object and its model. After executing the CCP approach, the ICP algorithm is employed to estimate the pose of the object accurately. Then, the system enables a manipulator to grip an object repeatedly based on the pose estimation result.

Conclusion:

The proposed system employs an eye-in-hand camera with active laser projection to reconstruct the point cloud of a pile of plumbing parts. After employing point cloud segmentation, the point cloud of the top object can be identified. To determine the pose of object successfully, the feature-geometric CCP is proposed to roughly estimate the pose of object and then the ICP algorithm is employed to precisely match the data and model point clouds. Thus, the pose of the object can be determined in order for the manipulator to grasp the object correctly. The experiments have been successfully validated since the system can correctly identify and then grasp the object. This system can be applied in industrial automation to improve the performance of existing production line.

3 Determination of 3D object pose in point cloud with CAD model

Abstract:

This paper introduces improvements to estimate 3D object pose from point clouds. We use point-pair feature for matching instead of traditional approaches using local feature descriptors. In order to obtain high accuracy stimation, a discriminative descriptor is introduced for point-pair features. The object model is a set of point pair descriptors computed from CAD model. The voting process is performed on a local area of each key-point to boost the performance. Due to the simplicity of descriptor, a matching threshold is defined to enable the robustness of the algorithm. A clustering algorithm is defined for grouping similar poses together. Best pose candidates will be selected for refining and final verification will be performed.

The robustness and accuracy of our approach are demonstrated through experiments. Our approach can be compared to state-of-the-art algorithms in terms of recognition rates. These high accurate poses especially useful for robot in manipulating objects in the factory. Since our approach does not use color feature, it is independent to light conditions. The system give accurate pose estimation even when there is no light in the area.

Introduction:
For years, the robot industry has been going through enormous development. Robot nowadays not only receives commands from the computer/human but also has the ability to make decision itself based on various input sources. With the appearance of depth sensors, robot can now recognize shape of objects. Object recognition, therefore, becomes an important part of robotic perception. On the other hand, the quality of depth sensors lead to different approaches in object recognition. In this work, the Kinect camera is used to capture depth image for recognition task. Kinect camera is affordable, and is available for a wide range of applications in both industry and academy.

One popular approach in object recognition is keypoint matching. It has been well known that stable keypoint descriptors successfully made their way in object recognition field [1], [2]. These keypoint descriptors are invariant to illumination changes and geometric transformation, and keypoint correspondences can be determined reliably. In order to obtain 6-DOF pose, the keypoint coordinates in 3D must be estimated. These coordinates can be determined using structure from motion [3], or back-projecting 2D keypoints to 3D CAD model [4]. Since these descriptors work for texture objects, they fail for textureless objects, which are the common cases. The edge feature turns out to be a solution for this case since it can represent object boundaries. The common approach is edge matching between the edge image with a set of edge image templates, known as priories. The recent work involving this approach is LINEMOD [5], [6]. LINEMOD exploits both depth and color images to capture the appearance and 3D shape of the object in a set of templates covering different views of the object. That being said, LINEMOD still suffers from the presence of false positives and noise.

With the rise of 3D data usage for perception tasks, 3D keypoint descriptors become useful tools in object recognition and pose estimation. A set of popular descriptors are available in PCL (Point Cloud Library) [7]. These descriptors encoding local information of the keypoint can be categorized as geometric based descriptors. Descriptors such as PFH (Point Feature Histogram) [8], FPFH (Fast Point Feature Histogram) [9], and VFH (Viewpoint Feature Histogram) [10] are in this group. These descriptors encode relative orientation of normals and sistances between point pairs in a fixed coordinate frame, which is constructed using the surface normal of the keypoint. Changes in the surface normal of keypoint can lead to significant changes in the descriptors. Thus, these descriptors are considered to be sensitive to noise. On the other hand, SHOT (Signature of Histogram of Orientation) [11] represents topological traits by encoding a signature of histogram. For each spherical grid sector, one dimensional histogram is constructed by accumulating angles between point normals and keypoint normal. Histograms is orderly juxtaposing according to the local reference frame. SHOT, therefore, is invariant to rotation and translation, and robust to noise and clutter. That being said, even SHOT descriptor suffers from the effect of noise if noise occurs near the keypoint. These descriptors then will be matched with the model, usually following by correspondence grouping to estimate 3D pose. This approach works for some experimental cases but it failed in our test, as we will show later in the manuscript.

Along with these, other local invariant features have been proposed such as surface curvature [12], spin image [13], and surface normal distribution [14]. The segmented point cloud can also be used for pose estimation. In another approach, Rusu et al. proposed Viewpoint Feature Histogram (VFH) descriptor [10] for this purpose. It encodes angular distributions of surface normals on a segmented point cloud. This makes VFH suffer from occlusion, and therefore, a full pose estimation cannot be obtained. A global descriptor called CVFH (Clustered Viewpoint Feature Histogram) [15] was proposed for this case. Although recognizing object poses efficiently, these approaches rely on the point cloud segmentation result. That being said, the noise and quality of point cloud still have great impacts on these approaches.

On the other hand, point-pair features (PPF) has been proved to be an efficient feature for 3D-recognition since it was introduced by Drost et al. [16]. Due to its simplicity, PPFs can be computed fast and be matched easily with the model. The object model is defined as a set of PPFs computed from model point cloud. For fast searching, those PPFs is stored in a hash table as in [17]. These PPFs can be seen as a successor of surflet pairs [18]. Points will be sampled from the scene and PPFs will be matched with the model. Every matched pair will vote for a pose hypothesis. Finally, the high votes represent possible object pose hypotheses. Recently, this approach was improved by incorporating visibility context [19], or considering object boundary information. It has been noted that these PPFs is well-suited for recognizing objects that have rich variations in the surface normals. It is not very efficient in representing planar surface or self-symmetric objects. Even though this problem can be solved through voting process, it greatly degrades the performance of the algorithm. Another improvement, which include color information in PPF descriptor, was introduced in [20] to over come this problem. With color information in the descriptor, the hash function can reduce number of PPFs that fall into the same slot. It employs the idea that the color information can prune potential false matches and thus improve the matching process. There are also various modified Hough transformation for estimating 3D object pose [21] using SRT distance [22].

Color information is useful in removing false matches and eliminating unnecessary checks in the hash table. It is, however, sensitive to light conditions as well as object material. In practice, the light reflection on object surface can significantly change the color of the object. The surface textures also contribute to variance in the color. In some cases, the material such as metal can be rusted and change the color which lead to missed, or wrong matches. In this work, we perform manipulation task on objects in the factory, which are mostly metal objects. The object is defined by a CAD model as in Fig. 1a. Fig. 1b shows color variation on the object surface. This cause difficulties for detection method based on color information such as [20]. In this work, we propose a discriminant point-pair feature, which is inspired by the work in [16], [20]. To improve performance, we define the local feature as a set of PPFs inside the sphere centered at the keypoint. In addition, we propose an alternative pose clustering algorithm to reduced number of poses to be verified. The following experiment section will demonstrate the performance of the proposed algorithm and the accuracy of the estimated poses.

4 Robot Assisted 3D Point Cloud Object Registration
Introduction:

The ability to identify and manipulate with objects in its environment is a crucial task for robots used in a wide variety of applications that assume unstructured working environment. In recent years depth cameras became a popular and accessible tool for capturing information about the robot environment. Range data can be acquired using a structured light camera [1], stereovision camera [2], LIDAR device [3] etc. Fore mentioned devices all generate discrete range measurements or point clouds from a fixed perspective. If a robot is working in an unstructured or dynamic environment the point cloud data can be used to detect objects and their position. When the position of a depth camera in a robot frame is known and the object pose is determined the robot can proceed to perform handling operations. Benefits of using point cloud data in robotic applications are numerous so implementations span from scientific and household to industrial. Current advances in 3D data acquisition and processing for industrial applications and potential research opportunities are covered in [4]. Article on industrial application of point clouds [5] describes an automatic system that uses robot motion control and a laser scanner for the purpose of reverse engineering. In the emerging field of service robotics 3D point clouds are mostly used for creating maps of household environments [6] and for reliable object grasping and manipulation [7]. An efficient tool for 3D point cloud processing comes in a form of c++ library called Point cloud library (PCL) [8]. To solve a problem of object recognition and six degree of freedom pose estimation from a point cloud several methods are available and presented in [9]. Matching an object to a scene requires a known object template. Object template can be acquired from a CAD model and transformed to a point cloud or it can be registered from a 3D scanning system. Real time free-hand scanning system used for 3D model reconstruction is described in [10]. The system registers and incorporates point clouds of an object and removes scanning errors. There are two methods for solving the registration problem: Feature based registration and Iterative closest point algorithm (ICP). Feature based registration includes usage of keypoints and feature descriptors when estimating correspondence between two point clouds. Keypoints are interest points derived from a set of points that best describe the scene. Feature descriptors are computed from each keypoint. Popular feature histograms are Persistent point feature histogram [11], Fast point feature histogram [12], Viewpoint feature histogram [13] and Viewpoint oriented color-shape histogram [14]. Iterative closest point method iterates steps: search for correspondences, reject bad correspondences and estimate a transformation using the good correspondences.

5 3D Pose Estimation of Daily Objects Using an RGB-D Camera

Introduction:

Object recognition and 6-DOF pose estimation are important tasks in robotic perception. For the last decade,stable keypoint descriptors [1], [2] have led to successful progress on object recognition. As these keypoint descriptors are invariant to changes in illumination and geometric transformation, keypoint correspondences over different images can be reliably determined. For robotic manipulation, 3D coordinates of keypoints are generally required as an object model so that full 6-DOF object pose can be recovered. These keypoint coordinates can be calculated via structure from motion [3] or back-projecting 2D keypoints to 3D CAD model [4]. The keypoint descriptors are suitable for textured objects, but a large number of daily objects still lack texture. For the less textured object, edge feature is preferred since it corresponds to the object boundaries. A common approach is that a set of edge image templates of an object is known a priori, and in testing phase the template images are matched with a given query edge image. In classic computer vision, the chamfer [5] and Hausdorff [6] distances were proposed as robust metric, and they were further enhanced by considering edge orientation [7], [8]. A common method to extract edge feature from an image is image gradient-based method, such as Canny edge detector [9]. However, this method often results in unnecessary edges coming from surface texture or non-Lambertian reflectance. To find useful edges from depth discontinuities, the multi-flash camera [10] was introduced to determine depth edges by casting shadows from multiple flashes and was successfully employed in several robotic pose estimation algorithms [11], [12]. As RGB-D sensors, which provide depth as well as color information in real-time, have recently been introduced at low cost, 3D information-based pose estimation can be more feasible than ever. Compared to 2D images, 3D data are more invariant to the geometric changes. The iterative-closest point (ICP) algorithm [13] is well-known for the registration of 3D point clouds, but it requires a good initial pose estimate. Rusu et al. [14] proposed the Viewpoint Feature Histogram (VFH) that encodes four angular distributions of surface normals on a segmented point cloud. As the VFH is not robust to occlusion and does not allow full pose estimation, the Clustered Viewpoint Feature Histogram (CVFH) was recently presented [15]. Lai et al. [16] proposed a tree structure for scalable object recognition and pose estimation, but the pose estimation is limited in that it can only estimate 1-DOF rotation of the object pose. Although these approaches can recognize object pose efficiently, they hinge upon perfectsegmentation from the background. All of these approaches are applicable for well structured table-top manipulation, but they are not robust for cluttered environments. For general object pose estimation, it is required to match an object model with a scene directly. Like local image keypoints, several local invariant features have been proposed based on the distribution of surface normal around a point [17], surface curvature [18], spin image [19], and relative angles between neighboring normals [20]. While these features are invariant to rigid body transformation, they are sensitive to noise and resolution difference of point clouds. Drost et al. [21] defined a pair feature using two points on surfaces and their normals. In the learning phase, a set of pair features from an object is calculated and saved in a hash table for fast retrieval. In the testing phase, points are randomly sampled from the sensor data, and each pair matched with pairs in the learned model votes for a pose hypothesis. After the voting process, a set of high votes over a certain confidence level are aggregated to form possible object pose hypotheses. The pair feature can be seen as a successor of the surflet pairs [22], and using a hash table for fast matching is also presented in [23]. This approach was recently enhanced by incorporating the visibility context [24] or considering object boundary information [25]. There are also several modified Hough transforms for 3D object pose estimation [26] using the SRT distance [27]. The surface point pair feature is well suited to recognize objects that have rich variations in surface normals. However, it is not very efficient in representing planar or self-symmetric objects because a lot of different point pairs fall into the same hash slot. Although this ambiguity could be solved via the voting process where different pose hypotheses are aggregated separately, this certainly degrades its efficiency. Moreover, when there are a large amount of background clutter in a test scene, a lot of the point pair features come from the clutter. If surface shapes of our object model and the clutter are similar each other, it is highly likely to have false feature matches and consequently results in false pose estimates. As such, we need to prune unnecessary feature matching for more efficient and accurate pose estimation. We exploit the RGB color information to prune potentially false matches based on the color similarity. To be more robust to illumination changes, the HSV (Hue, Saturation, and Value) color space is considered. By using these additional dimensions, the casted votes in an accumulator space are more likely to contribute to true pose hypotheses. Furthermore, the voting process is more efficient since unnecessary votes are skipped. These arguments are verified in the following experimental section.

6 Point Pair Features Based Object Detection and Pose Estimation Revisited

Introduction:

Many computer vision applications require finding the object of interest in either 2D or 3D scenes. The objects are usually represented with the CAD model or objects 3D reconstruction and typical task is detection of this particular object instance in the scenes captured with RGB/RGBD or a depth camera. Detection considers determining location of the object in the input image, usually denoted by the bounding box. However, in many scenarios, this information is not sufficient and complimentary 6DOF pose (3 degrees of rotation and 3 degrees of translation) is also required. This is typical in robotics and machine vision applications. Consequently, the joint problem of localization and pose estimation is much more challenging due to the high dimensionality of the search space. In addition, objects are often sought in cluttered scenes under occlusion and illumination changes and also close to real-time performance is usually required. In this paper, we rely only on depth data, which alleviates the problem of illumination changes. One of the most promising algorithms for matching 3D models to 3D scenes was proposed by Drost et al. [10]. In that paper, authors couple the existing idea of point-pair features (PPF), with an efficient voting scheme to solve for the object pose and location simultaneously. Given the objects 3D model, the method begins by extracting 3D features relating pairs of 3D points and their normals. These features are then quantized and stored in a hash table and used for representing the 3D model for detection. During run-time stage, the same features are extracted from a down-sampled version of a given scene. The hash-table is then queried per extracted/quantized feature and a Hough-like voting is performed to accumulate the estimated pose and location, jointly. In order to overcome complexity of the full 6DOF parametrization, assumption is made that at least one reference point in the scene belongs to the object. In that case if the correspondence is established between that reference point in the scene and one model point there, and if their normals are aligned, then there is only one degree of freedom, rotation around the normal, to be computed in order to determine the objects pose. Based on this fact, a very efficient voting scheme has been proposed. The great advantage of this technique lies in its robustness in presence of clutter and occlusion. Moreover, it is possible to find multiple instances of the same object, simply by selecting multiple peaks in the Hough space. While operating purely on 3D point clouds, this approach is fast and easy to implement. Due to its pros, aforementioned matching method immediately attracted attention of scholars and was plugged into many existing frameworks. Moreno et al. used it to constrain a SLAM system by detecting multiple repetitive object models [21]. They also devise a strategy towards an efficient GPU implementation. Another immediate industrial application is bin picking, where multiple instances of the CAD model is sought in a pile of objects [13]. Besides, there is a vast number of robotic applications [3, 20] where this method has been applied. The original method also enjoyed a series of add-ons developed. A majority of these works concentrated on aug- menting the feature description to incorporate color [5] or visibility context [14]. Choi et al. proposed using points or boundaries to exploit the same framework in order to match planar industrial objects [6]. Drost et al. modified the pair description to include image gradient information [9]. There are also attempts to boost the accuracy and performance of the matching, without touching the features. Figueiredo et al. made use of the special symmetric object properties to speed up the detection by reducing the hash-table size [8]. Tuzel et al. proposed a scene specific weighted voting method by learning the distinctiveness of the features as well as the model points using a structured SVM [22]. Unfortunately, despite being well-studied, method of Drost et al. [10] is often criticized by high dimensionality of the search space [4], being sensitive to 3D correspondences [16], having performance drops in presence of many outliers, and low density surfaces [19]. Furthermore, the succeeding works report to significantly outperform the technique in many datasets [12, 4]. Yet, these methods work with RGB-D data, cannot handle occlusions and heavily depend on the post-processing and pose refinement.

7 Going Further with Point Pair Features.

Related Work:

The literature on object detection and 3D pose estimation is very broad. We focus here on recent work only and split them in several categories.

Sparse Feature-Based Methods. While popular for 3D object detection in color or intensity images several years ago, these methods are less popular now as practical robotics applications often consider objects that do not exhibit many stable feature points due to the lack of texture.

Template-based methods. Several methods are based on templates [12, 21, 27], where the templates capture the different appearances of objects under different viewpoints. An object is detected when a template matches the image and its 3D pose is given by the template. [12] uses synthetic renderings of a 3D object model to generate a large number of templates covering the full view hemisphere. It employs an edge-based distance metric which works well for textureless objects, and refines the pose estimates using ICP to achieve an accurate 6D pose. Such template-based approaches can work accurately and quickly in practice. However, they show typical problems such as not being robust to clutter and occlusions.

Local patch-based methods. [5, 24] use forest-based voting schemes on local patches to detect and estimate 3D poses. While the former regresses object coordinatesGoing Further with Point Pair Features and conducts a subsequent energy-based pose estimation, the latter bases its voting on a scale-invariant patch representation and returns location and pose simultaneously. [4] also uses Random Forests to infer object and pose, but via a sliding window through a depth volume. In the experiment section, we compare our method against the recent approach of [5], which performs very well. Point-cloud-based methods. Detection of 3D objects in point cloud data has a very long history. A review can be found in [18]. One of the standard approaches for object pose estimation is ICP [17], which however requires an initial estimate and is not suited for object detection. Approaches based on 3D features are more suitable and are usually followed by ICP for the pose refinement. These methods include point pairs [9, 17], spin-images [14], and point-pair histograms [22, 25]. These methods are usually computationally expensive, and have difficulty in scenes with heavy clutter. However, we show in this paper that these drawbacks can be avoided. The point cloud-based method proposed in [9] is the starting point of our own method, and we detail it in the next section.

8 A 3D Object Detection and Pose Estimation Pipeline Using RGB-D Images

Introduction

3D object detection and pose estimation are of great significance to robotics because they allow robots to localize the objects. This capability enables robots to autonomously perform manipulation tasks such as pick and place, parts assembly, amongst other. Environmental uncertainty and lack of structure still pose important challenges to accurate and efficient object detection and pose estimation algorithms. Many algorithms have been developed to tackle this problem. Local image descriptors, such as SIFT [1] and SURF [2] are often used to match key points between the scene and textured objects. 2D keypoints are then back-projected to 3D, where the objects 6-DOF pose is retrieved based on the 3D point-to-point correspondences. However, these descriptors fail to extract stable feature points from texture-less objects common in industrial environments. Recently, learning-based methods [3], [4] use forest-based voting schemes on image patches to detect objects and estimate 3D poses. The former regresses object coordinates and estimates pose by minimizing an energy function. The latter integrates LINEMOD [5] template patches into random forests and jointly estimates objects positions and orientations. As more low-cost 3D cameras enter the market, the research focus has shifted to directly process 3D point clouds. The advantages of 3D-point-based algorithms are independence from object texture and invariance to illumination. The iterative closest point algorithm (ICP) [6] is a common approach to align model points with scene points, but can only be used for pose estimation. However, an objects initial pose is crucial to the outcome accuracy. A few of the 3D descriptor algorithms [7][12] have been devised recently. The goal of 3D point cloud descriptors is to establish correspondences between model points and scene points. 6-DOF transformation can then be calculated based the 3D-point correspondences. In [9], the authors use point pair features for point matching and proposes a voting scheme to recover an objects 3D pose. In [13], the point pair feature algorithm is enhanced by considering object boundaries. In [14], the authors compare the performance of popular 3D local descriptors on different datasets. These methods however still suffer from noise-sensitivity. They are prone to mismatch in cluttered scenarios or require rich variation in object geometry. Template matching is another common approach for object detection. During offline training, an objects template images are sampled from varying viewpoints. During on-line testing, templates are compared to a scene image by computing the similarity. The object is detected if a template is matched. The objects pose is determined based on the templates training pose.

9 Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes

Related Work

3D object detection and localization is a difficult but important problem with a long research history. Methods have been developed for detection in photometric images and range images, and more recently, in registered color/depth images. We discuss these below.

Camera Images. We can divide image-based object detection into two broad categories: learning-based and template approaches. Learning-based systems generalize well to the objects of particular class like human faces [6], cars [7, 8], or Model Based Training, Detection and Pose Estimation of 3D Objects other objects [9]. Their main limitations are the limited set of object poses they accept, and the large training database and time. In general, they also do not return an accurate estimate of the object 3D pose. To overcome these limitations, researchers tried to learn the object appearance from 3D models [7, 8, 10]. The approach of Stark et al. [7] relies only on 3D CAD models of cars and Liebelt and Schmid [8] combine geometric shape and pose priors with natural images. Both of these approaches work well and also generalize to object classes, but they are not real-time capable, require expensive training and cannot handle clutter and occlusions well. In [10] authors use a number of viewpoint-specific shape representations to model the object category. They rely on contours and introduce a novel feature called BOB (bag of boundaries), which at a given point in the image is a histogram of boundaries from image contours in training images. This feature is later used in the shape context descriptor for template matching. While it generalizes well, it is far from real-time and cannot find a precise 3D pose. In contrast, our method is real-time capable, can learn new objects online from 3D models, can handle large amount of clutter and moderate occlusions and can detect multiple objects simultaneously. As discussed in [1], template-based approaches [1114] typically do not require large training sets or time, as the templates are acquired quickly from views of the object. However, all these approaches are either susceptible to background clutter or too slow for real-time performance.

Range Images. Detection of 3D objects in range data has a long history; a review can be found in [15]. One of the standard approaches for object pose estimation is ICP [16]; however this approach requires an initial estimate and is not suited for object detection. Approaches based on 3D features are more suitable and are usually followed by ICP for the pose refinement. Some of these methods (which assume that a full 3D model is available) include spin-images [17], point pairs [18, 19], and point-pair histograms [20, 21]. These methods are usually computationally expensive, and have difficulty in scenes with clutter. The method of Drost et. al [18] can deal with clutter; however, its efficiency and performance depend directly on the complexity of the 3D scene, which makes it difficult to use in real-time applications.

RGBD Images. In recent years, a number of methods that rely on RGBD sensors have been introducedamong them [22] which is subject to object classification, pose estimation and reconstruction. Similar to us the training data set is composed of depth and image intensity cues and the object classes are detected using a modified Hough transform. While being quite effective in real applications these approaches still require exhaustive training on large data sets. In [23] Lei et al. study the recognition problem at both the category and the instance level. In addition they provide a large data set of 3D objects. However, they have neither demonstrated that their approach work on heavily cluttered scenes in real time nor that it returns 3D pose as our method does.

10 Model Globally, Match Locally: Efficient and Robust 3D Object Recognition

Related Work

The problem of detecting and recognizing free-form objects in three-dimensional point clouds is well studied. Methods for refining a coarse registration, such as ICP [26], are able to optimize a coarse registration. However, we are only interested in methods for object registration that do not need a coarse pose as input. Several global methods for 3D object detection have been proposed. However, they either detect only shapes such as planes, cylinders and spheres, or require a segmen- tation of the scene. Wahl et al. [23] introduce an object identification scheme that identifies segmented free-form objects by computing the histogram of oriented point relations, called surflet pairs. The two-point feature used in our method is based on the idea of surflet pairs. Several ap- proaches detect objects using a variant of the Generalized Hough Transform [8, 14, 25] but are limited to primitive objects as the recovery of a full 3D pose with 6 degrees of freedom is computationally too expensive. Schnabel et al. [18] detect primitives in point clouds by using an efficient variant of RANSAC. Park et al. [13] detect objects in range images by searching for patterns of the object created from multiple directions. They parallelize their algorithm on the GPU in order to obtain matching times of around 1 second. By contrast, our approach works with general 3D point clouds and is equally efficient without parallelization.

A second class of methods, local methods, usually use a pipeline that first identifies possible point to point correspondences between the model and the scene. Multiple correspondences are then grouped to recover the pose of the model. A typical way of finding the correspondences is the use of point descriptors that describe the surface around a certain point using a low-dimensional representation. The descriptors need to be as discriminating as possible while being invariant against a rigid movement of the surface, robust against clutter and noise and embedded in a framework that can deal with occlusion. Point correspondences are built by comparing the descriptors of the model to those of the scene. Extensive surveys over different descriptors are given in [2, 9, 12], which is why only a few selected ones are covered here.

Point descriptors can be categorized by the radius of influence that affects them. Local descriptors exploit the geometric properties around a surface point, most notably by using different representations of the surface curvature [1, 5] or by fitting polynomials [15] or splines [24]. Regional descriptors try to capture the surface shape around the reference point. Splashs [20] describe the distribution of normal orientations around a point. Point Signatures [4] use the distance of neighbouring points to the normal plane of the reference point. Point Fingerprints [21] use geodesic circles around the reference point as description. Gelfand et al. [6] introduced an integral descriptor for point cloud registration that describes the intersecting volume of a sphere around the reference point with the object. Rusu et al. [17] build a Point Feature Histogram of two-point descriptors for all neighbouring points of the reference point. Chen and Bhanu [3] introduced a local surface patch representation that is a histogram of shape index values vs. the angle between normals. Johnson and Hebert [7] introduced spin images, which are histograms of the surface created by rotating a half-plane around the normal of the reference point and summing the intersecting surface. In [19] the spin images were used as an index into a database of objects with subsequent pose detection using a batch RANSAC. Ruiz Correa et al. [16] defined a symbolic surface signature to detect classes of similar objects.

Mian et al. [10] use two reference points to define a coordinate system where a three-dimensional Tensor is built by sampling the space and storing the amount of surface intersecting each sample. The Tensors are stored using a hash table that allows an efficient lookup during the matching phase. The Tensors can be used not only for matching, but also for general point cloud registration. The problem with point descriptors is that they are often sensitive to occlusion, noise and local clutter. Since they describe the surface locally, its hard for them to discriminate self-similar surface parts, such as planar patches, spheres and cylinders. Increasing the radius of influence increases the discrimination capabilities, but makes the descriptors more sensitive to missing surface parts and clutter. We will show that our approach has no problem with such self-similar objects. Also, we do not require a postprocessing step such as RANSAC for grouping the correspondences. Finally, descriptor-based approaches require a dense surface representation for calculating the features, while our method efficiently matches objects in sparse point clouds. This allows our method to be much more efficient in terms of detection time than descriptor-based methods

11 Point Pair Feature based Object Detection for Random Bin Picking

INTRODUCTION

The automatic handling of objects by industrial robots iscommon practice. However, if the objects pose is unknownbeforehand, it needs to be measured. Indeed, vision-guidedindustrial robots are one of the key ingredients of state-of-the-art manufacturing processes. Everyone is aware of the factthat future production processes will be increasingly flexibleand less labor intensive. Purely mechanical singulation instal-lations, such as vibration feeders, no longer meet flexibilityrequirements or are no longer profitable, and manual work isbecoming more expensive. One very cost-effective and flexiblesolution is the supply of parts in bulk, as illustrated in Figure 1and Figure 2, from which industrial robot arms pick out theobjects one by one in order to feed them to the rest of themanufacturing chain. This application is referred to as randombin picking. The goal is to locate one pickable object instanceat a time and determine its six degree of freedom (6D) pose, sothat the robots end effector can be moved towards the objectand grasp it. This paper is focused on the object detectionand localization task of such an industrial random bin pickingapplication.

Pose estimation is a widely researched computer visionproblem and various solutions based on 2D images, rangeimages or 3D pointclouds exist. However, very few are suitedfor the specific conditions of the real-world industrial bin-picking application at hand. As will be detailed in the nextchapters overview of relevant object detection algorithms,the lab examples mostly studied in literature differ quite alot from real industrial bin picking scenarios. In the lattercase, objects with a wide gamut of characteristics are en-countered: from nicely textured boxes to smooth shiny metaparts and from complex three dimensional free form shapes torotationally-symmetric bolts. No satisfactory general solutionto this problem exists yet. In this paper, point pair featuresare studied as a versatile object detection technique in randombin picking. Moreover, combined with the simple but verypowerful heuristic search space reduction that is proposedin this paper, the techniques computational demands remainwithin manageable bounds. We also propose a generic methodthat enables to use industrially available CAD models of theobjects to be detected as input to our detection pipeline.

In literature, a hodgepodge of different evaluation mecha-nisms for random bin picking are used. This paper proposesa new universal way of evaluating pose estimation algorithmsfor random bin picking, necessary for fair comparison acrossdifferent approaches. It consists of a completely automated procedure for the generation of realistic synthetic scenes andthe evaluation of the detection algorithm. As the procedureis automatic, it can be used in a closed loop to optimizethe detection algorithms parameters. The goal is to achieveoptimal performance across a set of widely varying objects.The remainder of this paper is organized as follows. Sec-tion II gives an extensive overview of the different 3D objectlocalization techniques for random bin picking that are proposed in literature, as well as the detection evaluation methodsavailable. Our point pair feature-based random bin pickingapproach is introduced in section III, composed by the datasetpreprocessing, object detection, heuristic search space reduction and evaluation steps we propose. Experimental results onrepresentative industrial objects are presented and discussed insection IV, and the conclusions follow in section V

RELATEDWORK

An extensive review of the state of the art in 3D objectdetection and pose estimation algorithms is provided. Thissection is split into a part discussing the point pair featurebased techniques and a part discussing algorithms based onother representations.A. 3D Object Detection MethodsIn some simple cases the problem of detecting and esti-mating the pose of objects can be addressed by segmentingthe scene and applying a global feature descriptor to thesegmented parts in order to recognize one of the segmentsas being the considered object. However, in our random binpicking case, this approach will not work, as it is not possibleto reliably segment an object in the presence of significantclutter and occlusion.A more sophisticated approach is to detect and describecertain parts or features of the object. Some techniques havebeen proposed to detect 3D objects from regular imagesbased on representations such as: 2D keypoints [1], 2D edgetemplates [2] [3] or line descriptors [4] [5]. Other techniqueswork on 3D representations such as: shape primitives (planes,cylinder, sphere, superquadrics, etc.) [6], 3D keypoints [7] [8][9], range image templates [10] or color gradient and normaltemplates [11] [12] [13].All these detection methods either create templates fromseveral object views, or extract some kind of features. Animportant downside to the methods relying on multiple objectviews is that they require a large amount of dense templatesand as such, are computationally expensive to match. There aretwo important issues with feature based methods, the first isthat they are not very general: they can only represent objectsthat contain the specific type of feature they use. Another issueis that the feature descriptors (e.g. 3D keypoint descriptors) arequite sensitive to noise.

Point Pair FeaturesIn the previous section methods relying on several typesof features were discussed. A lot of early work focusedon describing the relations between a set of features of an object. The feature vector used in point pair features (seeSection III-B) is very similar to some of these formulations[17], however, the important difference is that point pairfeatures are calculated for all point pairs on a mesh, not justfor specific features. This means they can be used to representfree form objects.Point pair features were first introduced for 3D objectclassification [18]. A histogram of the features occurring on asurface allowed to classify the object. This technique was latercombined with a clustering algorithm to allow the detectionof objects in pointclouds [19]. An efficient procedure forstoring the features in hashtables and calculating matches wasproposed by Winkelbach et al. [20], which was later extendedto an object detection technique [21] [22].An efficient voting scheme to determine the object pose wasdescribed by Drost et al. [23]. The same authors later proposeda modified point pair feature using only points paired withgeometric edge pixels, which are extracted from multimodaldata [24]. This makes the method suitable for object detection(instead of recognition).Papazov et al. [25] [26] reduced the dimensions of thehashtable and the number of points to match by only usingpoint pairs within a certain relative distance range from eachother. Kim et al. [27] extended the point pair feature with avisibility context, in order to achieve a more descriptive featurevector. Choi et al. proposed the use of more characteristic andselective features, such as boundary points and line segments[28] and extended the feature point vector with a color feature[29] [30]. A method to learn optimal feature weights wasproposed by Tuzel et al. [31]. Several extensions are proposedby Birdal et al. [32]: they introduce a segmentation method,add a hypothesis verification stage and weigh the voting basedon visibility.

12 Pose Estimation of Rigid Object in Point Cloud

INTRODUCTION

6D Pose estimation is an important and challenging task in the field of robot vision, which is widely used in military guidance, visual navigation, robot, intelligent transportation, public safety and son on. In the fields of driving development in this area, robotics, in particular, has a powerful need for computationally efficient approaches, as the accurate information about the structure of the scene and the objects are required to perform operational tasks. One of the biggest limit of a normal camera is the difficulty of extracting depth information from the recorded data, However, in the last year, the depth camera has become a popular and accessible tool for acquiring information about the environment. Depth data can be got from stereovision camera, LIDAR device and so on. Using the depth information to estimate the object pose is available and more precisely. The research on the pose estimation issue involves a wide range of algorithms and approaches. The mainly vary according to the features used to describe the object, the mathematical approach to solve the problem and the format of the data given as input to the algorithm representing the model of the object to find. The localization of an object in a 3D scene is mainly addressed by using features descriptors which univocally identity an object or a part of the object. In particular with features we can refer both to the result of a neighborhood operation applied to the image (such as FPH and FPFH) [1] or to specific structures of the image (edges, corners, blobs, ridges, shape).

However, sometimes the data acquired by 3D camera is pretty rough and the number of points on the object is small. Besides that, the resolution of point cloud data is low that the key point is not consistent. So, in this paper, using (l60x 120) resolution point cloud dataset, we present a method that can perform well for pose estimation. So, in this paper, we present a method based on iterative closest point algorithm. After obtaining the 3D point cloud of scene, we remove the outlier points using StaticalOutlierRemoval filter, which make more accurate registration of the object. Then, we detect the object from the background using Euclidean cluster algorithm. When object caught, the registration between two adjacent images is done, so that we can get the six degrees information of the target.