Latent-Class Hough Forests for 3D Object Detection and Pose Estimation of Rigid Objects

Alykhan Tejani

Department of Electronic and Electrical Engineering

Imperial College London

This dissertation is submitted for the degree of

Master of Philosophy

November 2014

To my family for their unwavering support and encouragement over the years

DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other University. This dissertation is the result of my own, except where specifically indicated in the text.

Alykhan Tejani

November 2014

ACKNOWLEDGEMENTS

I would sincerely like to thank my supervisor Dr. Tania Stathaki for her unwavering support and understanding throughout my research. I would also like to thank my co-supervisor Dr. T-K. Kim for the many technical discussion we shared. Finally, I would like to thank my family and my soon to be wife, for their support, understanding and encouragement throughout this journey.

ABSTRACT

In this thesis we propose a novel framework, *Latent-Class Hough Forests*, for the problem of 3D object detection and pose estimation in heavily cluttered and occluded scenes. Firstly, we adapt the state-of-the-art template-based representation, LINEMOD [34, 36], into a scale-invariant patch descriptor and integrate it into a regression forest using a novel template-based split function. In training, rather than explicitly collecting representative negative samples, our method is trained on positive samples only and we treat the class distributions at the leaf nodes as latent variables. During the inference process we iteratively update these distributions, providing accurate estimation of background clutter and foreground occlusions and thus a better detection rate. Furthermore, as a by-product, the latent class distributions can provide accurate occlusion aware segmentation masks, even in the multi-instance scenario. In addition to an existing public dataset, which contains only single-instance sequences with large amounts of clutter, we have collected a new, more challenging, dataset for multiple-instance detection containing heavy 2D and 3D clutter as well as foreground occlusions. We evaluate the Latent-Class Hough Forest on both of these datasets where we outperform state-of-the art methods.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In this thesis, we tackle the problem of object detection and pose estimation of rigid 3D objects in heavily cluttered and occluded scenes using multi-modal input, namely color and depth images (see Fig. 1.1). We are interested in techniques that do not explicitly depend on artificial markers nor textural patterns. Furthermore, we are interested in an approach that does not rely on *a priori* knowledge about the detection environment, in particular background clutter and foreground occluder objects.

1.1 Motivation

Accurate localization and pose estimation of 3D objects in everyday scenes is of great importance to many higher level tasks such as augmented reality and household and industrial robotics to name a few. In particular, algorithms that are robust to the detection environment are of great interest. For example, for real world use, an algorithm must be able to detect coffee cups not only in an office environment, but in a kitchen and a park, in the day time as well as at night. Whilst these tasks are simple for the human visual system, the drastic changes in illumination and the unconstrained variation in the environment make this a highly challenging task for machine vision.

Below, we give an overview of some impacted domains, highlight key difficulties and

(a) (b)

Fig. 1.1 Detection and pose estimation of 3 object instances under occlusion in a heavily cluttered scene. (a) The RGB image with augmented 3D axis from pose estimation. (b) The colourized depth image of the scene. Note, whilst the objects have a textural pattern, this is not used to aid the detection.

further motivate the need for our study.

1.1.1 Augmented Reality

Augmented Reality (AR) refers to the technique of augmenting the real world with computer generated content, this is done by superimposing it such that it appears to be part of reality. AR thus allows the real world (reality) to become interact-able and digitally manipulated, which has been exploited successfully in many applications such as advertising, commerce, education and industrial design (see Fig. 1.2 for examples).

The ability to seamlessly augment reality requires two things (i) a region to augment must be detected in the image (object detection) and (ii) the rotation for the augmented content must be estimated (pose estimation). Traditionally this has been achieved using marker-based techniques, where artificial 2D markers (as shown in Fig. 1.3(a)) are manually applied to the scene or object of interest in large planar areas. These markers use high contrast patterns and can easily and efficiently be detected, even in bad illumination.

(a) (b) (c)

Fig. 1.2 Examples of augmented reality used in practice. (a) Shows an example for advertising (courtesy of Blippar) (b) an example for interior design (courtesy of Ikea) and (c) an example used for education purposes (courtesy of iSkull).

Successful detection of markers allows for the rotation to be estimated, by minimizing a re-projection error (via RANSAC [25] or a similar method) satisfying both requirements. However, the main drawback of marker-based AR is that markers are invasive and must be manually placed on the object of interest. This directly counters our ambition of being able to be used in unconstrained, real-world scenes.

Thus, a second common variant used is known as marker-less AR. Marker-less AR uses 2D textural features which lie on the object, to find 2D-3D correspondences and subsequently estimate pose [3, 18, 31, 46, 52, 59]. (see Fig. 1.3(b) for an example). While this approach has shown great results on highly textured objects [2, 5, 11, 48, 61], in the case of low-textured objects, unseen background clutter can provide significant false positive correspondences. Furthermore, in the case of texture-less objects (see Fig. 1.3(c)) or when key textural regions of low-textured objects are occluded this method can completely fail.

Therefore, we believe that for augmented reality to become a ubiquitous feature in life, there is a definite need for detection and pose estimation algorithms that are able to work on low-textured and even texture-less objects and are robust against unknown or unseen background objects and foreground occluders.

(a) (b) (c)

Fig. 1.3 Examples of augmented reality techniques (courtesy of ARLab). (a) An example of marker-based AR (b) An example of marker-less AR. (c) A texture-less object, for which 2D feature based marker-less AR cannot be used.

1.1.2 Robotics

The domain of robotics has various sub-fields that could vastly benefit from accurate detection and pose estimation of objects. For example, one prominent field is that of industrial part placement, sometimes referred to as bin picking (see Fig. 1.4(a)). In these scenarios, the task is to separate all parts, for which accurate pose estimation is crucial in calculating grasping strategies. Whilst environmental factors such as lighting and presence of unseen objects can be controlled, traditional techniques are still not relevant. For example, artificial marker-based approaches will not work as they are intrusive and would need to be manually placed on each object, which would also need to have large planar surfaces. Furthermore, in these scenarios object occlusion is common and detrimental to marker detection. Conversely, most industrial parts are reflective and texture-less, which also means marker-less approaches based on 2D textural features also fail to work. Thus highlighting the need for approaches geared toward texture-less objects under partial occlusion.

Another notable field is that of household robotics (See Fig 1.4(b)1.4(c)), for which robots must be able to navigate around unconstrained human environments. A key element of this is the ability to recognize objects in previously unseen environments, and accurately estimate their pose so that grasping or avoidance strategies can be calculated. In contrast

to industrial applications, in this scenario the environment is completely unknown and thus it is important to have methods that are highly robust to changing, dynamic environments. Again, marker based detection would be inappropriate for this domain and similarly to the industrial environment many household objects are often texture-less thus limiting the effect of 2D texture-based marker-less approaches.

(a) (b) (c)

Fig. 1.4 Exemplary robotic applications (a) An example of bin picking (courtesy of AH Automation) (b) & (c) Examples of household robotics (courtesy of Technische Universitat Munchen)

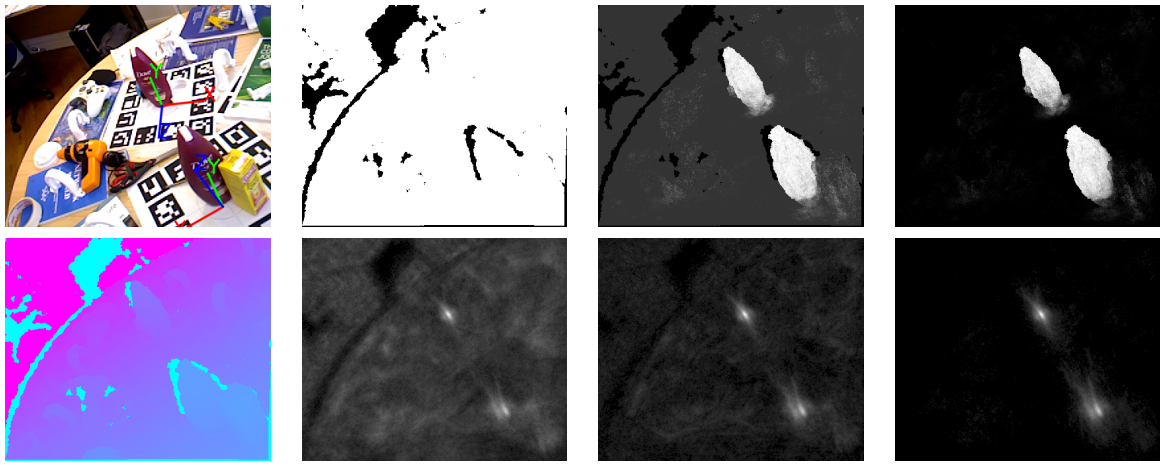
1.2 Challenges

It is clear that marker-less methods are more suited for the task of object detection and pose estimation in unknown environments as they require no manual intervention and placement of markers. Furthermore, we argue that marker-less systems are more generalized as objects in the real world are indeed marker-less. However, as we are interested in methods suited for texture-less objects, traditional marker-less approaches based solely on 2D textural features will not suffice. Nevertheless, The recent introduction of consumer-level depth sensors have allowed for substantial improvement over traditional 2D approaches as finer 3D geometrical

features can be captured. However, there still remain several challenges to address including heavy 2D and 3D clutter, large scale and pose changes due to free-moving cameras as well as partial occlusions of the target object.

In the field of 2D object detection, part or patch-based methods, such as Hough Forests [26], have had much success. In addition to being robust against foreground occlusions, they remove detection disambiguation by clustering votes over many local regions into mutually consistent hypothesis. Furthermore, these methods typically separate foreground regions from background clutter/occluders by a discriminatively learnt model, additionally reducing the rate of false positives. However, for practical use, this requires the collection of a representative negative training set which is also able to generalize to unseen environments. At present there is a huge disparity in the number of RGB-D image datasets vs. 2D image datasets and furthermore, it is not clear how to select such a representative set in order to not create a unintentional bias to particular environments. In fact, many studies have shown that classifiers can show a significant drop in performance when evaluated on images outside of the training domain [20, 58, 79].

State-of-the-art approaches in 3D detection and pose estimation [21, 36] avoid this issue by training just from 3D models of the target object. Whilst previously the requirement of 3D models may have been a disadvantage, with recent innovations in surface reconstruction techniques [54, 82] these can now be obtained easily and efficiently using hand-held RGB-D cameras. Using these models, 3D features, either simple point-pair features [21] or holistic templates [36] are extracted from the model and matched to the scene at test time providing promising results, even for texture-less objects in heavily cluttered environments. While these results are encouraging, these methods have only been evaluated under little or no occlusion and under the assumption of only one instance present per image. However, as these methods have no knowledge of the background distribution in training, heavy background clutter can cause false regions to have significant responses. While this is a more



RGB-D images #iterations=0 #iterations=2 #iterations=10

Fig. 1.5 An illustration of the algorithm used to update the latent class distributions. Columns 2-4 show intermediate results from different number of iterations of the algorithm, where row 1 shows the foreground confidence map, $\mathcal{Z}$, and row 2 shows the resulting Hough voting space. Note the contrast of the vote images have been enhanced for visualization.

prominent issue in the point-to-point methods, as planar regions of target objects are easily matched to background clutter, holistic template matching is by no means immune to this.

1.3 Main Contributions

Motivated by these issues, we present the Latent-Class Hough Forest; a framework for 3D object detection and pose estimation. Unlike the traditional Hough Forest [26], which explicitly exploits classification labels during training, we train only from positive samples and use only the regression term. However, unlike a regression forest [23] we maintain class distributions at leaf nodes. During testing, these distributions are considered as latent variables that are iteratively updated, providing more accurate voting results. Furthermore, as a by-product, our method also produces accurate occlusion-aware figure-ground segmentation masks, which are useful for further post-processing procedures such as efficient occlusion-aware registration [84]. Fig. 1.5 illustrates this iterative procedure, the effect it has on the output voting results and the figure-ground segmentation masks.

Our main contributions can be summarized as follows:

- We propose the Latent-Class Hough Forest, a novel patch-based approach to 3D object detection and pose estimation; It performs one-class learning at the training stage, and iteratively infers latent class distributions at test time.
- We adapt the state-of-the-art 3D holistic template feature, LINEMOD [34, 36], to be a scale invariant patch descriptor and integrate it into the random forest framework via a novel template-based splitting function.
- During the inference stage, we jointly estimate the objects 3D location and pose as well as a pixel wise visibility map, which can be used as an occlusion aware figure-ground segmentation for result refinement.
- We provide a new, more challenging public dataset for *multi-instance* 3D object detection and pose estimation, comprising *near and far range 2D and 3D clutter* as well as *foreground occlusions*

1.4 Outline Of Thesis

In the remainder of this thesis we first discuss related literature in Chapter 2 before introducing our method and providing a quantitative and qualitative evaluation in Chapter 3. Finally, in Sec. 4, we conclude with some final remarks and a discussion of future work.

CHAPTER 2

RELATED WORK

Image features and object representation play a crucial part of any detection algorithm and there are many differing methods used. Traditionally, edge and gradient features have been used to describe local image textures [2, 5, 11, 49, 61] and object-boundaries [17, 35, 66] as they are less sensitive to illumination change than color information alone. However, methods relying solely on 2D image features can be heavily degraded by slight changes in illumination and contrast and additionally, 2D image features frequently fail to capture discriminative information from texture-less objects.

Recently there has been a surge of interest in using depth information for object detection as this cue offers several advantages over traditional 2D features such as invariance to illumination, colour and texture and additionally allows us to exploit local surface properties. Furthermore, with the introduction of consumer-level real-time depth cameras, such as Microsoft's Kinect [51], this a much more appealing and practical option. Similarly to the 2D, RGB modality, there have been many local 3D point descriptors [14, 21, 39, 63, 70, 81] and larger surface descriptors [34, 42, 74] proposed. In fact many representations use a combination of both modalities to achieve superior results [34, 42, 60].

Objects are represented by the extracted image features, either as a collection of many local features in a part-based representation, or holistically as a template-based representation.

Template-based representations model the features jointly, and as such are very information rich. For example, Dalal and Triggs introduced the Histogram of Oriented Gradients [17] which describes the local distribution of image gradients on a grid across the object to form a holistic representation. This representation has further been extended to incorporate depth gradients [42] and normal vectors [74]. However, the representation of features is very important for efficiency and accuracy and often there is a trade-off between descriptiveness and time of computation and comparison. Recently Hinterstoisser *et al.*[35] proposed using only locally dominant orientations instead of local gradient distributions. They encode these points in a bitwise manner similar to [78] and obtain much greater efficiency without compromising accuracy. This method has further been extended to use 3D information such as normal vectors to further improve the descriptiveness of the template [34]. Rios-Cabrera and Tuytelaars [60], further extend this multi-modal representation by using a negative data set to select more prominent points to represent the object, further increasing efficiency and accuracy.

The prevalent approach for detection with template-based representations is the sliding windows approach, in which you centre a bounding box at varying scales and locations and test for object presence. Usually, template-based representations are used in conjunction with a vast background set to train classifiers to label each window as either background or foreground [17, 71, 80]. These approaches quantize templates together and can thus offer greater generalization ability, however they are not able to transfer context to the detection such as 3D pose.

Therefore, another successful approach to detection with template-based representations is by computing a direct template similarity at each location. One prominent method is to use the similarity of image edges to template edges to measure similarity [7, 28, 37, 62], which, while effective, rely on extracting contour edges, via the Canny detector [12] for example, which are sensitive against illumination, motion blur and background noise.

Furthermore, many of these methods do not account for edge direction and [56] show that not using orientation information can lead to a large amount false positives from background clutter.

Steger [69] attempts to remedy this by using image gradients instead of edges, however this method is not robust even against small deformations. However, Hinterstoisser *et al.*[35] successfully adapt this similarity metric to account for local deformations and translations and apply it to texture-less object. They show how to integrate multiple modalities for increased robustness [34] and additionally, show that how, when given a 3D model, a discrete set of templates can be learnt, annotated with their 3D pose and efficiently located using this similarity measure, thus allowing simultaneous detection and pose estimation [36]. For a full review on prior template matching techniques we refer the reader to [10].

Nevertheless, it is important to note that holistic template-based representations are inherently rigid, and, even with techniques such as orientation binning and gradient spreading [17, 34], they are only robust to small deformations; As a consequence, matching usually degrades rapidly in the presence of foreground occlusions [34, 69]. Although there have been approaches that incorporate occlusion reasoning, they either try to learn occlusion patterns from data [27, 41] which can exhibit severe dataset bias, or make strong assumptions about the occluder shape and location [38] which does not generalize well to unconstrained environments.

Alternatively, part-based representations provide a high level of robustness to local deformations and unconstrained foreground occlusions. The simplest form is to assume a set of 2D-3D part or point correspondences and use the individual detection of these to jointly estimate location and pose [3, 18, 31, 46, 52, 59]. However, detection of such points often relies on the presence of 2D texture, which is often not representative of many real world objects.

Rather than model objects as a collection of unrelated features, other part-based repre-

sentations model the relationship between these local features, either explicitly [15, 21, 22, 24, 45] by their spatial relationships, or implicitly [26, 43, 50, 55, 66] by a spatial relationship to an anchor point such as the object centre (not necessarily a feature). In recent studies, these representations have been used in conjunction with a form a generalized Hough voting [4] to perform detection that is robust to intra-class variations, deformations and foreground occlusions. Under this framework, detected local parts can transfer contextual knowledge for object detection and pose estimation. This is done by individual parts voting, in a Hough space, for the object anchor point(s) [16, 26, 43, 55, 67, 71–73] or the object 3D rotation [21, 23, 26] and local maxima are chosen as hypotheses. Furthermore, under these schemes local features are usually quantized together which in turn often generalizes better to slight variations in translation, local shape and viewpoint.

Hough-based methods have worked extremely well in the case where there is no background clutter present [23, 67, 72, 73] and even in the case of previously unseen backgrounds [14, 21, 43]. However, as local features are naturally less discriminative, they can often be strongly matched in background clutter. As a consequence, these methods are often combined with vast amounts of background information within a learning framework [26, 55, 71], in which an explicit background/foreground separation is learnt parametrically causing far less false positives to be generated. However, the efficacy of these approaches are heavily dependent on how representative the background training data is of the "real world", and this benefit does not always transfer across different domains. In fact, it has been shown that significant performance degradation can occur when the negative training set is not representative of target domain [20, 58, 79].

One-class classification is a branch of learning based methods focussed on learning only from the positive/target class. The aim of one-class classification is to discriminate the positive class from unknown outliers by learning only from positive samples. This branch of learning was first coined by by Moya *et al.*[53] but has also be viewed as outlier rejection

or novelty detection depending on the specific application domain.

Classical approaches to one-class classification include those introduced by Tax and Duin [75, 76], which try to learn closed decision boundaries, in the form of a tight hyper-sphere, around the target class in the feature space. However, these methods suffer when the data dimensionality is high or if there are low density regions in the positive data, in which case these regions may be wrongly rejected as outliers. An alternate approach, as described by Schölkopf *et al.*[64, 65] is to use hyper-planes, instead of a hyper-sphere, to separate non-target regions. However, these methods try to make the hyper-plane maximally distant from the origin, in turn using the origin as a priori information about the outlier distribution. Conversely, other approaches attempt to artificially generate a negative, outlier, dataset [19, 32, 77], however the generalization ability can depend heavily on the procedure used for data generation and carry the risk of adding unintentional bias into the negative set. For an in-depth review of once-class classification techniques, we refer the reader to [40].

In this thesis, we tackle the problem of object detection and pose estimation of rigid, textureless, 3D objects in heavily cluttered and occluded scenes, without a priori knowledge about the detection environment, in particular background clutter and foreground occluder objects. In table 2.1 we summarise some of the more related works to this problem highlighting the aspects that each cover. As can be seen, LINEMOD [34] and the method of Drost *et al.*[21] are very closely related to the goals of this thesis. LINEMOD [34] is inherently a holistic template matching system, which while providing a rich description of the object, can suffer from partial occlusions. On the other hand the method of Drost *et al.*performs object matching at a local level, using point-pair features, which can be much more robust to occlusion but can suffer from many false positives as large planar regions of target objects are easily matched to background clutter. In this thesis, we introduce Latent-Class Hough Forests, which aim to leverage the benefits of both approaches by using rich, but local, patch based template descriptors (see Sec. 3.0.1). Furthermore, similar to [26], we

exploit background information to aid classification performance, but unlike [26] we train only from foreground samples and instead infer the foreground-background distributions at test time (see Sec. 3.0.2).

Approach	**3D Pose Estimation**	**Cluttered Environment**	**Occlusion Robustness**	**Requires Knowledge of Background**	**Performance**
LINEMOD [34]	Yes	Yes	No	No	83.0% on dataset of [36]
Drost *et al.*[21]	Yes	Yes	Yes	No	79.3% on dataset of [36]
Rios-Cabrera *et al.*[60]	Yes	Yes	No	Yes	97.2% on dataset of [36]
Fanelli *et al.*[23]	Yes	No	No	Yes	90.4% on ETH Face Pose Dataset [9]
Gall *et al.*[26]	No	Yes	Yes	Yes	98.6% on UIUC-Multi Car Dataset [1]

Table 2.1 A summary of related works, highlighting the aspects they cover and their performance.

CHAPTER 3

METHODOLOGY

Our goal is to achieve accurate 3D object detection and pose estimation via one-class training, whilst being robust to background clutter and foreground occlusions. To this end, we use only synthetic renderings of a 3D model for training. To leverage the inherent robustness to foreground occlusions, we adopt the state-of-the-art patch-based detector, Hough Forests [26], and for the patch representation we use the state-of-the-art 3D template descriptor, LINEMOD [34, 36]. However, combining these components naively does not work for the following reasons: i) The absence of negative training data means that we cannot leverage the classification term of the Hough Forest, thus, relinquishing the ability to filter out false results caused by background clutter. ii) It is not clear how to integrate a template-based feature into the random forest framework; The main issue is that the synthetic training images have empty space in the background whereas the testing patches will not. Thus, doing a naive holistic patch comparison, or the two-dimenson/ two-pixel tests (as used in [23, 67, 73]) can lead to test patches taking the incorrect route at split functions. iii) LINEMOD, in its current form, is not a scale-invariant descriptor; this gives rise to further issues, such as should we train detectors for multiple scales and how finely should we sample these scales in both the training and testing phases.

To address these issues, we propose the Latent-Class Hough Forest (LCHF); an adapta-

tion of the conventional Hough Forest that performs one-class learning at the training stage, but uses a novel, iterative approach to infer latent class distributions at test time. In Sec. 3.0.1 we discuss how to build a LCHF, in particular we discuss how to adapt LINEMOD into a scale-invariant feature and how to integrate it into the random forest framework via a novel template-based split function. Following this, In Sec 3.0.2, we discuss how testing is performed with the LCHF and how we can iteratively update the latent class distributions and use them to refine our results.

It is important to note that the applicability of Latent-Class Hough Forests has a few prerequisites. Namely, the existence of a 3D mesh model of the target object for training and input from an RGB-D camera at test time.

3.0.1 Learning

Latent-Class Hough Forests are an ensemble of randomized binary decision trees trained using the general random forest framework [8]. During training, each tree is built using a random subset of the complete training data. Each intermediate node in the tree is assigned a split function and threshold to optimize a measure of information gain; this test is then used to route incoming samples either left or right. This process is repeated until some stopping criteria is met, where a leaf node containing application-specific contextual information is formed. Each stage in this learning process is highly application dependent and we will discuss each in turn below.

Training Data

In order to capture reasonable viewpoint coverage of the target object, we render synthetic RGB and depth images by placing a virtual camera at each vertex of a subdivided icosahedron of a fixed radius, as described in [33]. A tree is trained from a set of patches, $\{\mathcal{P}_i = (c_i, D_i, \mathcal{T}_i, \theta_i)\}$, sampled from the training images, where $c_i = (x_i, y_i)$ is the cen-

tral pixel, D_i is the raw depth map of the patch, $\mathcal{T}_i$ is the template describing the patch and $\theta_i = (\theta_x, \theta_y, \theta_z, \theta_{ya}, \theta_{pi}, \theta_{ro})$ is the 3D offset from the patch center to the object center and the 3 Euler angles representing the object pose. The patch template is defined as $\mathcal{T}_i = \left(\{\mathcal{O}_i^m\}_{m \in \mathcal{M}}, \Delta_i\right)$, where $\mathcal{O}_i^m$ are the aligned reference patches for each modality, m, which are either the image gradient or normal vector orientations and $\Delta_i = \{(r, m)\}$, where $r = (\lambda \cdot x, \lambda \cdot y)$ is a discrete set of pairs made up of the 2D offsets (x, y) scaled by λ which is equal to the templates depth at the central pixel, and modalities, m, of the template features. The template features are evenly spread across the patch; features capturing the image gradients are taken only from the object contours and features capturing the surface normals are taken from the body of the object, the collection and representation of template features is the same as described in [34, 36].

Split Function

Given a set of patches, $\mathcal{S}$, arriving at a node, a split function, h_i, is created by choosing a random patch, $\mathcal{P}_i$, and evaluating its similarity against all other patches, $\mathcal{P}_j \in \mathcal{S}$. Along with a randomly chosen threshold, τ_i, the incoming patches can be split into two distinct subsets $\mathcal{S}_l = \{\mathcal{P}_j | h_i(\mathcal{P}_j) \leq \tau_i\}$ and $\mathcal{S}_r = \mathcal{S} \setminus \mathcal{S}_l$. The original similarity measure of [34], adapted to work over patches, is formulated as:

$$\varepsilon(\mathcal{P}_i, \mathcal{P}_j) = \sum_{(r,m)}^{\Delta_i} \left(\max_{t \in \mathcal{R}(c_j + r)} f_m\left(\mathcal{O}_i^m(c_i + r), \mathcal{O}_j^m(t)\right) \right) \tag{3.1}$$

where $\mathcal{R}(x)$ defines a small search window centred at location x and $f_m\left(\mathcal{O}_i^m(x), \mathcal{O}_j^m(y)\right)$ computes the dot product between quantized orientations at locations x and y for modality, m. Note, for clarity we keep the max operator and the explicit function, f_m in the formulation, however, we refer the reader to [34] for a discussion on how to compute these with constant time complexity using pre-processing techniques.

As neither the patch description, $\mathcal{P}_i$, nor the similarity measure, ε, account for scale, this

similarity measure will only work if the patches, $\mathcal{P}_i$ and $\mathcal{P}_j$ are of the same scale. To remedy this, inspired by [67], we achieve scale-invariance by using the depth of the patch center to scale the offsets, r. More formally, we define a scale-invariant similarity measure, ε', as:

$$\begin{cases} \varepsilon'\left(\mathcal{P}_i, \mathcal{P}_j\right) & = \sum\limits_{(r,m)}^{\Delta_i} \left(\underset{t \in \mathcal{R}(\varsigma_j(c_j+r))}{max} f_m\left(\mathcal{O}_i^m\left(\varsigma_i\left(c_i+r\right)\right), \mathcal{O}_j^m\left(t\right)\right)\right), \\ \varsigma_x(c_x, r) & = c_x + \frac{r}{D_x(c_x)} \end{cases} \tag{3.2}$$

where $D_x(a)$ is the depth value at location a in patch $\mathcal{P}_x$.

This similarity measure is still not sufficient, as given two patches, both representing the same part of the target object, one synthetically generated and one from a testing image (containing background noise), the functions $f_m\left(\mathcal{O}_i^m\left(\varsigma_i\left(c_i+r\right)\right), \mathcal{O}_{train}^m\left(t\right)\right)$ and $f_m\left(\mathcal{O}_i^m\left(\varsigma_i\left(c_i+r\right)\right) \mathcal{O}_{test}^m\left(t\right)\right)$ will produce significantly different values if any template features, $(r, m) \in \Delta_i$, from the selected template falls in the empty, background, space in the training patch or on to a foreground occluder in the testing patch. This can then cause the two patches to proceed down the tree in different directions, see Fig. 3.1 for an illustration of this issue. To this end, we alter the similarity function is as follows:

$$\begin{cases} \varepsilon''\left(\mathcal{P}_i, \mathcal{P}_j\right) & = \sum\limits_{(r,m)}^{\Delta_i} \left(\underset{t \in \mathcal{R}(\varsigma_j(c_j+r))}{max} \iota(\mathcal{P}_i, \mathcal{P}_j, r) \cdot f_m\left(\mathcal{O}_i^m\left(\varsigma_i\left(c_i+r\right)\right), \mathcal{O}_j^m\left(t\right)\right)\right), \\ \iota\left(\mathcal{P}_i, \mathcal{P}_j, r\right) & = \delta\left(||D_i\left(\varsigma_i\left(c_i+r\right)\right) - D_i\left(c_i\right)| - |D_j\left(\varsigma_j\left(c_j+r\right)\right) - D_j\left(c_j\right)|| < \epsilon\right) \end{cases} \tag{3.3}$$

where $\iota(\mathcal{P}_i, \mathcal{P}_j, r)$ is an indicator function that removes template features that are not spatially consistent with the patch's 3D surface from having an effect on the similarity score. The efficacy of this indicator function is illustrated in Fig 3.1. Finally, we can express the split function of a node as $h_i\left(\mathcal{P}_j\right) = \varepsilon''\left(\mathcal{P}_i, \mathcal{P}_j\right)$.

The effectiveness of a particular splitting function is evaluated by the information gain, however, as no negative data is present at training we cannot use the formulation of the Hough Forest [26]. Instead, we measure only the entropy of the offset and pose regression

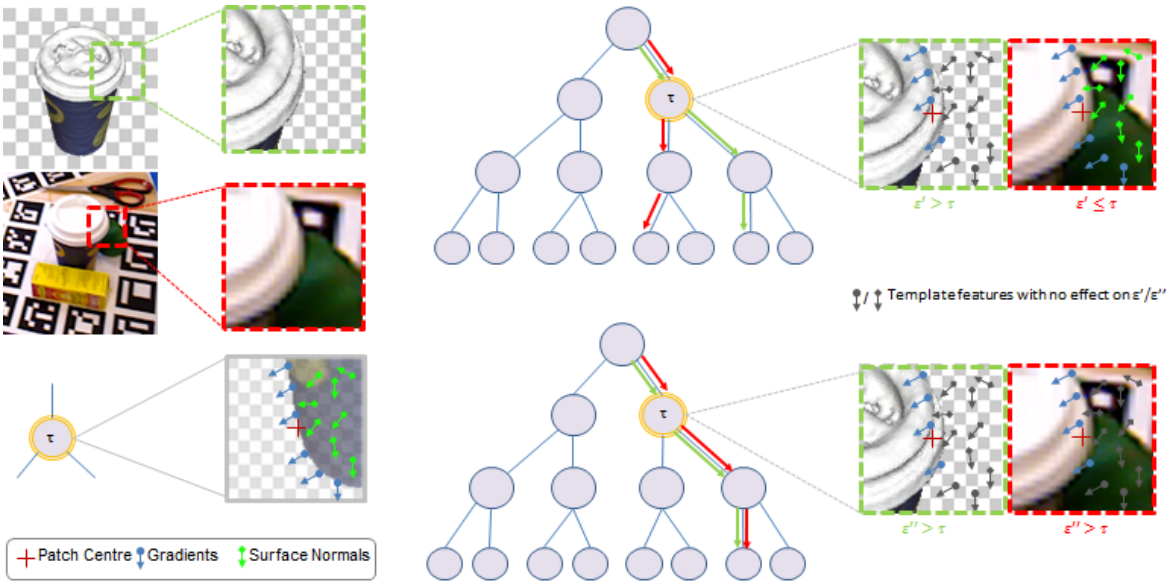


Fig. 3.1 A conceptual view of how our similarity measurement works. Left: rows 1 & 2 show two patches, one from training and one from testing; both are of different scale which is handled by Eq 3.2. Row 3 shows the learnt template, $\mathcal{T}$, at the highlighted node. Right: Shows how without the indicator function (Eq 3.3) the two patches can go down different paths in the tree, leading to wrong results.

by using the trace of the covariance matrix described by Fanelli *et al.*[23]. This process is then repeated multiple times and the split, (h_i, τ_i), producing the highest information gain is selected as the nodes split function.

Constructing Leaf Nodes

The training data is recursively split by this process until the tree has reached a maximum depth or the number of samples arriving at a node fall below a threshold. When this criteria is met a leaf node is formed from the patches reaching it. The leaf node stores votes for both the center position of the object, $(\theta_x, \theta_y, \theta_z)$, and the pose, $(\theta_{ya}, \theta_{pi}, \theta_{ro})$. Following the approach of Girshick *et al.*[29] we only store the modes of the distribution which we find efficiently via the mean shift algorithm. Finally, similar to the Hough Forest [26], we create a class distribution at the leaf, however, as no background information reaches the leaves during training this distribution is initialized to $p_{fg} = 1$ and $p_{bg} = 0$ for the foreground and

background probabilities respectively.

3.0.2 Inference

We want to estimate the probability of the random event, $E(\theta)$, that the target object exists in the scene under the 6 degrees of freedom pose $\theta = (\theta_x, \theta_y, \theta_z, \theta_{ya}, \theta_{pi}, \theta_{ro})$. We can calculate this by aggregating the conditional probabilities $P(E(\theta)|\mathcal{P})$ for each patch, $\mathcal{P}$. As we only model the effect that positive patches have on the pose estimation, the existence of an estimation at θ in the pose space assumes the vote originates from a foreground patch, that is $p_{fg} = 1$, which is assumed for all patches initially. Thus, for a patch $\mathcal{P}$ evaluated on tree $\mathcal{T}$ and reaching leaf node l, we can formalise the conditional probability as:

$$\begin{aligned} p\left(E\left(\theta\right)|\mathcal{P};\mathcal{T}\right) &= p\left(E\left(\theta\right), p_{fg}^{l} = 1|\mathcal{P}\right) \\ &= p\left(E\left(\theta\right)|p_{fg}^{l} = 1, \mathcal{P}\right) \cdot p\left(p_{fg}^{l} = 1|\mathcal{P}\right) \end{aligned} \tag{3.4}$$

where p_{fg}^{l} is the foreground probability at the leaf node, l. Finally, for a forest, $\mathcal{F}$, we simply average the probabilities over all trees:

$$p\left(E\left(\theta\right)|\mathcal{P};\mathcal{F}\right) = \frac{1}{|\mathcal{F}|}\sum_{t}^{|\mathcal{F}|} p\left(E\left(\theta\right)|\mathcal{P};\mathcal{T}_t\right) \tag{3.5}$$

The first factor, $p\left(E\left(\theta\right)|p_{fg} = 1, \mathcal{P}\right)$, can be estimated by passing each patch down the forest and accumulating the votes stored at the leaf, in which votes from multiple trees can be combined in an additive manner, this gives us the same probabilities as in Eq. 3.5 up to a constant factor. The estimation is then deferred to the ability of locating local maxima in this aggregated space which, traditionally, has been done by either exhaustively searching the space combined with non-max suppression or by treating each vote as a point in the space and using mean shift to locate the modes. However, these approaches are usually applied to low dimensional (2D) spaces [26, 44, 57] or in cases where many of the data points

(pixels) have already been removed via some pre-processing [23, 73]. In our case, the pose voting space is 6 dimensional and in the case of evaluating all patches in a VGA image for several trees in the forest, the number of data points becomes very large and this solution is highly inefficient. To this end, we propose a three-stage localization technique; We initially aggregate all votes into a 2D voting space i.e. $p\left(E\left((\theta_x, \theta_y)\right)|\mathcal{P}\right)$ and use this space to locate the hypothesis with non-max suppression. We then further process the votes from patches within the bounding box of these hypothesis to locate modes in the 3D translation space, $(\theta_x, \theta_y, \theta_z)$, and finally use the patches to find the modes in the rotation space, $(\theta_{ya}, \theta_{pi}, \theta_{ro})$, given the estimated translation.

The second factor of Eq (3.4), $p\left(p_{fg} = 1|\mathcal{P}\right)$, is traditionally estimated from the learnt class distribution at the leaf nodes. However, in the LCHF this is a latent distribution and all leaf nodes are in initially set to have $p_{fg} = 1$. Therefore, we propose a method similar to the co-training concept to iteratively update these distributions from the observable unlabelled data in the scene.

Co-training [6] is a technique, that has seen much success in many applications [13, 47, 83], where the main idea is to have two independent classifiers, in which each iteratively predicts labels for the unlabelled data and then uses these labels to update the other classifier. In the seminal work of Blum & Mitchell [6] it was stated that each classifier should be trained from different views/feature representations of the data, but it was later shown that using two classifiers trained originally on the same view will suffice [30], and this is the variant most similar to our method. In our work we make the assumption that as a Latent-Class Hough Forest is a classifier, a subset of Latent-Class Hough Trees can make a sub-forest which is a classifier in its own right. Furthermore as the trees are trained independently, two sub-forests can be viewed as two independent classifiers. Thus, to obtain classifiers for each iteration of the co-training, we randomly partition the Latent-Class Hough Forest, $\mathcal{F}$, into two forest subsets, $\mathcal{F}_1$ & $\mathcal{F}_2$.

Following this, given a forest, $\mathcal{F}$, we select a random subset of the image patches and predict their labels by evaluating Eq (3.4) to obtain an initial object hypotheses set, $\Theta = \{\theta^i\}$. For the N most likely hypotheses, we backproject the contributing votes to their corresponding patches to obtain a consensus patch set, K_i as done in [43]. This patch set is then further reduced to a consensus pixel set, Π, as follows:

$$\begin{cases} \Pi & = \bigcup\limits_{\theta^i \in \Theta} \left(\bigcup\limits_{\mathcal{P}_j \in K_i} g\left(\mathcal{P}_j, \theta^i\right) \right), \\ g\left(\mathcal{P}_j, \theta^i\right) & = \{p_j \in \mathcal{P}_j | d\left(c_j, \theta_i\right) \leq \alpha \varnothing \wedge d(p_j, c_j) \leq \beta \varnothing\} \end{cases} \tag{3.6}$$

where p_j are pixels, d is the euclidean distance function, $\varnothing$ is the diameter of the target objects 3D model and α and β are scaling coefficients. The consensus pixel set contains the pixels from patches that vote for the selected hypotheses and are also spatially consistent with the hypothesis that they vote for.

All pixels in Π are then labelled as foreground pixels and all others as background, thus producing two labelled datasets from the patches extracted around those pixels, $\mathcal{P}^+$ and $\mathcal{P}^-$. These datasets are then passed as input to the second classifier, $\mathcal{F}_j$, where each leaf node, l, accumulates the patches that arrive at it, $\mathcal{P}_l$, and updates the leaf probability distribution as follows:

$$p_{fg}^l = \frac{|\left\{\mathcal{P}_i \mid \mathcal{P}_i \in (\mathcal{P}_l \cap \mathcal{P}^+)\right\}|}{|\mathcal{P}_l|} \tag{3.7}$$

This process is then repeated for a fixed number of iterations. Once finished, the final hypotheses set is produced by passing all patches down the complete forest, $\mathcal{F}$ and evaluating Eq (3.5) using the newly learnt p_{fg}^l. The overall principle of this co-training algorithm is depicted in Algorithm 1 and in Fig. 1.5.

Additionally, as a by-product of this process, we can produce a pixel-wise foreground confidence map, $\mathcal{Z}$, of the input image by labelling each pixel by the average p_{fg}^l (see Fig 1.5). Using the confidence map, $\mathcal{Z}$, and the final set of hypotheses, Θ, we can produce a

Algorithm 1 Update Latent-Class Distributions

Require: An input image, $\mathcal{I}$; A Latent-Class Hough Forest, $\mathcal{F}$

1: **repeat**
2: Randomly draw a subset of trees $\mathcal{F}_i$ from $\mathcal{F}$; $\mathcal{F}_j = \mathcal{F} \setminus \mathcal{F}_i$.
3: Randomly sample a set of patches $\mathbb{P}$ from I.
4: Propagate $\mathbb{P}$ down $\mathcal{F}_i$ collect hypotheses set Θ with Eq (3.5).
5: Backproject top N hypotheses to obtain a consensus set Π (Eq. (3.6)).
6: Partition $\mathcal{P} \in \mathbb{P}$ into positive and negative sets using the consensus set.

$$\mathcal{P}^+ = \{\mathcal{P} | \mathcal{P} \in \Pi\}$$

$$\mathcal{P}^- = \mathbb{P} \setminus \mathcal{P}^+$$

7: Propagate $\mathcal{P}^+$ and $\mathcal{P}^-$ down $\mathcal{F}_j$ and update the leaf node distributions with Eq (3.7).
8: **until** Maximum iteration

final image segmentation mask, $\mathcal{M}$, by

$$\mathcal{M} = \bigcup_{\theta^i \in \Theta} \left(\mathbb{B}\left(\theta^i\right) \cap \mathcal{Z} \right) \tag{3.8}$$

where $\mathbb{B}(\theta)$ is a function that computes the bounding box from the hypothesis θ. This final segmentation, although not currently used, is useful for further refinement of the hypotheses, for example by using it as input for an occlusion-aware ICP alignment.

It is important to note this co-training algorithm does not have an objective function that it is attempting to minimize and instead just runs a fixed number of times. Whilst this has shown to be effective from our experimental evaluation, using a carefully designed objective function would likely improve accuracy. One such objective function could be to measure the error between the RGB-D information and a projection of the target object 3D model under the current estimation, however this is left as a task for future work.

Furthermore, there is no guarantee that the method will converge to a set of stable hypotheses or result in all pixels being labelled as background data. This is because it is heavily reliant on how accurately each sub-forest classifies pixels as this will effect the classification results of other sub-forests. Thus, in this work we use a fixed number of iterations of this

algorithm and a study of the convergence properties of this algorithm is an interesting task left for future work.

3.1 Experiments

We perform experiments on two 3D pose estimation datasets. The first is the publicly available dataset of of Hinterstoisser *et al.*[36], which contains 13 distinct objects each associated with an individual test sequence comprising of over 1,100 images with close and far range 2D and 3D clutter. Each test image is annotated with ground truth position and 3D pose.

For further experimentation, we propose a new dataset consisting of 6 additional 3D objects. We provide a dense 3D reconstruction of each object obtained via a commercially available 3D scanning tool [68]. For each object, similarly to [36], we provide an individual testing sequence containing over 700 images annotated with ground truth position and 3D pose. Testing sequences were obtained by a freely moving handheld RGB-D camera and ground truth was calculated using marker boards and verified manually. The testing images were sampled to produce sequences that are uniformly distributed in the pose space by $[0^\circ - 360^\circ]$, $[-80^\circ - 80^\circ]$ and $[-70^\circ - 70^\circ]$ in the yaw, roll and pitch angles respectively. Unlike the dataset of [36], our testing sequences contain *multiple object instances* and *foreground occlusions* in addition to near and far range 2D and 3D clutter, making it more challenging for the task of 3D object detection and pose estimation. Some example frames from this dataset can be seen in Fig 3.2.

In Sec. 3.1.1 we perform self comparison tests highlighting the benefits of adding scale-invariance to the template similarity measure (Eq. (3.2)) and using co-training to update the latent class distributions (Algorithm 1). Following this, in Sec. 3.1.2 we present a comparison of our method against the state of the art methods, namely LINEMOD [34, 36] and the method of Drost *et al.*[21].

In all experiments, Latent-Class Hough Forests are trained only from synthetically gen-

Fig. 3.2 Example frames from the testing sequence of our new dataset. As can be see, the testing images are taken from different viewpoints and scales and include both near and far range 2D and 3D clutter as well as partial occlusions. From top to bottom: Camera, Coffee Cup, Joystick, Juice Carton, Milk and Shampoo.

erated images from the 3D model of the target object (as described in Sec. 3.0.1) and tested on all images present in the test set in a frame-by-frame manner.

In all tests we use the metric defined in [36] to determine if an estimation is correct. More formally, for a 3D model $\mathcal{M}$, with ground truth rotation $\boldsymbol{R}$ and translation $\boldsymbol{T}$, given an estimated rotation, $\hat{\boldsymbol{R}}$ and translation, $\hat{\boldsymbol{T}}$, the matching score is defined as

$$m = \underset{\boldsymbol{x} \in \mathcal{M}}{avg} ||(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{T}) - (\hat{\boldsymbol{R}}\boldsymbol{x} + \hat{\boldsymbol{T}})|| \tag{3.9}$$

for non-symmetric objects and

$$m = \underset{\boldsymbol{x_1} \in \mathcal{M}}{avg} \min_{\boldsymbol{x_2} \in \mathcal{M}} ||(\boldsymbol{R}\boldsymbol{x_1} + \boldsymbol{T}) - (\hat{\boldsymbol{R}}\boldsymbol{x_2} + \hat{\boldsymbol{T}})|| \tag{3.10}$$

for symmetric objects. An estimation is deemed correct if $m \leq k_m d$, where k_m is a chosen coefficient and d is the diameter of $\mathcal{M}$.

Unlike [36], in which only the top detection from each image is selected, we compute precision-recall curves and present the F1-Score which is the harmonic mean of precision and recall. This is a more accurate form of comparison for generic object detection, as directly comparing detections is inaccurate as some images may be harder than others, which is especially true in the case of occlusion and heavy clutter (as in our new dataset). For example, one image may have a top score of 0.6 for the target object, whereas another may have a top score of 0.9 for a false positive and a score of 0.89 for the target object; by only selecting the top scores, the 0.6 detection is correct and the 0.89 is treated as incorrect. Therefore, similarly to [60], we argue a more meaningful evaluation is to sort all detection scores across all images and calculate the general performance of the detector, which is given by the precision-recall curves.

In all experiments, unless otherwise stated, the parameters for our method are as follows. For each object class we train a Latent-Class Hough Forest comprising of 10 trees with a maximum depth of 25 trained from randomly selected set of training patches. Image patch templates centred at a pixel consist of 20 features in both the color gradient and normal

channel. These features are selected anywhere in a search window centred at the central pixel with a maximum size of, but not further than, $\frac{1}{3}$ of the bounding box size in any direction and are chosen randomly in the same method as described in [34, 36]. For the co-training stage we set the number of iterations empirically as 10 and the number of hypothesis to be backprojected per iteration as $N = 5$.

The main parameters that effect performance are the size of the feature search window, which, if set too small, can fail to capture sufficient detail about the object, but if too large can become susceptible to foreground occlusions. Additionally, the number of hypothesis to be backprojected in the co-training stage must be set greater than the number of instances present in all datasets. Furthermore, the choice of the coefficient, k_m can effect the evaluation performance as it controls the correctness threshold. We set k_m to the value of 0.15 in all experiments which, for example, allows the estimation of an object, with a diameter of 10 cm, to only have a 15 mm error. We find this value allows for a small error tolerance whilst still providing visually correct results.

It is important to note that these parameters were chosen empirically by testing on a small subset of the test set and then fixed and used for the final evaluation. It is of interest to see how the overall performance on these datasets varies with these parameters and is left as exercise for future work.

3.1.1 Self Comparisons

We perform two self comparisons on the dataset of Hinterstoisser *et al.*[36]. Firstly we compare the results of our method with and without updating the latent class distributions. As can be seen in Fig. 3.3 our approach with updating distributions improves the F1-Score by 2.8% on average and up to 8.2% on some objects. The biggest gains are seen in objects which have large amounts of indistinct planar regions, for which background clutter can easily be confused at the patch level. For example, the biggest improvements are seen in the

Camera, Holepuncher and Phone objects which contain large planar regions. Furthermore, in Fig. 3.3 we also compare the results of LINEMOD using holistic templates with the original similarity measure (Eq. (3.1)) and the scale-invariant similarity measure (Eq. (3.2)). As the scale-invariant version is trained using only one scale, the performance is increased 6-fold (623 templates as opposed to 3738). Furthermore, the performance is also increased by 7.2% on average, this is due to the fact that templates are able to be matched at scales not seen in the normal template learning stage.

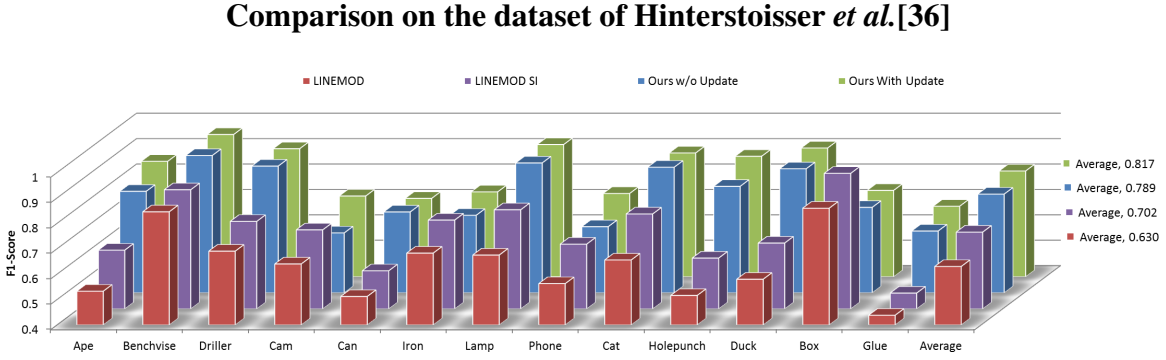


Fig. 3.3 F1-Scores for the 13 objects in the dataset of Hinterstoisser *et al.*[36]. We compare our approach with and without updating the latent class variables (Sec. 3.0.2). We additionally show results of the scale-invariant LINEMOD templates vs. non-scale invariant templates.

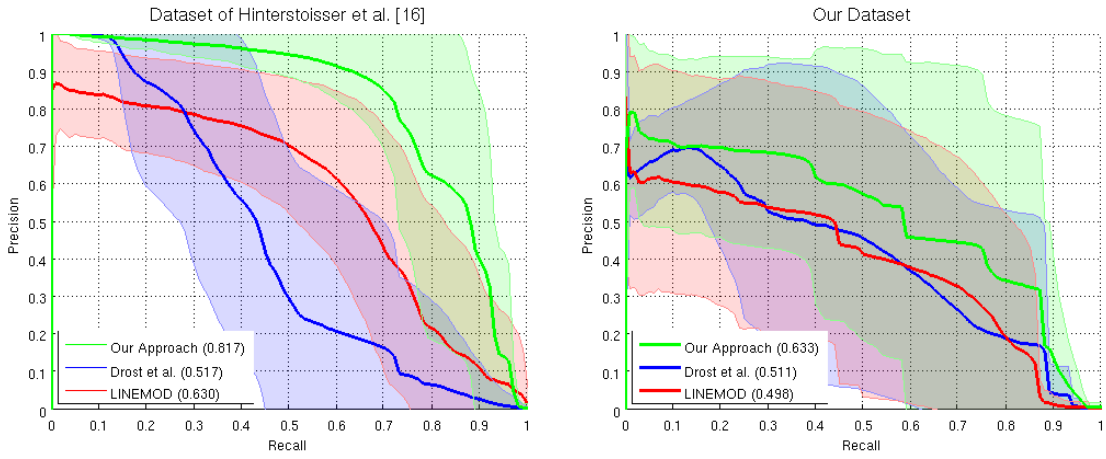
3.1.2 Comparison to State-of-the-arts

We compare our method to two state-of-the-art methods, namely LINEMOD [34, 36] and the method of Drost *et al.*[21]. For LINEMOD, we use our own implementation for based on [34] and using the improvements of template generation from [36], however it is important to note that we can not achieve the same performance as reported by the authors [1]. Additionally, for the method of Drost *et al.*[21], we use a binary version kindly provided by the author and set the parameters to the recommended defaults. Furthermore, for the method

[1] Unfortunately, the authors state that the available, OpenCV version of their algorithm (which ours is based on) will perform much worse, without further elaboration.

of Drost *et al.*[21] we remove points further than 2000mm to reduce the effect of noise, as recommended by the authors. Note, this should not effect accuracy as all target objects are safely within this range.

In Fig. 3.4 we show the average precision-recall curves across all objects in both datasets respectively and in Tables 3.1 and 3.2 we show the F1-Score per object for each dataset. All methods show worse performance on the new dataset, which is to be suspected due to the introduction of occlusions as well as multiple object instances. As can be seen we outperform both state-of-the-arts in both datasets. However, a point to note is that by just picking the top detection from each image, as done in [36], the method of Drost *et al.*and LINEMOD are shown to be almost equal in accuracy (see [36] for this comparison), however, when considering the precision-recall curve, as we do, the method of Drost *et al.*has considerably lower precision values. This is due to the fact that this method does not take object boundaries into consideration, thus large planar regions of the target object can have a large surface overlap in the background clutter causing many false positives in addition to the true positives. Conversely, our method maintains high levels of precision at high recall which is due to the inferred latent class distributions simplifying the Hough space. In Fig. 3.5 we present some qualitative results on both datasets.



(a) (b)

Fig. 3.4 Average Precision-Recall curve over all objects in the dataset of Hinterstoisser *et al.*[36] (a) and our dataset (b). The shaded region represents one standard deviation above and below the precision value at a given recall value.

Approach	**LINEMOD**	**Drost *et al.***	**Our Approach**
Sequence (# images)	**F1-Score**		
Ape(1235)	0.533	0.628	**0.855**
Bench Vise (1214)	0.846	0.237	**0.961**
Driller (1187)	0.691	0.597	**0.905**
Cam (1200)	0.640	0.513	**0.718**
Can (1195)	0.512	0.510	**0.709**
Iron (1151)	0.683	0.405	**0.735**
Lamp (1226)	0.675	0.776	**0.921**
Phone (1224)	0.563	0.471	**0.728**
Cat (1178)	0.656	0.566	**0.888**
Hole Punch (1236)	0.516	0.500	**0.875**
Duck (1253)	0.580	0.313	**0.907**
Box (1252)	**0.860**	0.826	0.740
Glue (1219)	0.438	0.382	**0.678**
Average (15770)	0.630	0.517	**0.817**

Table 3.1 F1-Scores for LINEMOD, the method of Drost *et al.*and our approach for each object class for the dataset of Hinterstoisser *et al.*[36].

Approach	**LINEMOD**	**Drost *et al.***	**Our Approach**
Sequence (# images)	**F1-Score**		
Coffee Cup (708)	0.819	0.867	**0.877**
Shampoo (1058)	0.625	0.651	**0.759**
Joystick (1032)	0.454	0.277	**0.534**
Camera (708)	**0.422**	0.407	0.372
Juice Carton (859)	0.494	0.604	**0.870**
Milk (860)	0.176	0.259	**0.385**
Average (5229)	0.498	0.511	**0.633**

Table 3.2 F1-Scores for LINEMOD, the method of Drost *et al.*and our approach for each object class for our new dataset.

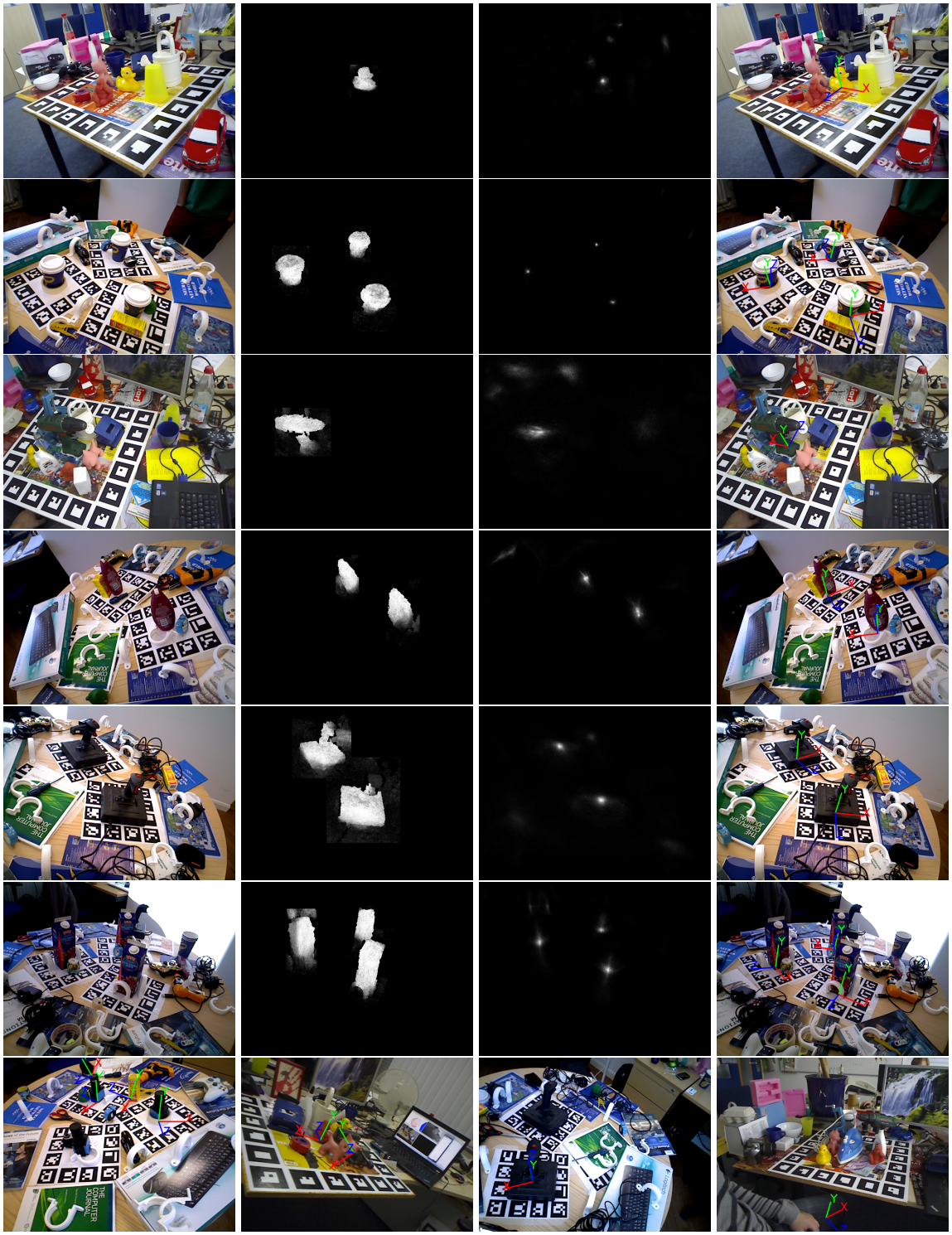


Fig. 3.5 Some qualitative results on both datasets. Rows 1-6 show, from left to right, the original RGB image, the final segmentation mask, the final Hough vote map and the augmented 3D axis of the estimated result. The final row shows some incorrect results.

CHAPTER 4

CONCLUSIONS

In this thesis we have introduced a novel framework for accurate 3D object detection and pose estimation of multiple object instances in cluttered and occluded scenes. We have demonstrated that these challenges can be efficiently met via the adoption of a state-of-the-art template-based representation into a patch-based regression forest. During training we employ a one-class learning scheme, i.e. training with positive samples only rather than involving negative examples. In turn, during inference, we engage the proposed Latent-Class Hough Forest that iteratively produces a more accurate estimation of the clutter/occluder distribution by considering class distribution as latent variables. As a result, apart from accurate detection results we can, further, obtain highly representative occlusion-aware masks facilitating further tasks such as scene layout understanding, occlusion aware ICP or online domain adaption to name a few. Our method is evaluated using both the public dataset of Hinterstoisser *et al.* [36] and our new challenging one containing foreground occlusion and multiple object instances. Experimental evaluation provides evidence of our novel Latent-Class Hough Forest outperforming all baselines highlighting the potential benefits of part-based strategies to address the issues of such a challenging problem.

4.1 Future Work

While the experimental results are a promising step in the direction of accurate 3D object detection and pose estimation in cluttered and occluded scenes, there is still work left to do. For example, the current method does not engage any optimization techniques and as a result is far from real-time, which is a must for real world use. However, one task for future work would be to utilize SSE instructions as described in [34] into the patch-based feature to provide massive speed-ups. Furthermore, whilst the occlusion-aware segmentation masks are currently a by-product of the system, a promising direction for future research would be to use these to refine results of both detection and pose estimation in a feedback manner. Furthermore, the current method works on a frame-by-frame basis and all latent-class variables are re-initialized on each frame. Whilst this is useful if the scene is highly dynamic, in most real world situations objects need to be tracked against static scenes. In such scenarios, rich models of the background can be built over time further improving the detection and pose estimation results as well as run-time speed.

AUTHOURED AND CO-AUTHORED PUBLICATIONS

During the duration of my study I have authored and co-authored the following works:

A. Tejani, D. Tang, R. Kouskouridas, T-K. Kim, **Latent-Class Hough Forests for 3D Object Detection and Pose Estimation**, *Proc. of European Conference on Computer Vision (ECCV)*, Zurich, Switzerland, 2014

D. Tang, H.J. Chang, A. Tejani, T-K. Kim, **Latent Regression Forest: Structural Estimation of 3D Articulated Hand Posture**, *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Columbus, Ohio, USA, 2014

REFERENCES

[1] Agarwal, S., Awan, A., and Roth, D. (2004). Learning to detect objects in images via a sparse, part-based representation. *PAMI*.

[2] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *CVPR*.

[3] Ansar, A. and Daniilidis, K. (2003). Linear pose estimation from points or lines. *TPAMI*.

[4] Ballard, D. H. (1981). Generalizing the hough transform to detect arbitrary shapes. *PR*.

[5] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *ECCV*.

[6] Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT*. ACM.

[7] Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *TPAMI*.

[8] Breiman, L. (2001). Random forests. *ML*.

[9] Breitenstein, M. D., Kuettel, D., Weise, T., Van Gool, L., and Pfister, H. (2008). Real-time face pose estimation from single range images. In *CVPR*.

[10] Brunelli, R. (2009). *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Online Library.

[11] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *ECCV*.

[12] Canny, J. (1986). A computational approach to edge detection. *TPAMI*.

[13] Chan, J., Koprinska, I., and Poon, J. (2004). Co-training with a single natural feature set applied to email classification. In *WIC*.

[14] Choi, C. and Christensen, H. I. (2012). 3d pose estimation of daily objects using an rgb-d camera. In *IROS*.

[15] Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In *CVPR*.

[16] Criminisi, A., Shotton, J., Robertson, D., and Konukoglu, E. (2011). Regression forests for efficient anatomy detection and localization in CT studies. In *MICCAI*.

[17] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

[18] Dementhon, D. F. and Davis, L. S. (1995). Model-based object pose in 25 lines of code. *IJCV*.

[19] Désir, C., Bernard, S., Petitjean, C., and Heutte, L. (2012). A random forest based approach for one class classification in medical imaging. In *MLMI*.

[20] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2009). Pedestrian detection: A benchmark. In *CVPR*.

[21] Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *CVPR*.

[22] Elidan, G., Heitz, G., and Koller, D. (2006). Learning object shape: From drawings to images. In *CVPR*.

[23] Fanelli, G., Gall, J., and Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *CVPR*.

[24] Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *IJCV*.

[25] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*.

[26] Gall, J., Yao, A., Razavi, N., Van Gool, L., and Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *TPAMI*.

[27] Gao, T., Packer, B., and Koller, D. (2011). A segmentation-aware object detection model with occlusion handling. In *CVPR*.

[28] Gavrila, D. M. and Philomin, V. (1999). Real-time object detection for "smart" vehicles. In *ICCV*.

[29] Girshick, R., Shotton, J., Kohli, P., Criminisi, A., and Fitzgibbon, A. (2011). Efficient regression of general-activity human poses from depth images. In *ICCV*.

[30] Goldman, S. and Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. In *ICML*.

[31] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.

[32] Hempstalk, K., Frank, E., and Witten, I. H. (2008). One-class classification by combining density and class probability estimation. In *ECML PKDD*.

[33] Hinterstoisser, S., Benhimane, S., , Lepetit, V., and Navab, N. (2008). Simultaneous recognition and homography extraction of local patches with a simple linear classifier. In *BMVC*.

[34] Hinterstoisser, S., Holzer, S., Cagniart, C., Ilic, S., Konolige, K., Navab, N., and Lepetit, V. (2011). Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*.

[35] Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., and Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects. In *CVPR*.

[36] Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., and Navab, N. (2012). Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*.

[37] Holzer, S., Hinterstoisser, S., Ilic, S., and Navab, N. (2009). Distance transform templates for object detection and pose estimation. In *CVPR*.

[38] Hsiao, E. and Hebert, M. (2012). Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*.

[39] Johnson, A. E. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes. *TPAMI*.

[40] Khan, S. S. and Madden, M. G. (2013). One-class classification: Taxonomy of study and review of techniques. *KER*.

[41] Kwak, S., Nam, W., Han, B., and Han, J. H. (2011). Learning occlusion with likelihoods for visual tracking. In *ICCV*.

[42] Lai, K., Bo, L., Ren, X., and Fox, D. (2012). Detection-based object labeling in 3d scenes. In *ICRA*.

[43] Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV*.

[44] Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *IJCV*.

[45] Leordeanu, M., Hebert, M., and Sukthankar, R. (2007). Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR*.

[46] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *IJCV*.

[47] Liu, R., Cheng, J., and Lu, H. (2009). A robust boosting tracker with minimum error bound in a co-training framework. In *ICCV*.

[48] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *ICCV*.

[49] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*.

[50] Maji, S. and Malik, J. (2009). Object detection using a max-margin hough transform. In *CVPR*.

[51] Microsoft Corp. (2011). Kinect.

[52] Moreno-Noguer, F., Lepetit, V., and Fua, P. (2008). Pose priors for simultaneously solving alignment and correspondence. In *ECCV*.

[53] Moya, M., Koch, M., and Hostetler, L. (1993). One-class classifier networks for target recognition applications. In *INNS*.

[54] Newcombe, R. A., Davison, A. J., Izadi, S., Kohli, P., Hilliges, O., Shotton, J., Molyneaux, D., Hodges, S., Kim, D., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*.

[55] Okada, R. (2009). Discriminative generalized hough transform for object dectection. In *ICCV*.

[56] Olson, C. F. and Huttenlocher, D. P. (1997). Automatic target recognition by matching oriented edge pixels. *IP*.

[57] Opelt, A., Pinz, A., and Zisserman, A. (2008). Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*.

[58] Perronnin, F., Sánchez, J., and Liu, Y. (2010). Large-scale image categorization with explicit data embedding. In *CVPR*.

[59] Quan, L. and Lan, Z. (1999). Linear n-point camera pose determination. *TPAMI*.

[60] Rios-Cabrera, R. and Tuytelaars, T. (2013). Discriminatively trained templates for 3d object detection: A real time scalable approach. In *ICCV*.

[61] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. In *ICCV*.

[62] Rucklidge, W. J. (1997). Efficiently locating objects using the hausdorff distance. *IJCV*.

[63] Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. In *ICRA*.

[64] Schölkopf, B., Williamson, R., Smola, A., and Shawe-Taylor, J. (1999a). Sv estimation of a distribution's support. *NIPS*.

[65] Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., and Platt, J. C. (1999b). Support vector method for novelty detection. In *NIPS*.

[66] Shotton, J. (2007). Contour and texture for visual recognition of object categories. *Doctoral of Philosphy, Queen's College, University of Cambridge, Cambridge*.

[67] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *ACM*.

[68] Skanect (2014). skanect.manctl.com/.

[69] Steger, C. (2001). Similarity measures for occlusion, clutter, and illumination invariant object recognition. In *PR*.

[70] Stein, F. and Medioni, G. (1992). Structural indexing: Efficient 3-d object recognition. *TPAMI*.

[71] Tang, D., Liu, Y., and Kim, T.-K. (2012a). Fast pedestrian detection by cascaded random forest with dominant orientation templates. In *BMVC*.

[72] Tang, D., Tejani, A., Chang, H. J., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*.

[73] Tang, D., Yu, T.-H., and Kim, T.-K. (2013). Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*.

[74] Tang, S., Wang, X., Lv, X., Han, T. X., Keller, J., He, Z., Skubic, M., and Lao, S. (2012b). Histogram of oriented normal vectors for object recognition with a depth sensor. In *ACCV*.

[75] Tax, D. M. and Duin, R. P. (1999a). Data domain description using support vectors. In *ESANN*.

[76] Tax, D. M. and Duin, R. P. (1999b). Support vector domain description. *PR Letters*.

[77] Tax, D. M. and Duin, R. P. (2002). Uniform object generation for optimizing one-class classifiers. *JMLR*.

[78] Taylor, S. and Drummond, T. (2009). Multiple target localisation at over 100 fps. In *BMVC*.

[79] Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR*.

[80] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*.

[81] Wahl, E., Hillenbrand, U., and Hirzinger, G. (2003). Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification. In *3DIM*.

[82] Weise, T., Wismer, T., Leibe, B., and Van Gool, L. (2009). In-hand scanning with online loop closure. In *ICCV Workshops*.

[83] Yu, S., Krishnapuram, B., Rosales, R., Steck, H., and Rao, R. B. (2007). Bayesian co-training. In *NIPS*.

[84] Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *IJCV*.