中京大学博士審査学位論文

大学院情報科学研究科

論文題目： Real-Time 2D/3D Object Detection and Pose Estimation based on Template Matching

（テンプレートマッチングに基づく 2 次元及び 3 次元リアルタイム物体位置姿勢認識に関する研究）

2018 年 10 月 8 日申請

小西 嘉典

CHUKYO UNIVERSITY

DOCTORAL THESIS

Real-Time 2D/3D Object Detection and Pose Estimation based on Template Matching

---

Author:
Yoshinori KONISHI

Supervisor:
Dr. Manabu HASHIMOTO

A thesis submitted in fulfillment of the requirements
for Doctoral Degree of Computer Science

in the

Graduate School of Computer and Cognitive Sciences

March 12, 2019

Chukyo University

Abstract

Graduate School of Computer and Cognitive Sciences

Doctoral Degree of Computer Science

Real-Time 2D/3D Object Detection
and Pose Estimation
based on Template Matching

by Yoshinori KONISHI

2D/3D object detection and pose estimation is one of the essential techniques of computer vision and is critical for various real applications such as factory automation (FA) and autonomous driving. The object detection and pose estimation is classified into broad two categories, one is general object (class) detection and pose estimation and another is specific object (instance) detection and pose estimation. The target of this thesis is specific object detection and pose estimation which is mainly used in FA applications such as visual inspections and robotic manipulations.

Three kinds of algorithms for specific object detection and pose estimation are required to cover various applications. The first is 2D object detection and pose estimation of planar objects on conveyors and tabletops. The pose of the object is constrained by planes and the algorithm estimates 4 parameters (X/Y translations, in-plane rotations and scales) from a monocular image. The second is 3D object detection and pose estimation from a monocular image. This estimates rough 3D position and pose (6 parameters - X/Y/Z translations and rotations) mainly for visualization in AR/MR applications where a small and fast monocular camera is preferred. The third is 3D object detection and pose estimation from 3D point clouds captured by a 3D (range) sensor. This estimates precise 3D object position and pose (6 parameters) mainly for robotic manipulations.

The algorithms for specific object (instance) detection and pose estimation are further categorized into three kinds of approaches, a global descriptor (template matching) based approach, a local descriptor based approach and a learning based approach. The global descriptor based approach can handle any kinds of objects and is robust against background clutters, but it is fragile to occlusions and transformations. The local descriptor based approach is robust against occlusions and transformations, but can only be applied to the objects where rich features (textures and shapes) are extracted. The learning based approach is superior to the former two approaches in performance, but this requires large training dataset for each object and scene. The target of this thesis is a research on the practical algorithms which can be applied to real FA applications. In these applications, many of the target objects are rigid, occluded objects are not inspected or grasped, and it is almost impossible for customers to collect large dataset for each object and scene. For these reasons, the global descriptor (template matching) based approach is employed in this thesis.

Proposed Algorithm 1: It has been shown that the template matching based on discretized gradient orientations could handle texture-less objects. Though the matching conditions based both on gradient positions and orientations are strict and robust against background clutters, the similarity scores decrease largely even when the appearance of target object is slightly changed. To tackle this problem, we propose COF (Cumulative Orientation Feature) which is robust to appearance changes induced by object pose changes and at the same time is enough discriminative to detect target objects against cluttered backgrounds. At first, many images are generated based on 2D geometric transformations of a model image using randomized parameters for X/Y translations, rotation angles and scales. Then orientation histograms are calculated at each pixel and pixel-wise dominant orientations are extracted as features. Our proposed method was evaluated on publicly available dataset and achieved higher detection rate and faster speed compared to state of the art.

Proposed Algorithm 2: The 3D object detection and pose estimation based on the template based approach tends to be slower when the number of templates amounts to tens of thousands for handling a wider range of 3D object pose. To alleviate this

problem, we propose a novel image feature and a tree-structured model. Our proposed perspectively COF (PCOF) is developed from COF and extracted from randomly generated 2D projection images from a 3D CAD, and the template based on PCOF explicitly handle a certain range of 3D object pose. The hierarchical pose trees (HPT) is built by clustering 3D object pose and reducing the resolutions of templates, and HPT accelerates 6D pose estimation based on a coarse-to-fine strategy with an image pyramid. In the experimental evaluation on our texture-less object dataset, the combination of PCOF and HPT showed higher accuracy and faster speed in comparison with state-of-the-art techniques.

Proposed Algorithm 3: We propose PCOF-MOD (multimodal PCOF), balanced pose tree (BPT) and optimum memory rearrangement for a coarse-to-fine search in order to make the template based 3D object detection and pose estimation from a RGB-D image faster. Firstly, PCOF-MOD is developed from PCOF by adding the discretized orientations of surface normals. As with PCOF, the model templates of PCOF-MOD explicitly handle a certain range of 3D object pose and the fewer number of templates can cover wider range of 3D object pose. Secondly, a large number of templates are organized into a coarse-to-fine 3D pose tree (BPT) in order to accelerate 6D pose estimation. Predefined polyhedra for viewpoint sampling are prepared for each level of an image pyramid and 3D object pose trees are built so that the number of child nodes of every parent node are almost equal in each pyramid level. Lastly, two kinds of binary features at the lower pyramid levels are rearranged so that nearby features are linearly aligned on a memory and these vectorized features are processed at one time using SIMD instructions. In the experimental evaluation of 6D object pose estimation on publicly available tabletop and our own bin picking dataset, our template based method showed higher accuracy and faster speed in comparison with the existing techniques including recent CNN based methods.

概要

画像や3次元点群から物体の2次元あるいは3次元の位置と姿勢を認識する技術は，工場自動化や自動運転など様々なアプリケーションで必要とされる画像認識の基本技術の一つである．物体位置姿勢認識技術は大きく二つに分類することができ，一つは顔や人体など物体クラスを対象とする一般物体位置姿勢認識，もう一つは特定の物体（インスタンス）を対象とする特定物体位置姿勢認識である．本論文では工場での外観検査やロボットによる把持・組立に用いられることが多い特定物体位置姿勢認識を対象とする．

特定物体位置姿勢認識はアプリケーションによって三種類のアルゴリズムが必要であると考えられる．一つ目はコンベアや机の上に置かれた平面的な物体の位置姿勢を認識するアルゴリズムである．この場合の物体姿勢変化は平面上に限定されるため，単眼カメラのみを用いて並進（XY成分）、回転、スケールの4つのパラメータを推定する．二つ目はAR/MRなど3次元表示を目的とした物体の概略3次元位置姿勢を認識するアルゴリズムである．この場合は処理速度や可搬性の観点から単眼カメラを用い，並進（XYZ成分）、回転（XYZ成分）の6つのパラメータを推定する．三つ目はロボットによる把持・組立等を目的としたより高精細な3次元位置姿勢を認識するアルゴリズムである．高精度認識のため距離センサにより計測した3次元点群を入力として用い，3次元位置姿勢の6つのパラメータを推定する．本論文ではこれら三つのアルゴリズムに関し，テクスチャ無しや単純形状を含むあらゆる物体に適用可能で高速かつ外乱に対してロバストな手法を提案する．

特定物体位置姿勢認識アルゴリズムは，大きく三つの手法に分類することができる．一つ目はテンプレートマッチングに基づく手法，二つ目は局所特徴量に基づく手法，三つ目は機械学習に基づく手法である．テンプレートマッチングに基づく手法は，あらゆる物体に適用可能で外乱に対してロバストであるが変形や隠れに弱い．局所特徴量に基づく手法は，隠れや物体の変形に対してロバストであるがテクスチャや形状などの特徴量が多く抽出できる物体にしか適用できない．機械学習に基づく手法は性能面では前者二つの手法と比較して優位に立っているものの，対象となる物体や背景について多くの学習データを収集する必要がある．本論文では工場自動化やロボットビジョンといった実アプリケーションに適した特定物体位置姿勢認識アルゴリズムの研究を目的としている．こういったアプリケーションにおいては対象とする物体は幅広いがその多くは剛体である，隠れている物体は検査や把持の対象とならない，物体や環境ご

とに大量の学習データを収集することは現実的でないといった理由から，本論文ではテンプレートマッチングに基づく手法を採用した．

提案手法 1：テクスチャの少ない物体にも適用可能な 2 次元物体位置姿勢認識手法として，輝度勾配方向特徴量を用いたテンプレートマッチングが提案されてきた．しかし勾配方向を照合条件として用いることで複雑背景下においても頑健な照合が可能である一方，対象物体自身の見えがわずかに変化した場合には照合スコアが大きく低下してしまうという課題があった．そこで本論文では，物体の姿勢変動による見えの変化を考慮した累積勾配方向特徴量（COF: Cumulative Orientation Feature）を提案する．提案手法ではまず，一定範囲内でランダムに発生させた平行移動，回転角度，スケールパラメータを用い，1 枚のモデル画像に対して幾何学的変換を適用して多数の画像を生成する．次に各画像において算出した量子化勾配方向特徴量を用いて画素毎に勾配方向ヒストグラムを作成し，頻度の大きい勾配方向のみを用いて特徴量を抽出した．実際の画像に対して照合処理行い，提案手法が対象物体と背景を識別する性能を維持したまま物体自身の見えの変動を許容できることを確認した．またテクスチャレス物体の公開画像データセットを用いた 2 次元物体位置姿勢認識の実験を行い，提案手法が認識正確性及び処理速度において既存手法を上回ることを示した．

提案手法 2：単眼カメラ画像から 3 次元物体位置姿勢を高速に認識する手法においては，認識対象となる 3 次元姿勢範囲が広い場合に照合に用いるテンプレートの数が膨大になり処理速度が低下するという課題があった．この課題に対して本論文では，透視投影に基づく累積勾配方向特徴量（PCOF: Perspectively COF）と階層的姿勢探索木（HPT: Hierarchical Pose Tree）の二つの手法を提案する．PCOF は COF を拡張した特徴量であり，対象物体の 3 次元 CAD を様々な視点から見た 2 次元投影画像を生成して特徴抽出を行う．このことにより，3 次元姿勢変化による対象物体の見えの変化に対する許容性と複雑背景に対する頑健性の両立を実現した．HPT は様々な視点において作成された大量のテンプレートに対し，類似度に基づいたクラスタリングとテンプレートの低解像度化を繰り返すことで作成する．HPT を用いて画像ピラミッド上を探索することにより，数万個の 3 次元姿勢候補の中から最も類似度の高いテンプレートを高速に絞り込むことが可能になる．9 種類の金属部品を様々な方向から撮影したデータセットを用いて評価実験を行い，PCOF と HPT を組み合わせた提案手法が 3 次元物体位置姿勢認識の高速性・正確性両面において既存手法を上回ることを確認した．

提案手法 3 : 距離画像や RGB-D 画像から 3 次元物体位置姿勢認識を行う場合においても，単眼カメラからの認識と同様に照合に用いるテンプレートの数が多く処理速度が

遅くなるという課題があった．この課題に対して本論文では，透視投影に基づく RGB-D 累積勾配方向特徴量（PCOF-MOD: Multimodal PCOF），平衡姿勢探索木（BPT: Balanced Pose Tree），特徴量再配置による粗密探索高速化の三つの要素技術からなる高速・ロバストな 3 次元物体位置姿勢認識手法を提案する．一つ目の PCOF-MOD は PCOF にデプス画像特徴量を加えた特徴量であり，一定範囲内においてランダムに設定した 3 次元視点位置から対象物体の 3 次元 CAD を見た場合のデプス画像を多数生成し，それらからデプス勾配方向と表面法線方向について画素ごとに方向ヒストグラムを作成し特徴抽出を行う．これにより，PCOF-MOD は視点を設定した範囲内の 3 次元姿勢変化による見えの変化のみを照合時に許容可能な特徴量となる．二つ目の要素技術である BPT は，画像ピラミッドの階層ごとに頂点数の異なる多面体の頂点を視点位置として使用することで，画像内 2 次元位置の粗密探索と 3 次元姿勢の粗密探索を同時に実施可能とした探索木である．全ての探索木の深さは等しく，親ノードに連結する子ノードの数もほぼ均一であるため探索効率が高いという特徴を備えている．三つ目の特徴量再配置は，画像ピラミッドの最上位階層以外では一つ上の階層で検出された正解候補周辺の画素においてのみ特徴量照合を行うという粗密探索の特性を活用している．即ち，照合対象となる周辺画素の特徴量がメモリ上で連続するように再配置した特徴量マップを作成し，連続データに適用可能な CPU 命令（SIMD 命令）を用いて一括照合を行うことで粗密探索の高速化を実現する．これら三つの要素技術を組み合わせた 3 次元物体位置姿勢認識手法について公開 RGB-D データセットと我々が構築したバラ積み部品データセットにおいて性能評価を行い，提案手法が 3 次元物体位置姿勢認識の高速性・正確性両面において近年の CNN ベース手法を含む既存手法を上回ることを示した．

Acknowledgements

First of all, I would like to thank my supervisor Prof. Manabu Hashimoto, Dean of School of Engineering at Chukyo University for giving me the opportunity to research object pose estimation algorithms at Intelligent Sensing Laboratory. He gave me many advices on writing this thesis, supported and encouraged me to pursue my PhD in parallel with working at the company.

This thesis was written during my employment at Vision Sensing Laboratory of OMRON Corporation in Kyoto, Japan. I received a lot of support while writing this thesis and I am deeply grateful to the former and current director of the laboratory, Mr. Masato Kawade and Mrs. Yuki Hasegawa for that. I would also like to thank my colleagues at the laboratory for helping me collecting the evaluation dataset and implementing a part of my computer programs.

Finally, I would like to thank my parents and sister, Hiroshi, Hiroko and Hitomi who always supported me during my whole life, letting me grow up in a loving environment. An acknowledgment page would be incomplete if I did not mention my wonderful wife Misato for unceasing support, trust, and love. I would also like to thank my children, Yui and Taku who are my little sunshine. They always cheered me up with their smiles and made me feel relaxed at home.

Yoshinori Konishi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Detecting object position and pose is one of the essential techniques in image processing or computer vision and is widely used in various applications. For example, face detection for digital equipments, pedestrian detection for video surveillance, vehicle/obstacle detection for autonomous driving, and detection of mechanical/electronic parts for robotic manipulation (Figure 1.1). Due to these broad-ranging applications, the research on this topic has been extensively conducted over many years.

There are two kinds of researches on object detection and pose estimation. One is detection of generic object or object class, and another is detection of specific object or instance. Face detection [1, 2] and pedestrian detection [3, 4] belong to generic object detection. In recent years, CNN based methods such as Faster R-CNN [5],

Figure 1.1 Typical applications of object detection: robot picking for packaging (left) and pedestrian detection for vehicle safety (right).

SSD [6] and YOLO [7] have achieved remarkable performances on generic object detection. On the other hand, the researches on specific object detection have also been extensively conducted for many years in the context of image retrieval [8], augmented reality (AR) [9], robotic grasping [10] and so on. Especially in the applications including physical interactions with real world such as AR and robotic grasping, estimation of object pose as well as object position is crucial.

This thesis pursue research on object detection and pose estimation of specific object or instance. Object detection and pose estimation are further divided into 2D and 3D. In 2D object detection and pose estimation, four parameters of X/Y positions, a rotation angle and a size in an image coordinate system should be detected from an input image. For example in common applications of factory automation, mechanical/electronic parts and products on a conveyor belt are picked by robots or visually inspected by cameras. In such cases, the degree of freedom for the target objects is three (X/Y translations and rotation) and these parameters are detected using 2D object detection and pose estimation algorithm. The reference model for 2D object detection and pose estimation usually made from a captured image or 2D CAD of the target object.

On the other hand, six parameters of X/Y/Z positions and rotation angles should be detected when the pose of objects are not constrained. The robotic grasping and AR applications usually require 3D position and pose of target objects. 3D object position and pose (6 degrees of freedom pose) can be recovered from a single monocular camera and from a 3D sensor or a RGB-D sensor, and the reference model for 6-DoF pose estimation is usually made from 3D CAD or real data captured from multiple viewpoints. The terms 3D sensor and RGB-D sensor here include various measurement principles such as passive/active stereo, 3D laser scanning, phase shifting and time-of-flight. Although the precision and robustness of detected position and pose are improved by using depth information from 3D sensors, they are usually bigger, heavier, slower and more expensive than monocular cameras. Moreover, they are more sensitive to illumination conditions and object materials/surfaces, and require cumbersome 3D calibrations. For those reasons, a single monocular camera and a 3D sensor for 3D object detection and pose estimation are used in their appropriate situations or applications. More concretely, a single monocular camera is used

for detecting rough 3D position and pose of isolated objects in grasping by consumer robots and AR applications. A 3D sensor is often used for detecting precise 3D position and pose under cluttered conditions in grasping and assembling by industrial robots.

To summarize, three algorithms are required for specific object detection and pose estimation.

- 2D object detection and pose estimation using a monocular camera which is suitable for detecting objects whose pose is constrained by a plane such as a conveyor belt in a factory.
- 3D object detection and pose estimation using a monocular camera which is suitable for detecting rough pose of isolated objects.
- 3D object detection and pose estimation using a 3D sensor which is suitable for detecting precise pose under cluttered conditions.

In this thesis, these three algorithms are researched and our proposed algorithms are evaluated. Although our proposed algorithms are not designed for a specific application, we often assume the algorithms are used in the applications for factory automation (FA) including robotic applications because the current market of machine vision for FA is huge (approximately 20 billion in USD) and will keep on expanding in future.

1.2 Problem Statement

In real applications of specific object detection and pose estimation, the algorithm should handle various target objects those are seen in various scenes such as home, office and factory. Additionally, the algorithm should work under various conditions and the time for detection is important for the usability and productivity of the applications. Therefore, the requirements for the algorithm are three-fold: target variation, robustness and detection speed.

Figure 1.2 Various target objects in factory scenes such as textured/texture-less, simple-/complex-shaped, shiny/matte/translucent/transparent surface.

Target Variation

It is desirable that a single algorithm handles as many objects as possible. As an example of real applications, the possible target objects in factory automation are shown in Figure 1.2. These packages, foods and mechanical/electronic parts are grasped/assembled/inspected by robots and cameras.

The factors which influence the performance of pose estimation algorithm are texture on surface, shape, material and size of the objects. Regarding the texture on surface, there are rich-textured objects such as the packaged food, tube and bottle where rich image features are extracted. Contrastingly, many of the objects in FA applications has little texture such as the connector, bearing and screw where the object contours are only clues for object pose estimation.

Similar to textures on object surfaces, rich 2D/3D features are extracted from the complex-shaped objects such as the connector and gear, and poor features are extracted from the simple-shaped objects such as the bearing and capacitor. Regarding object materials, metallic, translucent and transparent objects are difficult for the algorithm to detect. The appearance of these objects heavily depends on lighting conditions due to the light reflection and absorption around the object surface. This phenomena also make it hard for 3D sensors to measure the distance to the surfaces. For these reasons, 2D/3D features of these materials are unstable and fragile to illumination conditions.

Lastly, small-sized objects like the capacitor are also difficult for object detection algorithms simply because the extracted features are too few to discriminate target objects from backgrounds and recognize subtle differences between objects of different classes or pose.

These factors on target variations are combined and influence on each other. Additionally, illumination conditions and background clutters complicate these factors and degrade the performance of pose estimation algorithms.

Robustness

Partial occlusions, illumination conditions and background clutters usually pose problems to specific object detection and pose estimation. However in FA applications, partial occlusion is not much of a problem for object detection because occluded objects are difficult to be grasped and inspected. The occluded objects are ignored or resolved by other devices. Contrastingly, the robustness against illumination conditions and background clutters is crucially important for FA applications because many of the target objects are small, texture-less, simple-shaped and metallic. Only poor 2D/3D features are extracted on small/texture-less/simple-shaped objects and this degrades the performance under background clutters due to many false positives on the background (top left of Figure 1.3). The reflection light on metallic objects often creates false edges and textures, and this induce false positives inside the object area (top right of Figure 1.3).

The robustness against the changes in appearance and shape from reference models is also important. Although many of the target objects in FA applications are

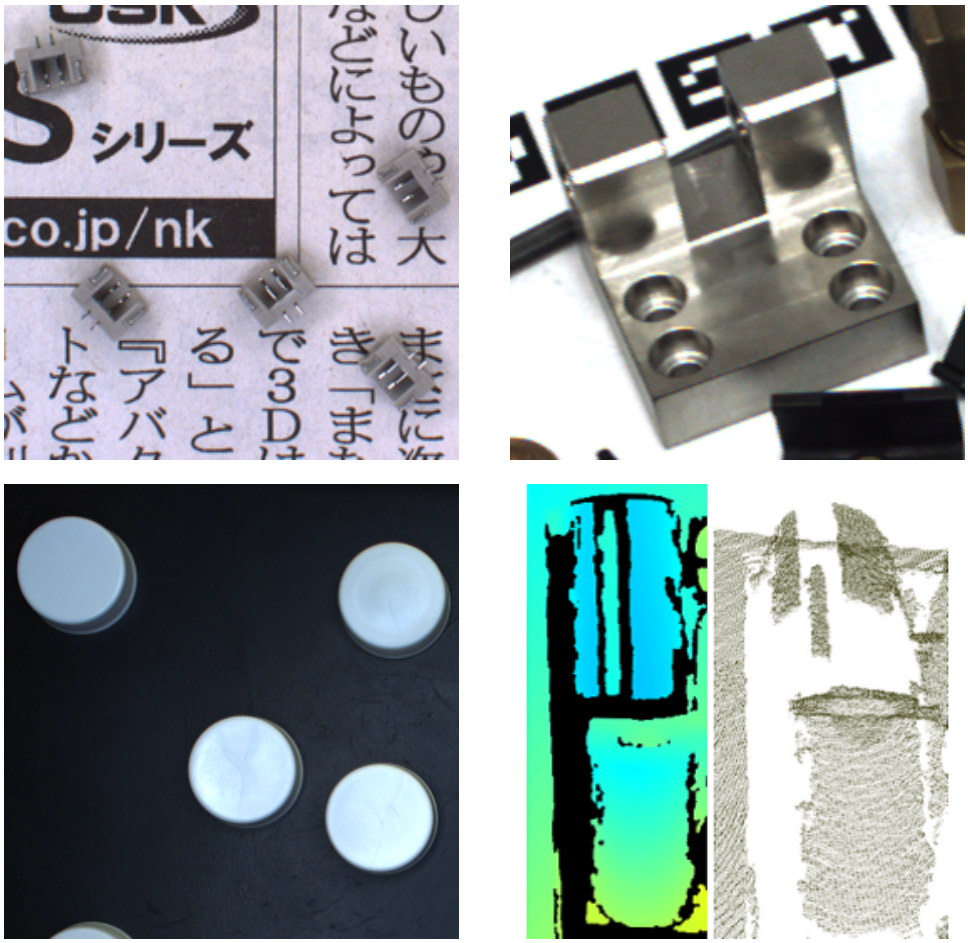


Figure 1.3 Example images of disturbances to specific object detection. 1st row: Small connectors on newspaper (cluttered background) and the reflection on metallic surface. 2nd row: Perspective distortion and the 3D measurement errors on a translucent tube.

rigid, the difference between reference model and the input image or 3D point cloud occur by various factors. For example, each part and product is slightly different in its shape from CAD by poor machining accuracy, the appearance is perspectively distorted (bottom left of Figure 1.3), and the measured 3D point clouds contain noise and errors (bottom right of Figure 1.3). The algorithm should detect target objects ignoring these differences and errors.

Detection Speed

Detection speed determines the productivity of factory lines and the faster is better. Although GPUs are getting faster and cheaper in recent years, the GPUs which meet the requirements of FA applications such as durability for 24/7 running and

long-period supply have not been available yet. Moreover, embedded systems are preferred for the usability and compact size of machine vision products. Thus the algorithm should run fast even on low-end CPUs using high-resolution cameras under background clutters.

Not only the time for detection but also the time for training is important for real applications of specific object detection. The training of classifiers for generic object detection is executed offline using huge training samples because the object classes to be detected in the applications are determined beforehand. However, the target objects in the applications of specific object detection such as FA applications are different in every factory line. Therefore, it is difficult for engineers to train all models beforehand and the model for each application should be trained on-site. It is preferred in real applications that the model is trained only from a captured image or CAD of the target object and the training takes a few minutes at the longest.

1.3 Contribution of the Thesis

This thesis presents fast and robust 2D/3D detection and pose estimation of specific objects or instances mainly for factory automations including robotic applications. Three algorithms are proposed to handle various situations: 2D detection and pose estimation from a monocular image, 3D detection and pose estimation from a monocular image and 3D detection and pose estimation from a RGB-D image. Though these three algorithms are designed for different purposes and scenes, all of these are based on some common technical components. Fist of all, our proposed algorithms are all based on template matching which has many advantages in real applications of FA and robotics over other methods like local descriptor and machine learning (the details will be discussed in Chapter 2). Moreover, our algorithms consist of three common technical components. First, 2D/3D binary features which are fast to compute similarity scores, and robust against background clutters, lighting conditions and small 2D/3D pose changes. Second, tree-based data structures for model templates which are efficient for search in 2D/3D position and pose space, and costs small memory footprint. Third, a memory rearrangement algorithm which makes a coarse-to-fine search faster using SIMD instructions.

Due to our template matching based approach with fast and robust features, our algorithms can handle wider ranges of objects such as textured, texture-less, simple-shaped, complex-shaped, shiny, matte objects. They also are robust against background clutters, illumination changes and small changes of object pose. Though template based approaches tend to be slower due to the large numbers of model templates to be matched, our tree-structured model and memory rearrangement make them faster and realize real-time processing. Additionally, 2D/3D models are trained only from a image or 3D CAD and the training takes less than a few minutes. These enable users to train the models of their target objects on-site.

Typical detection results of our proposed three algorithms are shown in Figure 1.4. 1st row shows the input and result image of our 2D detection and pose estimation algorithm. The multiple texture-less objects (air nozzle) are detected under background clutters. 2nd row shows the input and result image of our 3D detection and pose estimation algorithm from a monocular image. The texture-less and shiny object (side clamp) is detected under background clutters and partial occlusion. 3rd row shows the input depth and grayscale image where the result of our 3D detection and pose estimation from a RGB-D image is rendered on. The multiple translucent tubes whose measured 3D point cloud are partially incomplete are detected correctly.

The key contributions of this thesis are as follows:

- 2D/3D fast and robust features which tolerate only the small changes in 2D/3D object pose without degrading the robustness against background clutters. First, the feature is introduced in 2D object detection and pose estimation. The 2D feature is based on the discretized orientation of image gradients and is robust to the appearance changes by object pose changes (X/Y translation, in-plane rotation and object scale). This feature is developed to handle object pose changes in X/Y/Z translation and rotation for 3D object detection and pose estimation from a monocular image. Then it is applied to 3D object detection and pose estimation from a RGB-D image by adding the discretized orientations of surface normals as a 3D feature.

- Tree-based data structures which enable efficient search both in position and

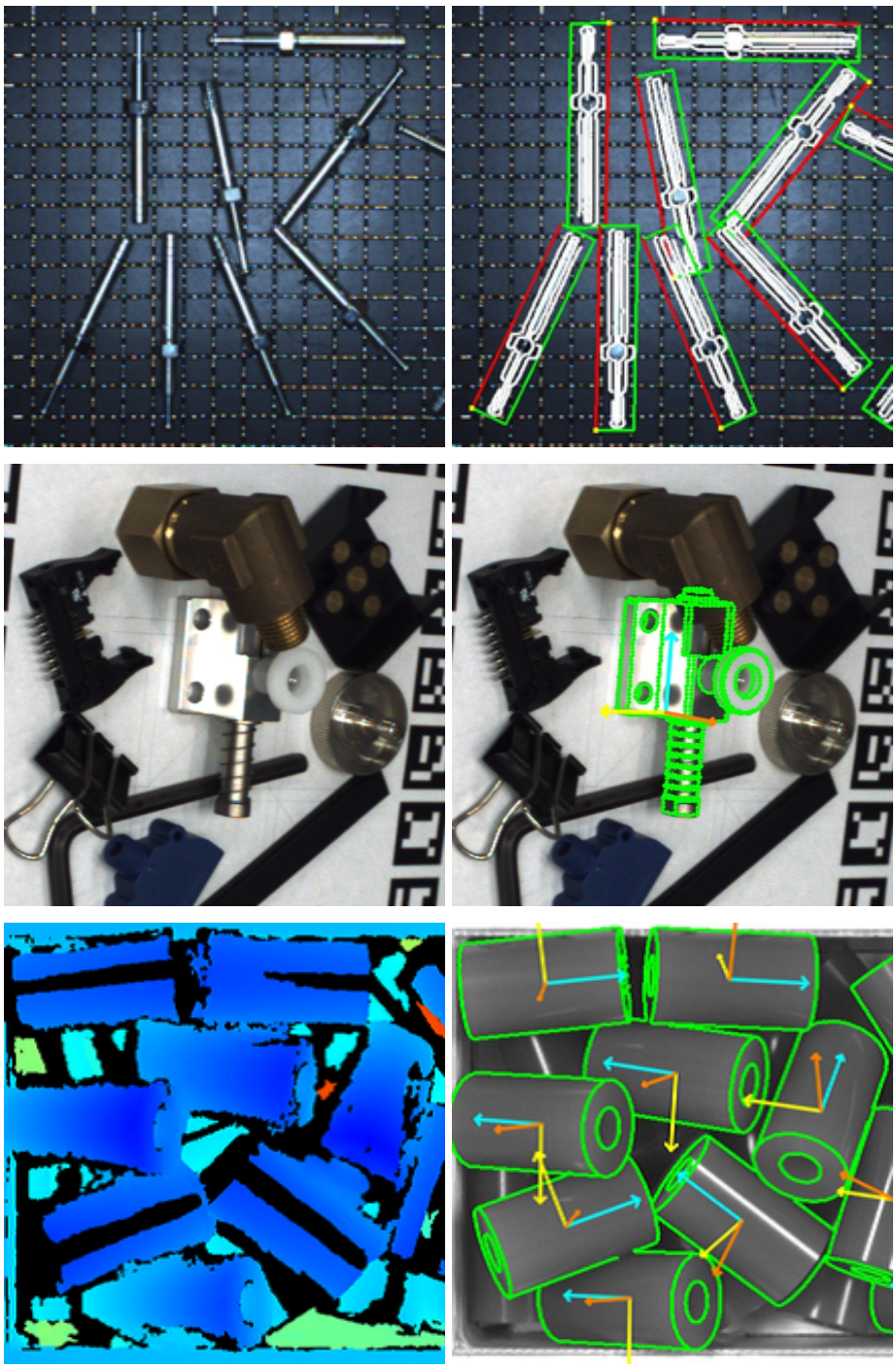


Figure 1.4 Example images of the detection results by our proposed three algorithms. Object edges and bounding boxes/3D coordinate axes are drawn based on the detected pose on the right panel. 1st row: Air nozzles are detected under background clutters by our 2D object detection and pose estimation algorithm from a monocular image. 2nd row: The shiny mechanical part is detected by our 3D object detection and pose estimation algorithm from a monocular image. 3rd row: The translucent tubes where the measured point clouds are deficient are detected by our 3D object detection and pose estimation algorithm from a RGB-D image.

pose are proposed. The data structure is firstly introduced in 3D detection and pose estimation from a monocular image in order to boost the detection speed using large numbers of templates. It is based on the combination of a coarse-to-fine search with an image pyramid for efficient search in an image space and pose clustering for efficient search in 3D pose space. This is optimally modified for 3D detection and pose estimation from a RGB-D image where 3D feature can be extracted and is more discriminative than 2D feature for recognizing 3D pose. Though these data structures are not applied to 2D object detection and pose estimation in this thesis, they are also effective for fast 2D object detection.

- Optimum memory rearrangement is introduced in order to make a coarse-to-fine search faster in 3D detection and pose estimation from a RGB-D image. The feature map which is optimized for template matching using SIMD instructions is created so that two types of features of neighboring pixels (e.g. 4 by 4) are linearly aligned. Though this technique can be applied to any detection algorithms which utilize a coarse-to-fine search with an image pyramid, this is most effective in 3D object detection from a RGB-D image because large numbers of templates for two types of features should be scanned within a image.

- Complete pipelines for three types of specific object detection including pose refinement are implemented. They are evaluated on publicly open dataset and our own dataset which resembles realistic FA and robotic scenes. Our proposed algorithm and the whole pipelines are thoroughly compared with existing methods, and our (dis-)advantage over them and the remaining problems are discussed.

Though our proposed algorithms are evaluated and compared with the existing work mainly on FA and robotic scenes or dataset, the algorithms are also useful and effective for other applications such as augmented reality and medical imaging systems due to their preciseness, fast speed and robustness against background clutters and illumination changes. Our three main technical components: the fast

and pose-robust features, the tree-structured models and the optimum memory rearrangement for a coarse-to-fine search can be independently used and effective for various applications.

1.4 Outline of the Thesis

The remaining of this thesis is organized as follows: Chapter 2 summarizes current state-of-the-art techniques for 2D/3D object detection and pose estimation of specific objects. The researches on 3D object detection and pose estimation include both of those from a monocular image and a RGB-D image. The existing research on these topics are surveyed and categorized into some groups. The pros and cons of these groups are listed and compared. After determining one group which is most suitable for FA applications, the main limitations are identified.

Chapter 3 describes our proposed algorithm of 2D object detection and pose estimation. In addition to the main idea on 2D image feature which is robust against the changes in object pose, whole object detection pipeline including pose refinement is implemented. The robustness of our proposed image features is compared with existing orientation features. The accuracy and speed of our object detection pipeline are evaluated on two public dataset and compared with existing methods. The estimation errors are also evaluated on our own dataset.

Chapter 4 presents our proposed algorithm of 3D object detection and pose estimation from a monocular image. The 2D image feature proposed in Chapter 3 is modified to be robust against the changes in 3D object pose. The robustness of the proposed feature is evaluated and compared with existing orientation features. Additionally, the tree-structured model which enables the search both in 2D image and 3D pose space faster simultaneously is proposed. Our 3D object detection and pose estimation algorithm consists of the image feature and data structure. Our detection pipeline including pose refinement based on 2D edges is evaluated and compared with existing methods on our own dataset for FA applications.

Chapter 5 explains our proposed algorithm of 3D object detection and pose estimation from a RGB-D image. The 2D image feature proposed in Chapter 4 is further modified to handle RGB-D images by adding the orientations of surface normals as

3D features. The tree-based data structure in Chapter 4 is also modified so that the searching is more efficient by building the balanced trees. Moreover, the memory rearrangement algorithm for faster coarse-to-fine search is introduced. The whole pipeline includes these three components and ICP-based 3D pose refinement. Our pipeline is evaluated on two public dataset for tabletop scene and on our own built dataset for bin-picking scene.

Chapter 6 summarizes and gives a conclusion of the thesis. Furthermore, an outlook on future work is provided.

Chapter 2

Related Work and State of the Art

This chapter presents existing work on 2D/3D object detection and pose estimation of specific object instances. The researches have been often divided into two groups, global- and local-descriptor based methods. In recent years, many learning-based approaches have been proposed for specific object detection. Though the learning-based methods include both global- and local- descriptor based methods, we categorize these researches into new third group because they have different properties from other two groups and often show improved performance. After the details of each group is described and surveyed, we discuss pros and cons of these approaches and the suitabilities for real applications in factory scenes.

2.1 Global Descriptor Based Approach

The global descriptor based method uses a target object as a whole for detection and pose estimation. The descriptor is compared with the segmented region or scanned in an image, and the position and pose are detected. This approach is also known as template matching or pattern matching and has been intensely studied since the beginning of image processing research. The research started with 2D object detection and pose estimation from a monocular image in 1960s. The research on 3D object detection and pose estimation from a monocular image appeared in 1980s and then the research using a depth image appeared in 1990s. These researches are surveyed in the following two subsections, one is on 2D detection and pose estimation and another is on 3D detection and pose estimation.

2.1.1 2D Object Detection and Pose Estimation

The most basic image feature is a brightness value at each pixel and the vector of brightness values is compared between a model image and an input image based on similarity measures such as normalized cross correlation (NCC), sum of absolute difference (SAD) and sum of squared difference (SSD). To localize the position and pose of a target object, many templates of various sizes and rotation angles are exhaustively scanned in an input image (full search). Many full search equivalent algorithms were proposed to speed up this detection process and these are well surveyed in [11]. Moreover, many approximating full search algorithms have also been proposed to make detection faster, and some of them were based on hierarchical image pyramids [12], simple window testing [13], hash tables [14] and nearest neighbor search [15].

The brightness value of each pixel is heavily influenced by illumination conditions and the performance of object detection degrades when the brightness patterns on object surfaces are changed due to shadows and reflections. The edge based image features were proposed to improve the robustness against illumination changes. The edges which represent only the shapes and textures of the target object are extracted from a model image and compared with the edges from an input image using the similarity measures like Chamfer distance [16, 17] and Hausdorff distance [18, 19]. Moreover, the similarity measures which use not only the positions of edges but also the directions of their normal vectors were proposed in order to make template matching robust against background clutters [20, 21]. The edges are extracted using edge detection algorithms such as Canny [22] and LSD [23], and their speed and repeatability also influence the performance of object detection.

The direction of image gradient has also been proposed as the image feature which is robust against occlusion, clutter and illumination [24]. The calculation of image gradients based on derivative filters like Sobel and Prewitt is faster and the result is more stable compared to the edge extraction algorithms. Ullah et al. have discretized the direction of image gradients into eight orientations and showed that template matching based on it is also robust against background clutters and illumination changes [25]. Hinterstoisser et al. [26] have proposed DOT (Dominant

Orientation Template) where only the dominant orientations within a certain range of pixel grid were discretized and represented as a binary digit number. It has been shown that DOT was fast to compute the similarity score using logical operations and was robust against small deformations and object pose changes due to the grid-based feature extraction.

The robustness to partial occlusions and object transformations are also critical problems for object detection. The occlusion aware algorithms which were based on occlusion reasoning [27] and RANSAC based hashing [28] have been presented to handle partial occlusions. Though the transformation or deformation invariant template matching e.g. rotation [29], affine transformation [30] and non-rigid deformations [31, 32, 33] have been proposed, these algorithms are not appropriate for the application where the precise object pose is required.

The detection results of the global descriptor based method are shown in Figure 2.1. Our 2D detection and pose estimation algorithm which uses quantized gradient orientations (introduced in Chapter 3) was used for detection. The method can handle textured (1st row) and texture-less objects (2nd row) even on a cluttered background (3rd row). However, the similarity scores become lower when the object is occluded (4th row) and perspectively distorted (5th row).

2.1.2 3D Object Detection and Pose Estimation

The research on global descriptor based approach for 3D object detection and pose estimation has started with monocular image in 1980s. The appearance of a target object from various viewpoints were represented as aspect graph [34], interpretation tree [35], relational graph [36] and aspect tree [37]. The model templates are searched and found in an input image, then 3D position and pose are retrieved from the viewpoint where the model template is made. The matching between models and inputs was done based on line features [38], edges and silhouettes [39], and shock graphs and curves [40]. The search space of 3D object detection and pose estimation is huge (6-DoF) and many algorithms which reduced the search space have been proposed, for example, template matching on the 3D pose manifold [41, 42] and pose estimation of the objects on a conveyor belt [43]. Moreover, the additional equipment and pre-/post-processing algorithms such as multi-flash camera

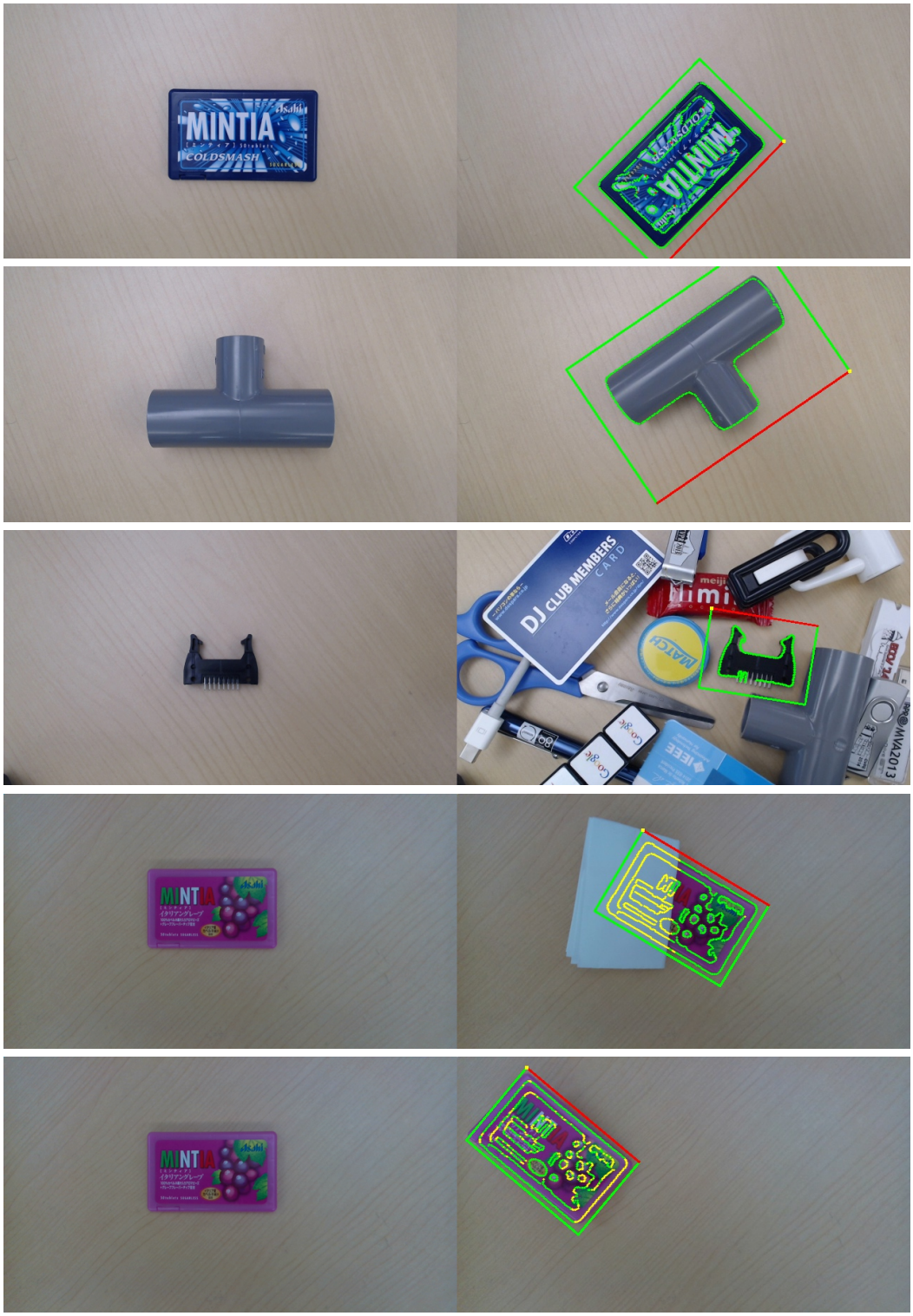


Figure 2.1 Detection results of our global descriptor based method presented in Chapter 3. Top row: A textured package on a simple background. 2nd row: A texture-less pipe on a simple background. 3rd row: A texture-less connector on a cluttered background. 4th row: An occluded package. 5th row: A textured package with perspective distortion. The left images are models and the right images are the detection results where the bounding boxes and extracted edge from the models are drawn. The yellow dots represent the feature points whose correspondences are not found in the input images.

for eliminating the false edges due to reflections and shadows [10], matching of hierarchically segmented contours (gPb) [44, 45] and two-level hypothesis verifications [46] have been introduced.

Full search among 6-DoF pose space only from a monocular image is prone to fail and take long time because the number of model templates for covering full 3D object pose is large and the edge/line features extracted from images are fragile to background clutters. Ulrich et al. [47] proposed pose clustering based on similarity scores between neighboring viewpoints and template matching based on normalized gradient vectors. It was shown that their proposed method was fast enough for covering full 3D object pose and robust against background clutters. Hinterstoisser et al. [48] also presented that their LINE2D algorithm which was based on fast matching of binarized orientation features using vectorized memory map achieved fast and robust 6-DoF pose estimation.

Adding depth information makes 3D object detection and pose estimation more robust against background clutters. Various global descriptors based on 3D shapes of objects have been proposed, for example, surface signatures [49], shape distributions [50], cone curvature [51] and depth gradient images [52]. These descriptors represent the 3D curvatures of object surfaces or relationships between points on surfaces, and the computation is rather complex and takes long time. For robotic applications, more simple and faster, but viewpoint dependent descriptors have been proposed, such as viewpoint feature histogram (VFH) [53] and HOG over depth image [54]. In order for more robust and efficient 3D object detection and pose estimation, VFH was extracted only on clustered stable regions (CVFH) [55], CVFH was computed in unique reference frames (OUR-CVFH) [56], and VFH was combined with a 2D gradient based detector [57].

Hinterstoisser et al. [58, 48, 59] have proposed LINEMOD where discretized surface normals were added to LINE-2D and showed that LINEMOD was more robust against cluttered background and faster than the existing methods. LINEMOD has been extended using hashing [60, 61], matching of GPU-optimized feature vectors for scalability to increasing kinds of objects [62], and compensation for bias of multimodal features for handling simple-shaped objects [63].

2.2 Local Descriptor Based Approach

The local descriptor based object detection is mainly divided into two components, first is keypoint detection and second is descriptor matching. The keypoint is a distinctive point or area in an image, for example, edges, corners and blobs. The local descriptor represents the local image patches or object surfaces around the keypoint, and the similarity of the descriptor is measured between a reference model and an input image. The position and pose are usually recovered from multiple matching results of local descriptors based on pose clustering or Hough voting. For estimation of 3D object pose, it can be recovered by solving PnP problem based on 2D-3D correspondences those are extracted from local descriptor matching [64]. The research on keypoint detection and local descriptor matching have been done from 1980s and their application to object detection and pose estimation has started from late 1990s. These researches are surveyed in the following two subsections, Subsection 2.2.1 for 2D object detection and pose estimation and Subsection 2.2.2 for 3D object detection and pose estimation.

2.2.1 2D Object Detection and Pose Estimation

The research on local descriptor based object detection has been rapidly become popular since the successful and impressive result of SIFT [65]. The local descriptor based method is robust to partial occlusions and deformation due to its locality. Additionally, it is scalable to higher resolution of images and increasing number of object class because the features are extracted only from the interested regions. Then, many successor of SIFT those were faster, more robust and compact like SURF [66] and ORB [67] have been proposed. Various keypoint detectors and local descriptors have been well evaluated and summarized in the survey paper[68, 69, 70].

Many of the local descriptors extract brightness patterns or gradient histograms from local regions. This requires rich textures on every local surfaces of target objects and cannot be applied to texture-less objects. To handle texture-less objects, the local descriptors based on edge orientation histograms [71] and line features [72] were proposed. However, these simple features were less discriminative and fragile to cluttered background. A pair of line segments was used to make the matching

of edge based local descriptor more robust to background clutters [73, 74]. Furthermore, the aggregation of line segments into oriented rectangles [75] and multi-layered binary nets [76] were proposed for more robust and faster matching. These extension of local descriptor to texture-less objects often use line segments as their feature description, and the repeatability and speed of the line segment detector heavily influence on the performance of object detection and pose estimation.

The voting based methods such as generalized Hough transform [77] and geometric hashing [78] are often classified into the local descriptor based approach. However, object detection based on voting from densely sampled features like edges and lines is rather similar to global descriptor (template matching) based approaches because the model features for voting is extracted from whole object area. Therefore, the voting based approaches are applicable to texture-less objects and robust against background clutters, but fragile to partial occlusions and deformations.

The detection results of the local descriptor based method (SIFT [65] was used) are shown in Figure 2.2. The test images are same as in Figure 2.1. The number of detected keypoints and matched descriptors on texture-less objects (2nd and 3rd rows) are much fewer than that on the textured object (1st row). This implies that the local descriptor based object detection tends to fail on texture-less and small objects. The 3rd row shows that there are many false correspondences due to the background clutters and this makes it difficult for the local descriptor based method to handle cluttered backgrounds. On the other hand, there are sufficient number of correct correspondences on the occluded (4th row) and perspectively distorted object (5th row). This is because the local areas are not so changed compared to the whole object.

2.2.2 3D Object Detection and Pose Estimation

The keypoint detectors and local descriptors have been applied to 3D object detection and pose estimation. Collet et al. [79] have proposed 3D pose estimation based on the combination of RANSAC and mean shift from the matched SIFT descriptors. They also proposed optimized framework for object pose estimation where robust performance with iterative feature clustering and partition was achieved [80]. Wagner et al. [81] have presented fast 6-DoF pose detection and tracking on mobile

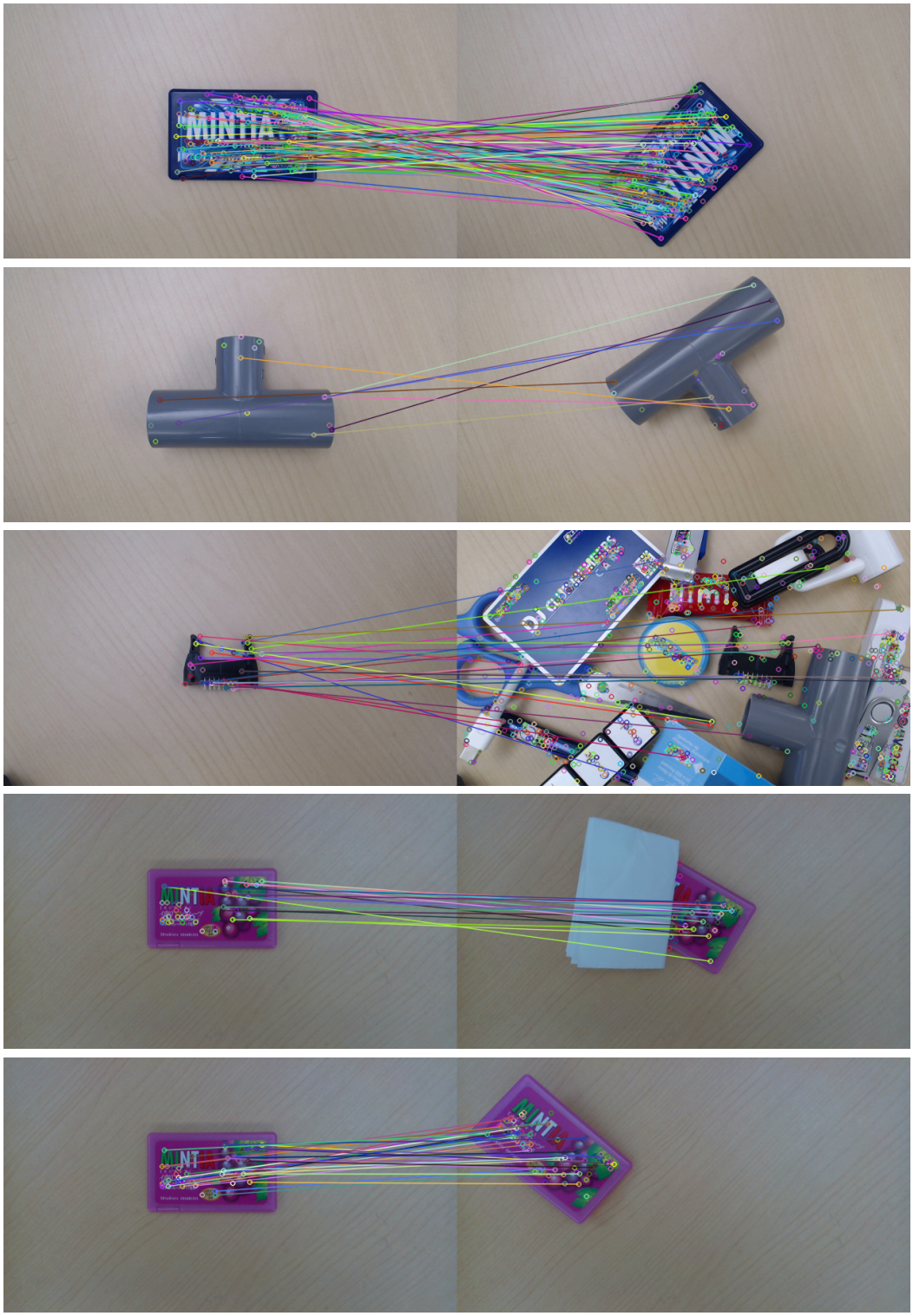


Figure 2.2 Detection results of local descriptor based method (SIFT [65] was used). 1st row: A textured package on a simple background. 2nd row: A texture-less pipe on a simple background. 3rd row: A texture-less connector on a cluttered background. 4th row: An occluded package. 5th row: A textured package with perspective distortion. The left images are models and the right images are the detection results where the circles represent the detected keypoints and matched keypoints are connected by straight lines.

devices based on FAST corner detectors and SIFT/Ferns descriptors. Hinterstoisser et al. [82] have proposed the keypoint detection based on the pairs of Harris corner points and 3D pose estimation based on the geometric and photometric consistencies of the point pairs. It has been shown that these approaches worked well only on rich textured objects with high resolutions.

Similar to 2D object detection and pose estimation (Subsection 2.2.1), the edge and line features are required for handling texture-less objects. Various approaches have been proposed based on the correspondence of line features [72], Chamfer matching of the edge template [83], and tracing paths of edgelet constellations [84]. However, edges and lines are too simple to discriminate those of objects from those of background and it have been shown that these approaches were effective for texture-less objects with high resolutions at simple backgrounds.

Using depth information is of course helpful for more robust and accurate 3D object detection and pose estimation like in global descriptor based approaches. Various 3D local descriptors have been proposed from 1990s and they were well surveyed and experimentally evaluated in [85, 86, 87]. Spin image [88] which describes the spatial distributions of neighboring points is the first successful 3D local descriptor. It has been often cited and applied to many applications. Afterwards FPFH [89] and SHOT [90] which describes the geometric attributes of the surfaces around keypoints using histograms have been introduced and they showed superior performance.

Many pose hypothesis are generated by matching of local descriptors and verification methods of these pose hypothesis have been proposed. Zach et al. [91] have introduced early rejection of outliers using dynamic programming. Aldoma et al. [92] have proposed the framework for verification of pose hypotheses based both on global and local descriptors. Buch et al. [93] have presented the voting in a 1-DoF rotational subgroup and achieved efficient and robust pose estimation.

The point pair feature (PPF) [94] is a most successful and well known 3D local descriptor ever and it has often been extended and modified since the original algorithm was published in 2010. For example, using visibility context for robustness against background clutters [95], selecting points and making the pair of boundary to boundary (B2B), surface to boundary (S2B), line to line (L2L) [96], adding a

color component (CPPF) [97], using the points on geometric edges [98], calculating PPF on segmented point clouds [99] and modified point sampling and voting [100] have been proposed. The performance evaluation of PPF and its variants has been detailed in [101]. In PPF, 3D pose and positions are voted from dense point pair features. Therefore, the features of PPF is more similar to the global descriptor based approaches than the local descriptor based approaches, which is described in Subsection 2.2.1.

2.3 Learning Based Approach

Machine learning algorithms such as SVM and boosting have widely been applied to generic object detection and pose estimation in order to handle intra-class variability. On the other hand, they have rarely used for specific object detection and pose estimation because there was only one positive example for the learning. Though Malisiewicz et al. [102] have proposed exemplar-SVMs where a single positive instance and millions of negatives were trained, they applied the ensemble of them to general object detection. This trend has changed since the breakthrough of CNN on the task of general object recognition on ImageNet [103]. Then the deep learning algorithms has begun to be applied to feature/descriptor design (Subsection 2.3.1) and pose classification (Subsection 2.3.2) for specific object detection and pose estimation. The learning of general purpose descriptors and object pose manifold requires only positive samples and the classifier learning for detection requires both positive and negative samples.

2.3.1 Learning of Feature and Descriptor

The deep learning methods have been applied to local descriptor matching such as comparison of image patches [104], designing discriminative image patches [105], and end-to-end learning of whole pipeline [106]. However, Balntas et al. [70] have shown that the tuned hand-crafted descriptors achieved almost the same performance as deep learning based descriptors on their new large dataset. They have also been utilized to extract discriminative and compact 3D local descriptors [107,

108]. The end-to-end framework for joint learning of keypoint detector and descriptor [109], and learned descriptors based on point pair feature [110, 111] have also been proposed. Though these local descriptors were learned based on large dataset, it was unclear whether they work on totally different scenes.

The learning methods for object specific feature extraction have been presented. For example, the learned voting weights for point pair feature [112], the discriminative features for 3D object pose and class learned by CNN [113], and the learning of 3D object pose differences by CNN [114] were proposed. The manifold learning of 3D object pose by dimensionality reduction has also been proposed, for example, learning of RGB-D patches using convolutional auto encoder [115] and learning of appearance of whole object in RGB image using a self-supervised augmented auto encoder [116]. Pavlakos et al. [117] have proposed the object specific keypoint detector learned by CNN and it was used in their pipeline for 6-DoF pose estimation.

2.3.2 Learning of Object Class and Pose

The learning based pose classification has been often used for 3D object detection and pose estimation because 3D pose space was too large for exhaustive search. Rodorigues et al. [118] have proposed 6-DoF pose voting based on random ferns from RGB patches generated by their multi light imaging system. Tejani et al. [119] have proposed 6-DoF pose estimation from RGB-D patches using Hough forest and inference of latent class distribution. To make pose estimation robust against background clutters, some researches use background class for their training. Rios-Cabrera et al. [120] used the coefficients learned by linear SVM for feature selection and matching weights. Brachmann et al. [121] have learned 126 classes (discretized 125 pose plus background) by random forest and estimated 6-DoF pose based on coordinate regression. Later, this method has been improved by exploiting label uncertainty [122]. Kehl et al. [123] have combined SSD-like CNN architecture for general object detection with 6-DoF pose estimation. They estimated simultaneously 2D position of object bounding box, object class, viewpoint and 2D rotation angle using multi-class detector. Xiang et al. [124] have proposed the quaternion regression based on two kinds of loss functions for handling symmetric objects.

In recent years, it has become popular that 6-DoF object pose was recovered from the detection results of the projected 2D corner points of 3D bounding boxes. This method estimates 6-DoF pose based on 2D-3D correspondences only from a RGB image and the pose is refined using depth information if it is available. Crivellaro et al. [125] have trained CNN to detect parts of objects based on the projected 3D control points. Rad et al. [126] have estimated the projected corner points of bounding boxes using the appearance of a whole object. Tekin et al. [127] have proposed similar but faster method based on YOLO-like CNN architecture.

The learning based approach has also been employed for 6-DoF pose refinement and hypothesis verification. Krull et al. [128] have proposed the pose hypothesis verification by analysis by synthesis and learned CNN for the comparison between rendered and observed images. Michel et al. [129] have presented the globally optimum hypothesis generation based on fully-connected conditional random field. Regarding 6-DoF pose refinement, the gradient based optimization using reinforcement learning [130] and the prediction of relative pose transformation from the difference between the rendered image and the observed image [131] have been proposed.

2.4 Suitability for Real Applications

In previous sections, three kinds of approaches for 2D/3D specific object detection and pose estimation are surveyed. The pros and cons of these approaches are summarized in Table 2.1. The global descriptor based approach can handle any kinds of objects and robust against background clutters as shown in Figure 2.1. However, the feature points are extracted from whole object and their correspondence should be found. Then the global method is fragile to occlusions and transformations (see 4th and 5th row in Figure 2.1). Additionally, the processing time for exhaustive search of model templates increases linearly with the number of kinds of objects and image resolutions.

Contrastingly, the local descriptor based approach is robust against occlusions and transformations because the matching of each descriptor is done independently and the descriptor represents a part of object. Moreover, the local descriptors are

Table 2.1 The pros and cons of global descriptor based, local descriptor based and learning based approaches for specific object detection and pose estimation.

	global	local	learning
texture-less and simple-shaped	✓		✓
background clutter	✓		✓
small size	✓		✓
occlusion		✓	✓
deformation		✓	✓
scalability		✓	✓
immediate training	✓	✓	

extracted only on the detected keypoints and the processing time does not depend on the number of kinds of objects and image resolutions. However, the local part of object should be discriminative enough for finding correspondences correctly. This requires that the object is texture-rich/complex-shaped and rather large in an image. For those reasons, the local descriptor based approach cannot handle texture-less/simple-shaped/small objects and is fragile to background clutters (see 2nd and 3rd rows in Figure 2.2).

The learning based approach is superior to the former two approaches in performance because the features are designed and selected based on training samples so that they are discriminative and robust against background clutters, occlusions and transformations. It can handle texture-less/simple-shaped/small objects and is scalable to increasing number of kinds of objects and image resolutions. However, it requires large training dataset of positive and negative samples. It takes too much time and cost for users to prepare and annotate training samples for each object and scene.

The target of this thesis is a research on the practical algorithms which can be applied to real factory scenes. In these applications, many of the target objects are rigid and occluded objects are not inspected nor grasped. For those reasons, it is not so important to handle occlusions and object deformations. However, the immediate on-site training of target objects is crucially important because it is almost impossible for customers to collect large dataset for each application. To summarize, we have no choice but to employ the global descriptor (template matching) based approach

which includes the methods based on voting from dense features.

Chapter 3

2D Detection and Pose Estimation of Texture-less Objects

In this chapter, we introduce a fast and robust algorithm for estimation of 2D object position and pose. As described in Section 2.4, we employ the template matching based approach in order to handle wider range of objects including texture-less objects such as mechanical/electronic parts. Our proposed image feature COF (Cumulative Orientation Feature) relaxes the matching condition only for the appearance changes by the changes in 2D object pose without losing the robustness against background clutters. Due to this, our model template with COF can handle wider range of 2D object pose and our detection speed becomes faster using less number of templates.

The remaining contents of this chapter are organized as follows: Section 3.1 presents the existing work regarding COF. After explaining COF and whole pipeline of our detection algorithm in Section 3.2, Section 3.3 shows the experimental results including the comparison with the state-of-the-art methods and our failure cases. Section 3.4 concludes this chapter.

3.1 Related Work

As described in Subsection 2.1.1, it has been shown that the gradient direction vectors [24] and the quantized gradient orientations [25] were robust against cluttered backgrounds and illumination changes. However, it was pointed out that the similarity scores based on these features rapidly declined even if only slight changes

in object pose occurred. To overcome this problem, dominant orientations within a grid of pixels (DOT) [26] and spread orientation which allowed some shifting in matching [48] were proposed. DOT and spreading orientation are robust to the pose changes and slight deformations of target objects. However, they relax matching conditions both in foregrounds and backgrounds, and this possibly degrade the robustness against cluttered backgrounds.

Berg et al. [132] have proposed Geometric Blur which blurred both model and input image using position-dependent blur kernels and they showed that it made template matching more robust to affine distortions compared to normal blurring based on isotropic kernels like a Gaussian filter. The spreading orientation isotropically copy the orientation feature to neighboring pixels and is considered as a Gaussian filter for quantized image features. We propose the position-dependent spreading algorithm for quantized orientation features, which resembles Geometric Blur for continuous values.

3.2 Proposed Method

This section describes our proposed 2D object detection and pose estimation pipeline using COF. After COF is introduced in Subsection 3.2.1, the coarse-to-fine search algorithm using COF templates is explained in Subsection 3.2.2. Then the refinement of 2D object pose is described in Subsection 3.2.3.

3.2.1 COF: Cumulative Orientation Feature

There are few features on the surfaces of texture-less objects and the features representing their shapes such as object contours are important for detection and pose estimation. COF is designed based on histograms of gradient orientations in order to make it more relevant to object shapes as well as more robust to transformations of objects. HOG [133] is also based on the histograms of gradient orientations and has been successfully applied to many tasks such as object class recognition and pedestrian detection. The main difference between HOG and COF is that COF uses the histogram at each pixel while HOG uses the histogram at local regions called cells. This enables COF to determine the positions and poses of objects more precisely.

(a) (b) (c)

Figure 3.1 (a) The model image of the connector. (b) Colored gradient directions of the model image. (c) Quantization of gradient directions disregarding their polarities.

Moreover, COF is represented as a binary string which describes only the dominant orientations of the histograms [26] and this leads to fast matching of the features based on logical operations.

The way how to extract COF is explained using a connector shown in Figure 3.1 (a) which is a typical texture-less object. Firstly many training images are synthesized based on randomized 2D pose parameters (X/Y translations, in-plane rotations and object scales). The range of randomized parameters should be limited so as to a single template can handle the appearance changes caused by the randomized parameters. In our research, the range of randomization were experimentally determined and those were ± 1 pixel in X/Y translations, ± 15 degrees of in-plane rotations and ± 5 % in object scales. Total of N images are generated.

Secondly image gradients of all the generated images are computed using Sobel operators (the maximum gradients among RGB channels are used). To reduce false edges due to image noise and illumination changes, only the gradient directions whose magnitudes are larger than a certain threshold are used for feature extraction. The colored gradient directions of the model image is shown in Figure 3.1(b).

To extract features which tolerate only the appearance changes by different object pose, the orientation histograms are built at each pixel using all the generated images of gradient vectors. The gradient vector is quantized into eight orientations (Figure 3.1(c)) and voted to the histogram (add 1) at each pixel. The added values to the neighboring bins are interpolated based on the ratios of the differences between the gradient directions and the centers of the bins. Only the orientations with

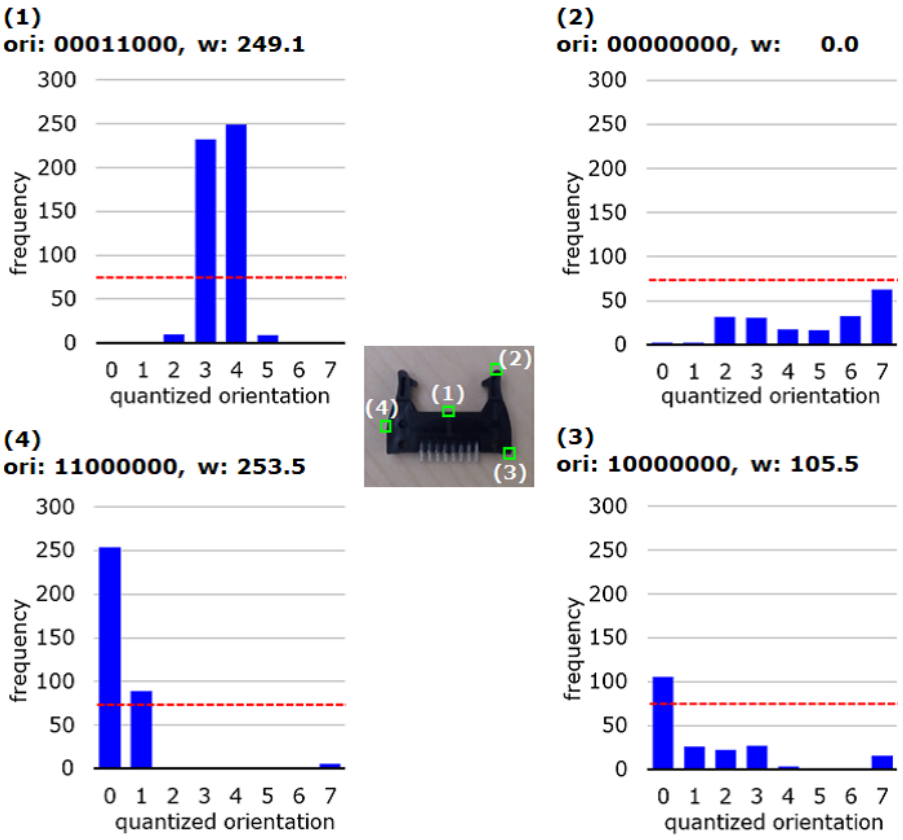


Figure 3.2 Orientation histograms, cumulative orientation feature (ori) and their weights (w) those were extracted at four exemplar pixels of the model image. The dotted red lines of histograms showed thresholds for feature extraction.

the higher magnitude than the threshold are voted and the maximum frequency is equal to N. Lastly the dominant orientations at each pixel whose frequencies are larger than a threshold (Th) are extracted and they are represented by 8-bit binary strings. This is the cumulative orientation feature (COF) and COF represents the gradient orientations which are probably observed when object pose is changed in a compact form. The maximum frequencies of the histograms are also extracted and used as the weighting factor for matching of features because the orientation features with high frequencies are stable against the appearance changes by slightly different object pose.

The orientation histograms, cumulative orientation features (ori) and their weights on exemplar four pixels are shown in Figure 3.2. In our study, the number of generated images (N) was 500 and the frequency threshold value (Th) was 75. The votes were concentrated on a few orientations at the pixels along lines or arcs such as pixel

Figure 3.3 The weighting factor of COF at each pixel. The brighter pixel values represented the larger weighting factors.

(1) and (4). At these pixels the important features with large weights were extracted. On the contrary, the votes were scattered among many orientations at the pixels on corners such as pixel (2) and (3). At these pixels the features with small or zero weights were extracted. These tendencies are also observed on the image (Figure 3.3) where the brighter pixel values represents the larger weighting factors.

3.2.2 2D Object Detection and Pose Estimation

A model template which consists of COF can handle appearance changes caused by different object pose generated in training stage (± 15 degrees of in-plane rotation and ± 5 % of object scale in our study). To cover a wider range of object pose parameters, additional templates are made using rotated and resized model images. We prepared model image in every 20 degrees of in-plane rotation and in every 7.5 % of object scale in order for whole 2D object space to be redundantly covered.

To make our detection and pose estimation faster, coarse-to-fine search [17] was utilized. The image pyramid of the model image is made and COF is calculated on the image of every in-plane rotation angle and object scale at every level of the pyramid. The ranges for randomization of the pose parameters are large at the higher levels of the image pyramid because the appearance changes by different rotation angles and object scales look smaller comparing with those at lower levels. The steps for rotation angles and scales are doubled when the resolution of image is halved (one step up to the higher level) and the number of templates is reduced by one fourth. This leads to the efficient search of 2D position and pose at the same time.

The j-th model template at level i is represented as

$$T_{ij} : \{x_k, y_k, ori_k, w_k | k = 1, ..., n\} . \tag{3.1}$$

Each template has n COF features whose weighting factors are larger than a threshold and each feature has position at model image (x_k, y_k), quantized orientation ori_k, and weighting factor w_k.

In testing, the quantized orientation features are firstly extracted on each level of the input image pyramid. Secondly, the whole area of the top level of the image pyramid is scanned exhaustively using the templates of the top level of the template pyramid. The possible candidates for the correct object pose whose similarity scores between the input features and the model templates are larger than a search threshold are further scanned using the templates of the lower levels of the pyramid. Lastly, the 2D position and pose (in-plane rotation and scale) are determined by non-maximum suppression algorithm at the bottom level. To avoid discarding promising candidates at the upper levels, the search threshold at the upper levels is decreased to 80% of the threshold at the bottom level. The similarity score is calculated as

$$score(x, y) = \frac{\sum_{k=1}^{n} \delta_k(ori^I_{(x+x_k, y+y_k)} \in ori^T_k)}{\sum_{k=1}^{n} w_k}. \tag{3.2}$$

ori^I is a quantized orientation feature of an input image and ori^T is COF of a model template. δ_k is a function which determines whether these orientations are same and this is quickly calculated by bitwise operation.

$$\delta_k(ori^I \in ori^T) = \begin{cases} w_k & \text{if } ori^I \wedge ori^T > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

w_k is a weighting factor of COF and $\wedge$ represents bitwise AND. This bitwise operation is further boosted by using SIMD instructions. We utilized Intel SSE2 intrinsics which is capable of 128-bit registers for matching of 16 features at one time.

3.2.3 Pose Refinement

The obtained 2D position and pose parameters can be further refined by registration of 2D edge point pairs between the model and the input image. The edge points and normal vectors of model images are extracted by Canny edge detector [22] in training phase. After the detection based on the hierarchical template matching in testing phase, the model edge points are projected onto an input image using the detection results as initial pose parameters. The paired edge points on the input image (x', y') are searched along the normal vectors (n_x, n_y) of the model points (x, y) and they are found as local maxima of image gradients along the search lines. After N_e edge point pairs are collected, the sum of inner products between the normal vectors and the estimated tangent vectors of the model edge points is minimized.

$$p = \text{argmin} \sum_{i=1}^{N_e} n_x(M(x') - x) + n_y(M(y') - y) \tag{3.4}$$

$M()$ represents the 2D transformation based on four 2D pose parameters (X/Y translation, in-plane rotation and scale). This minimization problem is solved in a closed form and the paired edge points on an input image are re-searched based on the updated pose parameters p. This procedure is iterated until the update is less than a certain threshold.

3.3 Experimental Evaluation

Five experiments were executed to evaluate our proposed algorithm. The influence of the parameters of COF on its robustness against background clutters and the appearance changes by object transformations was evaluated in Experiment 1 (Subsection 3.3.1). In Experiment 2, the robustness of COF was compared with that of existing orientation features (Subsection 3.3.2). The accuracy and speed of the detection algorithm based on COF was evaluated and compared with existing algorithms using D-Textureless dataset in Experiment 3 (Subsection 3.3.3) and CMU_KO8 dataset in Experiment 4 (Subsection 3.3.4). In Experiment 5 (Subsection 3.3.5), the detection errors were investigated in the repeatability, linearity and rotation tests. Finally,

Test image 1 Test image 2

Figure 3.4 Test image 1 and Test image 2 used in Experiment 1 and Experiment 2. The only difference between these two images are in-plane rotation angle of the connector (approximately by 10 degrees).

the failure cases of our algorithm mainly in Experiment 3 and 4 are introduced and discussed in Subsection 3.3.6.

3.3.1 Experiment 1: Parameters for COF

In Experiment 1, we tested the influence of two parameters for extraction of COF on the discriminative power of the feature. One of the parameters was the number of training images (N) and another was threshold for histogram frequencies (Th). Test image 1 and Test image 2 in Figure 3.4 were prepared and scanned using the model image (Figure 3.1(a)) when N and Th were varied. In Test image 1, the connector was placed in a cluttered background while the in-plane rotation angle and size of the connector were almost the same as those of the model image. In Test image 2, the scene was almost the same as that of Test image 1 except for the in-plane rotation angles of the connector. We evaluated the difference between a maximum score in the foreground (FG) and in the background (BG) of the test images. The larger this difference is, the feature is more discriminative and this leads to higher accuracy in object detection and pose estimation.

The score differences when N was changed from 1 to 1000 is shown in Figure 3.5(a). Th was fixed as 15% of N. Both on Test image 1 and Test image 2, the score difference changed significantly under $N = 200$ and then converged to almost the same value around $N = 500$. This is because larger number of training images with various 2D pose parameters contribute to the invariant score when the appearance of target object is changes by object transformations.

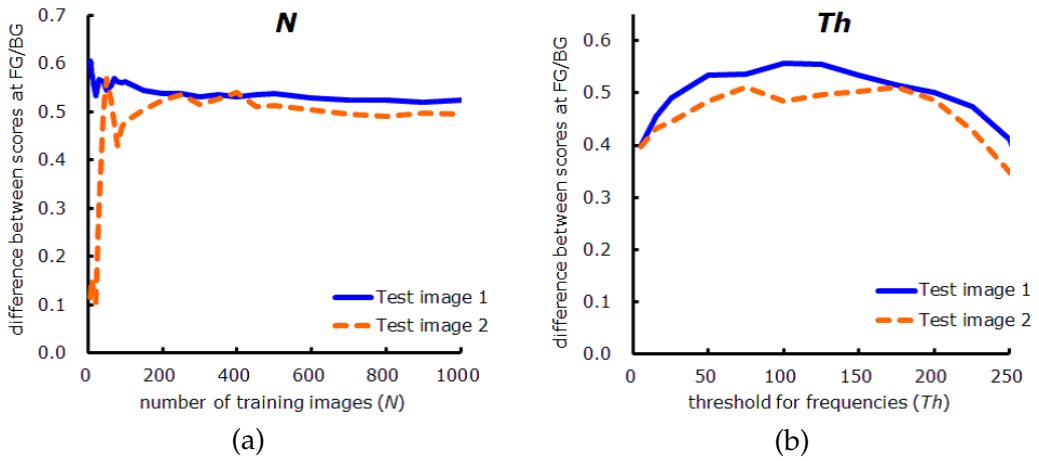


Figure 3.5 Differences between maximum scores at foreground (FG) and background (BG) of Test image 1 and Test image 2 when (a) the number of synthesized images and (b) the threshold for histogram frequencies were changed.

The score difference of Test image 2 changed more drastically than that on Test image 1 because normal model image without any transformations was always used as first synthesized image. Therefore the score difference of Test image 1 was high and that of Test image 2 was low when N is small. The larger N was, the more stable the score differences of both test images were. The score differences did not changed any more when N was larger than 500 and we used $N = 500$ in our research.

The score differences when Th was changed from 0 to 250 is shown in Figure 3.5(b). N was fixed as 500. The differences of Test image 1 and Test image 2 increased as Th grew from $Th = 0$. They reached to their maximum when Th was from 75 to 150 and then started decreasing. More gradient pixels and more quantized orientations are included into COF when Th is lower. These loose COF are matched with any orientation features even on background. On the contrary, less pixels and less quantized orientations are included into COF when Th is higher. These strict COF cannot be matched with the orientation features of target object even with small 2D transformation of object pose. To summarize, there is optimum Th which balance robustness against background clutters and tolerance to appearance changes of target objects. $Th = 75$ was used in our research.

3.3.2 Experiment 2: Evaluation of Orientation Features

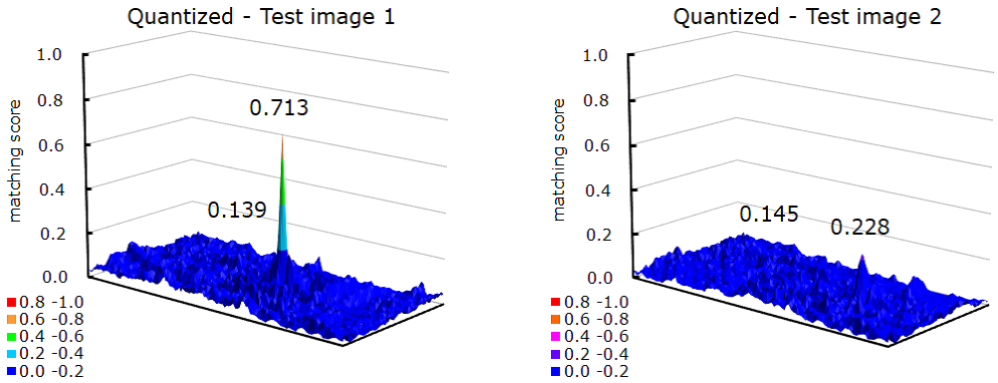
In Experiment 2, COF with and without weighting factors were compared with existing quantized orientation [25], normalized gradient vector [24], and spreading orientation [48]. The model image shown in Figure 3.1(a) and two test images shown in Figure 3.4 were used for the comparison. The 2D histograms of similarity scores calculated based on four orientation features on Test image 1 (left column) and Test image 2 (right column) are shown in Figure 3.6. The difference between maximum scores at foreground and at background on both test images are also shown in Table 3.1.

Quantized orientation

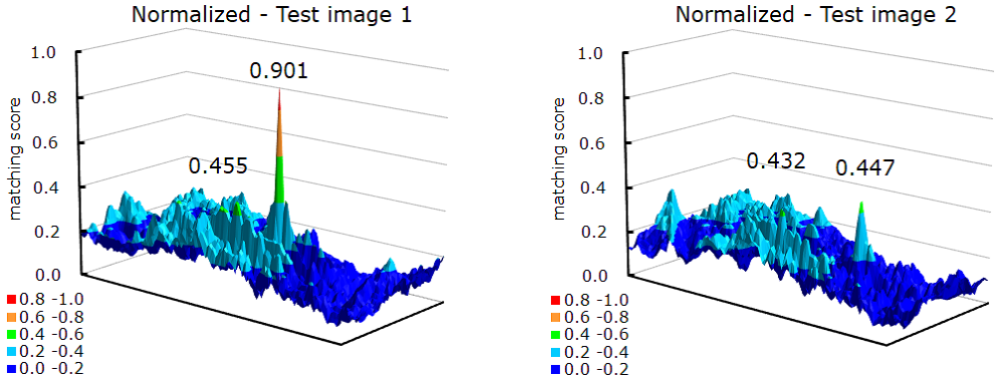
Quantized orientation is calculated by quantizing the image gradient direction (Figure 3.1(b)) into eight orientations (Figure 3.1(c)). The similarity score is calculated by dividing the number of pixels which have same orientation between a model and a test image by the number of all pixels of the model image. The similarity scores at all pixels of two test images were shown in Figure 3.6(a). Though the maximum scores at background were low on both test images, the maximum scores at foreground were substantially decreased on Test image 2. Then the maximum score difference (FG - BG) on Test image 2 was lower than that of COF (Table 3.1). These results shows that quantized orientation is robust to background clutters but fragile to appearance changes of target objects.

Normalized gradient vector

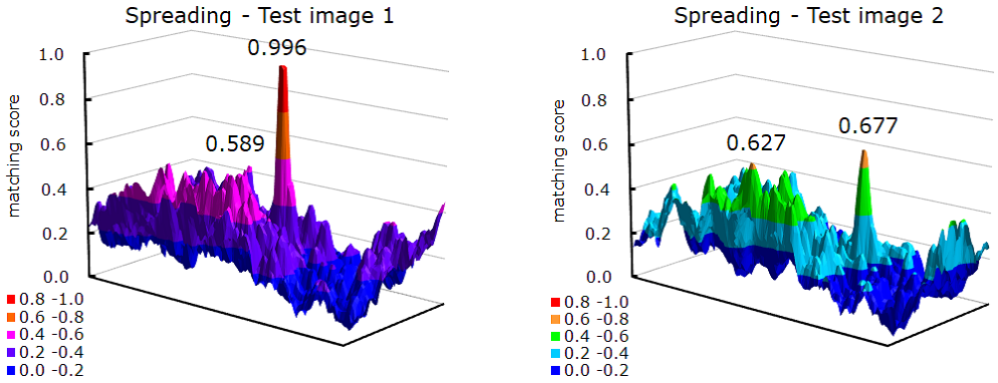
Steger et al. [24] have proposed normalized gradient vector which is a unit vector of an image gradient and have shown that the sum of inner products of normalized gradient vectors was occlusion, clutter and illumination invariant. This was demonstrated on our experimental result on Test image 1 (Figure 3.6(b)). However, the maximum score at foreground was substantially decreased on Test image 2 and this indicates that this feature does not handle appearance changes of target objects. This was also supported by the significant decrease of score difference between FG and BG on Test image 2 shown in Table 3.1.


Quantized - Test image 1
matching score
1.0
0.8
0.6
0.4
0.2
0.0
0.713
0.139
0.8 -1.0
0.6 -0.8
0.4 -0.6
0.2 -0.4
0.0 -0.2
Quantized - Test image 2
matching score
0.145
0.228

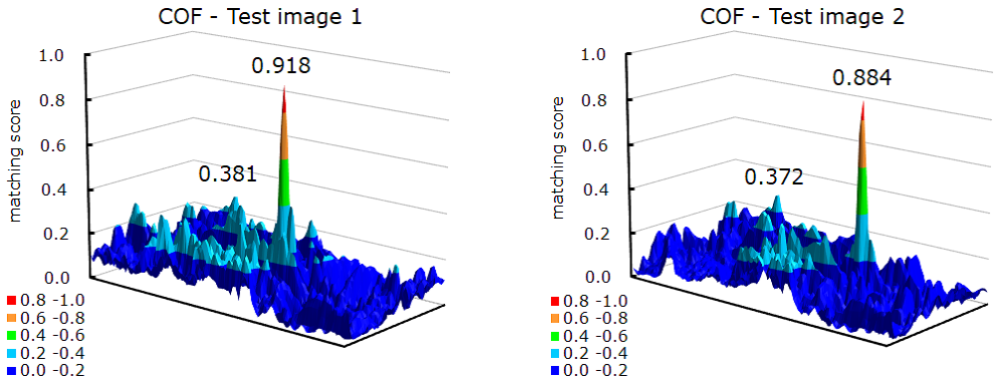
(a)


Normalized - Test image 1
matching score
0.901
0.455
0.8 -1.0
0.6 -0.8
0.4 -0.6
0.2 -0.4
0.0 -0.2
Normalized - Test image 2
matching score
0.432
0.447

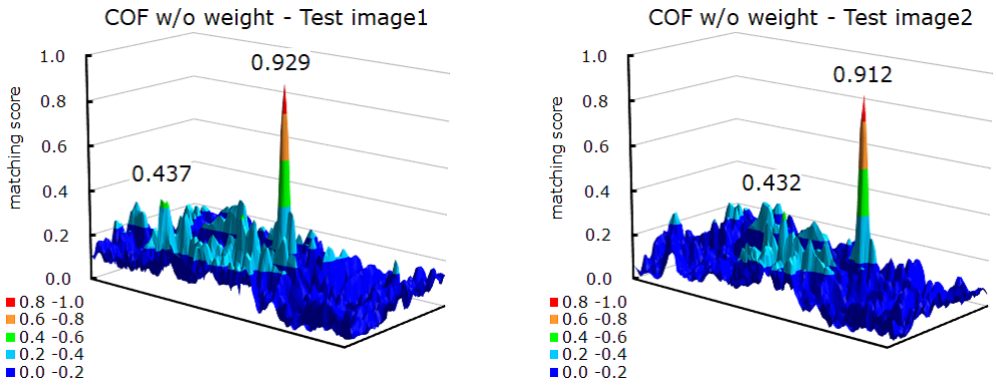
(b)


Spreading - Test image 1
matching score
0.996
0.589
0.8 -1.0
0.6 -0.8
0.4 -0.6
0.2 -0.4
0.0 -0.2
Spreading - Test image 2
matching score
0.627
0.677

(c)


COF - Test image 1
matching score
0.918
0.381
0.8 -1.0
0.6 -0.8
0.4 -0.6
0.2 -0.4
0.0 -0.2
COF - Test image 2
matching score
0.884
0.372

(d)



(e)

Figure 3.6 2D histograms of similarity scores on Test image 1 (left column) and Test image 2 (right column) based on (a) quantized orientation, (b) normalized gradient vector, (c) spreading orientation, (d) COF and (e) COF without weighting factors. The maximum scores at foreground and background are presented in each figure.

Spreading orientation

Hinterstoisser et al. [48] proposed spreading orientation to relax the matching condition for quantized orientations by copying them to neighboring pixels of a test image. As they described in the paper, the quantized orientations on the test images were spread in the range of ± 4 pixels and combined them by bitwise ADD. The similarity scores between the spread orientation of the test images and the quantized orientation of the model image were calculated and shown in Figure 3.6(c). The maximum scores at background are higher than those of quantized orientation on both test images and the score difference on Test image 1 is the lowest in Table 3.1. These results indicate that spreading orientation is fragile to background clutters. Though the maximum scores at foreground were high on both test images and the tolerance to appearance changes of target objects was improved, the score difference was the lowest also on Test image 2 (Table 3.1). This is because spreading operation relaxes the matching conditions both at foreground and background.

COF

COF (Subsection 3.2.1) was extracted on the model image and the model template is scanned at the quantized orientation features on test images. The similarity scores at foreground and background on Test image 1 were increased (Figure 3.6(d)) from

Table 3.1 Differences between maximum scores at foreground and background ($FG - BG$) on Test image 1 and Test image 2.

	Test image 1	Test image 2
Quantized	**0.574**	0.083
Normalized	0.446	0.015
Spreading	0.407	0.050
COF	0.537	**0.512**
COF w/o weight	0.492	0.480

those of quantized orientation and the score difference was a little decreased from quantized orientation (Table 3.1). These results demonstrated that COF was robust against background clutters. Additionally, the scores at foreground and background, and their difference on Test image 2 were not so changed from those on Test image 1. This shows that COF can improve the tolerance to the appearance changes of target objects without degrading the robustness against background clutters.

COF without weighting factors

To test the effect of weighting factors on COF, the similarity scores were calculated using equal weights for all features and shown in Figure 3.6(e). The scores at background on both test images were increased and the differences between FG and BG scores on both test images were decreased compared with those of COF with weights. The COF with low weights tend to have many orientation as shown in Figure 3.2 and these features are inclined to be matched with any orientations both at foreground and background. Using equal weights enlarges the influence of such features and this degrades the robustness against background clutters. Therefore, weighting factors of COF are important for its discriminative power.

3.3.3 Experiment 3: Evaluation on D-Textureless Dataset

Experimental settings

In Experiment 3, our proposed detection algorithm based on COF was evaluated on a publicly open D-Textureless dataset [74]. The dataset consists of nine model images and 54 test images. The target objects are texture-less such as a nipper and

Figure 3.7 Nine model images of D-Textureless dataset.

Table 3.2 The parameters used in Experiment 3. The number of generated images (N) and threshold of orientation histograms (Th) for COF extraction. The intervals, ranges and numbers of templates for rotations and scales in template generation.

N	Th	rotation (deg)	scale (%)
500	75	20° in $\pm 180^\circ$ (18)	7.5% in $\pm 30\%$ (9)

a spanner shown in Figure 3.7. The resolution of the images is 640×480 pix and the test images include the changes of the objects' in-plane rotation and scale under background clutters and partial occlusions. Some of test images are shown in Figure 3.8. The bounding boxes and model edge points are drawn based on the detected results by our algorithm.

To compare our proposed method with existing research, Steger's algorithm based on inner products of gradient direction vectors [24], LINE-2D based on spreading orientation [48] and BOLD based on local line segments [74] were also evaluated on the dataset. We utilized the implementation of Steger's algorithm "shape based matching" in "HALCON 11.0" (MvTEC, Germany) which is commercial software library for machine vision. We also utilized LINE-2D implemented in the open source library "OpenCV 2.4.11" and the binary software of BOLD which was provided by the authors. When the model images were trained in HALCON and COF, the images were rotated in ± 180 degrees and resized in $\pm 30\%$. The model data prepared in the dataset was used for LINE-2D. The parameters for our algorithm are summarized in Table 3.2. All the programs were run on a same PC (Core i7 3770 3.4GHz and 8GB

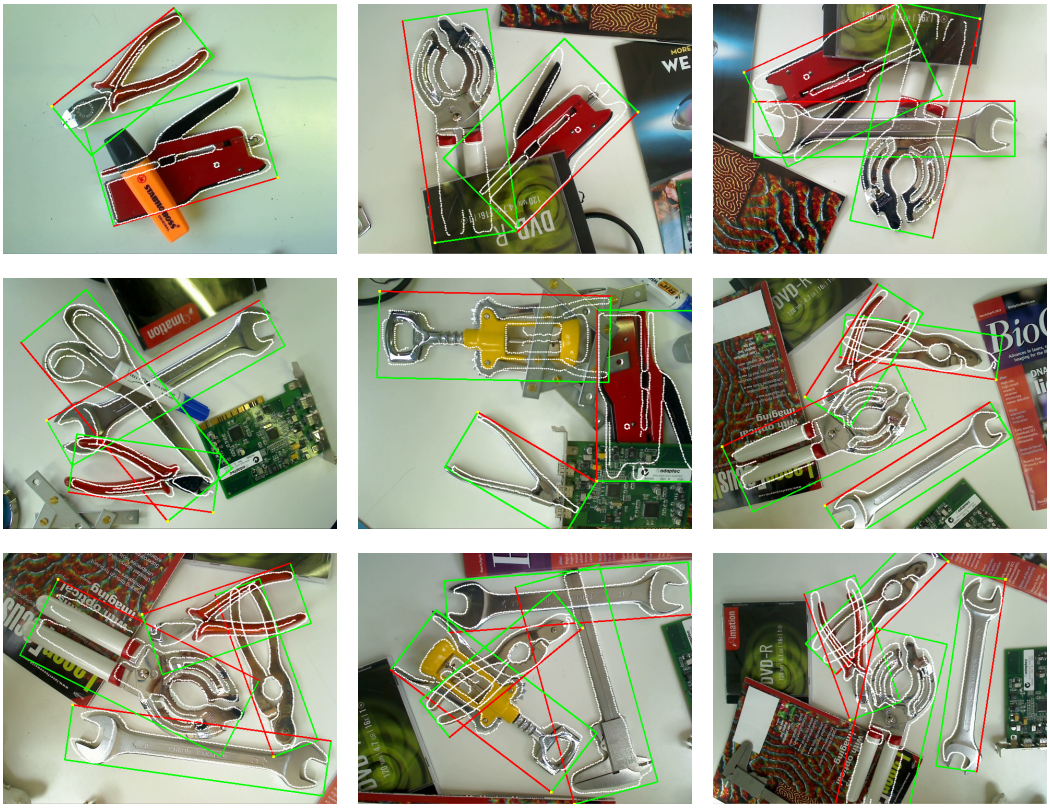


Figure 3.8 Example images of D-Textureless dataset used in Experiment 3. The edges of the objects extracted from the model images (white lines) and the bounding boxes (red and green lines) are drawn based on the detection results by our proposed method.

RAM) using a single CPU core. When the bounding box of detection results (BB_{dt}) overlap sufficiently that of ground truth (BB_{gt}), the result was counted as correct [3]. The threshold for overlap was 0.7 in our research.

$$\frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.7 \tag{3.5}$$

Detection accuracy

The graph in Figure 3.9 shows the relationships of four detection methods between correct detection rate (DR) and false positives per image (FPPI). These data were plotted when the search thresholds were changed. Better detection results were plotted on left-upper area. Correct detection rates and processing times when FPPI = 1.0 were shown in Table 3.3. From these results, our proposed method showed higher detection accuracy than existing template-based methods such as Steger's method

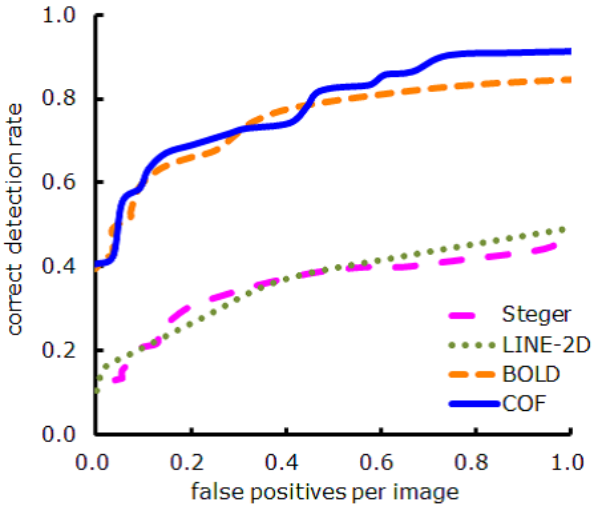


Figure 3.9 DR - FPPI curve on D-Textureless dataset in Experiment 3.

and LINE-2D. Furthermore, its detection accuracy was comparable to that of BOLD which was based on local descriptors.

The major difference among three template-based algorithms (Steger's, LINE-2D and COF) is the feature used. Our algorithm used COF, Steger's used normalized gradient vector, and LINE-2D used spreading orientation. Normalized gradient vector is robust to background clutters but not tolerate to the appearance changes of target objects as described in Experiment 2 (Subsection 3.3.2). On the contrary, spreading orientation is tolerate to the appearance changes of target objects but not robust to background clutters. COF can balance the robustness and the tolerance, and this is why our proposed method surpass these two template-based existing method in detection accuracy.

In general, the detection algorithms based on local descriptors such as BOLD are more robust to partial occlusions and transformation of target objects than template matching based algorithms. The tolerance to the appearance changes of target objects are improved in COF by extracting orientations from geometrically transformed images and the model templates are prepared so that the detection range is covered with overlapped ranges of each COF template. Regarding the robustness to occlusion, the local descriptors on edges of texture-less objects does not have enough discriminative power as the descriptors on textures of texture-rich objects. Therefore it is considered that local descriptor based algorithm for texture-less object detection does not have much advantage about robustness against occlusions over template

Table 3.3 Correct detection rate and processing time (ms) when FPPI = 1.0 on D-Textureless dataset.

	detection rate	processing time
Steger	0.466	396.4
LINE-2D	0.492	47.7
BOLD	0.846	177.9
COF	**0.916**	**31.8**

based algorithms. From these reasons, our proposed method showed almost the same detection accuracy as BOLD.

Processing time

The speed of our detection algorithm was faster than the existing algorithms in Table 3.3. Steger's algorithm uses the inner product of normalized gradient vectors as similariry score and this calculation is based on a floating point number. COF uses 8-bit binary number as their orientation feature and the similarity score is calculated quickly by SIMD instructions for logical operations. This is why COF exceeds Steger's algorithm in detection speed. Regarding the processing time of BOLD, it took about 106 ms in average for extracting local descriptors based on LSD [23]. COF is based on image gradients which require only Sobel filtering as preprocessing and this lead to the large difference between COF and BOLD in detection speed.

LINE-2D also uses 8-bit binary number for encoding quantized orientations and the similarity score based on logical operations. Furthermore, the arrangement of orientation features on a computer memory are re-aligned for sequential access and the similarity score is computed just by adding the pre-computed values by utilizing look-up tables. These boost the detection speed of LINE-2D. However, the model template of COF can handle wider range of rotation angles and object scales compared to spreading orientation because COF balances the robustness against background clutters and the tolerance to the appearance changes of target objects as shown in Experiment 2. This allows us to reduce the number of templates for detection. In Experiment 3, the number of templates of COF were 162 (18 rotation and 9 scales) and this was fewer 75 % than the number of templates of LINE-2D

Figure 3.10 Eight model images of CMU_KO8 dataset. Top row: Bakingpan and Colander. 2nd row: Cup and Pitcher. 3rd row: Saucepan and Scissors. Bottom row: Shaker and Thermos. The mask images for training are also shown.

(650 in average). To summarize, the fast detection speed of LINE-2D comes from the efficient computation of similarity scores and that of COF comes from the fewer number of templates used for matching.

3.3.4 Experiment 4: Evaluation on CMU_KO8 Dataset

Experimental setting

In Experiment 4, our proposed detection algorithm was evaluated on a publicly open CMU_KO8 dataset [134]. The target objects are eight texture-less objects which are used in daily life such as mugs and pans. The dataset consists of two scenes; single viewpoint and multiple viewpoints. The single-view dataset consists of one model image and 100 test images from a single viewpoint. The multi-view dataset consist of 25 model images and 100 test images from multiple viewpoints. The model images of single-view dataset is shown in Figure 3.10. The resolution of the images is 640×480 pix and the target objects are heavily occluded in the test images. Some of test

Figure 3.11 Examples of test images from CMU_KO8 dataset (single-view). In each panel, the left is an input image and the right is result image where the bounding box and matched model edge points detected by our algorithm are drawn.

Table 3.4 The parameters used in Experiment 4. The number of generated images (N) and threshold of orientation histograms (Th) for COF extraction. The ranges and numbers of templates for rotations and scales in template generation.

N	Th	rotation (deg)	scale (%)
500	75	$\pm 0^\circ$ (1)	$\pm 0\%$ (1)

images from single-view dataset and detection results are shown in Figure 3.11. The bounding boxes and model edge points are drawn based on detected results by our algorithm in the result images.

Our detection program ran on the same PC as in Experiment 3 (Core i7 3770 3.4GHz, single CPU core was used). The parameters for our algorithm are summarized in Table 3.4. The criteria for correct detection result was also the same (Equation 3.5) but the threshold was changed from 0.7 to 0.5 which was used in the paper which introduced CMU_KO8 dataset [134].

Existing algorithms

The authors [135] provided the evaluation results of existing methods on CMU_KO8 dataset and these methods were compared to our algorithm. The evaluated algorithms are listed below:

LINE-2D [48] is fast and robust template matching for detection of texture-less objects, which is based on the spreading orientation and the linearized memory.

Robust LINE-2D (rL2D) [134] is a modified version of original LINE-2D. rL2D allows only the matching between the model and the input images those have same quantized orientation. This makes template matching more robust against cluttered background than LINE-2D.

Robust LINE-2D with gPb (rL2D-gpb) [135] uses gPb edge detector based on texture and color segmentations [136] instead of Sobel filter used in original LINE-2D.

Oriented Chamfer Matching (OCM) [137] extended Chamfer matching [16] to penalize the dissimilarity of edge orientations. They added the penalty term for the dissimilarity of edge orientations to the distance to the nearest edge pixel.

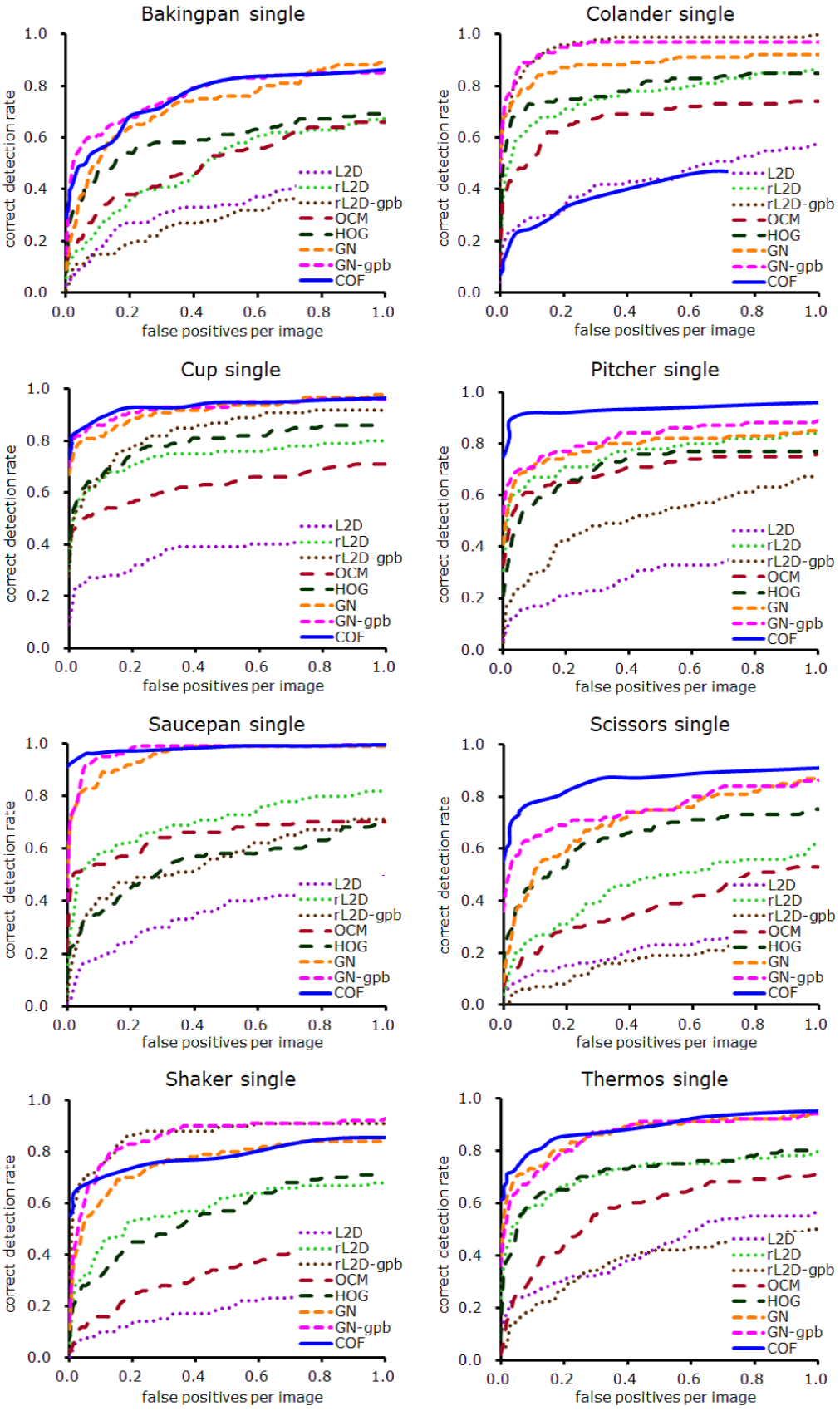


Figure 3.12 DR - FPPI curve on CMU_KO8 dataset (single-view).

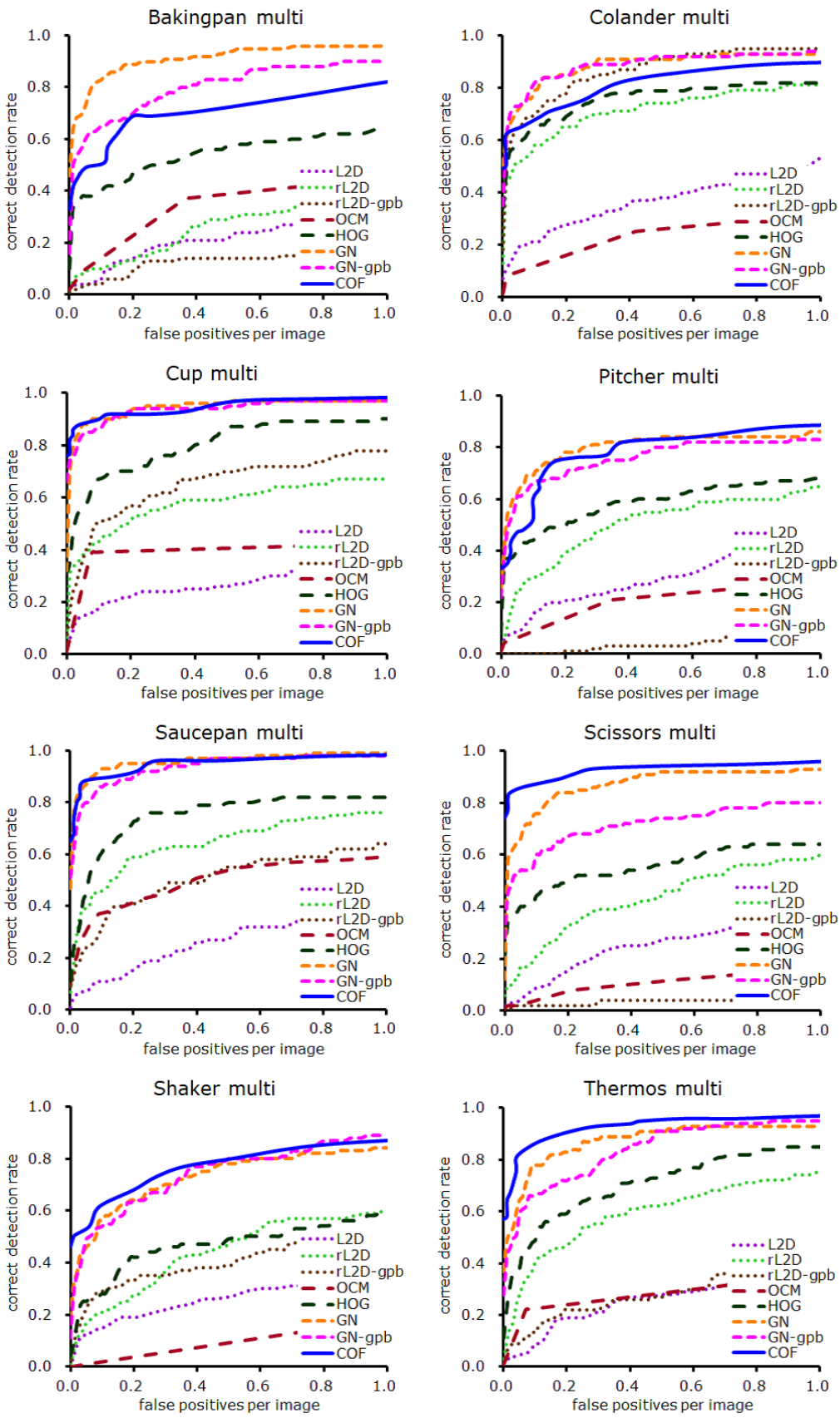


Figure 3.13 DR-FPPI curve on CMU_KO8 dataset (multi-view).

Histogram of Oriented Gradient (HOG) [133] represents an object as grids of gradient histograms. HOG of training image is learned using exemplar SVM [102] for each object. In testing, HOG of the input image is convolved with the learned template.

Gradient Network (GN) [135] describes the local connectivity based on gradient orientations, color and edge potentials. The shape similarity is evaluated through message passing algorithm.

Gradient Network with gPb (GN-gpb) [135] uses gPb edge detector for computing the edge potentials.

Detection accuracy

The graphs showing the relationships between correct detection rate (DR) and false positives per image (FPPI) for eight methods on single-view dataset were presented in Figure 3.12. The same graph on multi-view dataset were presented in Figure 3.13. The detection rates when FPPI = 1.0 on single-view dataset were shown in Table 3.5 and those on multi-view dataset were shown in Table 3.6. These results show that the detection accuracy of our algorithm based on COF is comparable to GN or GN-gpb and higher than other existing methods on both of single- and multi-view dataset. Only the detection rate on single-view colander is much lower than other existing method. This is because the dataset includes small viewpoint changes and the appearance of some test images are different from the model image. When the model images of multi-view dataset are used for testing of single-view colander, the detection rate increased to 0.836.

The existing algorithms other than GN uses the similarity scores based on the summation of a matching result at each pixel or grid. Our proposed COF is also based on per-pixel matching and the main difference from these existing methods is the robustness to background clutters and the tolerance to the appearance changes of target objects. As shown in Experiment 2, LINE-2D and its variants uses the spreading orientation and their robustness to background clutters is lower than COF. OCM and HOG are also computed based only on single model image per view point and their method for improving the tolerance (distance map of edges and spatial grids for gradient histogram) are isotropic. These improvements are not specialized for

Table 3.5 Correct detection rate when FPPI = 1.0 on CMU_KO8 dataset (single-view).

	L2D	rL2D	rL2D-gpb	OCM	HOG	GN	GN-gpb	**COF**
Bakingpan	0.460	0.679	0.410	0.660	0.690	**0.890**	0.859	0.863
Colander	0.579	0.870	**1.000**	0.740	0.850	0.920	0.970	0.491
Cup	0.450	0.800	0.931	0.710	0.860	**0.980**	0.960	0.967
Pitcher	0.449	0.840	0.670	0.760	0.770	0.850	0.888	**0.960**
Saucepan	0.495	0.820	0.710	0.700	0.694	0.990	**1.000**	0.995
Scissors	0.290	0.621	0.270	0.530	0.750	0.870	0.863	**0.907**
Shaker	0.292	0.680	0.910	0.492	0.720	0.840	**0.928**	0.850
Thermos	0.569	0.796	0.500	0.710	0.800	0.940	0.940	**0.951**
Mean	0.448	0.763	0.675	0.663	0.767	0.910	**0.926**	0.873

Table 3.6 Correct detection rate when FPPI = 1.0 on CMU_KO8 dataset (multi-view).

	L2D	rL2D	rL2D-gpb	OCM	HOG	GN	GN-gpb	**COF**
Bakingpan	0.324	0.411	0.190	0.450	0.650	**0.968**	0.900	0.822
Colander	0.530	0.810	**0.950**	0.314	0.820	0.930	0.940	0.894
Cup	0.340	0.671	0.780	0.424	0.900	0.970	0.970	**0.983**
Pitcher	0.430	0.650	0.110	0.283	0.680	0.860	0.830	**0.883**
Saucepan	0.410	0.760	0.640	0.592	0.820	**0.990**	0.980	0.982
Scissors	0.366	0.598	0.070	0.167	0.640	0.930	0.800	**0.959**
Shaker	0.338	0.610	0.500	0.184	0.590	0.840	**0.890**	0.869
Thermos	0.377	0.750	0.400	0.357	0.850	0.930	0.950	**0.971**
Mean	0.389	0.657	0.455	0.347	0.744	**0.927**	0.908	0.920

the shapes of target objects and lead to degraded robustness against background clutters.

Gradient networks (GN) calculates the probability how each pixel of an input image is similar to the model template using neighboring pixels and evaluates the contiguity of pixels which have high probabilities. This makes it robust to background clutters. Though COF calculates the similarity score just by adding the matching result at each pixel, each feature point has a weight representing the probability that the orientation is observed after 2D transformation of the target. This suggested that each feature point of COF represents the gradient orientations of its local area and this leads to similar effect of GN's similarity score which evaluates the contiguity of

Table 3.7 Processing time (milliseconds) when FPPI = 1.0 on CMU_KO8 dataset (single- and multi-view).

	Single view	Multi view
Bakingpan	3.5	15.5
Colander	2.6	22.3
Cup	2.3	22.7
Pitcher	3.6	26.4
Saucepan	3.3	19.1
Scissors	3.5	39.6
Shaker	4.1	37.6
Thermos	5.6	21.6
Mean	3.5	25.6

similar features. This is why the detection accuracy of COF is comparable to GN.

Processing time

The processing times when FPPI = 1.0 are presented in Table 3.7. It takes on average 3.5 milliseconds for single-view dataset and 25.6 milliseconds for multi-view dataset. These are shorter than the processing time on D-Textureless dataset in Experiment 3 because CMU_KO8 dataset does not include any rotations and changes in size of object and the numbers of model templates are less than those on D-Textureless dataset.

Though the authors of CMU_KO8 dataset did not provide details of processing time in their experimental evaluation, they described that GN took on average 1 second per image and this is much slower than our COF-based detection. The results in Experiment 3 show that COF is faster than LINE-2D. Regarding OCM and HOG, they include pre-processing steps such as distance transform and building normalized gradient histograms. Additionally, their similarity scores are based on floating point arithmetic which is slower than logical operation of binarized features in COF and LINE-2D. From these results, our object detection pipeline is the fastest among these existing methods.

3.3.5 Experiment 5: Detection Errors

The assembly applications in factory automation require precise alignment between parts and machines approximately by 100 µm or less. This is equal to sub-pixel/sub-degree alignment accuracy at an image coordinate system. We investigated the detection errors of our detection algorithm in three types of test: repeatability, linearity and rotation.

Experimental setting

We used five objects and one alignment mark for the error testing. These objects are captured using industrial USB camera (STC-MC33USB, resolution: 640×480, OMRON SENTECH CO., LTD.) with 16 mm lens. The example images are shown in Figure 3.14. For the repeatability test, 40 images were taken per object while the object is still in the same position. For the linearity test, 20 images were taken per object while the object on a mechanical stage was moved along X axis of the image approximately by 0.1 pixel. For rotation test, the image was rotated by 0.1 degree based on bilinear interpolation, which amounted to 3599 images per object.

Result

The detection errors in the repeatability, linearity and rotation tests are shown in Table 3.8. The error of repeatability is calculated as the standard deviation of the estimated positions along x axis. The errors of linearity and rotation are calculated as the root mean square errors of the residuals from least squares fitting for X-translations and rotation angles. The error values shown in the table are similar to those of [24]. These errors in pixels are equal to 6 µm in real world (0.3 mm per pixel in our camera) and these are small enough for the precise alignment required in factory automations even if there might be various disturbances like noise, blur and partial occlusions.

3.3.6 Failure Cases

Typical examples of our failure cases in the experimental evaluation are presented in Figure 3.15. These failures are mainly due to the following reasons.

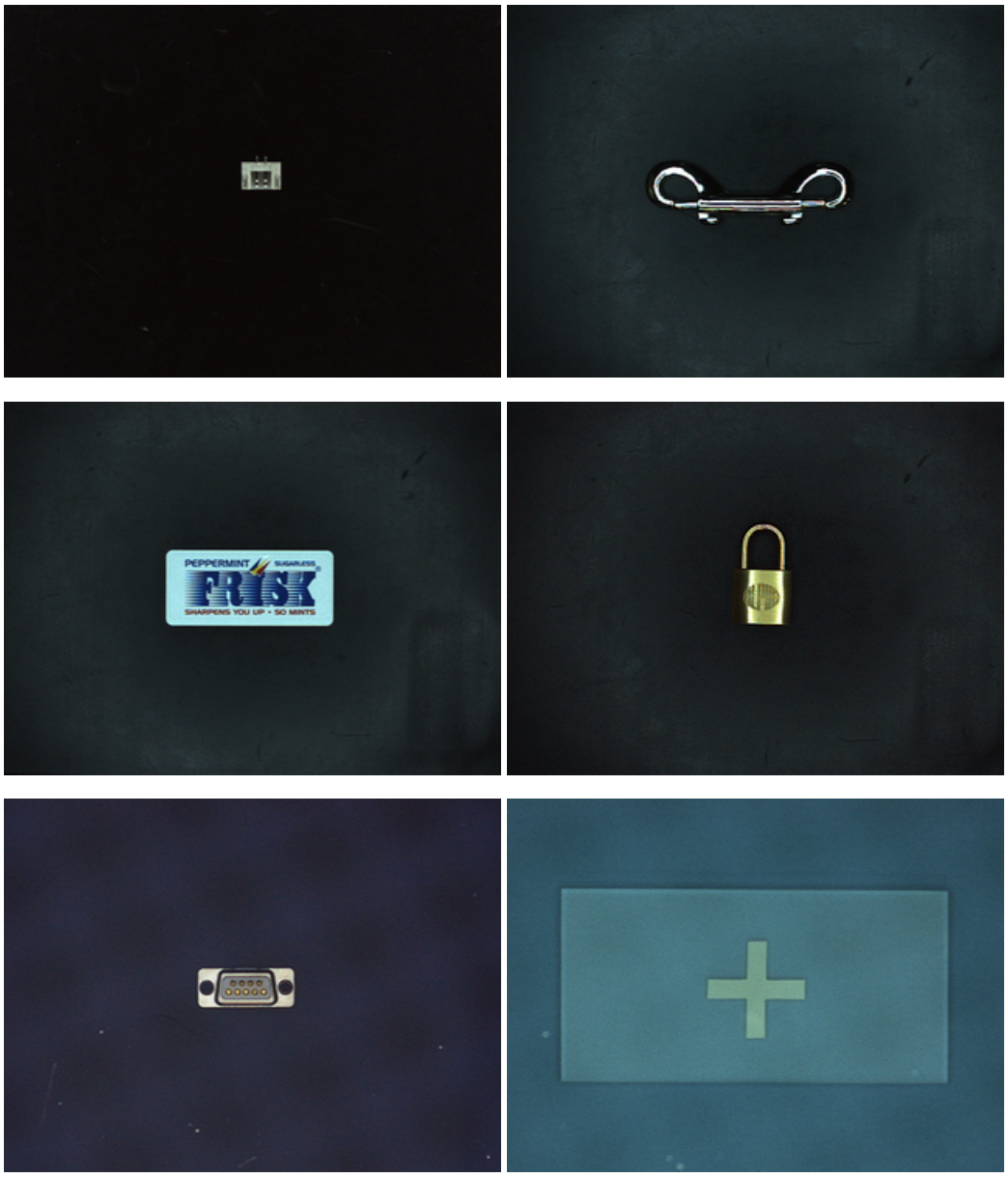


Figure 3.14 Example images of the detection error tests. 1st row: Connector and Key. 2nd row: Frisk and Padlock. 3rd row: Dsub and Cross.

Table 3.8 Detection errors in repeatability, linearity and rotation.

	repeatability (pix)	linearity (pix)	rotation (deg)
Connector	0.006	0.022	0.023
Key	0.005	0.020	0.010
Frisk	0.006	0.018	0.038
Padlock	0.007	0.014	0.016
Dsub	0.005	0.017	0.017
Cross	0.013	0.015	0.013
Mean	0.007	0.018	0.019

Partial occlusion

Our proposed algorithm sometimes fails to detect occluded objects (1st row of Figure 3.15). We use the orientation of gradients as a feature for matching and clear gradients are often observed around the outline of the objects. Therefore, the occlusions of object outlines have large influence on our detection rate. In D-Textureless dataset, the objects are not detected when approximately 40 % of the object outline is occluded. Contrastingly in CMU_KO8 dataset, the objects whose outline is occluded by more than 60 % are correctly detected (Figure 3.11). This is because the robustness against partial occlusions depends on the variation of the appearance of model templates. In-plane rotations and changes in scale of the objects also should be searched in D-Textureless dataset, but not in CMU_KO8 dataset. More variations of model templates tend to increase false positives at the background and higher thresholds for the similarity scores are required to suppress them. This degraded our robustness against partial occlusions in D-Textureless dataset.

Background clutter

The examples of wrong matches in background are shown in 2nd row (D-Textureless dataset), 3rd row (CMU_KO8 dataset) and 4th row (our additional dataset) of Figure 3.15. The wrong matches occur at the objects in backgrounds which have similar shapes or patterns of texture. Moreover, shadows and light reflections also sometimes cause false positives as shown in the center and right of 4th row. The simple

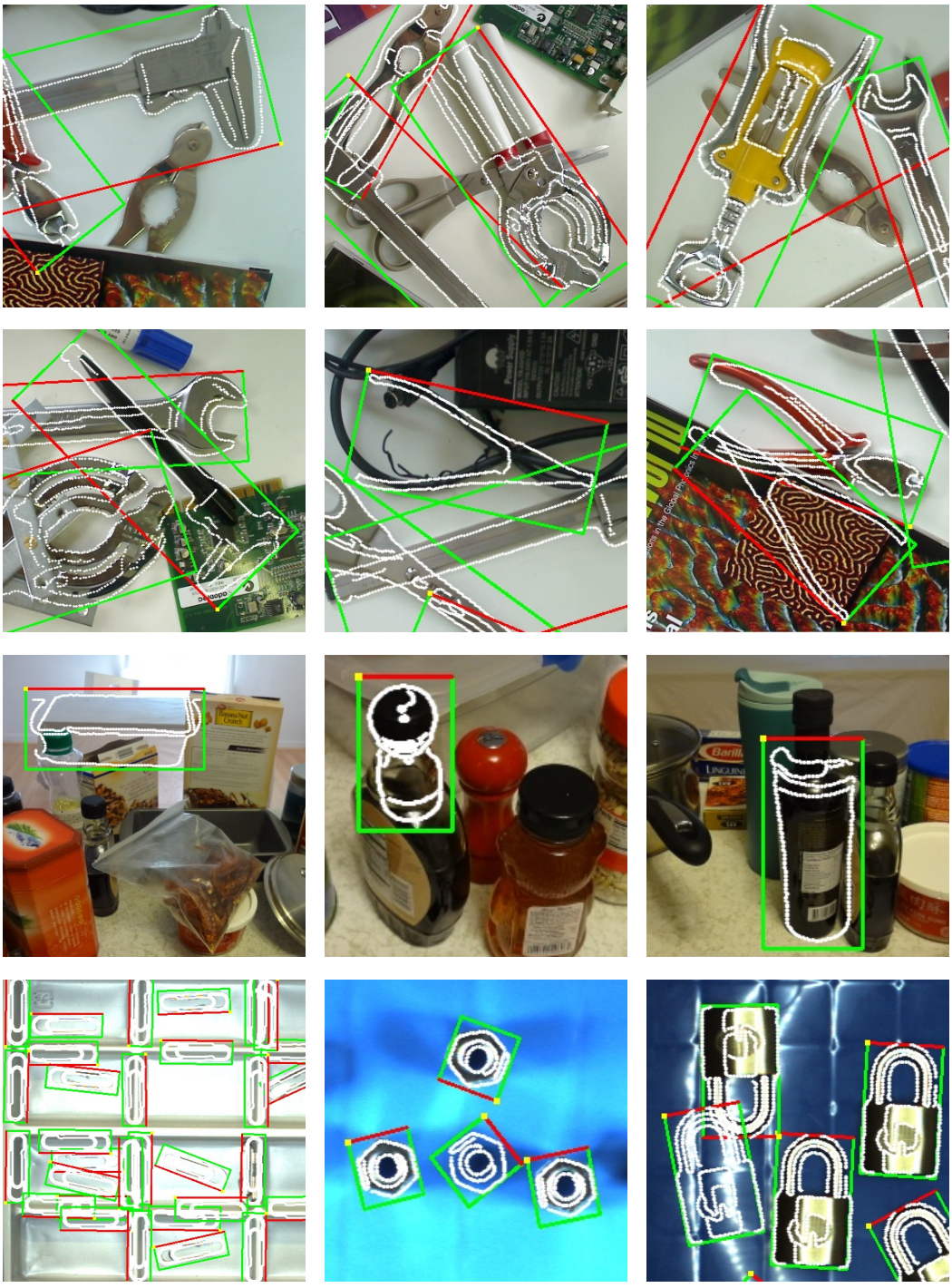


Figure 3.15 Example images of the failure cases of our proposed method. 1st row: The target objects from D-Textureless dataset were not recognized due to partial occlusions. The examples of false positives from D-Textureless dataset (2nd row), CMU_KO8 dataset (3rd row) and our additional dataset (4th row) due to background clutters, shadows and light reflections.

shape of the target object like paper clips (left of 4th row) is naturally prone to be matched with the background.

3.4 Conclusion

In this chapter, we proposed the cumulative orientation feature (COF) which is extracted from hundreds of randomly 2D-transformed images. The experimental results showed that our COF tolerated the appearance changes caused by the changes in 2D object pose without degrading the robustness against background clutters compared to the existing orientation features. We also proposed the hierarchical object detection pipeline using COF. Our proposed method was evaluated on two publicly open dataset and compared with the existing detection algorithms. The results showed that our proposed method was faster and more robust against background clutters and partial occlusions than the existing methods. Moreover, the detection errors of our algorithm is approximately 0.007 pixel in the repeatability test, 0.018 pixel in the linearity test and 0.019 degrees in the rotation test. These errors are small enough for any alignment applications of parts and machines in factory automation.

Chapter 4

3D Object Detection and Pose Estimation from a Monocular Image

In this chapter, a fast and robust algorithm for estimation of 3D object position and pose (6-DoF pose) from a monocular image is presented. Our proposed algorithm is based on template matching and the model templates are made only from 3D CAD of target objects. As described in Section 2.4, the template matching based approach can handle texture-less objects such as hammers, mugs and mechanical/electronic parts those are often seen in daily lives and factories. The model training which requires less time and effort is also important for on-site training in real applications using service/industrial robots.

We propose a novel image feature and a tree-structured model for fast template based 6-DoF pose estimation. The former is Perspectively Cumulated Orientation Feature (PCOF) extracted using 3D CAD data of target objects. PCOF is robust to the appearance changes caused by the changes in 3D object pose, and the number of templates are greatly reduced without loss of pose estimation accuracy. The latter is Hierarchical Pose Tree (HPT), which is also introduced for efficient 6-DoF pose search. HPT consists of hierarchically clustered templates whose resolutions are different at each level, and it accelerates the subwindow search by a coarse-to-fine strategy with an image pyramid.

The remaining contents of this chapter are organized as follows: Section 4.1 presented related work on image features for texture-less object detection and data

structures for efficient search. Section 4.2 introduces our proposed PCOF, HPT and 6-DoF pose estimation algorithm based on them. Section 4.3 evaluates our proposed method and compare it with state-of-the-art methods. Section 4.4 concludes this chapter.

4.1 Related Work

This section presents the existing researches which are closely related to our proposed PCOF and HPT. PCOF is developed from COF (Cumulative Orientation Feature) which is introduced in Subsection 3.2.1. Though COF is robust against the appearance changes caused by the changes in 2D object pose, it does not explicitly handle appearance changes caused by the changes in 3D object pose.

Search strategies and data structures are also important for template based approaches. The tree-structured models are popular in the nearest neighbor search for image classification [138, 139, 140] and for joint object class and pose recognition [54]. These tree-structured models were also used in joint 2D detection and 2D pose recognition [141] and joint 2D detection and 3D pose estimation [142]. Though they offered efficient search in 2D/3D object pose space, they were not efficient in 2D image space (X/Y translations). The coarse-to-fine search [12, 17] is well-known efficient search in 2D image space. Ulrich et al. [47] have proposed a hierarchical model which combined the coarse-to-fine search and the viewpoint clustering based on similarity scores between templates. However, their model is not fully optimized for the search in 3D pose space when 2D projection images from separate viewpoints are similar, as is often the case with texture-less objects.

4.2 Proposed Method

Our proposed method consists of a image feature for dealing with the appearance changes caused by the changes in 3D object pose (Subsection 4.2.1) and a hierarchical model for an efficient search (Subsection 4.2.2). The template based 6-DoF pose estimation algorithm using both PCOF and HPT is described in Subsection 4.2.3.

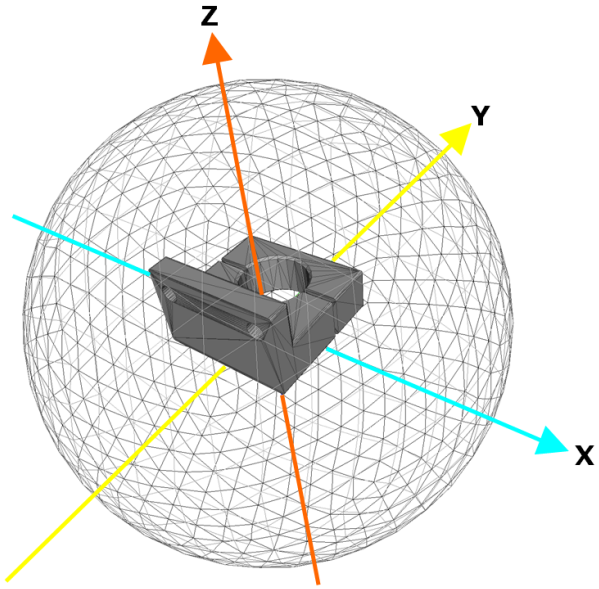


Figure 4.1 3D CAD of L-Holder, its coordinate axes and a sphere for viewpoint sampling.

4.2.1 PCOF: Perspectively Cumulated Orientation Feature

Our proposed PCOF is developed from COF [143]. COF can handle appearance changes induced only by 2D object pose changes (in-plane rotations and scales) and the possible application is detection and pose estimation of planar objects on flat tables or conveyor belts. On the other hand, PCOF explicitly handle appearance changes caused by 3D object pose changes and the possible application could be extended to detection and pose estimation of objects with various shapes those are randomly piled in a bin.

The way how to extract PCOF is explained using L-Holder shown in Figure 4.1 which is a typical texture-less object. Firstly many 2D projection images are generated using 3D CAD from randomized viewpoints. The viewpoints are determined by four parameters those are rotation angles around X/Y axes, a distance from the center of the object and a rotation angle around a optical axis. The range of randomized parameters should be limited so as to a single template can handle the appearance changes caused by the randomized parameters. In our research, the range of randomization were experimentally determined and those were ± 12 degrees around X/Y axes, ± 40 mm in the distance and ± 7.5 degrees around the optical axis. Figure

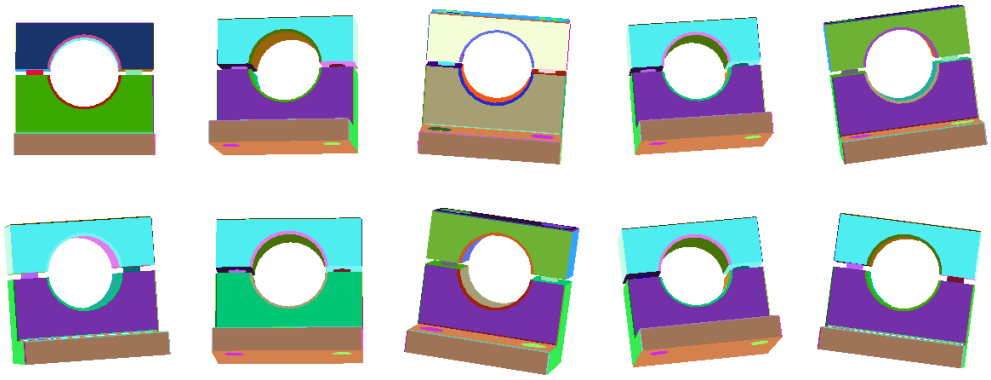


Figure 4.2 Examples of the generated projection images from randomized viewpoints around the viewpoint on z-axis (upper-left image). Surfaces of objects are drawn by randomly selected colors in order to extract distinct image gradients.

4.2 shows examples of generated projection images. The upper-left image is the projection image from the viewpoint where all rotation angles are zero and the distance from the object is 680 mm, and this viewpoint is at the center of these randomized examples. In generation of projection images, the neighboring meshes where the angle between them is larger than a threshold value are drawn by different color in order to extract distinct image gradients. In this thesis, the threshold was 30 degrees.

Secondly image gradients of all the generated images are computed using Sobel operators (the maximum gradients among RGB channels are used). We use only the gradient directions and discard the gradient magnitudes because the magnitudes depend on the randomly selected mesh colors. The colored gradient directions of the central image (the upper-left in Figure 4.2) are shown in Figure 4.3(a). Then the gradient direction is quantized into eight orientations disregarding its polarities (Figure 4.3(b)), and the quantized orientation is used for voting to the orientation histogram at each pixel. The quantized orientations of all the generated images are voted to the orientation histograms at the corresponding pixels. Lastly the dominant orientations at each pixel are extracted by thresholding the histograms and they are represented by 8-bit binary strings [26]. The maximum frequencies of the histograms are used as weighting factors in calculating a similarity score.

The orientation histograms, extracted binary features and their weights on arbitrarily selected four pixels are shown in Figure 4.4. In our study, the number of

(a) (b)

Figure 4.3 (a) Colored gradient directions of the upper-left image in Figure 4.2. (b) Quantization of gradient directions disregarding their polarities.

generated images was 1,000 and the threshold value was 120. The votes were concentrated on a few orientations at the pixels along lines or arcs such as pixel (2) and (3). At these pixels, the important features with large weights were extracted. On the contrary, the votes were scattered among many orientations at the pixels on corners and complicated structures such as pixel (1) and (4). At these pixels, the features with small or zero weights were extracted. These tendencies are also observed on the image in Figure 4.5 which represents the feature weights as pixel values. The template T with n PCOF (excluding the pixels with zero-weight) represented as follows:

$$T : \{x_i, y_i, ori_i, w_i | i = 1, ..., n\}, \tag{4.1}$$

and the similarity score is given by following equation,

$$score(x, y) = \frac{\sum_{i=1}^{n} \delta_k(ori^I_{(x+x_i, y+y_i)} \in ori^T_i)}{\sum_{i=1}^{n} w_i}. \tag{4.2}$$

If the quantized orientation of the test image (ori^I) is included in the PCOF template (ori^T), the weight (w) is added to the score. The delta function in Eqn. (4.2) is calculated quickly by a bitwise AND operation (the symbol $\wedge$). Additionally, this calculation can be accelerated using SIMD instructions where multiple binary features are matched by a single instruction.

$$\delta_i(ori^I \in ori^T) = \begin{cases} w_i & \text{if } ori^I \wedge ori^T > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

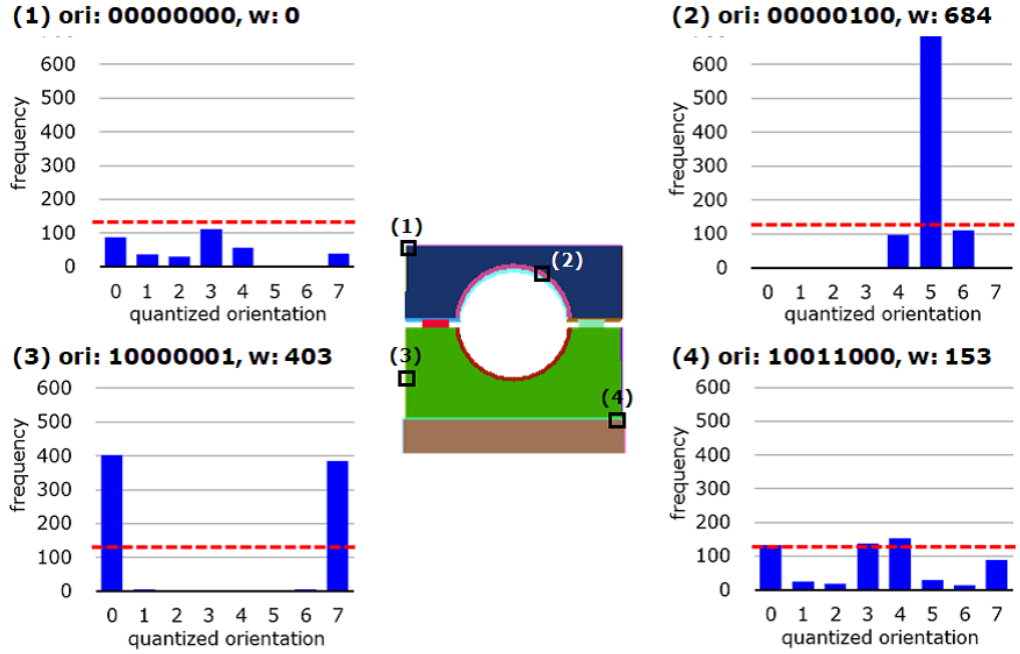


Figure 4.4 Examples of the orientation histograms, binary features (ori) and their weights (w) on arbitrarily selected four pixels. Red dotted lines show the threshold for feature extraction.

4.2.2 HPT: Hierarchical Pose Tree

A single PCOF template can handle the appearance changes caused by 3D pose changes generated in training (± 12 degrees around X/Y axes, ± 40 mm in the distance and ± 7.5 degrees around the optical axis). To cover a wider range of 3D object pose, additional templates are made at every vertices of the viewpoint sphere in Figure 4.1 which contains 642 vertices as a whole and two adjacent vertices are approximately 8 degrees apart. Additionally, the templates are made in every 30 mm

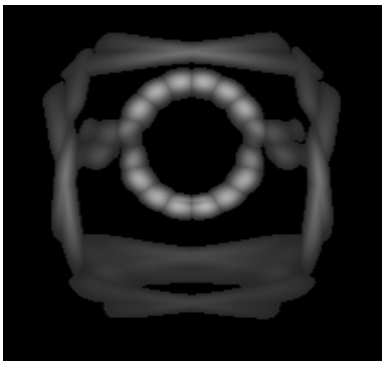


Figure 4.5 The weights of PCOF. This image represents the feature weights for L-Holder as pixel values.

Algorithm 1 Building hierarchical pose trees

Input: a number of PCOF templates T and their orientation histograms H
Output: hierarchical pose trees

```
T'_0 ← T
H'_0 ← H
i ← 1
loop
  C_i ← cluster the templates in T'_{i-1}
  for each cluster C_ij do
    H_ij ← add histograms at each pixel of H'_{i-1} ∈ C_ij
    H_ij ← normalize histograms H_ij
    T_ij ← thresholding H_ij and extract new binary features and weights
  end for
  for each T_ij and H_ij do
    H'_ij ← add histograms of nearby 2 × 2 pixels
    H'_ij ← normalize histograms H'_ij
    T'_ij ← thresholding H'_ij and extract new binary features and weights
  end for
  N'_i ← minimum number of feature points in T'_i
  if N'_i < N_min then
    break
  else
    i ← i + 1
  end if
end loop
```

in the distance to the object and in every 5 degrees around the optical axes. These PCOF templates can redundantly cover the whole 3D pose space.

Most of the image gradients are extracted around object boundaries of texture-less objects and the feature vectors of the projection images from totally different viewpoint are often similar. This is often the case with coarse image levels of the image pyramids for a coarse-to-fine search. We utilize this in order to make search of both 3D pose and 2D position more efficient and propose our hierarchical pose tree (HPT) which are built by integration and hierarchization of templates based solely on the similarities between them.

HPT is built in a bottom-up way starting from a lot of PCOF templates and their orientation histograms. The algorithm is shown in Algorithm 1 and it consists of three steps: clustering, integration and reduction of resolutions. Firstly all the templates are clustered based on the similarity scores (Eqn. 3.2) between templates using X-means algorithms [144]. In X-means clustering, the optimum number of clusters are estimated based on Bayesian information criteria (BIC). Secondly the orientation

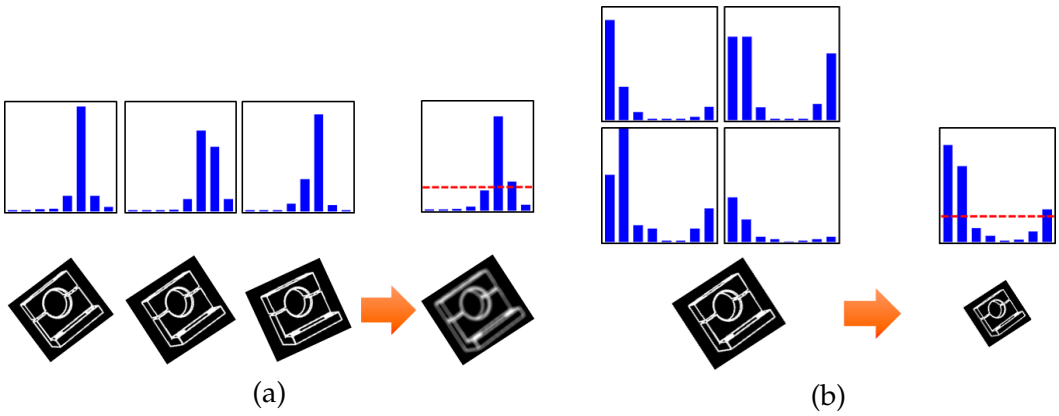


Figure 4.6 (a) Integration of orientation histograms. (b) Hierarchization of orientation histograms.

histograms which belong to a same cluster are added and normalized at each pixel. Then the clustered templates are integrated to new templates by extracting the binary features and the weights from these integrated orientation histograms (Figure 4.6(a)). Lastly the resolutions of the histograms are reduced to half by adding and normalizing histograms of neighboring 2×2 pixels (Figure 4.6(b)). Then the low-resolution features and weights are extracted from these histograms. These procedures are iterated until the minimum number of feature points contained in low resolution templates is less than a threshold value (N_{min}). In this thesis, N_{min} was 50.

Part of HPT are shown in Figure 4.7. When the range of 3D pose was as same as the settings of experiment 2 ($\pm$ 60 degrees around X/Y axes, 660 mm – 800 mm in the distance from the object and $\pm$ 180 degrees around the optical axis), the total number of PCOF templates amounted to 73,800 (205 viewpoints $\times$ 5 distances $\times$ 72 angles around the optical axis). These initial templates were clustered and integrated into 23,115 templates at the end of first round in Algorithm 1, and the number of templates was further reduced to 4,269 at second round and to 233 at third round. In this experimental setting, the iteration of hierarchization stopped at third round.

4.2.3 Pose Estimation and Refinement

In 6-DoF pose estimation, firstly the image pyramid of a test image is made and the quantized orientations are calculated on each pyramid level. Then the top level of the pyramid is scanned using the root nodes of HPT (e.g. the number of root

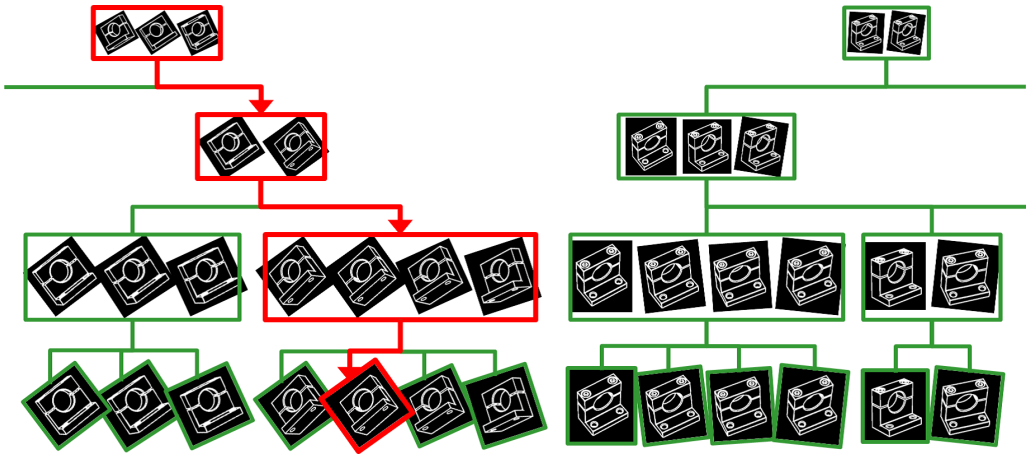


Figure 4.7 Part of hierarchical pose trees are shown. Green and red rectangles represent templates used for matching. The bottom templates are originally created PCOF templates and the tree structures are built in a bottom-up way by clustering similar templates, integrating them into new templates and decreasing the resolutions of the templates. In estimation of object pose, HPT is traced from top to bottom along the red line, and the most promising template which contains the pose parameters is determined.

nodes was 233 in experiment 2). The similarity scores are calculated based on Eqn. 4.2. The promising candidates whose scores are higher than a search threshold are matched with the templates at the lower levels, and they trace HPT down to the bottom. Finally the estimated results of 2D positions on a test image and the matched templates which have four pose parameters (three rotation angles and a distance from a camera) are obtained after non-maximum suppressions. 6-DoF object pose of these results are calculated by solving PnP problems based on the correspondences between 2D feature points on the test image and 3D points of CAD [64].

The obtained 6-DoF object pose is not precise due to the sampling of viewpoints, camera roll angles and distance to the object. Using this pose parameters as initial values, 6-DoF pose is refined based on the registration of 2D edge points pairs between the model and the input image. Firstly, 3D CAD is projected to the input image plane using the initial 6-DoF pose parameters and internal camera parameters. Then the image gradients are calculated using Sobel filter both on the projected and input images. The model edge points (x, y) and normal vectors (n_x, n_y) are extracted based on the local maxima of the gradients. The corresponding edge points (x', y') on the input image are found as a local maxima of the input gradients along the model normal vectors. Lastly, N point pairs are found and the sum of inner

products between the normal vectors and the estimated tangent vectors of the model edges is minimized using Levenberg-Marquardt algorithm.

$$p = \text{argmin} \sum_{i=1}^{N} n_x(M(x') - x) + n_y(M(y') - y) \tag{4.4}$$

$M()$ represents the combination of 3D transformation based on 6-DoF pose parameters and 2D projection based on internal camera parameters. The pose parameters p are updated and 3D CAD is projected again using new pose parameters. This optimization is iterated until the update is less than certain thresholds.

4.3 Experimental Evaluation

We carried out two experiments. One is to evaluate the robustness of PCOF against cluttered backgrounds and the appearance changes caused by the changes in 3D object pose. Another is to evaluate the accuracy and the speed for our combined PCOF and HPT to estimate 6-DoF pose of texture-less objects. Both experimental evaluation include the comparison with state-of-the-art methods. Additionally, the effects of perspective distortions on our method are described in Subsection 4.3.3 and the failure cases of our proposed method are introduced in Subsection 4.3.4.

Nine kinds of metallic parts are prepared for the evaluation (Figure 4.8). These objects are texture-less and some of them have shiny surfaces. All images were captured by industrial USB camera (STC-MC33USB, resolution: 640×480 pix, OMRON SENTECH CO., LTD.) with 16mm lens.

4.3.1 Experiment 1: Evaluation of Orientation Features

Experimental settings

In experiment 1, we evaluated four kinds of orientation features on test images of nine kinds of objects shown in Figure 4.8. The target object in cluttered background was captured by a monocular camera from the randomized viewpoints described in Figure 4.2 (the center of viewpoint range was on $z = 680$ mm and the ranges are ± 12 degrees around X/Y axes, ± 40 mm in the distance and ± 7.5 degrees around

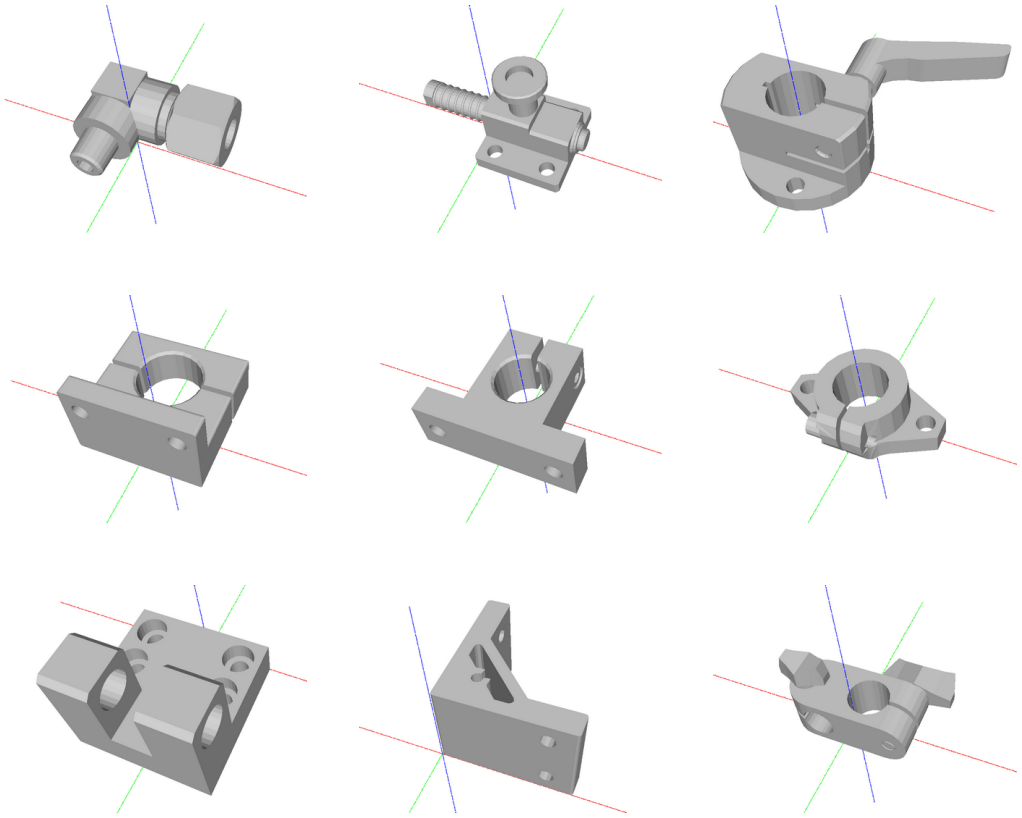


Figure 4.8 3D CAD of target objects used in experiment 1 and 2. Top: Connector, SideClamp and Stopper. Middle: L-Holder, T-Holder and Flange. Bottom: HingeBase, Bracket and PoleClamp. The red (X axis), green (Y axis) and blue (Z axis) lines represent object coordinate system.

the optical axis). The number of test images per object was approximately 100. An example image for each object was shown in Figure 4.9.

Our proposed PCOF was compared with three existing orientation features: normalized gradient vector [24], spread orientation [48] and cumulative orientation feature (COF) [143]. Existing methods used the upper-left image in Figure 4.2 as the model image.

Results

Similarity scores based on four kinds of orientation features were calculated at every pixel. The maximum score at the foreground (FG: inside object boundary) and at the background (BG: outside object boundary) were shown in Table 4.1. The difference between scores at FG and BG were shown in Table 4.2. This difference represents how discriminative each feature is against cluttered backgrounds under appearance

Figure 4.9 Example images used in experiment 1. A target object whose 3D pose is slightly transformed (less than approximately 10 degrees around X/Y/Z axes) is captured under background clutters. Nine kinds of texture-less objects are tested. Top: Connector, SideClamp and Stopper. Middle: L-Holder, T-Holder and Flange. Bottom: HingeBase, Bracket and PoleClamp.

changes induced by randomized viewpoints. The larger the score difference is, the more discriminative the feature is.

Discussion

Normalized gradient vector [24] is a unit vector of an image gradient. Though it was shown that the sum of inner products of normalized gradient vectors was occlusion, clutter and illumination invariant, this feature does not handle appearance changes of object itself. Our experimental results showed that FG scores were lower (Table 4.1) than other three features and this demonstrated that normalized gradient vector was fragile to the changes in 3D object pose. The score differences ($FG - BG$) of this feature were also lower than others (Table 4.2) and it was shown that this is the least discriminative feature in our comparison.

Table 4.1 The mean values of maximum scores at foreground (FG) and background (BG) in experiment 1.

	Normalized		Spreading		COF		**PCOF**	
	FG	BG	FG	BG	FG	BG	FG	BG
Connecotor	0.567	0.504	0.614	0.491	0.616	0.492	0.707	0.559
SideClamp	0.535	0.486	0.584	0.469	0.443	0.325	0.547	0.403
Stopper	0.507	0.421	0.581	0.400	0.707	0.360	0.800	0.398
L-Holder	0.610	0.492	0.739	0.479	0.742	0.374	0.874	0.440
T-Holder	0.610	0.470	0.773	0.466	0.751	0.378	0.879	0.480
Flange	0.596	0.510	0.670	0.470	0.732	0.316	0.852	0.406
HingeBase	0.548	0.487	0.658	0.512	0.606	0.466	0.703	0.530
Bracket	0.501	0.442	0.528	0.469	0.702	0.397	0.921	0.578
PoleClamp	0.542	0.468	0.619	0.418	0.719	0.325	0.803	0.386
Mean	0.557	0.475	0.641	0.464	0.669	0.381	0.787	0.464

Spreading orientation [48] was designed to make its similarity score robust to small shifts and deformations. The quantized orientations of test images are efficiently spread by shifting them over the range of $\pm 4 \times \pm 4$ pixels and merging them with bitwise OR operations. The model template consists from quantized orientation (being not spread) and was matched to the input features. In our experimental results in Table 4.1, FG scores were higher than normalized gradient and it was shown that spreading of orientation made its score robust to the changes in 3D object pose. However, the differences between FG and BG scores were lower than both of COF and PCOF (Table 4.2) and this indicates that simple spreading operation is insufficient for making the feature robust both to cluttered backgrounds and changes in 3D object pose.

Cumulative orientation feature (COF) [143] was proposed to make orientation features robust both to cluttered backgrounds and the appearance changes caused by the changes in 2D object pose. Following their paper, many training images were generated by transforming the model image using randomized geometric transformation parameters (within the range of ± 1 pixel in X/Y translations, ± 7.5 degrees of in-plane rotation and ± 5 % of scale) and COF was calculated at each pixel by extracting dominant orientations from the orientation histogram. The model template

Table 4.2 The mean values of differences between scores at FG and BG in experiment 1. The larger the score difference is, the more discriminative the feature is.

	Normalized	Spreading	COF	**PCOF**
Connecotor	0.064	0.122	0.125	**0.148**
SideClamp	0.049	0.115	0.118	**0.144**
Stopper	0.087	0.181	0.346	**0.402**
L-Holder	0.118	0.260	0.368	**0.434**
T-Holder	0.140	0.307	0.373	**0.399**
Flange	0.086	0.200	0.416	**0.446**
HingeBase	0.061	0.146	0.140	**0.172**
Bracket	0.059	0.059	0.305	**0.343**
PoleClamp	0.075	0.201	0.393	**0.417**
Mean	0.082	0.177	0.287	**0.323**

using COF was matched to the quantized orientations extracted on test images. The matching results in Table 4.1 and Table 4.2 showed that COF could relax the matching condition when the object pose changes and maintained the robustness to background clutters. However, the score differences of COF was still lower than those of PCOF because COF is not designed to make it robust against the changes in 3D object pose.

Perspectively cumulated orientation feature (PCOF) was calculated as described in Section 4.2.1 and matched to the quantized orientations extracted on the test images. The differences between FG and BG scores in Table 4.2 were higher than other three features and this shows that PCOF is robust both to cluttered backgrounds and the changes in 3D object pose. Due to this robustness, the template which consist of PCOF can handle a certain range of 3D object pose (approximately 8 degrees in out-of-plane rotation angles) without loss of the robustness to cluttered backgrounds. This advantage enables PCOF templates to handle a wider range of 3D object pose with fewer number of templates than other image features.

4.3.2 Experiment 2: Evaluation of 6-DoF Pose Estimation

Experimental settings

In experiment2, we evaluated the accuracy and the speed of our 6-DoF pose estimation algorithm on our texture-less object dataset (Mono-6D dataset). Nine kinds of texture-less objects (Figure 4.8) were captured from various viewpoints within the range of ± 60 degrees around X/Y axes, ± 180 degrees around the optical axis and 660 mm – 800 mm in distance from the center of the object. Approximately 500 images were taken per object where cluttered backgrounds and partial occlusions were contained. The ground truth of 6-DoF object pose were estimated based on the surrounding AR markers printed on the board where the target objects were placed on. The AR markers were recognized using ArUco library [145]. We counted the estimated 6-DoF pose as correct if the errors of the result were within 10 mm along X/Y axes, 40 mm along Z axis, 10 degrees around X/Y axes and 7.5 degrees around Z axis. The example images of our dataset are shown in Figure 4.10. The estimated results by our proposed method are drawn on the images.

The existing 6-DoF pose estimation algorithms by Ulrich et al. [47], Hinterstoisser et al. (LINE-2D) [48] and Konishi et al. (COF) [143] were also evaluated on the dataset. We used the function "find_shape_model_3d" in the machine vision library "HALCON 11" (MVTec Software Gmbh in Germany) as an implementation of [47], LINE-2D implemented in OpenCV 2.4.11 and the source code of COF which was provided by the authors. We prepared 2D projection images from the same viewpoints as PCOF (total of 205 images per object) and used them for the training of LINE-2D and COF. Regarding our algorithm, the parameters are summarized in Table 4.3. All the programs were run on a PC (Core i7 3770 3.4GHz and 8GB RAM) using a single CPU core.

Estimation accuracy

Figure 4.11 shows the curves representing the relation between the success rate of correctly estimated 6-DoF pose (vertical axis) and false positives per image (FPPI, horizontal axis). The estimation results with various search thresholds are plotted on the graphs. Even when the threshold is low and FPPI is high, the success rate

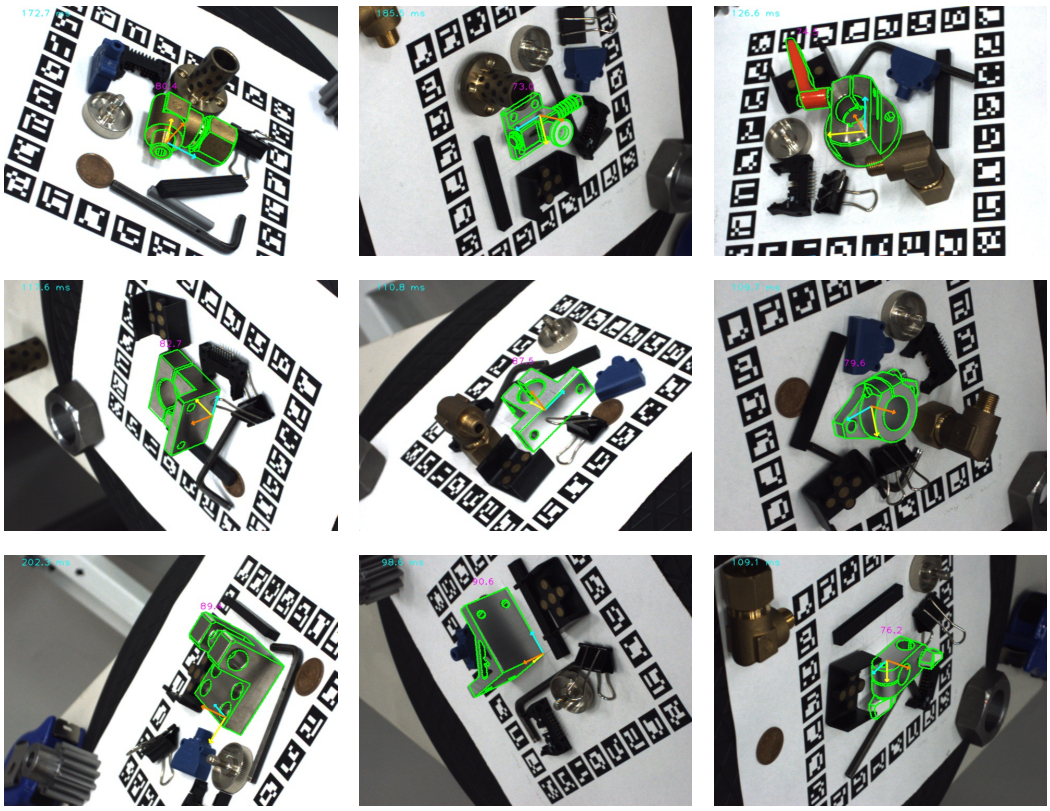


Figure 4.10 The example images of Mono-6D dataset are presented. The dataset consists of nine texture-less objects and contains cluttered backgrounds and partial occlusions. Top: Connector, SideClamp and Stopper. Middle: L-Holder, T-Holder and Flange. Bottom: HingeBase, Bracket and PoleClamp. The edges of the objects extracted from 3D CAD (green lines) and the coordinate axes (three colored arrows) are drawn on the images based on the estimated 6-DoF pose by our proposed method.

for each object is never close to 1.0. This is because 6-DoF pose estimation requires not only correct positions but also correct rotation angles around X/Y/Z axes, and the estimated rotation angles do not depend on the search thresholds. Thus some false estimation results of object pose are remained when the search threshold is zero. All the graphs indicate that our proposed method achieves higher accuracy in comparison with other existing methods.

As shown in experiment 1 (Section 4.3.1), COF and spread orientation of LINE-2D are not robust to the appearance changes caused by the out-of-plane rotations of the object. The numbers of viewpoints for making model templates are same in COF, LINE-2D and PCOF. Thus the differences in the success rate between these three methods are mainly due to the different image features.

Table 4.3 The parameters used in Experiment 2. The number of generated images (N) and threshold of orientation histograms (Th) for PCOF extraction. The intervals, ranges and number of templates for rotations (X/Y and Z) and distances to the camera in template generation.

N	Th	X/Y rot. (deg)	Z rot. (deg)	distance (mm)
1000	120	8° in $\pm 60^\circ$ (205)	5° in $\pm 180^\circ$ (72)	30 in $660 - 800$ (5)

In the algorithm of Ulrich et al. [47], the templates using normalized gradient vectors of Steger et al. [24] are made at the viewpoints sampled more densely than other three methods. Then the viewpoints are clustered based on the similarity scores between the templates. Thus the viewpoint sphere is divided into some aspects which are optimized for a single template to keep its similarity score higher than a certain threshold. This viewpoint sampling is better than the regularly spaced sampling as in COF and LINE-2D, and the success rate of Ulrich et al. is higher than those of COF and LINE-2D. However, a single template represented each aspect (clustered viewpoint) and the similarity score should be degraded at the edges of the aspect. This is because our method surpass Ulrich et al. in the success rate of correctly estimated 6-DoF object pose.

PCOF is calculated using many synthetic images from randomized viewpoints while other existing features are calculated only at a single viewpoint (a vertex of geodesic sphere). Therefore PCOF is especially effective in the situation where the appearance of an object drastically changes only by small movements of a viewpoint. This is often the case with the objects with irregular/uneven shapes such as HingeBase, Bracket, SideClamp and Stopper in our dataset (Figure 4.8). PCOF showed large advantages over existing method regarding these three objects other than HingeBase (in Figure 4.11).

The reason why the advantage of PCOF is small for HingeBase is that it has shiny surface and leads to many false matchings due to the reflection of ambient light and background clutters. This demonstrates that the material of object surface as well as the shape of object has large influence on the performance of pose estimation.

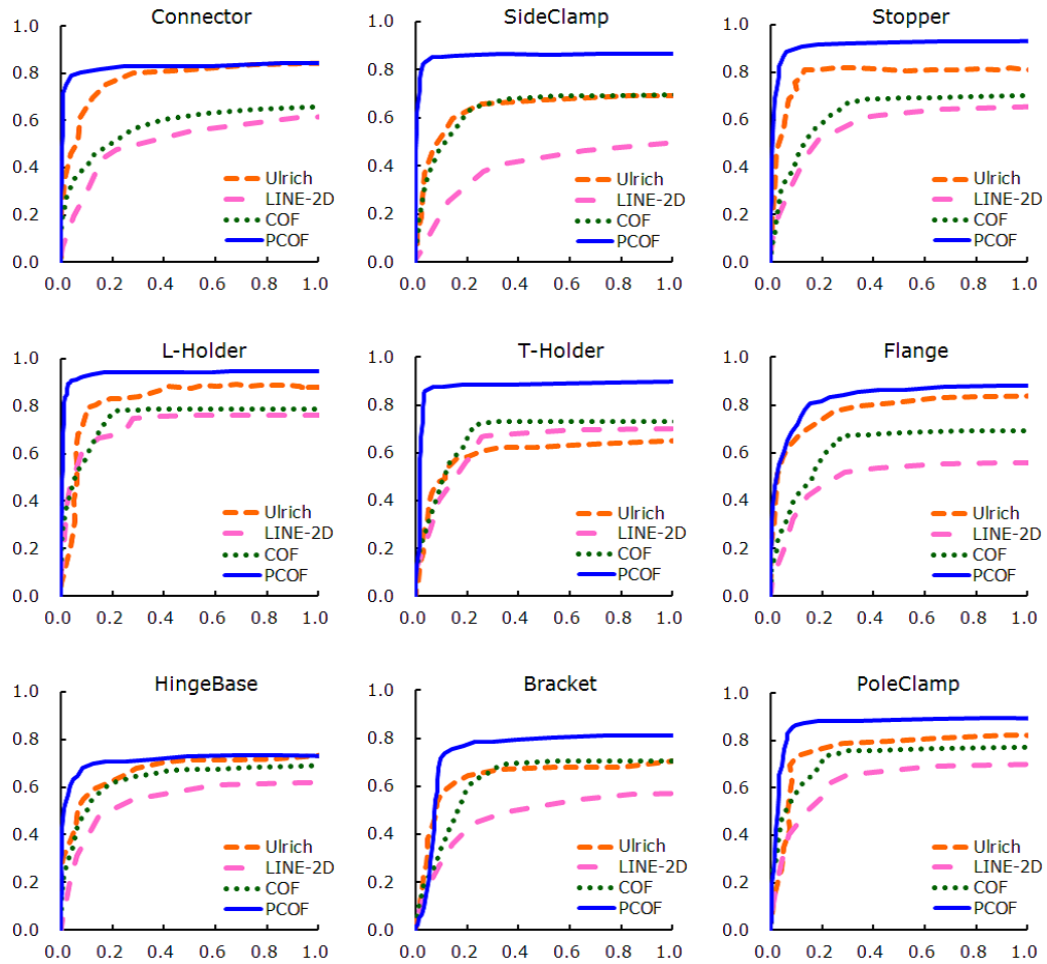


Figure 4.11 The graphs showing the relation between the success rate of correctly estimated 6-DoF pose (vertical axis) and false positives per image (FPPI, horizontal axis) are presented. There are nine graphs for each object in the dataset and the curves by four methods (Ulrich et al. [47], LINE-2D [48], COF [143] and PCOF (ours)) are drawn on each graph.

Processing time

The processing times (ms) for 6-DoF pose estimation when FPPI is 0.5 are shown in Table 4.4. Our proposed method achieved faster speed compared with the existing methods. PCOF and COF [143] use the same similarity scores calculated by bitwise ADD operations of binary features, and the main difference between them influencing the processing time is their search data structures. In COF the 2D object pose is estimated at each viewpoint independently, and the search strategy is optimized only in 2D pose space and not in 3D pose space. This is why the speed of COF was slower by approximately ten times than PCOF. The data structure of model templates in LINE-2D [48] is also not efficient for search in 3D pose space. However,

Table 4.4 The processing times (ms) for 6-DoF pose estimation in experiment 2 when FPPI is 0.5 are presented. The mean value is also shown at the bottom row.

	Ulrich	LINE-2D	COF	**PCOF**
Connector	964.1	375.8	1258.5	**167.1**
SideClamp	2724.4	383.2	1387.5	**220.4**
Stopper	2703.0	345.7	1149.9	**129.9**
L-Holder	963.8	357.1	1015.8	**122.6**
T-Holder	912.2	376.3	1140.1	**137.5**
Flange	973.0	390.5	1238.1	**137.4**
HingeBase	1137.1	348.9	1124.6	**226.1**
Bracket	792.4	358.5	961.4	**127.1**
PoleClamp	1439.0	375.9	1320.1	**137.4**
Mean	1401.0	368.0	1177.3	**156.2**

Table 4.5 Recognition rate (FPPI = 0.5) and processing time (ms) for L-Holder with and without HPT.

	recognition rate	processing time
with HPT	0.942	122.6
without HPT	0.977	126727.7

the similarity score of LINE-2D is calculated just by summing up the precomputed response maps where the memory is linearized for reducing a cache miss, and this is much faster than the computation of scores based on bitwise operations. Thus LINE-2D is much faster than COF.

Ulrich et al. [47] uses the normalized gradient vectors [24] which is not robust to the changes in 3D object pose, and their method requires more templates than PCOF in order to handle the same range of 3D object pose. Add to this, their search model is constructed by merging the neighboring viewpoints, and this is not fully efficient in the case that 2D views from separate viewpoints are similar, as is often the case with texture-less objects. Their similarity score which is based on floating-point arithmetic possibly lead to slower matching of templates. From these reasons, 6-DoF pose estimation of Ulrich et al. is slower by five to ten times than PCOF.

We have also tested how large HPT contributed to our efficient search in 6-DoF

pose space. Table 4.5 shows the recognition rate (FPPI = 0.5) and processing time (ms) for recognizing 6-DoF pose of T-Holder when all 73,800 PCOF templates were scanned without using HPT. This result demonstrates that HPT boosted the search speed by more than 1,000 times, while at the same time the recognition rate was decreased by 3.5 %. The reason for this degradation was that some of correct candidates for 6-DoF pose were discarded at higher levels (lower image resolutions) of the image pyramid. Though the recognition rate with HPT will be increased if lower score thresholds are used at the higher levels, the speed will be slower because more candidates should be matched with the model templates at the lower levels. There is a trade-off between speed and score threshold, as is often the case with hierarchical search algorithms.

Our proposed algorithm takes approximately 3 minutes for the training while the other three existing methods takes 10 seconds or so. This is because our method render a thousand of depth images at each viewpoint while the existing methods use one depth/gray image per viewpoint. Although ours takes longer time than existing methods, 3 minutes for training is quick enough for the on-site training in real applications.

Estimation error

Mean absolute errors of estimated 3D positions along X/Y/Z axes in mm for Ulrich and our proposed algorithm on Mono-6D dataset are shown in Table 4.6 and errors of estimated rotation angles around X/Y/Z axes in degrees are shown in Table 4.7. These errors are averaged only among the successful results using the threshold values of 10 mm along X/Y axes, 40 mm along Z axis, 10 degrees around X/Y axes and 7.5 degrees around Z axis, and the numbers of samples in Ulrich et al. and PCOF for computation of mean errors are different.

The errors of Ulrich et al. and our algorithm are almost the same both for translations and rotations. This is because the estimation errors depend on the registration algorithm and both Ulrich et al. and ours use the least square minimization of 2D-3D point correspondences for 6-DoF pose refinement.

The errors along Z axis are larger approximately by ten times than those along X/Y axes. This is because the projected distance along X/Y axes on the image plane

Table 4.6 Mean absolute errors of estimated positions along X/Y/Z axes in mm for Ulrich et al. and our algorithm (PCOF) on Mono-6D dataset.

	Ulrich et al.			**Ours**		
	tra X	tra Y	tra Z	tra X	tra Y	tra Z
Connector	1.120	1.120	10.936	0.978	1.038	8.679
SideClamp	1.090	1.047	13.042	0.939	0.918	9.890
Stopper	1.170	1.121	13.854	1.137	0.999	13.107
L-Holder	0.920	0.881	10.655	0.836	0.794	11.050
T-Holder	1.071	0.857	14.251	0.913	0.795	12.645
Flange	0.891	0.842	13.675	0.684	0.623	13.743
HingeBase	1.212	1.050	9.220	1.155	1.001	11.567
Bracket	1.151	1.074	9.617	1.127	1.007	10.553
PoleClamp	1.127	1.037	12.229	1.029	0.901	12.128
Mean	1.083	1.003	11.942	0.978	0.898	11.485

are much smaller than the distance along Z axis. For example, the focal length of our camera in Experiment 2 is 2210 ($f = 2210$) and the working distance is 680 mm ($Z = 680$). The projected 2D point on the image plane (x, y) of a 3D point (X, Y, Z) is calculated using the pinhole camera model:

$$\begin{aligned} x &= \frac{fX}{Z} \\ y &= \frac{fY}{Z}. \end{aligned} \tag{4.5}$$

In our case, the translation of 1 mm along X/Y axes is equal to 3.25 pixels on the image. Contrastingly, the translation of 1 mm along Z axis at X = 50 mm (average size of the target objects in Experiment 2) is equal to 0.24 pixels on the image. For this reason, it is so hard to estimate precise 3D position along Z axis from the found 2D positions at the image coordinate system. For similar reasons, the errors of rotation angles around X/Y axes (of-the-plane rotations) are larger than those around Z axis (in-plane rotations).

Due to these errors (especially translation error along Z axis) of estimated pose,

Table 4.7 Mean absolute errors of estimated rotation angles around X/Y/Z axes in degrees for Ulrich et al. and our algorithm (PCOF) on Mono-6D dataset.

	Ulrich et al.			**Ours**		
	rot X	rot Y	rot Z	rot X	rot Y	rot Z
Connector	3.039	2.526	1.550	2.921	2.327	1.483
SideClamp	2.663	1.971	1.620	2.957	2.073	1.732
Stopper	2.676	2.263	1.692	2.832	2.548	1.941
L-Holder	2.443	2.038	1.297	2.553	2.139	1.299
T-Holder	2.672	2.568	1.398	2.669	2.244	1.284
Flange	2.977	2.372	2.160	2.610	2.211	1.979
HingeBase	2.564	2.177	1.367	2.769	2.510	1.569
Bracket	2.551	2.096	1.258	2.702	2.283	1.452
PoleClamp	2.495	2.404	1.431	2.937	2.511	1.740
Mean	2.675	2.268	1.530	2.772	2.316	1.609

the application of 6-DoF pose estimation from a monocular image is somewhat limited. For example, augmented reality where the estimated pose is used for displaying some information and robotic grasping using vacuum grippers where positioning errors are allowed to some extent.

Sampling interval of viewpoints on geodesic sphere

Sampling interval of viewpoints on geodesic sphere has large influence on our pose estimation performance. When the viewpoints are sampled sparsely, the number of templates for 6-DoF pose recognition reduced and the processing time is shortened. At the same time, PCOF template should cover wider range of 3D object pose because the distances between neighboring viewpoints become longer. PCOF which is made from wider range of 3D object pose becomes less discriminative and fragile to background clutters because it includes more orientations as its features. Contrastingly, dense sampling of viewpoints leads to slower search speed due to the increased number of templates and more discriminative PCOF which is made from narrower range of 3D object pose.

We tested 6-DoF pose estimation of L-Holder when the sampling interval is doubled and halved. The range of 3D object pose when generating training images for

Table 4.8 Recognition rate (FPPI = 0.5) and processing time (ms) for L-Holder using three different numbers of viewpoints.

num. of views (sampling int.)	recognition rate	processing time
54 (approx. 15 deg)	0.803	76.7
205 (approx. 8 deg)	0.942	122.6
835 (approx. 4 deg)	0.963	151.2

PCOF was adjusted based on the sampling interval of viewpoints. The recognition rate and the processing time (ms) are shown in Table 4.8. This table shows the trade-off between sampling intervals and recognition performance, that is to say, denser sampling of viewpoints (from first to third row in the table) leads to higher recognition rate and longer processing time.

Number of templates used for matching

The reason why our proposed method uses fewer number of templates is twofold. One is that PCOF can handle a certain range of 3D object pose. Another is that HPT clusters templates based solely on the similarities between templates. The numbers of templates of L-Holder trained at each level by Ulrich's pose tree (clustering of templates from neighboring viewpoints) with Steger's image feature (normalized gradient vector), HPT with Steger, and HPT with PCOF are shown in Table 4.9. The table demonstrates that HPT reduces the number of templates at higher levels (lower image resolution) because more templates become similar due to the lower resolution. It is also shown in the table that reduced number of templates at lower levels (higher image resolution) is mainly due to PCOF which has wider coverage of 3D object pose and the neighboring templates tend to be more similar compared to the templates of normalized gradient vector (Steger's).

4.3.3 Handling of Perspective Distortion

The optical axis of the virtual camera always go through the origin of object coordinate when the projection images are synthesized for extraction of PCOF. This means that the target objects are always drawn at the centers of images in training and PCOF is not extracted from the appearances of the objects at the corners of images.

Table 4.9 The number of viewpoints and number of templates of L-Holder at each level of image pyramids.

	num. of views	level 1	level 2	level 3
Ulrich with Steger	73,800	39,835	8,937	1,024
HPT with Steger	73,800	39,245	8,010	435
HPT with PCOF	73,800	23,115	4,269	233

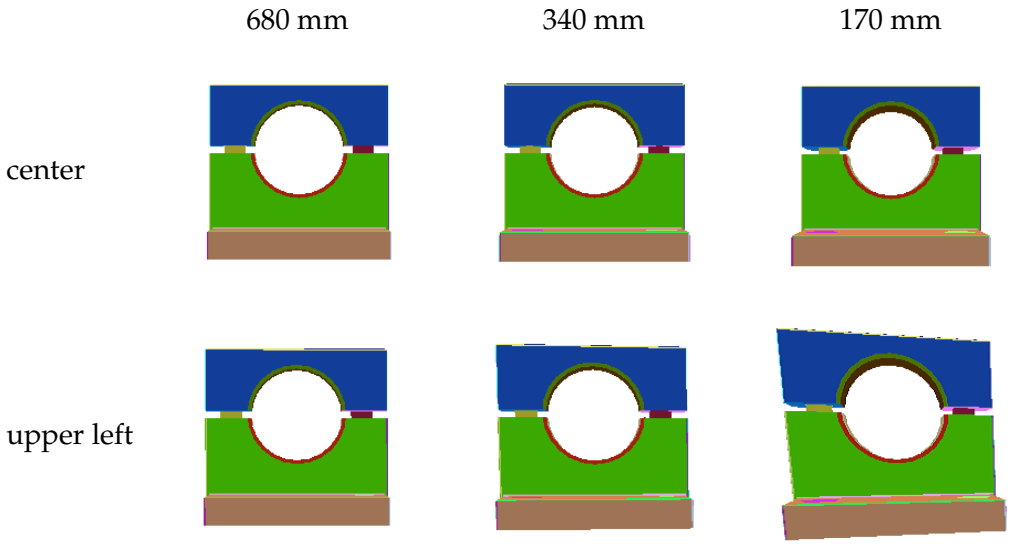


Figure 4.12 2D projection images of L-Holder which are rendered at the center (upper row) and the upper left (lower row) of the images from 3 different distances (left: 680 mm, center: 340 mm, right: 170 mm) are presented.

Therefore, PCOF might not be able to estimate object pose correctly when the objects are perspectively distorted at the corners of images. In order to test how large influence the perspective distortion has on the performance of 6-DoF object pose estimation, the projection images of L-Holder which are rendered at the center and the upper left of the images from 3 different distances (680 mm, 340 mm, and 170 mm) are shown in Figure 4.12.

The distance between the camera and the target object was 680 mm in our experimental settings. In this case the appearances of the objects at the center and at the upper left are almost the same (left column of Figure 4.12). The projected object from half distance (340 mm) is slightly distorted at the upper left corner of the image (center column of Figure 4.12) and from the shorter distance (170 mm) the object is

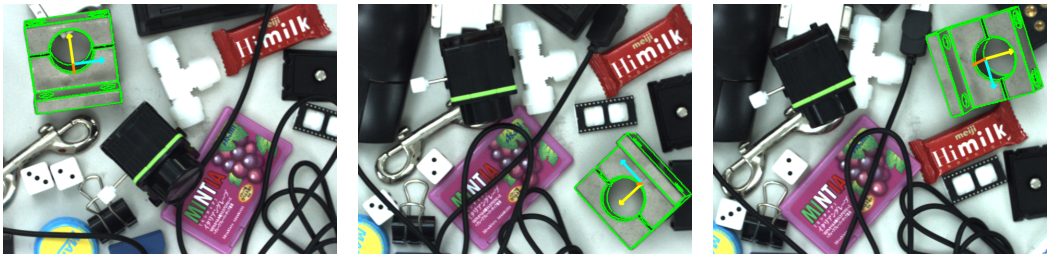


Figure 4.13 Example images of 6-DoF pose estimation results when the objects are around the corners of images.

extremely distorted (right column of Figure 4.12). This shows that PCOF cannot recognize target objects at corners of images due to perspective distortions when they are captured from close range using a lens with a short focal length. In our experimental setting (16 mm lens and working distance is 680 mm), the distortion can be ignored and PCOF can recognize 6-DoF object pose correctly as shown in Figure 4.13.

When the working distance is short (e.g. 170 mm using 4 mm lens), our proposed method can handle perspective distortions by building HPT based on PCOF templates at each local area of an image. Though this uses more memory for storing model data, the estimation speed is not degraded at all.

4.3.4 Failure Cases

Typical examples of our failure cases in Experiment 2 are presented in Figure 4.14. These failures are mainly due to the following reasons.

Partial occlusion

Our proposed algorithm sometimes fails to estimate 6-DoF pose of occluded objects (1st row of Figure 4.14). We use the orientation of gradients as a feature for matching and clear gradients are often observed around the outline of the objects. Therefore, the occlusions of object outlines significantly degrades the estimation accuracy of our algorithm.

Figure 4.14 Example images of the failure cases of our proposed method. 1st row: Stopper, T-Holder and HingeBase were not recognized due to partial occlusions. 2nd row: There were false positives of SideClamp, T-Holder and Flange due to background clutters. 3rd row: The 3D pose of AirNozzle, FluoroConnector and UrethaneTube were erroneously estimated due to less-visible edges. 4th row: 3D pose estimation of Connector, Bracket and PoleClamp were failed due to partial correspondences. The target objects in 3rd row are from an additional experiment and others are from Mono-6D dataset.

Background clutter

The examples of wrong matches in background are shown in 2nd row of Figure 4.14. SideClamp, L-Holder and Flange in Mono-6D dataset are fitted to other objects in background. The clear image gradients are usually extracted around object's outlines and the false matches are prone to occur when there is a shape in background which is similar to the outlines of the target object from any viewpoints. Moreover, metallic objects in background produce many false edges and gradients, which easily output more false positives as shown in the example of Flange (right of 2nd row).

Less-visible edges

As described above, the gradient orientation features extracted from the outlines of objects are crucial for our pose estimation algorithm. However, the image gradients around the ridges and corners of the objects are also important for determining 3D pose. These edges and gradients are sometimes less-visible and this often produces false pose matching as shown in 3rd row of Figure 4.14. The typical objects are ones with dark color (AirNozzle) and translucent objects (FluoroConnector and UrethaneTube). Additional lighting might make these ridges and corners more clearly visible and possibly alleviate this problem.

Partial correspondence

Though the last case is similar to the previous one, the edges and gradients of the objects are visible (4th row of Figure 4.14). These failures are due to the similar appearances between wrong 3D pose and input object. To overcome this failure, the classifier which is trained so that it can discriminate small differences of the appearances would be a good solution. After the rough position of the target object is detected by our template based algorithm, the pose classifier (or regressor) which is specifically trained to estimate 3D pose of the object is applied to the detected area. The pose classifier can be prepared and applied only for the ambiguous poses in order to prevent pose estimation from becoming slower for real applications.

4.4 Conclusion

In this chapter, we proposed PCOF and HPT for template based 6-DoF pose estimation of texture-less objects from a monocular image. PCOF is extracted from randomly generated 2D projection images using 3D CAD to explicitly handle a certain range of 3D object pose. HPT is built by clustering 3D object pose based solely on the similarities between 2D views and reducing the resolutions of PCOF features to accelerate 6-DoF pose estimation using a coarse-to-fine search. The experimental evaluation demonstrated that PCOF was robust both to cluttered backgrounds and the appearance changes caused by the changes in 3D object pose. Another experimental result showed that our 6-DoF pose estimation algorithm based on PCOF and HPT achieved higher success rate of correctly estimated 6-DoF pose and faster speed in comparison with state-of-the-art methods on our Mono-6D dataset. Our model training requires only 3D CAD of target objects and takes 3 minutes, this is desirable for on-site training in real applications. However, the application is somewhat limited due to the estimation error of 1 cm in Z translation and the sensitivity to the occlusion of object outlines, the lighting conditions and the background clutters.

Chapter 5

3D Object Detection and Pose Estimation from a RGB-D Image

In this chapter, we introduce fast and accurate algorithm for estimation of 3D object position and pose (6-DoF pose) from a RGB-D image. As with the algorithm for 6-DoF pose estimation from a monocular image introduced in Chapter 4, we employ template matching based algorithm in order to handle texture-less/simple-shaped objects those are often seen in real applications. The model training requires only a 3D CAD of a target object and takes a few minutes, which is suited for on-site training.

Our proposed method consists mainly of three technical elements: Multimodal Perspectively Cumulated Orientation Feature (PCOF-MOD), Balanced Pose Tree (BPT) and optimal memory rearrangement for a coarse-to-fine search. PCOF-MOD and BPT are developed and modified based on PCOF (described in Subsection 4.2.1) and HPT (described in Subsection 4.2.2) in order to make them applicable and optimum to RGB-D images. Another idea of making a coarse-to-fine search faster using SIMD instructions is introduced in this chapter.

The remaining contents of this chapter are organized as follows: Section 5.1 presents the existing work regarding our proposed method. After explaining three technical elements and whole pipeline of our proposed method in Section 5.2, Section 5.3 shows the experimental results both in tabletop and bin-picking scenes which simulate manipulation tasks of service and industrial robots. Section 5.4 concludes this chapter.

5.1 Related Work

In this section, the existing researches which are closely related to our proposed PCOF-MOD, HPT and optimum memory rearrangement. Firstly, PCOF-MOD is developed from PCOF which was presented in Subsection 4.2.1 as an image feature which was extracted from randomly generated 2D projection images using 3D CAD to explicitly handle a certain range of 3D object pose. This idea is also useful for RGB-D images and some depth features should be added. Hinterstoisser et al. [58] have proposed the binary feature which represented discretized normal orientations. This feature represents the shapes of object surfaces and complements the gradient orientation which represents the object contours. We add this discretized normal orientations to PCOF by constructing the orientation histogram at each pixel using randomly transformed and synthesized depth images.

Secondly, BPT is modified from HPT which was presented in Subsection 4.2.2 as an efficient tree-based template data structure which was built by pose clustering based solely on 2D view similarity. However, the number of child nodes in HPT sometimes becomes large because a simple-shaped object has many similar 2D views, and this lead to slower computation. Moreover, adding depth features make the template more discriminative among object 3D poses. For those reasons, regularly sampled viewpoints where the numbers of child nodes of all parent nodes are almost the same (balanced tree) are more efficient for matching of PCOF-MOD templates than the clustered viewpoints.

Lastly, the optimum memory rearrangement for coarse-to-fine search is inspired by the following existing researches. Hinterstoisser et al. [48] have proposed LINE where the response map for every discretized orientation was precomputed and the similarity score was quickly calculated just by summing up the orientation response using the look-up tables. They also restructured the response maps into linearized vectors for further speed up. Cao et al. [62] have presented the efficient template matching on GPU, in which the model templates and an input image were concatenated and vectorized respectively. We apply these image restructuring to a coarse-to-fine search [12] which is often used for making template matching faster. In a coarse-to-fine search, the promising results which are detected at the higher levels

of image pyramid are further scanned at the lower levels only around the detected positions. We rearrange that these nearby features (e.g. within 2-by-2 pixels) are linearly aligned and accessed using SIMD instructions.

5.2 Proposed Method

This section introduces our proposed methods for 3D object detection and pose estimation or 6-DoF object pose estimation. Our proposed method consists of three components: PCOF-MOD, balanced pose tree and optimal memory rearrangement. We explain them in the following three subsections and the whole pipeline including the refinement of 6-DoF object pose in the last subsection.

5.2.1 PCOF-MOD: Multimodal Perspectively Cumulated Orientation Feature for RGB-D image

We modified PCOF (Perspectively Cumulated Orientation Feature) which is introduced in Subsection 4.2.1 to make it applicable to RGB-D images. PCOF is based on the orientation of gradients extracted from RGB images and it represents the shapes of object contours. We added the orientation of surface normals extracted from depth images which represents the shapes of object surfaces. We combined these two orientation features for accurate and robust 6-DoF object pose estimation and named it PCOF-MOD (multiMODal Perspectively Cumulated Orientation Feature).

We describe the details of PCOF-MOD using CAD of iron (Figure 5.1(a)) in ACCV-3D dataset (Subsection 5.3.1) as an example. Firstly, depth images are rendered from randomized viewpoints sampled on the spheres whose coordinate axes are aligned with those of target objects. Four parameters which determine the viewpoints (rotation angles around X, Y, optical axes, distance from the object) are generated using uniformly random number in a certain range. This range of randomization should be small enough for a single model template at a viewpoint can represent the distribution of features. In our research, we experimentally determined the ranges and they are ± 10 degrees around X/Y axes, ± 7.5 degrees around optical axis and ± 90 mm from objects. Internal camera parameters for rendering depth images should

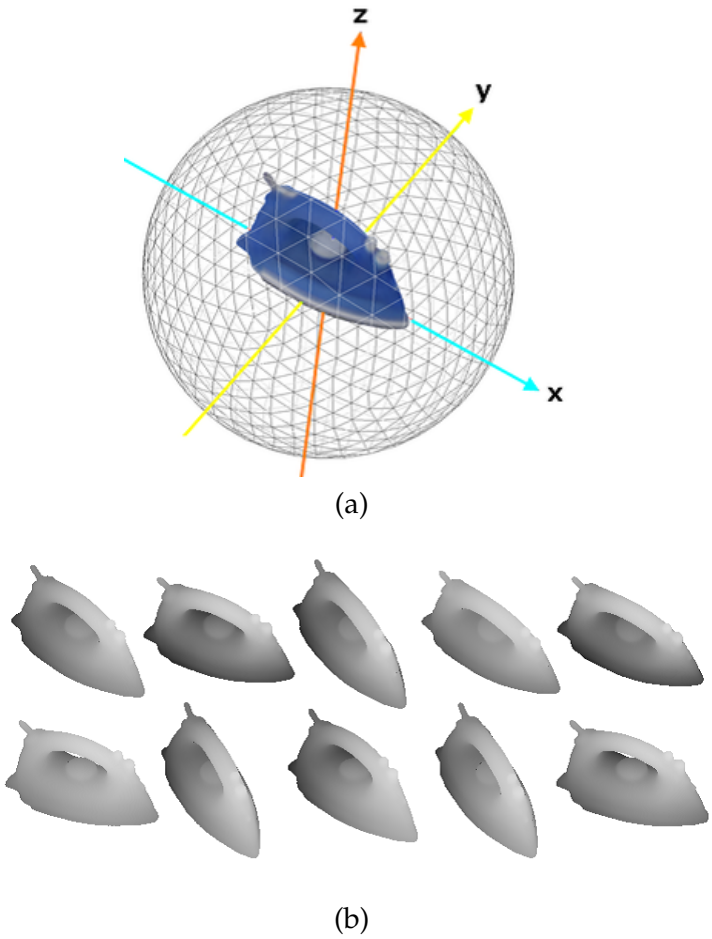


Figure 5.1 (a) 3D CAD of iron, its coordinate axes and a sphere for viewpoint sampling. (b) Examples of depth images from randomized viewpoints around a certain vertex.

be same ones of the RGB-D sensor used for pose estimation. Some of the depth images rendered using randomized viewpoints (the center of the range is at 33.9 deg around X axis, 25.5 deg around Y axis 0 deg around optical axis and 900 mm from the object) are shown in Figure 5.1(b). The upper left image is rendered at the center of the randomization range.

Secondly, gradients vectors and normal vectors are extracted from the rendered N depth images. The gradient vectors are computed only around object contours using Sobel filter and the normal vectors are computed by fitting planes to nearby pixels [58]. The colored gradients and normal vector orientations extracted from the upper left image in Figure 5.1(b) are shown in Figure 5.2(a) and (c).

Thirdly, the distributions of the gradient and normal orientations are computed

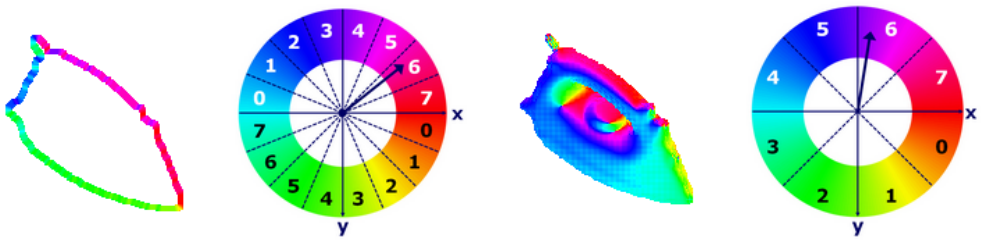


Figure 5.2 (a) Colored gradient orientations. (b) Quantization of gradient orientations. (c) Colored normal orientations. (d) Quantization of normal orientations.

at each pixel. The gradient and normal vector orientations are quantized into eight orientations (Figure 5.2(b) and (d)) and weights are added to corresponding bins. The weights are linearly interpolated between neighboring bins and added to them, for example the weights are added to bin 5 and 6 in Figure 5.2(b) and (d). When there is no depth value, no weight is added to the histogram at the pixel. Then two histograms are obtained per pixel whose maximum frequencies are N.

Lastly, we select dominant orientations whose frequencies are larger than a certain threshold (Th) and extract 8 bit binary digits where the bit of the dominant orientations are 1 and others are 0. The frequency values of the maximum bin are also extracted and used as the weighting factors for calculating similarity scores because the features with higher frequencies are more stably observed and more robust against the changes of object pose. The histograms without the dominant orientation are not used for matching.

Four examples of histograms, quantized orientation features (ori) and weights (w) are shown in Figure 5.3. These are calculated from the depth images shown in Figure 5.1(b). The pixel A and B are selected from the gradient orientation image and the pixel C and D are from the normal orientation image. The number of generated depth images (N) and the threshold for frequencies (Th) were experimentally determined, we used $N = 1000$ and $Th = 100$ for the gradient orientation and $N = 1000$ and $Th = 200$ for the normal orientation. Regarding the gradient orientations, the votes are distributed to many bins (orientations) and the dominant orientations are not obtained on the corners of objects like pixel A. Contrastingly, the votes are concentrated on a few bins and the dominant orientations with large weights are obtained on the smooth contours of objects like pixel B. Similarly on the

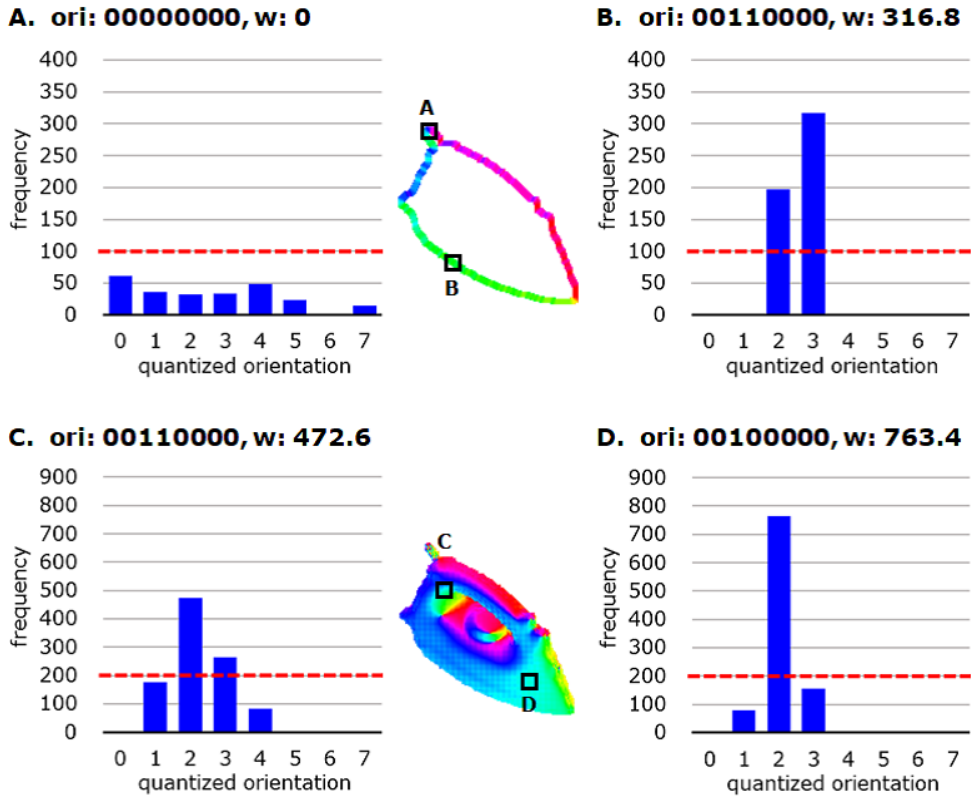


Figure 5.3 Examples of the orientation histograms, binary features (ori) and their weights (w) on arbitrarily selected pixels. Pixel A and B are extracted from gradient orientations, and pixel C and D are from normal orientations. Red dotted lines show the threshold for feature extraction

normal orientations, the orientations with smaller weights are extracted on the corner shapes like pixel C and the orientations with larger weights are extracted on the smooth surface like pixel D.

A PCOF template (T) consists of n quantized orientations (ori_i) and weights (w_i) of pixels (x_i and y_i) whose weights are larger than zero:

$$T : \{x_i, y_i, ori_i, w_i | i = 1, ..., n\}. \tag{5.1}$$

A similarity score at pixel (x, y) is calculated by following equations:

$$score(x, y) = \frac{\sum_{i=1}^{n} \delta_k(ori^I_{(x+x_i, y+y_i)} \in ori^T_i)}{\sum_{i=1}^{n} w_i}. \tag{5.2}$$

The weights are added to the score when any of the orientations of an input image are included in the orientations of model template. The delta function in Equation

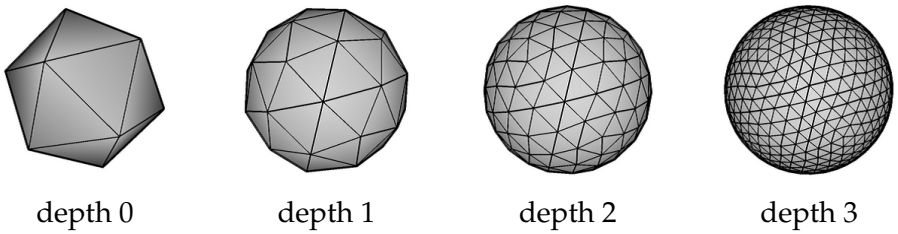


Figure 5.4 Icosahedron (left) and almost regular polyhedrons those are generated by recursive decompositions.

5.2 can be computed efficiently utilizing bitwise AND ($\wedge$).

$$\delta_i(ori^I \in ori^T) = \begin{cases} w_i & \text{if } ori^I \wedge ori^T > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.3}$$

The PCOF templates for both gradient orientations and normal orientation are made and stored when training. When testing, the similarity scores are computed separately and the sum of two scores are used for pose estimation.

5.2.2 BPT: Balanced Pose Tree

The PCOF template described in previous subsection is robust to small change in object 3D pose, e.g. within ± 10 degrees around X/Y axes, ± 7.5 degrees around optical axis and ± 90 mm of the distance to an object in our research. To cover full 3D object pose, PCOF templates are made at the viewpoints which are regularly sampled on the sphere (Figure 5.1 (a)). These viewpoints are the vertices of an almost regular polyhedron and are made by recursively decomposing an icosahedron [146]. Figure 5.4 shows this procedure where new vertices are made by dividing edges in half. It starts from the icosahedron (20 faces) shown in leftmost of Figure 5.4 and the polyhedrons with 80 faces, 320 faces and 1280 faces are obtained in sequence. The number of vertices (viewpoints) are 12, 42, 162 and 642 respectively.

We used the 1280 faced polyhedrons (642 vertices) for sampling viewpoints of PCOF templates because the angles between neighboring viewpoints are approximately 8 degrees around X/Y axes and one PCOF template (± 10 degrees) can fully cover this range. PCOF templates are also made at 70 mm intervals in distance to an

Algorithm 2 Building balanced pose tree

Input: Orientation histograms Hg_d, Hn_d, and balanced pose trees BPT with depth d
Output: Templates Tg_i, Tn_i $(i = 0, ..., d-1)$
  for $i \leftarrow d-1$ to 0 **do**
    $P_i \leftarrow$ parent viewpoints of ith level in BPT
    for each parent viewpoint $P_{i,j}$ **do**
      $C_{i+1,j} \leftarrow$ child viewpoints of $P_{i,j}$
      $Hg'_{i+1,j} \leftarrow$ add histograms of child viewpoints $Hg_{i+1} \in C_{i+1,j}$ at each pixel
      $Hn'_{i+1,j} \leftarrow$ add histograms of chile viewpoints $Hn_{i+1} \in C_{i+1,j}$ at each pixel
      $Hg''_{i+1,j} \leftarrow$ normalize histograms $Hg'_{i+1,j}$
      $Hn''_{i+1,j} \leftarrow$ normalize histograms $Hn'_{i+1,j}$
      $Hg_{i,j} \leftarrow$ add histograms of nearby 2×2 px of $Hg''_{i+1,j}$
      $Hn_{i,j} \leftarrow$ add histograms of nearby 2×2 px of $Hn''_{i+1,j}$
      $Hg'_{i,j} \leftarrow$ normalize histograms $Hg_{i,j}$
      $Hn'_{i,j} \leftarrow$ normalize histograms $Hn_{i,j}$
      $Tg_{i,j} \leftarrow$ thresholding $Hg'_{i,j}$ and extracting new binary features and weights
      $Tn_{i,j} \leftarrow$ thresholding $Hn'_{i,j}$ and extracting new binary features and weights
    end for
  end for

object and 6 degrees intervals around an optical axis so that a PCOF template (± 90 mm and ± 7.5 degrees) can fully cover these intervals.

We integrate all these PCOF templates into balanced pose tree (BPT) which consists of hierarchical templates with different resolutions and viewpoint intervals. It is well known that the coarse-to-fine search on image pyramids using hierarchical templates boosts object detection and pose estimation [17, 141]. We combine it with the coarse-to-fine sampling of viewpoints based on the hierarchical polyhedrons (Figure 5.4) because rough 3D pose estimation is enough for coarse image resolutions. This reduces the number of templates to be scanned at the coarse image layers and make pose estimation more efficient.

Our BPT consists of four layers (depth 0, 1, 2, 3) and the viewpoint sampling becomes denser at the deeper layer. We use the vertices of icosahedron shown in the left of Figure 5.4 as the root nodes of BPT and link each root node to its nearest vertices of depth 1 (80 faced polyhedron). Each parent node has three or four child nodes and this procedure is iterated from depth 1 to depth 2 and from depth 2 to depth 3. We also decrease the intervals by half for the rotation angles around optical axis and the distance to the object. Therefore, our BPT is B-tree of depth 3 where each parent node has 12 or 16 child nodes.

Each node of BPT consists both of the gradient and normal orientation templates. We have already described the way how to create the gradient orientation template (Tg_3) and the normal orientation template (Tn_3) at depth 3 in Subsection 5.2.1. The model templates at depth 2 and the upper levels are made using the templates of one level lower and the algorithm is shown in Algorithm 2. Firstly, the gradient and normal histograms (Hg_{i+1} and Hn_{i+1}) of the child nodes (C_{i+1}) which has same parent node (P_i) are added and normalized at each pixel. The number of child nodes are 12 or 16 and their 3D pose (including the angles around optical axis and the distance to the object) are slightly different. The added histograms represent wider distribution of orientation which should be handled by the parent node. Secondly, the resolution of the added histograms ($Hg''_{i+1,j}$ and $Hn''_{i+1,j}$) are reduced to half by adding and normalizing the histograms of nearby 2×2 pixels. Lastly, the binary gradient and normal orientation features and weights of the templates ($Tg_{i,j}$ and $Tn_{i,j}$) are extracted by thresholding the histograms. These procedures are iterated to depth 0 and we obtain the hierarchical templates whose resolutions of image space and 3D object pose are simultaneously reduced from the top to bottom level.

Part of BPT of iron are shown in Figure 5.5. This example does not include the templates of different rotation angles around optical axis and distance to objects, and a parent node has three or four child nodes. When model is trained, PCOF-MOD templates at depth 3 are created and then the templates whose 3D poses are similar are integrated into the templates at one level upper by adding and downsizing the orientation histograms (see Algorithm 2). In case of iron of ACCV-3D dataset (Experiment 1 in Subsection 5.3.1), the 3D object pose falls in the range of ± 90 degrees around X/Y axes, ± 45 degrees around optical axis and $650mm - 1150mm$ of distance to object. The numbers of viewpoints on hemisphere are 6, 21, 81, 321 at depth 0, 1, 2, 3. The numbers of rotation angles around optical axis are 2, 4, 8, 16 and distances to object are 1, 2, 4, 8. The numbers of templates at each depth are calculated by multiplying these numbers and amount to 12, 168, 2592, 41088.

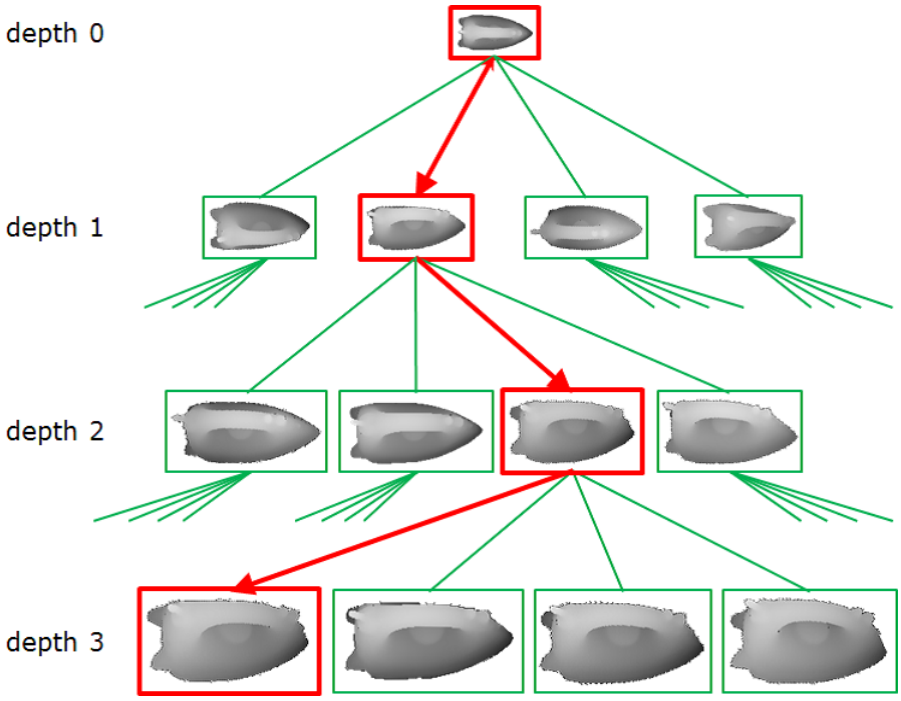


Figure 5.5 Part of the balanced pose tree of the iron are shown. The bottom templates are originally created PCOF-MOD templates and the tree structures are built in a bottom-up way by adding and downscaling of orientation histograms. In the estimation of object pose, the tree is traced from top to bottom along the red arrow

5.2.3 Pose Estimation and Refinement

In pose estimation, firstly, the image pyramids of RGB-D input are made. The quantized gradient orientations are computed at each level of RGB image and the quantized normal orientations are computed at each level of depth image. Secondly, the gradient and normal orientation template of the root nodes are scanned and matched against the top level of the gradient and normal orientation pyramids. The similarity scores of the gradients and the normals are computed using Equation 5.2 and the results whose sum of the scores are larger than a certain threshold are selected as the promising results. These results (pose and position) are further searched at the lower levels of the pyramids using the templates at the lower depth. At the bottom of the pyramids, the detected positions on the image and their 3D pose of the matched templates are obtained. Multiple results whose positions are contiguous are clustered and the results with non-maximum scores are suppressed. Lastly, the correspondences between the 2D points on the image and the 3D points on CAD are obtained and 6-DoF object pose is retrieved by solving PnP problem [64].

The obtained 6-DoF pose is not optimum because the 3D pose of the templates are spatially discretized. We refine the pose using ICP algorithm [147]. Model point clouds extracted from CAD are transformed using the initial 6-DoF pose parameters obtained by template matching and the corresponding points are searched in the input point clouds by normal shooting [148]. Then the point-to-plane metric (Equation 5.4) is minimized by linearizing the problem (assuming the rotation updates are so small that $sin(\theta) \approx \theta$ and $cos(\theta) \approx 1$) [149].

$$\mathbf{M}_{\mathrm{opt}} = \mathrm{argmin}_{\mathbf{M}} \sum_i ((\mathbf{M} \cdot \mathbf{s}_i - \mathbf{d}_i) \cdot \mathbf{n}_i)^2 \tag{5.4}$$

where M and M_opt are 4×4 3D rigid transformation matrices, s_i is model point, d_i is the corresponding input point and n_i is the unit normal vector at d_i. This procedure is iterated until the rotation updates are negligibly small.

5.2.4 Optimal Memory Rearrangement for a Coarse-to-Fine Search

Our pose estimation algorithm uses two kinds of binary features, one is quantized gradient orientations extracted from RGB image and another is quantized normal orientations extracted from depth image. The upper images of Figure 5.6 show the part of features extracted from 10-by-10 pixel size images. The numbers indicate the memory address which start at the top-left of the images.

When the template is scanned and matched exhaustively on the top of the image pyramid, the calculation of similarity scores is easily accelerated by using SIMD instructions. In case of Intel AVX intrinsics, 256 bit register is available and 32 model features (8 bit) are matched against the input features by one instruction (logical AND in Equation 5.3).

However, on the lower levels of the pyramid, the templates are searched only around the promising areas selected by the matching results of one level upper. On the image pyramids where the size of the lower image is increased to double, the templates are searched in 2-by-2 pixels, for example '0', '1', '10', '11' in Figure 5.6. In this case, the features to be matched against the templates are not linearly aligned and applying SIMD instructions is not efficient. We propose the algorithm which rearranges nearby features in a rectangular grid into a linearly aligned form and the

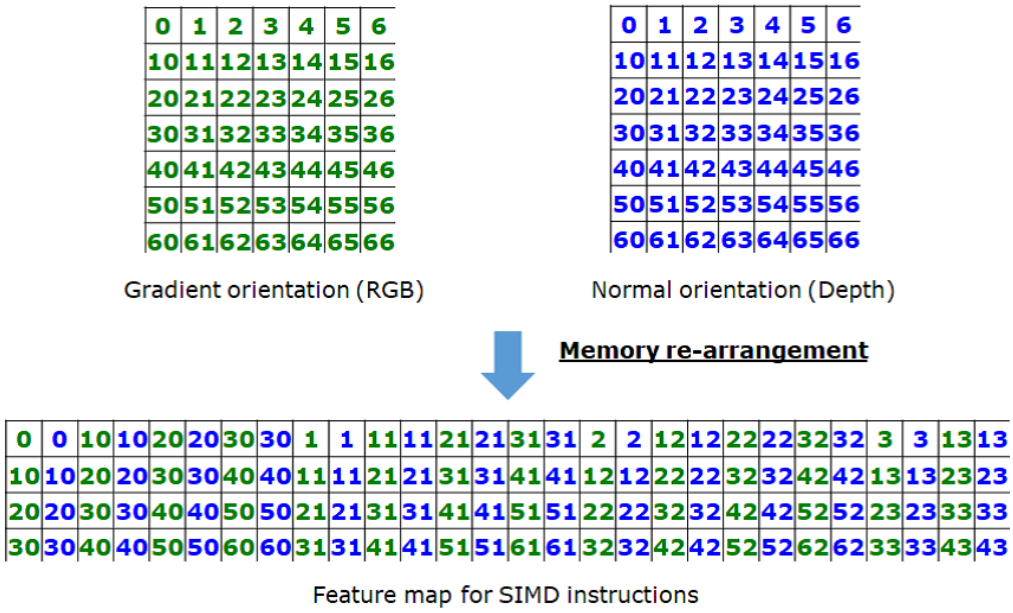


Figure 5.6 Our memory rearrangement strategy which enables highly efficient coarse-to-fine search. The upper two figures show the gradient orientation features (green) and normal orientation features (blue). The numbers indicate the memory address. These two features are mixed and re-arranged so that every 4 by 4 grid of these features are aligned (the lower figure).

template matching is done highly efficiently on the rearranged feature map using SIMD instructions.

In our research two kinds of 8 bit features are used and 32 features are processed at one time using Intel AVX instructions. In order to make full use of this, we rearrange these two kinds of features in 4-by-4 pixels into linearly aligned 32 features (the lower image in Figure 5.6). When there is promising results in 2-by-2 pixels at the upper level of the pyramids, the corresponding 4-by-4 pixels at the lower level are searched and any 4-by-4 features on the rearranged feature map can be accessed in a linearly aligned form.

Our proposed feature rearrangement for an efficient coarse-to-fine search can be applied to any length binary or floating-point features. We should note that the rearranged feature map consumes more memory footprint by 4 times than original input features.

5.3 Experimental Evaluation

We evaluated our proposed algorithm and compared it with state-of-the-art methods in different two scenes. One was tabletop scene which simulated the manipulation tasks of service robots in housing space (Experiment 1). Another was bin picking scene which simulated the manipulation task of industrial robots in factories (Experiment 2). In the last subsection, we summarize and discuss the failure cases in these two experiments.

5.3.1 Experiment 1: Evaluation on Public RGB-D Dataset in Tabletop Scenes

Experimental setting

In Experiment 1, ICVL dataset [119] (we used corrected annotation [115]) and ACCV-3D dataset [59] which were publicly available RGB-D dataset were used for the evaluation in tabletop scenes. ICVL dataset consists of 6 kinds of target objects and they have more than 500 RGB-D images per object those were captured by Carmine 1.09 (Primesense). ACCV-3D dataset consists of 15 kinds of target objects and they have more than 1000 RGB-D images per object those were captured by PSDK 5.0 (Primesense). Both of them provide CAD data of the target objects and the ground truth of 6-DoF object pose which were estimated using AR markers. Regarding ACCV-3D dataset, we evaluated 13 objects whose CAD are provided. Examples of depth images and RGB images of ICVL dataset are shown in Figure 5.7 and those of ACCV-3D dataset are shown in Figure 5.8.

The pose of objects in both dataset ranges from -90 to 90 degrees around X and Y axes, from -45 to 45 degrees around Z axis (optical axis). The distance from the camera ranges from 450 to 1100 mm for ICVL dataset and from 650 to 1150 mm for ACCV-3D dataset. For our proposed algorithm, model templates were trained in the ranges of the object pose. Our algorithm was implemented using C++ and ran on Windows PC (Core i7-7700 3.6GHz) using 4 cores. The parameters for our algorithm are summarized in Table 5.1. The estimation accuracy and speed was compared with existing template matching based [59] and learning based [119, 115, 123] methods.

We used the criteria which was defined in [59] to determine whether the estimated pose was correct. More formally, for a 3D model $\mathcal{M}$ which had ground truth

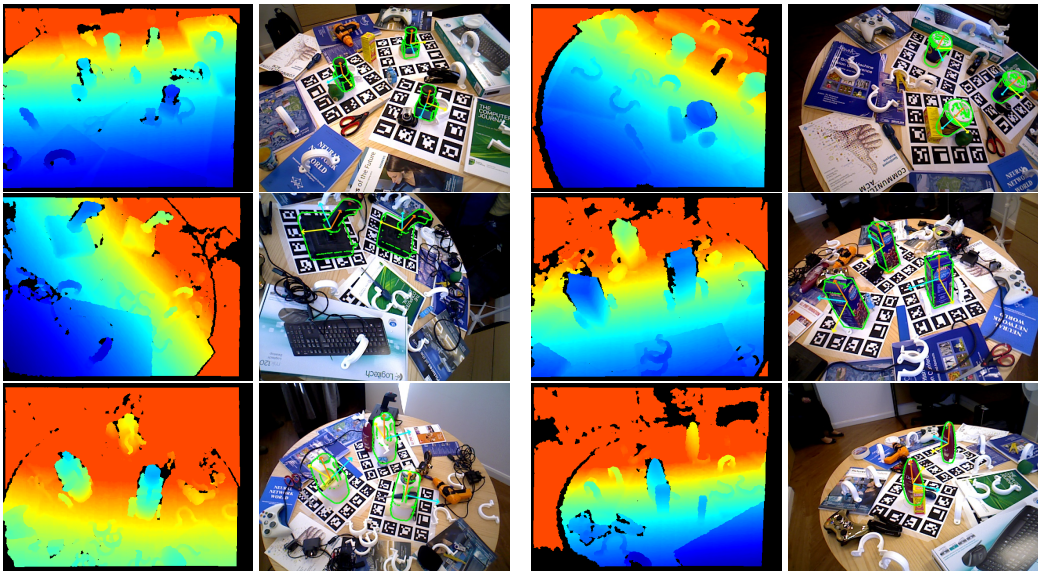


Figure 5.7 Example images of ICVL dataset in Experiment 1 (Top row: Camera and Cup, Middle row: Joystick and Juice, Bottom row: Milk and Shampoo). The depth image and RGB image are shown for each object. The edges of the objects extracted from 3D CAD data (green lines) and the coordinate axes (three colored arrows) are drawn on the images based on the estimated 6-DoF pose by our proposed method.

rotation $\boldsymbol{R}$, translation $\boldsymbol{T}$, the estimated rotation $\tilde{\boldsymbol{R}}$ and translation $\tilde{\boldsymbol{T}}$, the average distance between the model points $\boldsymbol{x}$ transformed by the ground truth and those by estimated pose is defined as:

$$m = \text{avg}_{\boldsymbol{x} \in \mathcal{M}} \|(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}) - (\tilde{\boldsymbol{R}}\boldsymbol{x} + \tilde{\boldsymbol{T}})\|. \tag{5.5}$$

for asymmetric objects and

$$m = \text{avg}_{\boldsymbol{x1} \in \mathcal{M}} \text{min}_{\boldsymbol{x2} \in \mathcal{M}} \|(\boldsymbol{R}\boldsymbol{x_1} + \boldsymbol{T}) - (\tilde{\boldsymbol{R}}\boldsymbol{x_2} + \tilde{\boldsymbol{T}})\|. \tag{5.6}$$

for symmetric objects such as Eggbox and Glue. If $k_m d \geq m$ where k_m is a chosen coefficient and d is the diameter of $\mathcal{M}$, we defined that the estimated pose is correct. The coefficient k_m was set to the value of 0.15 which was used in the existing research.

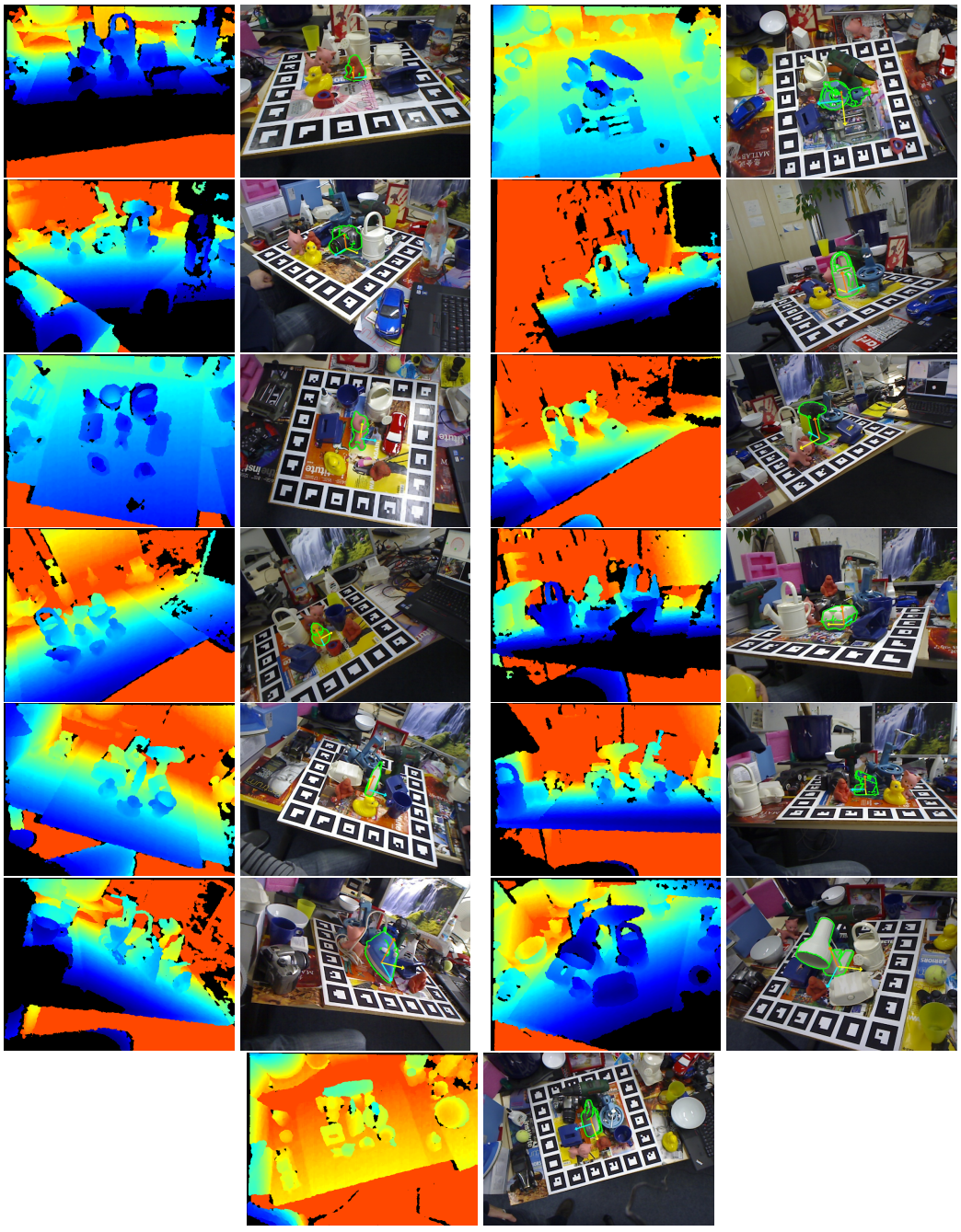


Figure 5.8 Example images of ACCV-3D dataset in Experiment 1 (Top row: Ape and Benchvise, 2nd row: Cam and Can, 3rd row: Cat and Driller, 4th row: Duck and Eggbox, 5th row: Glue and Holepuncher, 6th row: Iron and Lamp, Bottom row: Phone). The depth image and RGB image are shown for each object. The edges of the objects extracted from 3D CAD data (green lines) and the coordinate axes (three colored arrows) are drawn on the images based on the estimated 6-DoF pose by our proposed method.

Table 5.1 The parameters used in Experiment 1. The number of generated depth images (N) and threshold of gradient orientation histograms (Th_g) and of normal orientation histogram (Th_n) for PCOF-MOD extraction. The intervals, ranges and numbers of templates for rotations (X/Y and Z) and distances to the camera in template generation.

N	Th_g	Th_n	X/Y rot. (deg)	Z rot. (deg)	distance (mm)
1000	100	200	8° in $\pm 90^\circ$ (321)	6° in $\pm 45^\circ$ (16)	70 in $450 - 1100$ (10) for ICVL dataset, 70 in $650 - 1150$ (8) for ACCV-3D datase

Estimation accuracy

The graphs which represent the recall, precision and F1 score for each object of ICVL dataset are shown in Figure 5.9 and those of ACCV-3D dataset are shown in Figure 5.10. The graphs are obtained by evaluating the dataset using various search threshold. There are more true positives and false positives (higher recall) when the threshold is lower, and there are less true positives and false positives (higher precision) when the threshold is higher. F1 score is defined as a harmonic mean of precision and recall. For most objects, the highest F1 scores are obtained around the threshold value of 0.5.

The highest F1 scores for each object of ICVL dataset and mean values are shown in Table 5.2 and those of ACCV-3D dataset are shown in Table 5.3. The F1 scores by existing algorithms which were evaluated in [123] are also shown in the tables. Our proposed algorithm achieved the highest F1 score among the state-of-the-art methods on ACCV-3D dataset and second highest score on ICVL dataset.

There are two reasons why our algorithm has an advantage over existing template based (LINEMOD) and learning based (LC-HF and Deep patch) algorithms. One is that we sample viewpoints, roll angles and camera distances more densely for making model templates. On ACCV-3D dataset, our algorithm makes 41,088 templates (321 viewpoints, 16 roll angles and 8 distances) and the existing algorithms make 3,402 templates (81 viewpoints, 7 roll angles and 6 distances). Using more templates makes pose estimation more accurate and robust because the pose differences between objects in captured images and model templates become smaller.

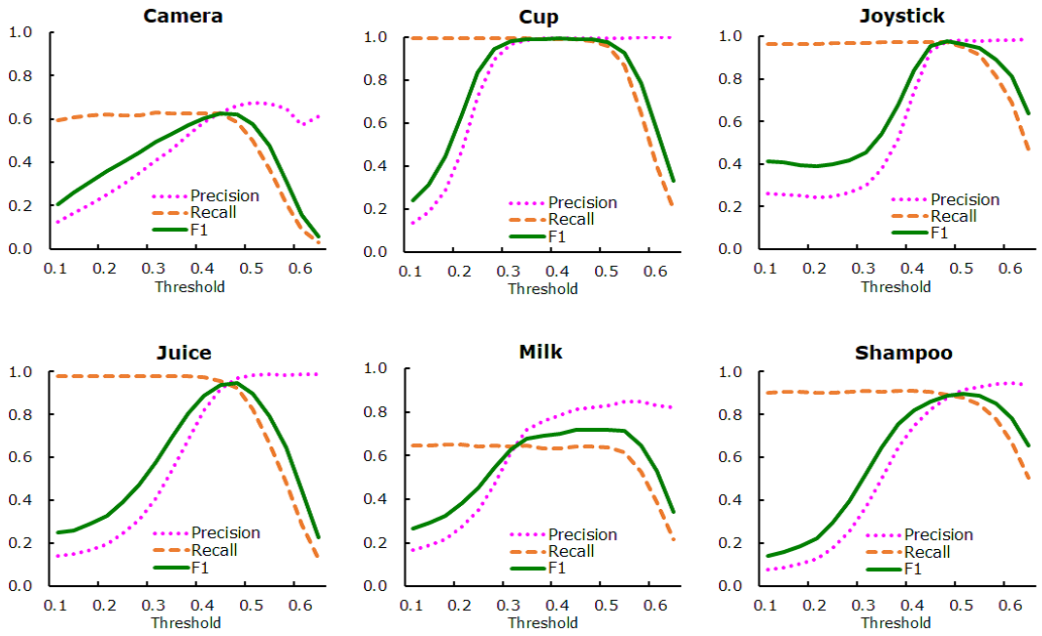


Figure 5.9 Plotting the precision, recall and F1 score for a varying threshold on ICVL dataset in Experiment 1. Top row: Camera, Cup and Joystick. Bottom row: Juice, Milk and Shampoo.

Another reason is that our proposed feature PCOF-MOD is extracted from a large number of depth images those are generated within a certain range of 3D object pose and the feature is robust to small pose changes of objects. Due to this, PCOF-MOD relax only the matching conditions for target objects without increasing false positives under cluttered background and is matched to the testing objects whose poses are slightly different from those of model templates.

Though SSD-6D is CNN based method and discriminates target objects from

Table 5.2 F1 scores on ICVL dataset for different algorithms.

	LINEMOD	LC-HF	Deep patch	SSD-6D	**Ours**
Camera	0.589	0.394	0.383	**0.741**	0.627
Cup	0.942	0.891	0.972	0.983	**0.992**
Joystick	0.846	0.549	0.892	**0.997**	0.975
Juice	0.595	0.883	0.866	0.919	**0.945**
Milk	0.558	0.397	0.463	**0.780**	0.719
Shampoo	**0.922**	0.792	0.910	0.892	0.897
Mean	0.740	0.651	0.747	**0.885**	0.859

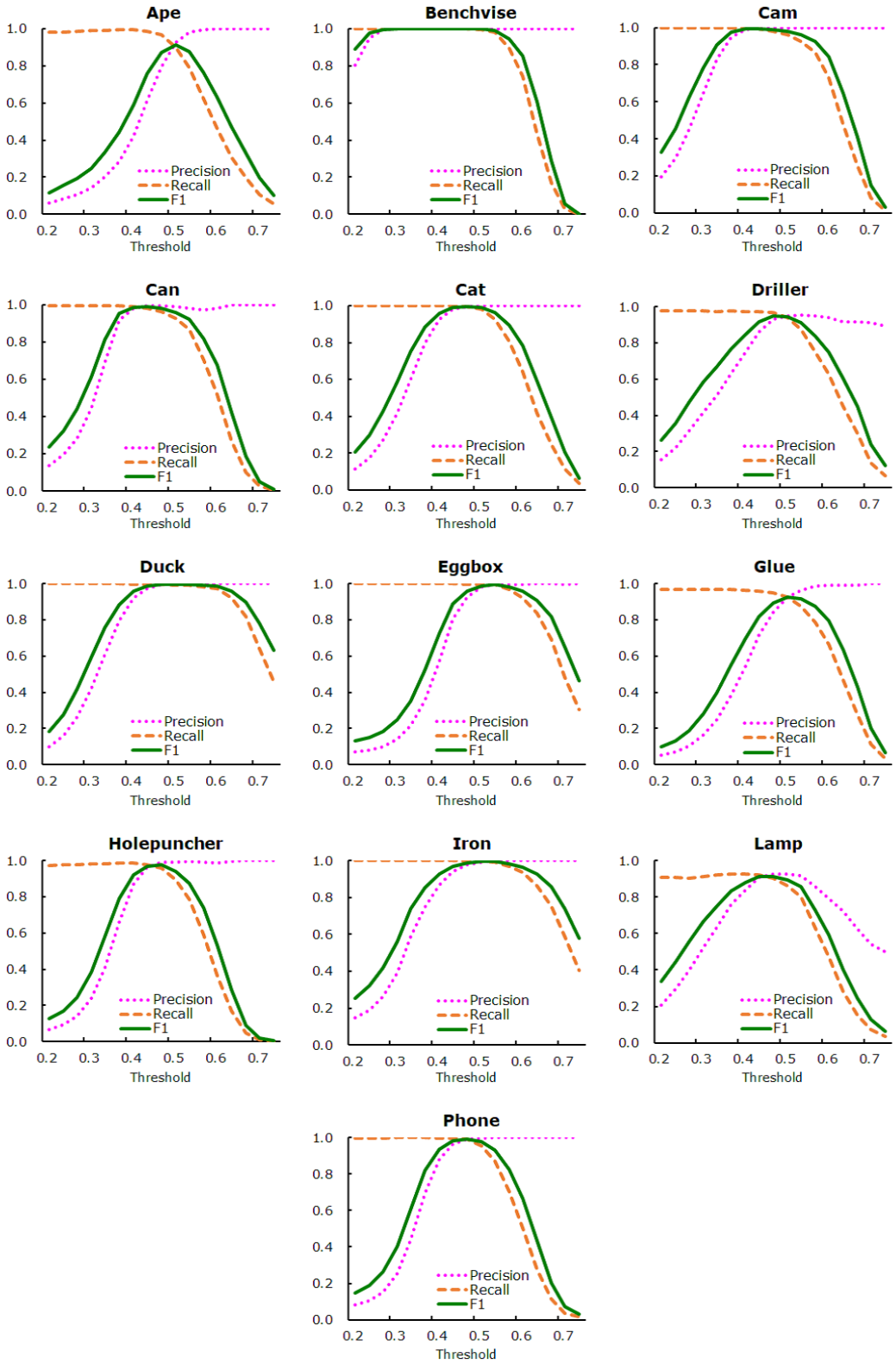


Figure 5.10 Plotting the precision, recall and F1 score for a varying threshold on ICVL dataset in Experiment 1. Top row: Ape, Benchvise and Cam. 2nd row: Can, Cat and Driller. 3rd row: Duck, Eggbox and Glue. 4th row: Holepuncher, Iron and Lamp. Bottom row: Phone.

Table 5.3 F1 score on ACCV-3D dataset.

	LINEMOD	LC-HF	Deep patch	SSD-6D	**Ours**
Ape	0.533	0.855	**0.981**	0.763	0.913
Benchvise	0.846	0.961	0.948	0.971	**0.998**
Cam	0.640	0.718	0.934	**0.922**	**0.995**
Can	0.512	0.709	0.826	0.931	**0.988**
Cat	0.656	0.888	0.981	**0.893**	**0.997**
Driller	0.691	0.905	0.965	**0.978**	0.947
Duck	0.580	0.907	0.979	0.800	**0.996**
Eggbox	0.860	0.740	**1.000**	0.936	0.995
Glue	0.438	0.678	0.741	0.763	**0.926**
Holepuncher	0.516	0.875	**0.979**	0.716	0.976
Iron	0.683	0.735	0.910	0.982	**0.995**
Lamp	0.675	0.921	**0.982**	0.930	0.914
Phone	0.563	0.728	0.849	0.924	**0.992**
Mean	0.630	0.817	0.929	0.885	**0.972**

background, it uses only RGB information. Hinterstoisser et al. [48] showed that using depth made pose estimation more robust against background clutters compared than using only RGB. This is why the F1 score of SSD-6D was lower than those of Deep patch and our proposed method which used both of RGB and depth on ACCV-3D dataset. Meanwhile, the F1 score of SSD-6D was highest on ICVL dataset. This is due to that ICVL dataset includes more occlusions compared to ACCV-3D dataset. Occlusions have influence both on RGB and depth features, and degrade more largely the performance of RGB-D based methods.

Processing time

The processing time of our proposed algorithm with and without the memory rearrangement are shown in Table 5.4 for ICVL dataset and in Table 5.5 for ACCV-3D dataset. Estimation speed is boosted by approximately 2 times for ICVL dataset and by approximately 3 times for ACCV-3D dataset when introducing the memory rearrangement technique. The memory rearrangement technique is more effective on ACCV-3D dataset than on ICVL dataset. This is because the target objects of ACCV-3D dataset are smaller in the images (e.g. Ape and Glue) than those of ICVL dataset.

Table 5.4 Processing time (ms) with and without the memory rearrangement (MemRea) on ICVL dataset.

	w/o MemRea	w/ MemRea
Camera	112.4	55.9
Cup	71.7	41.3
Joystick	57.9	23.3
Juice	71.7	34.1
Milk	58.0	31.8
Shampoo	71.8	46.8
Mean	73.9	38.9

When target objects in images are smaller, the features at top pyramid layer are less discriminative and there remained more candidates for the lower pyramid layers. Then these larger number of candidates are efficiently matched using SIMD instructions on the rearranged memory map.

Mean processing time of the existing methods for ACCV-3D dataset are shown in Table 5.6 (only the time of LC-HF were not described in the paper). These results are taken from each paper and the algorithms were executed on various computing environments. However, Deep patch and SSD-6D are CNN based methods and were evaluated using GPU (Geforce GTX Titan X for Deep patch and GTX 1080 for SSD-6D). These two methods should take more than a few seconds on CPU. When comparing ours with LINEMOD, ours is faster approximately by 3 times than LINEMOD on a rather faster CPU (Core i7-2820QM 2.3GHz for LINEMOD and Core i7-7700 3.6GHz for ours) using same number of cores (4 cores). From these, we can conclude that our proposed method is the fastest among the existing methods.

Our proposed algorithm takes approximately 3 minutes for the training and most part of it is consumed by rendering a thousand of depth images at each viewpoint. Similarly, existing template matching based [59, 60, 61] and local descriptor based methods [94, 100] take a few minutes or less for the training where templates at various viewpoints or 3D features are extracted. On the other hand, learning based methods [120, 119, 121, 122] take much longer time for training classifiers and collecting training images. Moreover, CNN based methods [128, 115, 126, 123] require more training time and samples.

Table 5.5 Processing time (ms) with and without the memory rearrangement (MemRea) on ACCV-3D dataset.

	w/o MemRea	w/ MemRea
Ape	256.5	66.5
Benchvise	73.8	28.1
Cam	132.4	57.2
Can	98.9	39.5
Cat	130.9	50.5
Driller	73.4	27.7
Duck	132.7	50.5
Eggbox	140.4	43.1
Glue	125.3	50.5
Holepuncher	180.1	55.1
Iron	62.6	25.9
Lamp	73.7	29.0
Phone	100.8	39.6
Mean	121.7	43.3

Table 5.6 Processing time on ACCV-3D dataset for various methods.

	LINEMOD	LC-HF	Deep patch	SSD-6D	**Ours**
Mean time	119	n/a	671	109	**43.3**

5.3.2 Experiment 2: Evaluation on Bin-Picking Dataset

Experimental setting

In Experiment 2, our algorithm was evaluated on our Bin-Picking dataset where target objects were piled randomly in a bin. Depth and grayscale images (1280×1024 resolution) of 6 kinds of mechanical parts were captured using an industrial 3D sensor (Ensenso X36, IDS Gmbh in Germany). A total of 60 images were captured per object which included 5 different patterns for piling, 4 rotation angles of a bin and 3 viewing angles of the 3D sensor. The pose of visible target objects (more than 70 % of the surface are captured by the sensor) were annotated by ICP refinement with manually annotated initial pose. These annotated pose were transformed to the images of the rotated bin and different viewing angles based on AR markers which surrounded the bin. 5 to 10 objects were annotated per bin and the total numbers

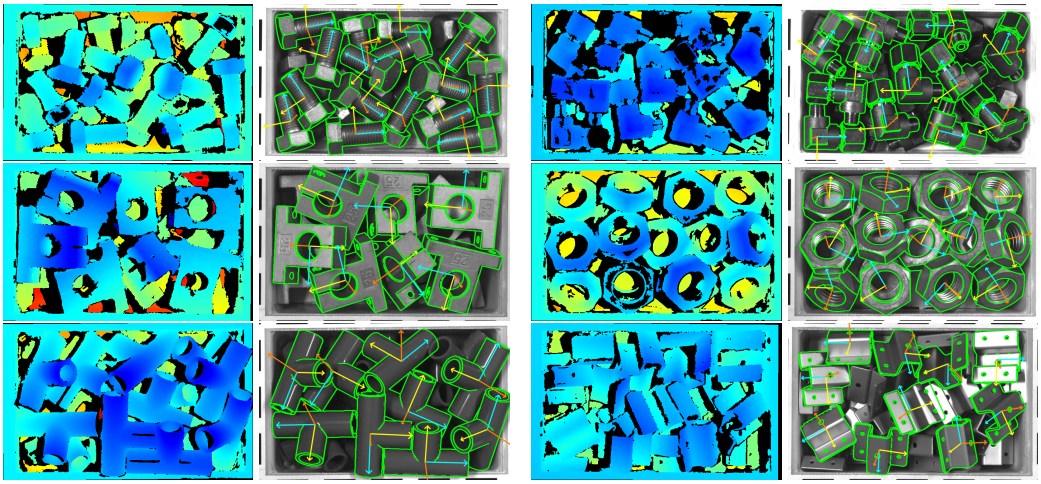


Figure 5.11 Example images of Bin-Picking dataset in Experiment 2. Top row: Bolt and Connector. Middle row: Holder and Nut. Bottom row: Pipe and SheetMetal. The depth and grayscale image are shown for each object. The edges of the objects extracted from 3D CAD data (green lines) and the coordinate axes (three colored arrows) are drawn on the images based on the estimated 6-DoF pose by our proposed method.

Table 5.7 The parameters used in Experiment 2. The number of generated depth images (N) and threshold of gradient orientation histograms (Th_g) and of normal orientation histogram (Th_n) for PCOF-MOD extraction. The intervals, ranges and numbers of templates for rotations (X/Y and Z) and distances to the camera in template generation.

N	Th_g	Th_n	X/Y rot. (deg)	Z rot. (deg)	distance (mm)
1000	100	200	8° in $\pm 180^\circ$ (642)	6° in $\pm 180^\circ$ (60)	70 in $700 - 900$ (4)

of annotated objects were 610 (Bolt), 430 (Connector), 210 (Holder), 412 (Nut), 381 (Pipe) and 388 (SheetMetal). Example images for each object are shown in Figure 5.11.

The existing method [94] which consists of point-pair feature (PPF) and generalized Hough transform was also evaluated on our Bin-Picking dataset. "Surface-based Matching" which implements PPF on the commercial machine vision software "Halcon13" (MvTec Gmbh in Germany) was used for the evaluation. Both of PPF and our method used 3D CAD in Figure 5.12 for training and were ran on same PC as Experiment 1 (Core i7-7700 3.6GHz) using 4 cores. The area for pose estimation were limited to inside the bin (approximately 700×400 pix).

In the training of our proposed method, the PCOF-MOD templates were made on the viewpoints of a sphere (Connector and SheetMetal) or a hemisphere (Bolt,

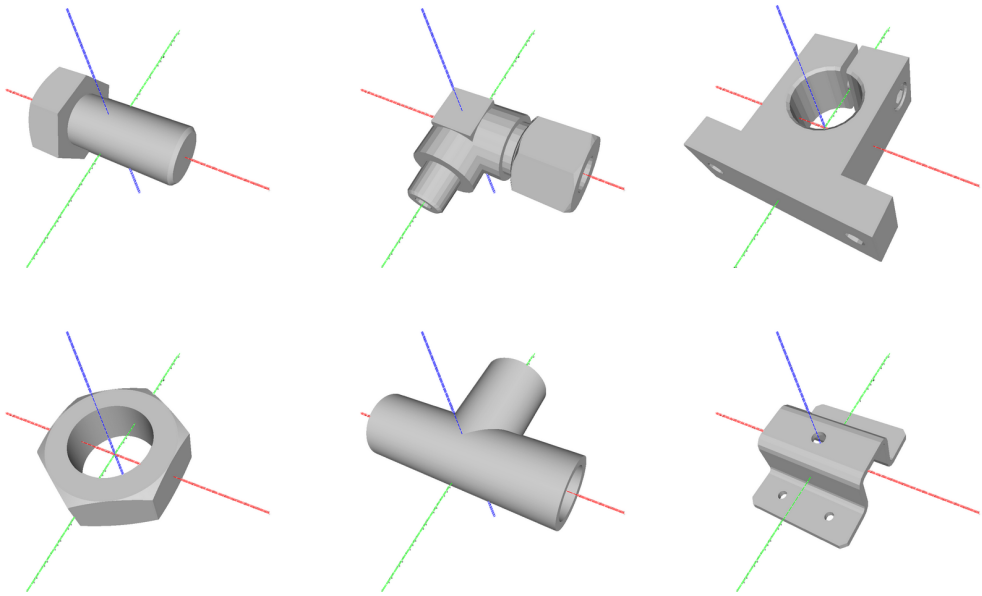


Figure 5.12 3D CAD of target objects in Bin-Picking dataset. Top: Bolt, Connector, Holder. Bottom: Nut, Pipe, SheetMetal. The coordinate axes are drawn on the images (red - X, green - Y, blue - Z.

Holder, Nut and Pipe which are symmetric about X/Y plane), ± 180 degrees around the optical axis and in 750mm - 900mm from the objects. Then the numbers of viewpoints are 12 (6), 42 (21), 162 (81), 642 (321) on sphere (hemisphere) at depth 0, 1, 2, 3. The numbers of rotation angles around optical axis are 8, 15, 30, 60 and distances to the object are 1, 1, 2, 4. The total numbers of templates are 96 (48), 630 (315), 9720 (4860), 154080 (77040). The parameters for our algorithm are summarized in Table 5.7.

Estimation accuracy

On Bin-Picking dataset, the pose estimation was regarded as correct one when the absolute differences of 6 pose parameters (X/Y/Z positions and rotation angles) between the estimated pose and the ground truth were smaller than threshold values were correct. The threshold values of 5 mm in positions and 7.5 degrees in rotations were used. The graphs which represent the recall, precision and F1 score of our proposed method for all objects of Bin-Picking dataset are shown in Figure 5.13, and the highest F1 scores of ours and PPF for all objects and mean values are shown in Table 5.8. The F1 scores of our method are higher than those of PPF on all objects. PPF describes only the surface features of objects and does not explicitly include any

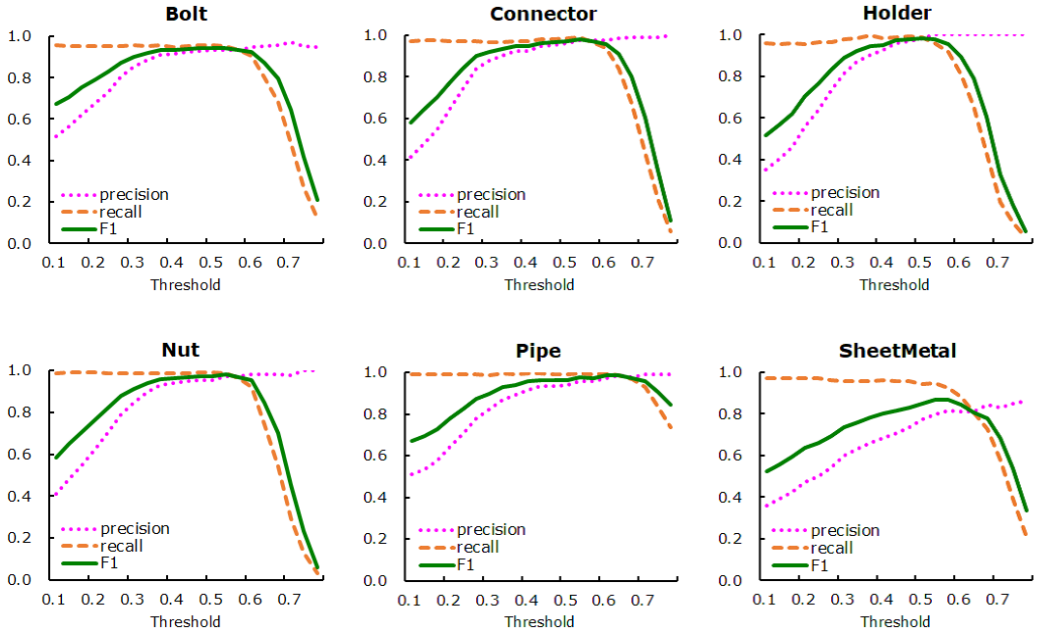


Figure 5.13 Plotting the precision, recall and F1 score for varying thresholds on Bin-Picking dataset in Experiment 2. Top row: Bolt, Connector and Holder. Bottom row: Nut, Pipe and SheetMetal.

contour information. This is why the performance of PPF is degraded when the target objects are industrial parts which commonly consists of simple primitive shapes like planes and cylinders.

Processing time

The processing time of PPF and our proposed algorithm with and without the memory rearrangement are shown in Table 5.9. PPF should be calculated on all pairs of neighboring 3D point clouds. This leads to longer computational time compared

Table 5.8 The highest F1 score on Bin-Picking dataset.

	PPF	**Ours**
Bolt	0.754	**0.944**
Connector	0.758	**0.982**
Holder	0.922	**0.983**
Nut	0.858	**0.980**
Pipe	0.879	**0.986**
SheetMetal	0.700	**0.867**
Mean	0.812	**0.957**

Table 5.9 Processing time (ms) with and without the memory rearrangement (MemRea) on Bin-Picking dataset

	PPF	w/o MemRea	w/ MemRea
Bolt	3334.1	230.6	107.2
Connector	2382.4	209.7	89.2
Holder	1343.1	99.6	42.9
Nut	4387.3	214.3	97.0
Pipe	1501.5	100.1	41.6
SheetMetal	3121.3	183.9	77.9
Mean	2678.3	173.0	76.0

to the pixel-based feature like LINEMOD and ours. Furthermore, our algorithm is accelerated by more than 2 times when using the memory rearrangement and this speed is faster by more than 20 times than PPF.

Though the image resolutions of two types of dataset (bin picking and tabletop) are almost the same, our algorithm takes longer time approximately by two times for pose estimation on Bin-Picking dataset than on the tabletop dataset. This is because larger number of target objects' poses should be estimated on Bin-Picking dataset.

Estimation error

Mean absolute errors of PPF and our proposed algorithm for 3D positions along X/Y/Z axes in mm on Bin-Picking dataset are shown in Table 5.10 and for rotation angles around X/Y/Z axes in degrees are shown in Table 5.11. These errors are averaged only among the successful results using the threshold values of 5 mm for translations and 7.5 degrees for rotations, and the numbers of samples of PPF and PCOF-MOD for averaging are different.

The errors of PPF and our algorithm are almost the same both for translations and rotations. This is because the estimation errors depend on the registration algorithm and both of PPF and ours use ICP algorithm for the registration which is a de-facto standard for 6-DoF pose refinement. The errors in translations are less than 0.5 mm and the errors in rotations are less than 1.0 degrees, those are small enough for robotic grasping.

Table 5.10 Mean absolute errors of estimated positions along X/Y/Z axes in mm for PPF and PCOF-MOD on Bin-Picking dataset.

	PPF			**Ours**		
	tra X	tra Y	tra Z	tra X	tra Y	tra Z
Bolt	0.546	0.441	0.399	0.392	0.312	0.421
Connector	0.512	0.446	0.437	0.443	0.391	0.448
Holder	0.675	0.583	0.401	0.685	0.647	0.406
Nut	0.412	0.278	0.331	0.384	0.274	0.378
Pipe	0.480	0.339	0.332	0.354	0.254	0.344
SheetMetal	0.629	0.540	0.363	0.662	0.615	0.397
Mean	0.542	0.438	0.377	0.487	0.415	0.399

Table 5.11 Mean absolute errors of estimated rotation angles around X/Y/Z axes in degrees for PPF and PCOF-MOD on Bin-Picking dataset.

	PPF			**Ours**		
	rot X	rot Y	rot Z	rot X	rot Y	rot Z
Bolt	2.397	0.710	0.787	1.968	0.687	0.722
Connector	1.444	1.388	1.150	1.372	1.302	0.991
Holder	0.585	0.748	0.897	0.532	0.742	1.029
Nut	0.851	0.755	1.557	0.841	0.765	1.486
Pipe	0.877	0.502	0.704	0.557	0.432	0.564
SheetMetal	0.744	0.690	1.024	0.671	0.571	0.945
Mean	1.150	0.799	1.020	0.990	0.750	0.956

The estimation errors of 6-DoF pose based on the 3D sensor are definitely smaller than those based only on the monocular camera shown in Table 4.6 and Table 4.7. Especially the error in Z translation is greatly reduced from 11.485 mm to 0.399 mm. This is due to the high accuracy in Z translation of the 3D sensor (Ensenso X36), which is approximately 0.2 mm in our experimental setting.

5.3.3 Failure Cases

Typical examples of our failure cases in Experiment 1 and 2 are presented in Figure 5.14. These failures are mainly due to the following reasons.

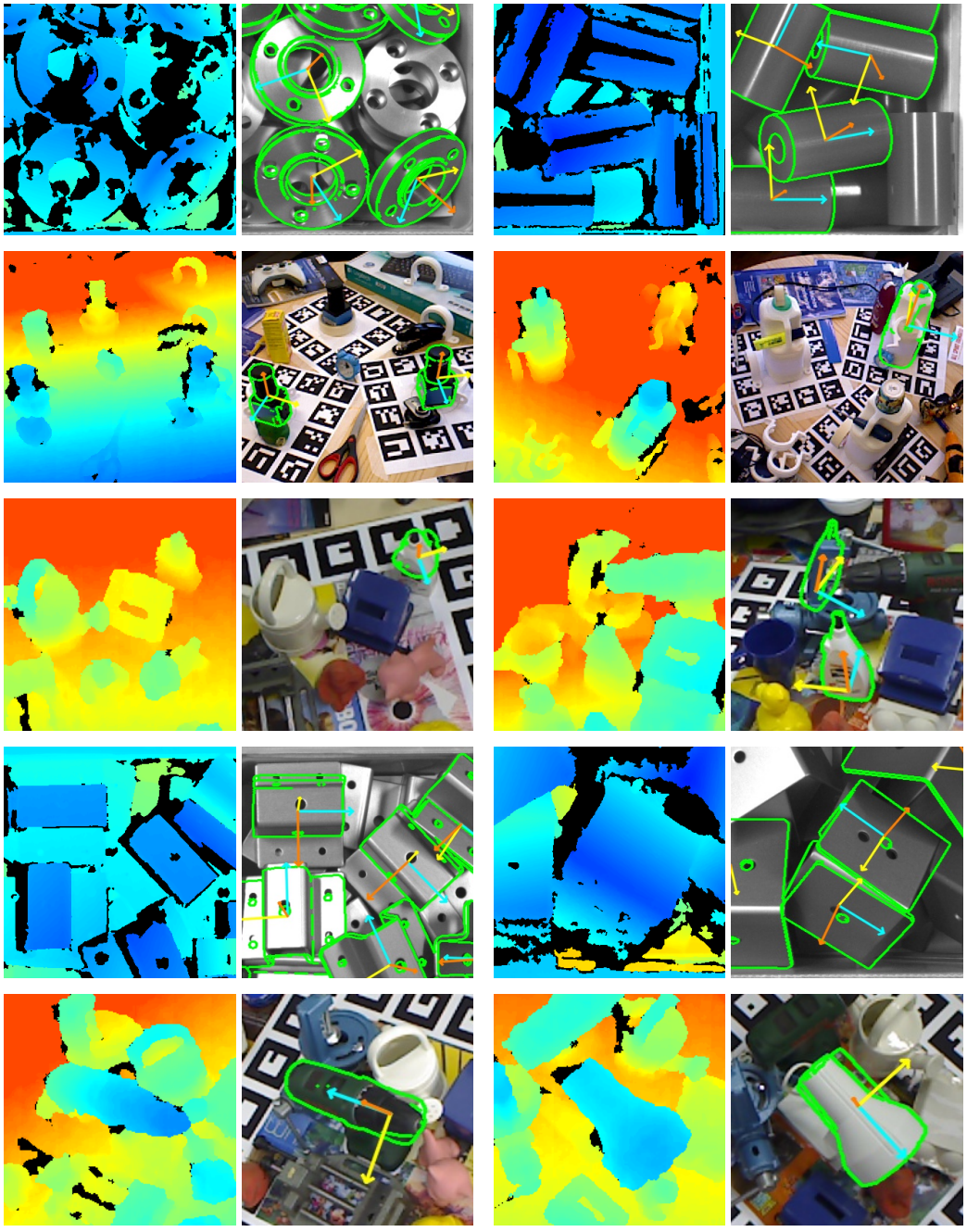


Figure 5.14 Example depth and grayscale images of the failure cases of our proposed method. 1st row: Some of BearingCover and UrethaneTube from Bin-Picking dataset were not recognized due to lack of 3D point clouds. 2nd row: Some of Camera and Milk from ICVL dataset were not recognized due to occlusions. 3rd row: There were false positives of Ape and Driller (ACCV-3D dataset) due to background clutters. 4th row: 3D pose estimations of SheetMetal and L-SheetMetal (Bin-Picking dataset) were failed due to partial correspondences. 5th row: 3D pose estimations of Glue and Lamp (ACCV-3D dataset) were failed due to partial correspondences.

Lack of measured 3D point clouds

The 3D sensor (Ensenso X36) sometimes cannot measure enough 3D point clouds of BearingCover and UrethaneTube from Bin-Picking dataset for 6-DoF pose estimation (1st row of Figure 5.14). The projector of Ensenso projects structured light to the objects and measures the depth based on stereo matching. However, the metalic surface of BearingCover and the translucent material of UrethaneTube make it difficult for the sensor to capture the reflected pattern of light correctly. Some other materials like black rubber and transparent glass possibly cause same problem. This problem might be partially solved by using 3D sensors those are based on other measurement principles such as phase shifting and 3D laser scanning.

Partial occlusion

There are some man-caused partial occlusions in ICVL dataset and these sometimes make it difficult for us to estimate object pose (2nd row of Figure 5.14). The template based pose estimation like ours is more sensitive to occlusions than learning based methods because the model template uniformly samples the features from whole object area. However, partial occlusions might not become big problems for pose estimation in real applications like robotic grasping because grasping occluded objects itself is too hard for a robot arm to execute.

Background clutter

The examples of wrong matches in background are shown in 3rd row of Figure 5.14. Ape and Glue in ACCV-3D dataset are fitted to other objects in background. These failures are prone to occur when the size of target object is small or thin in the captured image. This does not mean that the real size of the object is small/thin and these failures could be overcome by capturing the objects at close range so that the objects look large in the image.

Partial correspondence

The last case is caused by the partial correcpondense to a wrong pose. Two examples from Bin-Picking dataset and another two examples from ACCV-3D dataset are

shown in 4th and 5th row of Figure 5.14. This problem is almost the same as in 6-DoF object pose estimation from a monocular image (Subsection 4.3.4). Therefore, the classifier or regressor which is specifically trained so that it can discriminate small differences of the appearance and shape would be a good solution.

5.4 Conclusion

In this chapter, we introduced the fast and precise 6-DoF pose estimation from a RGB-D image which consists of three main part: PCOF-MOD (Multimodal Perspectively Cumulated Orientation Feature), BPT (Balanced Pose Tree) and the memory rearrangement for coarse-to-fine search. PCOF-MOD is a RGB-D feature which is robust to the change in object 3D pose. BPT is an tree-based data structure of model templates which reduces the search space of 2D position and 3D pose simultaneously. Additionally, two kinds of binary features are rearranged so that nearby features are matched at one time using SIMD instructions. The experimental evaluations were carried out on two different dataset, one is publicly open dataset in tabletop scenes for service robots and another is our own Bin-Picking dataset for industrial robots. The results showed that our proposed method was more accurate and fast than the state-of-the-art methods which include emerging CNN based methods. The errors of our estimated pose were less than 0.5 mm in translations and 1.0 degree in rotations, which are small enough for robot arms to grasp. Our algorithm takes a few minutes for the training and requires no training data other than CAD of target objects, and this is desirable property for real robotic applications where various target objects should be registered on site.

Chapter 6

Conclusion

This chapter provides concluding remarks on the contributions of this thesis and gives an outlook to potential future research and applications.

6.1 Conclusion

This thesis presents fast algorithms for three types of specific object detection and pose estimation, 2D object detection and pose estimation, 3D object detection and pose estimation from a monocular image and 3D object detection and pose estimation from a RGB-D image. All of our proposed algorithms are global descriptor (template matching) based approach which can handle various objects including texture-less and simple-shaped objects, and are robust against background clutters. Furthermore, the model templates are trained based only on a model image for 2D pose estimation and a CAD for 3D pose estimation, and the training takes less than a few minutes. These characteristics of the approach are suitable for real applications such as factory automation and AR/MR. Our proposed methods consist mainly of two components, pose robust features and efficient data structures for template matching.

Firstly, we have proposed image features which explicitly handle certain range of object pose. The way how to extracting the features is that hundreds of randomly transformed images are generated and only dominant orientations are extracted per pixel from the orientation histograms. For 2D object detection and pose estimation, we have applied this to the discretized orientations of image gradients which were robust against background clutters and illumination changes. Furthermore, the discretized orientations are efficiently represented as binary numbers and the

similarity score is quickly computed by logical operations. It has been experimentally shown that our proposed feature which was named Cumulated Orientation Feature (COF) could tolerated the appearance changes caused by 2D object pose changes without degrading the robustness against background clutters. Then COF was extended to handle 3D object pose changes for 3D object detection and pose estimation from a monocular image. The extended feature which was named Perspectively COF (PCOF) are based on the discretized orientations of image gradients and are extracted from a thousand of randomly 3D transformed depth images which were synthesized using 3D CAD of a target object. For 3D object detection and pose estimation from a RGB-D image, Multimodal PCOF (PCOF-MOD) has been introduced where the discretized orientation of surface normals were added to PCOF. It has experimentally been shown that PCOF-MOD was robust enough to be applied to the randomly piled objects (bin picking scene).

Secondly, we have presented the hierarchical pose trees for efficient coarse-to-fine search with image pyramids. The object pose is sampled regularly and the sampling intervals are changed depending on the image resolutions of the image pyramids, for example sparse pose sampling for the higher levels and dense pose sampling for the lower levels. For 2D object detection and pose estimation, the number of pose parameters is two, in-plane rotation angles and object scales, and the numbers of pose are halved at one level higher in the image pyramid. For 3D object detection and pose estimation from a monocular image, the number of pose is much larger than that of 2D pose and we have introduced Hierarchical Pose Tree (HPT) which hierarchically clustered model templates based only on the similarity scores between the templates. This greatly reduces the number of templates which should be matched against an input image and it has experimentally been shown that HPT made 3D pose estimation faster by more than 1,000 times. In 3D object detection and pose estimation from a RGB-D image, the discretized normal orientation feature which is extracted from a depth image is added (PCOF-MOD). This makes the template more discriminative among 3D pose compared to PCOF templates and we have proposed Balance Pose Tree (BPT) where 3D pose was regularly sampled without pose clustering based on the iteratively decomposed polyhedrons. In BPT, the numbers of child nodes of all parent nodes are almost equal and the hierarchical

search tracing along the trees was more efficient than in HPT. Furthermore, the optimum memory rearrangement for coarse-to-fine search has been proposed. At the lower levels of image pyramid, two kinds of orientation features (gradients and normals) within nearby pixels are restructured to be linearly aligned. The model templates are matched against these rearranged feature maps using SIMD instructions and it has experimentally shown that the rearrangement made 3D pose estimation faster by more than two times.

The pipelines of object detection and pose estimation based on the above technical components have also been developed and evaluated. Many experimental evaluations have been done on publicly open dataset and our own practical dataset for three types of specific object detection and pose estimation (2D, 3D from monocular and 3D from RGB-D). The processing times detection accuracies and were compared with the existing methods and it has been shown that our proposed methods were faster and more robust to background clutters. We also have evaluated the precision of the estimated pose and the errors were small enough for robotic grasping and assembly (e.g. less than 1.0 pixel for 2D and 1.0 mm for 3D). The failure cases of our proposed methods for three types of pose estimations have been shown respectively and the reasons and the possible solution were discussed.

6.2 Future Work

In this thesis, fast and robust algorithms for 2D/3D object detection and pose estimation are proposed. Although it is experimentally shown that our proposed algorithms are faster and more robust against background clutters, many more extensions and applications can be considered as outlined in the following.

Scalability

The processing times of our proposed algorithms increase linearly with the number of kinds of objects because the model templates for all objects should be scanned in an image. To alleviate this problem, hashing techniques has been introduced to 2D [14] and 3D [60] object detection and pose estimation. These hashing techniques can

be combined with our orientation features and are helpful for improving the scalability. Another option is to make hierarchical pose tree (HPT in Subsection 4.2.2) based on the templates from different object poses and classes. This clusters the templates whose similarity score is higher than a certain threshold even if the templates are made from different objects. HPT is effective for detection of multiple kinds of objects when the appearance and shape of the objects are similar. In such case, many templates of different objects at coarse layer become similar and they are clustered into fewer number of templates.

Handling intra-class variation

Though many of the target objects are rigid in factory scenes, the difference between reference model and the input image or 3D point cloud sometimes occur by various factors. Moreover in food industry, the algorithms should handle objects whose shapes are different individually, for example fruits and vegetables. To handle these intra-class variation, matching of multiple template is effective. Similar to deformable part model [150], the multiple template model consists of a global root template and several part model templates. Firstly, the root templates are exhaustively scanned in an image at low resolution. Secondly, the part templates are scanned within the limited area based on the relative position of the part to the object center at higher resolution. The part templates might be extracted from regular grids or selected manually so that they are discriminative in the shape and appearance.

Exploiting machine learning

The background is often fixed in factory scenes, for example belt conveyors, containers, pallets and baskets. When the background can be defined using a few images, the background images are utilized to learn discriminative features and classifiers. Like exemplar SVM [102], classifiers can be learned using one positive samples and many negative samples those can be extracted from a few background images. The ad hoc learning for feature selection and weight extraction [120] is more promising for FA and robotic applications. This will be easily combined our proposed features and hierarchical data structures, and the training based on linear SVM does not take

so long time. The machine learning can be used to recognize subtle difference in appearances among different object poses. The different object poses are regarded as different classes to be classified. This should be effective especially for estimation of 3D object pose because there are so many similar appearances in various 3D poses of objects and this induce many false positives as seen in Figure 4.14 and 5.14

Authored Publication

Journal

1. 小西嘉典, 井尻善久, 川出雅人, 橋本学. “累積勾配方向特徴量を用いたテクスチャレス物体検出”. 電子情報通信学会論文誌 D, vol. J99-D, no. 8 (2016), pp. 689–698.

2. 小西嘉典, 半澤雄希, 川出雅人, 橋本学. “階層的姿勢探索木を用いた単眼カメラからの高速 3 次元物体位置姿勢認識”. 電子情報通信学会論文誌 D, vol. J100-D, no. 8 (2017), pp. 711–723.

3. 小西嘉典, 服部宏祐, 橋本学. “平衡姿勢探索木を用いた RGB-D 画像からの高速 3 次元物体位置姿勢認識”. 精密工学会誌 vol. J84, no. 4 (2018), pp. 348–355.

Conference

1. Y. Konishi, Y. Kotake, Y. Ijiri, and M. Kawade. “Fast and precise template matching based on oriented gradients”. In: *Proceedings of European Conference on Computer Vision (ECCV) Demonstration*. 2012, pp. 607–610.

2. 小西嘉典, 井尻善久, 川出雅人. “摂動勾配方向特徴を用いたテクスチャレス物体検出”. 画像センシングシンポジウム (SSII). 2014, IS1-01.

3. Y. Konishi, Y. Ijiri, M. Suwa, and M. Kawade. “Textureless object detection using cumulative orientation feature”. In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 1310–1313.

4. 小西嘉典, 半澤雄希, 川出雅人, 橋本学. “階層的統合モデルを用いた単眼カメラからの高速 3 次元物体位置・姿勢認識”. ビジョン技術の実利用ワークショップ (ViEW). 2015, OS2-H2.

5. Y. Konishi, Y. Hanzawa, M. Kawade, and M. Hashimoto. “Fast 6D pose estimation from a monocular image using hierarchical pose trees”. In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2016, pp. 398–413.

6. 小西嘉典, 服部宏祐, 橋本学. “特徴量再配置による 3 次元物体位置姿勢認識の高速化”. 画像センシングシンポジウム (SSII). 2018, IS3-35.

Preprint

1. Y. Konishi, K. Hattori, and M. Hashimoto. “Real-time 6D object pose estimation on CPU”. arXiv:1811.08588 [cs.CV], 2018.

Bibliography

[1] S. Zafeiriou, C. Zhang, and Z. Zhang. "A survey on face detection in the wild". In: *Computer Vision and Image Understanding* 138.C (2015), pp. 1–24.

[2] S. Yang, P. Luo, C.C. Loy, and X. Tang. "WIDER FACE: A face detection benchmark". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 5525–5533.

[3] P. Dollar, C. Wojek, B. Schiele, and P. Perona. "Pedestrian detection: An evaluation of the state of the art". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.4 (2012), pp. 743–761.

[4] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele. "How far are we from solving pedestrian detection?" In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1259–1267.

[5] S. Ren, K. He, R. Girshick, and J. Sun. "Faster R-CNN: towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems (NIPS) 28*. 2015, pp. 91–99.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, and A.C. Berg. "SSD: Single shot multibox detector". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2016, pp. 21–37.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: unified, real-time object detection". In: *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788.

[8] L. Zheng, Y. Yang, and Q. Tian. "SIFT meets CNN: A decade survey of instance retrieval". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.5 (2018), pp. 1224–1244.

[9] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst. "Shape recognition and pose estimation for mobile augmented reality". In: *IEEE Transactions on Visualization and Computer Graphics* 17.10 (2011), pp. 1369–1379.

[10] M.Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T.K. Marks, and R. Chellappa. "Fast object localization and pose estimation in heavy clutter for robotic bin picking". In: *The International Journal of Robotics Research* 31.8 (2012), pp. 951–973.

[11] W. Ouyang, F. Tombari, S. Mattoccia, L. Di Stefano, and W.K. Cham. "Performance evaluation of full search equivalent pattern matching algorithms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.1 (2012), pp. 127–143.

[12] S. Tanimoto and T. Pavlidis. "A hierarchical data structure for picture processing". In: *Computer Graphics and Image Processing* 4.2 (1975), pp. 104–119.

[13] O. Pele and M. Werman. "Accelerating pattern matching or how much can you slide?" In: *Proceedings of Asian Conference on Computer Vision (ACCV)*. 2007, pp. 435–446.

[14] T. Dean, M.A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. "Fast, accurate detection of 100,000 object classes on a single machine". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 1814–1821.

[15] L. Talker, Y. Moses, and I. Shimshoni. "Efficient sliding window computation for NN-based template matching". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2018, pp. 409–424.

[16] H.G. Barrow, J.M. Tanenbaum, R.C. Bolles, and H.C. Wolf. "Parametric correspondence and chamfer matching: two new techniques for image matching". In: *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*. 1977, pp. 659–663.

[17] G. Borgefors. "Hierarchical chamfer matching: a parametric edge matching algorithm". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.6 (1988), pp. 849–865.

[18] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. "Comparing images using the Hausdorff distance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.9 (1993), pp. 850–863.

[19] W.J. Rucklidge. "Efficiently locating objects using the Hausdorff distance". In: *International Journal of Computer Vision* 24.3 (1997), pp. 251–270.

[20] C.F. Olson and D.P. Huttenlocher. "Automatic target recognition by matching oriented edge pixels". In: *IEEE Transactions on Image Processing* 6.1 (1997), pp. 103–113.

[21] M.Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. "Fast directional chamfer matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 1696–1703.

[22] J. Canny. "A computational approach to edge detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 (1986), pp. 679–698.

[23] R.G. von Gioi, J. Jakubowicz, J.M. Morel, and G. Randall. "LSD: A fast line segment detector with a false detection control". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.4 (2010), pp. 722–732.

[24] C. Steger. "Occlusion, clutter, and illumination invariant object recognition". In: *International Archives of Photogrammetry and Remote Sensing*. Vol. XXXIV, part 3A. 2002, pp. 345–350.

[25] Farhan Ullah and Shun'ichi Kaneko. "Using orientation codes for rotation-invariant template matching". In: *Pattern Recognition* 37.2 (2004), pp. 201–209.

[26] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. "Dominant orientation templates for real-time detection of texture-less objects". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 2257–2264.

[27] E. Hsiao and M. Hebert. "Occlusion reasoning for object detection under arbitrary viewpoint". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.9 (2014), pp. 1803–1815.

[28] S. Korman, M. Milam, and S. Soatto. "OATM: Occlusion aware template matching by consensus set maximization". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2675–2683.

[29] D.M. Tsai and C.H. Chiang. "Rotation-invariant pattern matching using wavelet decomposition". In: *Pattern Recognition Letters* 23.1 (2002), pp. 191–201.

[30] S. Korman, D. Reichman, G. Tsur, and S. Avidan. "FasT-Match: Fast affine template matching". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013, pp. 2331–2338.

[31] S. Belongie, J. Malik, and J. Puzicha. "Shape matching and object recognition using shape contexts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.4 (2002), pp. 509–522.

[32] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, and W.T. Freeman. "Best-buddies similarity for robust template matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 2021–2029.

[33] I. Talmi, R. Mechrez, and L. Zelnik-Manor. "Template matching with deformable diversity similarity". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1311–1319.

[34] W.H. Plantinga and C.R. Dyer. "An algorithm for constructing the aspect graph". In: *Proceedings of Annual Symposium on Foundations of Computer Science*. 1986, pp. 123–131.

[35] K. Ikeuchi. "Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks". In: *International Journal of Computer Vision* 1.2 (1987), pp. 145–165.

[36] P.J. Flynn and A.K. Jain. "CAD-based computer vision: from CAD models to relational graphs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.2 (1991), pp. 114–132.

[37] O. Munkelt. "Aspect-trees: Generation and interpretation". In: *Computer Vision and Image Understanding* 61.3 (1995), pp. 365–386.

[38] S. Lanser, O. Munkelt, and C. Zierl. "Robust video-based object recognition using CAD models". In: *Intelligent Autonomous Systems IAS-4*. 1995, pp. 529–536.

[39] J.H.M. Byne and J.A.D.W. Anderson. "A CAD-based computer vision system". In: *Image and Vision Computing* 16.8 (1998), pp. 533–539.

[40] C.M. Cyr and B.B. Kimia. "A similarity-based aspect-graph approach to 3D object recognition". In: *International Journal of Computer Vision* 57.1 (2004), pp. 5–22.

[41] H. Murase and S. K. Nayar. "Visual learning and recognition of 3-D objects from appearance". In: *International Journal of Computer Vision* 14.1 (1995), pp. 5–24.

[42] R.J. Campbell and P.J. Flynn. "Eigenshapes for 3D object recognition in range data". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 1999, pp. 505–510.

[43] C. von Bank, D.M. Gavrila, and C. Whler. "A visual quality inspection system based on a hierarchical 3D pose estimation algorithm". In: *Pattern Recognition: DAGM Symposium*. 2003, pp. 179–186.

[44] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. "Contour detection and hierarchical image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 898–916.

[45] M Zhu, K.G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis. "Single image 3D object detection and pose estimation for grasping". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 3936–3943.

[46] H. Cai, T. Werner, and J. Matas. "Fast detection of multiple textureless 3-D objects". In: *Proceedings of International Conference on Computer Vision Systems*. 2013, pp. 103–112.

[47] M. Ulrich, C. Wiedemann, and C. Steger. "Combining scale-space and similarity-based aspect graphs for fast 3D object recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.10 (2012), pp. 1902–1914.

[48] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. "Gradient response maps for real-time detection of textureless objects". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (2012), pp. 876–888.

[49] S.M. Yamany and A.A. Farag. "Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.8 (2002), pp. 1105–1120.

[50] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. "Shape distributions". In: *ACM Transactions on Graphics* 21.4 (2002), pp. 807–832.

[51] A. Adan and M. Adan. "A flexible similarity measure for 3D shapes recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.11 (2004), pp. 1507–1520.

[52] A. Adan, P. Merchan, and S. Salamanca. "3D scene retrieval and recognition with Depth Gradient Images". In: *Pattern Recognition Letters* 32.9 (2011), pp. 1337–1353.

[53] R.B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. "Fast 3D recognition and pose using the viewpoint feature histogram". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010, pp. 2155–2162.

[54] K Lai, L Bo, X Ren, and D Fox. "A large-scale hierarchical multi-view RGB-D object dataset". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 1817–1824.

[55] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R.B. Rusu, and G Bradski. "CAD-model recognition and 6DOF pose estimation using 3D cues". In: *Proceedings of IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 585–592.

[56] A. Aldoma, F. Tombari, R.B. Rusu, and M. Vincze. "OUR-CVFH – Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation". In: *Joint DAGM and OAGM Symposium: Pattern Recognition*. 2012, pp. 113–122.

[57] M. Muja, R.B. Rusu, G. Bradski, and D.G. Lowe. "REIN - A fast, robust, scalable REcognition INfrastructure". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 2939–2946.

[58] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 858–865.

[59] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G.R. Bradski, K. Konolige, and N. Navab. "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes". In: *Proceedings of Asian Conference on Computer Vision (ACCV)*. 2012, pp. 548–562.

[60] W. Kehl, F. Tombari, N. Navab, S. Ilic, and V. Lepetit. "Hashmod: A hashing method for scalable 3D object detection". In: *Proceedings of British Machine Vision Conference (BMVC)*. 2015, pp. 36.1–36.12.

[61] T. Hodan, X. Zabulis, M. Lourakis, S. Obdrzalek, and J. Matas. "Detection and fine 3D pose estimation of texture-less objects in RGB-D images". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 4421–4428.

[62] Z. Cao, Y. Sheikh, and N.K. Banerjee. "Real-time scalable 6DOF pose estimation for textureless objects". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 2441–2448.

[63] B. Bratanic, F. Pernus, B. Likar, and D. Tomazevic. "Real-time pose estimation of rigid objects in heavily cluttered environments". In: *Computer Vision and Image Understanding* 141 (2015), pp. 38–51.

[64] V. Lepetit, F. Moreno-Noguer, and P. Fua. "EPnP: An accurate O(n) solution to the PnP problem". In: *International Journal of Computer Vision* 81.2 (2009), pp. 155–166.

[65] D.G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* 60.2 (2004), pp. 91–110.

[66] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-up robust features (SURF)". In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[67] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 2564–2571.

[68] T Tuytelaars and K. Mikolajczyk. "Local invariant feature detectors: A survey". In: *Foundations and Trends in Computer Graphics and Vision* 3.3 (2008), pp. 177–280.

[69] S. Krig. "Interest point detector and feature descriptor survey". In: *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Apress, 2014, pp. 217–282.

[70] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3852–3861.

[71] K. Mikolajczyk, A. Zisserman, and C. Schmid. "Shape recognition with edge-based features". In: *Proceedings of British Machine Vision Conference (BMVC)*. 2003, pp. 779–788.

[72] P. David and D. DeMenthon. "Object recognition in high clutter images using line features". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, pp. 1581–1588.

[73] G. Kim, M. Hebert, and S.K. Park. "Preliminary development of a line feature-based object recognition system for textureless indoor objects". In: *Recent Progress in Robotics: Viable Robotic Service to Human: An Edition of the Selected Papers from the 13th International Conference on Advanced Robotics*. 2008, pp. 255–268.

[74] F. Tombari, A. Franchi, and L.D. Stefano. "BOLD features to detect textureless objects". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 1265–1272.

[75] J. Chan, J.A. Lee, and Q. Kemao. "BORDER: An oriented rectangles approach to texture-less object recognition". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2855–2863.

[76] J. Chan, J.A. Lee, and Q. Kemao. "BIND: Binary integrated net descriptors for texture-less object recognition". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3020–3028.

[77] "Generalizing the Hough transform to detect arbitrary shapes". In: *Pattern Recognition* 13.2 (1981), pp. 111–122.

[78] Y. Lamdan and H.J. Wolfson. "Geometric hashing: A general and efficient model-based recognition scheme". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 1988, pp. 238–249.

[79] A. Collet, D. Berenson, S.S. Srinivasa, and D. Ferguson. "Object recognition and full pose registration from a single image for robotic manipulation". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2009, pp. 48–55.

[80] A. Collet, M. Martinez, and Siddhartha S.S. "The MOPED framework: Object recognition and pose estimation for manipulation". In: *International Journal of Robotics Research* 30.10 (2011), pp. 1284–1306.

[81] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. "Real-time detection and tracking for augmented reality on mobile phones". In: *IEEE Transactions on Visualization and Computer Graphics* 16.3 (2010), pp. 355–368.

[82] S. Hinterstoisser, S. Benhimane, and N. Navab. "N3M: Natural 3D markers for real-time object detection and pose estimation". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2007, pp. 1–7.

[83] C. C. Choi and H.I. H. I. Christensen. "3D textureless object detection and tracking: An edge-based approach". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 3877–3884.

[84] D. Damen, P. Bunnun, A. Calway, and W. Mayol-cuevas. "Real-time learning and detection of 3D texture-less objects: A scalable approach". In: *Proceedings of British Machine Vision Conference (BMVC)*. 2012, pp. 23.1–23.12.

[85] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan. "3D object recognition in cluttered scenes with local surface features: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014), pp. 2270–2287.

[86] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N.M. Kwok. "A comprehensive performance evaluation of 3D local feature descriptors". In: *International Journal of Computer Vision* 116.1 (2016), pp. 66–89.

[87] A.G. Buch, H.G. Petersen, and N. Kruger. "Local shape feature fusion for improved matching, pose estimation and 3D object recognition". In: *SpringerPlus* 5.1 (2016), pp. 297–329.

[88] A.E. Johnson and M. Hebert. "Using spin images for efficient object recognition in cluttered 3D scenes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 433–449.

[89] R.B. Rusu, N. Blodow, and M. Beetz. "Fast point feature histograms (FPFH) for 3D registration". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2009, pp. 3212–3217.

[90] F. Tombari, S. Salti, and L. Di Stefano. "Unique signatures of histograms for local surface description". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2010, pp. 356–369.

[91] C. Zach, A. Penate-Sanchez, and M. Pham. "A dynamic programming approach for fast and robust object pose recognition from range images". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 196–203.

[92] A. Aldoma, F. Tombari, L.D. Stefano, and M. Vincze. "A global hypothesis verification framework for 3D object recognition in clutter". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.7 (2016), pp. 1383–1396.

[93] A.G. Buch, L. Kiforenko, and D. Kraft. "Rotational subgroup voting and pose clustering for robust 3D object recognition". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4137–4145.

[94] B. Drost, M. Ulrich, N. Navab, and S. Ilic. "Model globally, match locally: Efficient and robust 3D object recognition". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2010, pp. 998–1005.

[95] E. Kim and G. Medioni. "3D object recognition in range images using visibility context". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2011, pp. 3800–3807.

[96] C. Choi, Y. Taguchi, O. Tuzel, M. Liu, and S. Ramalingam. "Voting-based pose estimation for robotic assembly using a 3D sensor". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2012, pp. 1724–1731.

[97] C. Choi and H.I. Christensen. "3D pose estimation of daily objects using an RGB-D camera". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 3342–3349.

[98] B. Drost and S. Ilic. "3D object detection and localization using multimodal point pair features". In: *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission (3DIMPVT)*. 2012, pp. 9–16.

[99] T. Birdal and S. Ilic. "Point pair features based object detection and pose estimation revisited". In: *Proceedings of International Conference on 3D Vision (3DV)*. 2015, pp. 527–535.

[100] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige. "Going further with point pair features". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2016, pp. 834–848.

[101] L. Kiforenko, B. Drost, F. Tombari, N. Kruger, and A.G. Buch. "A performance evaluation of point pair features". In: *Computer Vision and Image Understanding* 166 (2018), pp. 66–80.

[102] T. Malisiewicz and A. Efros. "Ensemble of exemplar-SVMs for object detection and beyond". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2011, pp. 89–96.

[103] A. Krizhevsky, I. Sutskever, and G.E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS) 25*. 2012, pp. 1097–1105.

[104] S. Zagoruyko and N. Komodakis. "Learning to compare image patches via convolutional neural networks". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4353–4361.

[105] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. "Discriminative learning of deep convolutional feature point descriptors". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 118–126.

[106] K.M. Yi, E. Trulls, V. Lepetit, and P. Fua. "LIFT: Learned invariant feature transform". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2016, pp. 467–483.

[107] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, and T. Funkhouser. "3DMatch: Learning local geometric descriptors from RGB-D reconstructions". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 199–208.

[108] M. Khoury, Q. Zhou, and V. Koltun. "Learning compact geometric features". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 153–161.

[109] G. Georgakis, S. Karanam, Z. Ziyan Wu, J. Ernst, and J. Kosecka. "End-to-end learning of keypoint detector and descriptor for pose invariant 3D matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 1965–1973.

[110] H. Deng, T. Birdal, and S. Ilic. "PPFNet: Global context aware local features for robust 3D point matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 195–205.

[111] H. Deng, T. Birdal, and S. Ilic. "PPF-FoldNet: Unsupervised learning of rotation invariant 3D local descriptors". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2018, pp. 620–638.

[112] O. Tuzel, M.Y. Liu, Y. Taguchi, and A. Raghunathan. "Learning to rank 3D features". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2014, pp. 520–535.

[113] P. Wohlhart and V. Lepetit. "Learning descriptors for object recognition and 3D pose estimation". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3109–3118.

[114] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T. Kim. "Pose guided RGBD feature learning for 3D object pose estimation". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 3876–3884.

[115] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab. "Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2016, pp. 205–220.

[116] M. Sundermeyer, Z.C. Marton, M. Durner, Brucker M., and R. Triebel. "Implicit 3D orientation learning for 6D object detection from RGB images". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2018, pp. 712–729.

[117] G. Pavlakos, X. Zhou, A. Chan, K.G. Derpanis, and K. Daniilidis. "6-DoF object pose from semantic keypoints". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 2011–2018.

[118] J.J. Rodrigues, Jun-Sik Kim, M. Furukawa, J. Xavier, P. Aguiar, and T. Kanade. "6D pose estimation of textureless shiny objects using random ferns for bin-picking". In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 3334–3341.

[119] A. Tejani, D. Tang, R. Kouskouridas, and T-K. Kim. "Latent-class Hough forests for 3D object detection and pose estimation". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2014, pp. 462–477.

[120] R. Rios-Cabrera and T. Tuytelaars. "Discriminatively trained templates for 3D object detection: A real time scalable approach". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 2048–2055.

[121] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. "Learning 6D object pose estimation using 3D object coordinates". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2014, pp. 536–551.

[122] E. Brachmann, F. Michel, A. Krull, M.Y. Yang, S. Gumhold, and C. Rother. "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3364–3372.

[123] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 1521–1529.

[124] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes". In: *Robotics: Science and Systems*. 2018.

[125] A. Crivellaro, M. Rad, Y Verdie, K.M. Yi, P. Fua, and V. Lepetit. "A novel representation of parts for accurate 3D object detection and tracking in monocular images". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 4391–4399.

[126] M. Rad and V. Lepetit. "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 3828–3836.

[127] B. Tekin, S.N. Sinha, and P. Fua. "Real-time seamless single shot 6D object pose prediction". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 292–301.

[128] A. Krull, E. Brachmann, F. Michel, M.Y. Yang, S. Gumhold, and C. Rother. "Learning analysis-by-synthesis for 6D pose estimation in RGB-D images". In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 954–962.

[129] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother. "Global hypothesis generation for 6D object pose estimation". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 115–124.

[130] A. Krull, E. Brachmann, S. Nowozin, F. Michel, J. Shotton, and C. Rother. "PoseAgent: Budget-constrained 6D object pose estimation via reinforcement learning". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2566–2574.

[131] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. "DeepIM: Deep iterative matching for 6D pose estimation". In: *Proceedings of European Conference on Computer Vision (ECCV)*. 2018, pp. 695–711.

[132] A.C. Berg and J. Malik. "Geometric blur for template matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2002, pp. 607–614.

[133] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005, pp. 886–893.

[134] E. Hsiao and M. Hebert. "Occlusion reasoning for object detection under arbitrary viewpoint". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 3146–3153.

[135] E. Hsiao and M. Hebert. "Gradient networks: Explicit shape matching without extracting edges". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2013, pp. 417–423.

[136] A. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. "From contours to regions: An empirical evaluation". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 2294–2301.

[137] J. Shotton, A. Blake, and R. Cipolla. "Multiscale categorical object recognition using contour fragments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.7 (2008), pp. 1270–1281.

[138] D. Nister and H. Stewenius. "Scalable recognition with a vocabulary tree". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006, pp. 2161–2168.

[139] C. Silpa-Anan and R. Hartley. "Optimised KD-trees for fast image descriptor matching". In: *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008, pp. 1–8.

[140] M. Muja and D.G. Lowe. "Scalable nearest neighbor algorithms for high dimensional data". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014), pp. 2227–2240.

[141] D.M. Gavrila. "A Bayesian, exemplar-based approach to hierarchical shape matching". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.8 (2007), pp. 1408–1421.

[142] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. "Hand pose estimation using hierarchical detection". In: *Proceedings of European Conference on Computer Vision (ECCV) Workshop on HCI*. 2004, pp. 105–116.

[143] Y. Konishi, Y. Ijiri, M. Suwa, and M. Kawade. "Textureless object detection using cumulative orientation feature". In: *Proceedings of IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 1310–1313.

[144] D. Pelleg and A. Moore. "X-means: Extending K-means with efficient estimation of the number of clusters". In: *Proceedings of International Conference on Machine Learning (ICML)*. 2000, pp. 727–734.

[145] S. Garrido-Jurado, R. Munoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marin-Jimenez. "Automatic generation and detection of highly reliable fiducial markers under occlusion". In: *Pattern Recognition* 47.6 (2014), pp. 2280–2292.

[146] S. Hinterstoisser, S. Benhimane, V. Lepetit, P. Fua, and N. Navab. "Simultaneous recognition and homography extraction of local patches with a simple linear classifier". In: *Proceedings of British Machine Vision Conference (BMVC)*. 2008, pp. 10.1–10.10.

[147] P.J. Besl and N.D. McKay. "A method for registration of 3-D shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.

[148] Y. Chen and G Medioni. "Object modeling by registration of multiple range images". In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. 1991, pp. 2724–2729.

[149] S. Rusinkiewicz and M. Levoy. "Efficient variants of the ICP algorithm". In: *Proceedings of International Conference on 3-D Digital Imaging and Modeling (3DIM)*. 2001, pp. 145–152.

[150] P. Felzenszwalb, D. McAllester, and D. Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008, pp. 1–8.