A Dynamic Programming Approach for Fast and Robust Object Pose Recognition from Range Images

Christopher Zach
Toshiba Research Europe
Cambridge, UK
christopher.m.zach@gmail.com

Adrian Penate-Sanchez
CSIC-UPC
Barcelona, Spain
apenate@iri.upc.edu

Minh-Tri Pham
Toshiba Research Europe
Cambridge, UK
mtpham@crl.toshiba.co.uk

Abstract

Joint object recognition and pose estimation solely from range images is an important task e.g. in robotics applications and in automated manufacturing environments. The lack of color information and limitations of current commodity depth sensors make this task a challenging computer vision problem, and a standard random sampling based approach is prohibitively time-consuming. We propose to address this difficult problem by generating promising inlier sets for pose estimation by early rejection of clear outliers with the help of local belief propagation (or dynamic programming). By exploiting data-parallelism our method is fast, and we also do not rely on a computationally expensive training phase. We demonstrate state-of-the art performance on a standard dataset and illustrate our approach on challenging real sequences.

1. Introduction

Since the emergence of commodity depth sensors in the past few years, recognizing objects and estimating their pose using such depth sensors is an active research topic. Several approaches demonstrate that the results for this task can be improved over methods using only color images by combining RGB and depth features (e.g. [8, 17, 21]), but in many situations color cues are either not available, not informative or unreliable. In particular, in the context of automated manufacturing and mechanical assembling, objects may need to be recognized and their pose estimated from depth data only.

In contrast to color images, depth maps are usually far less discriminative in their appearance. While a good statistical model for color images is still an open research topic, a sensible and simple prior for depth images is given by a piecewise smooth regularizer. Consequently, we do not rely on any interest point detection in depth images and evaluate features densely (or quasi-densely by subsampling) in the query image. Further, real depth sensors exhibit several shortcomings at depth discontinuities, such as half-occlusions and foreground fattening occurring with triangulation-based sensors (passive stereo or Kinect-type active stereo), and mixed pixels with time-of-flight sensors. Overall, many depth sensing technologies report reliable and accurate depth values only in smooth regions of the true scene geometry. Beside that, the piecewise smooth appearance of range images also implies that extracting a full 3D local coordinate frame is not reliable, but at least estimating surface normals is rather stable. Thus, feature extraction can be easily made invariant with respect to two degrees of freedom (i.e. the surface normal) but not reliably invariant with respect to the remaining 2D rotation in the local tangent plane. We also believe that for the same reason predicting poses directly based on feature correspondences leads to large uncertainties in the estimates, and therefore we follow [20, 3] in predicting "object coordinates" (i.e. 3D vertices on the object of interest) and computing more certain and accurate poses from multiple correspondences.

Finally, objects of interest can be occluded and only be partially visible. A sensible principle to add robustness with respect to occlusions is to employ a compositional method, i.e. to detect the object and estimating its pose by detecting and aligning smaller parts. Due to the locally ambiguous appearance of depth images, we expect a much higher false-positive rate than with color images when matching features extracted in the query images with the ones in the training database, and it will be essential to maintain several predictions of object coordinates per pixel to address the amount of false positive matches. In summary, object detection solely from depth data is facing the following challenges: (i) few salient regions in range images, (ii) unreliable depth discontinuities, and (iii) uninformative features and descriptors.

Since depth cameras report 3D geometry, and our method is based on predicting 3D object coordinates for pixels in the range image, we are able to assess the internal consistency of putative object coordinates by comparing the distance between two observed 3D points (back-projected from the depth map) and the one between predicted object coordinates. Grossly deviating distances indicate that at least one of the predicted object coordinates is an outlier. Thus, one can easily avoid sampling and evaluating pose hypotheses from outlier-contaminated minimal sample sets by scoring this (pairwise) consistency between predictions and observed data.

If one interprets the object coordinate hypotheses per pixel as unknown (or latent) states, then the pairwise consistency of predicted object coordinates plays the role of pairwise potentials in a graphical model. Hence, it is natural to build on inference in graphical models in this setting in order to rank sets of putative object coordinates by computing respective (min-)marginals. In contrast to the standard use of graphical models, which usually defines a random field over the entire domain (i.e. image), we utilize many but extremely simple and local graphical models whose underlying graph has exactly the size of the required minimal sample set.

Robust geometric estimation is typically addressed by data-driven random sampling in computer vision. A standard RANSAC-type approach for rigid object pose estimation would randomly draw three object coordinate hypotheses (not necessarily using a uniform distribution) and evaluate the induced pose with respect to the given data. On a high level view RANSAC generates a large number of pose hypotheses and subsequently ranks these. We reverse the direction of computation: our method considers a large number of overlapping minimal sample sets and removes the ones clearly contaminated with outliers by utilizing the consistency criterion. Since the minimal sets are overlapping, applying the consistency criterion to a pair of putative correspondences is able to discard several minimal sample sets at once. We believe that our approach is an elegant solution to generate promising sample sets for robust (pose) estimation exhibiting very few inlier correspondences.

2. Related work

Object detection from 3D inputs has been widely researched during the past decade. Initially many solutions focused on trying to solve object detection from laser scans or even from synthetically generated meshes [10, 13, 4]. However, with the popularization of RGB-D sensors since 2010 there has been an increasing demand of algorithms [8, 17, 21, 3] that operate at interactive frame rates and that are able to cope with inputs that are less reliable than laser scans. Most of the latter algorithms rely heavily on RGB data to perform detection, which prohibits the application of these methods on 3D only inputs. Several approaches [8, 17, 21] use a global description of the object (using RGB edges and depth normals), hence these methods have difficulties in handling occlusions. Brachmann et al. [3] on the other hand compute features densely for each pixel, and subsequently apply a regression forest followed by pose scoring to determine detections. Because it uses a dense description of local features it is able to address occlusions. The biggest advantage of RGB-D algorithms over methods that rely solely on 3D or depth data is their capability to deliver up to real-time performance. Further, these methods are able to cope with noisier data returned by commodity depth sensors.

Methods that utilize only 3D data as input can be based on either global or local object representations. Several proposed methods based on a global object representation employ the Hough transform [11, 16, 23]. These approaches create a set of features that are accumulated in a Hough voting space and then select the pose which gathered the largest number of votes. Like in the RGB-D case, global descriptions suffer again when strong occlusions are present. Several local descriptor-based approaches find salient points in the point cloud and then obtain invariant descriptions of the regions around them [19, 10, 1, 18]. The main problem with this approaches is that 3D information, in contrast to RGB, is usually quite uninformative (many flat surfaces or similar curves) and one cannot find a sufficient number of reliable features in many situations.

Spin images [10] are among the successful local descriptors to recognize 3D shapes. Applied on 3D shapes the spin image is a revolution histogram describing the local surface, and (due to alignment with the local surface normal) it is invariant to 1D rotations in the tangent plane. Mian et al. [13] also use the normal to obtain an invariant descriptor: they fix the local coordinate frame by using two points on the model (in additional to the normal), and fill an occupancy grid given the local coordinate frame. In this work it was also shown that the use of occupancy grids is more discriminative than the use of spin images.

Similar to [13], Drost et al. [4] fix the local coordinate frame of the shape descriptor by choosing two vertices, but their descriptor is based on the distance and the geometric relation of the normals at the chosen surface points instead of using an occupancy grid. This descriptor design makes Drost's features less discriminative than Mian's but also faster to compute. Although Drost's features are less informative, by using all possible combinations of two points in the model, given an initial point, he was able to construct an overall more informative description. One of the most important contributions of this work is that it cannot be classified as either a local or a global description but rather as both; by taking pairs of points, Drost is able to obtain invariance to occlusion like other local approaches and at the same time by creating a feature using every point in the model it makes the approach robust to non informative objects. The true potential of Drost's feature is shown in [22], by applying several learning techniques they are able to re-

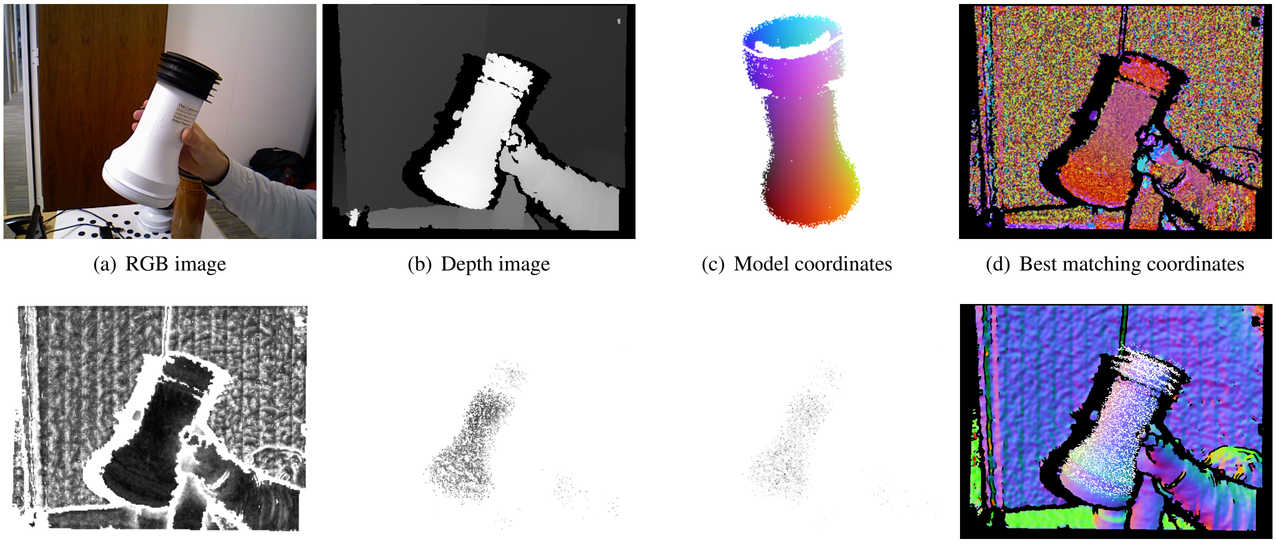
(a) RGB image
(b) Depth image
(c) Model coordinates
(d) Best matching coordinates
(e) Feature (Hamming) distance
(f) Self-consistency
(g) Pose score
(h) Superimposed model

Figure 1. Method overview: (a) input RGB image (for illustration purpose only); (b) input depth image; (c) view on the trained CAD model with color coded object coordinates; (d) best matching object coordinates for the input to illustrate the level of false positives; (e) the corresponding minimal feature distances, which also serve as unary potentials in Eq. 3; (f) the smallest min-marginals Eq. 5 per pixel; (g) the geometric pose scores (Eq. 9) after pose refinement; and (h) points of the model superimposed according to the best pose estimate.

duce the number of features required obtaining a reduced more informative set of pair features. When compared to all previous approaches they clearly outperform all of them in both computation time and accuracy.

Although these techniques using only 3D data as input obtain very good results, they are designed to work with relatively clean data such as laser scans. The effect of noise on detection rates is not assessed in most cases. Another big drawback are the computation times, since none of these algorithms is able to perform close to real time speeds. In our work we show that the proposed approach is able of handling a noisy sensor while performing at several frames per second. Another challenging aspect not explicitly addressed in [10, 13, 4, 22] is handling objects with highly self-similar local shape appearance (e.g. surfaces of revolution or objects with multiple symmetries).

If local minima were of no concern, then estimating the pose of a rigid object given depth data amounts to registering two meshes (a given object of interest and the current depth observation), which can be solved by the ICP algorithm [2] or one of its robust variants. It is well known that ICP requires a good initial estimate to converge to a sensible solution. Ultimately, all methods to detect 3D objects in either depth-only or RGB-D data aim to provide a good initializer for an ICP-like refinement procedure.

3. Our Approach

Before we describe our method in detail, we provide a high-level overview (see also Fig. 1): at test time the algorithm maintains a set of putative matching object coordinates for each pixel in the test image (Figs. 1(d,e)). Instead of sampling minimal sets of correspondences required for (rigid) pose computation, the utility of pairs of correspondences is assessed by using the consistency with the observed depth data. Triplets of correspondences are ranked (Fig. 1(f)), and finally promising ones are evaluated using a standard geometric criterion (Fig. 1(g)) to determine the best-scoring object pose (Fig. 1(h)).

3.1. Descriptor Computation

Given the nature of depth maps and the problem of detecting objects that occupy only a fraction of the image, we opt for a dense (or quasi-dense) computation of descriptors in order not to rely on unstable salient feature points.

A natural choice for a descriptor to represent (local) geometry is based on an implicit volumetric representation of range images and 3D surface meshes. We employ a binary occupancy grid to compute descriptors. A slightly more discriminative volumetric data structure would be a (truncated) signed distance function (TSDF), but we discard this option for efficiency reasons (proper TSDF computation is costly, and the descriptors would use several bits per voxel). We believe that using generalizations of successful gradient-based image descriptors to 3D shapes (such as 3D-SURF [11]) is not necessary, since the intensity values of the (3D) image are known to be only 0 and 1 for occupancy grids (and therefore invariance to intensity transformations is unnecessary). Consequently, our descriptor is a bit string of occupancies in the vicinity of a surface point.

In order to obtain some degree of invariance with respect to viewpoint changes, the z-axis of the local coordinate frame at a surface point is aligned with the (local) surface normal. Given the piecewise smooth characteristic of range images, normals can be estimated relatively reliably for most pixels (after running a Wiener filter to reduce the quantization artifacts observed in triangulation-based depth sensors). For the same reason computation of the second principal direction is highly unreliable and not repeatable. Therefore we compute several descriptors at each surface point by sampling the 2D rotation in the tangential plane (we sample in 20° steps resulting in 18 descriptors per surface point).

Instead of storing a full local occupancy grid (centered at a surface point) we use a subset of voxels (512 in our implementation, i.e. our descriptors are 512 bits long). We initially utilized a conditional mutual information based feature selection method [6] to determine the most informative set of voxels, but this procedure turned out to be rather slow even with the proposed lazy evaluation technique. The reason is that many voxels are not very discriminative, and their respective conditional mutual information is similar. By running feature selection on example training data, we observed that only voxel positions near the tangent plane are selected. Thus, we decided to randomly sample voxel positions in a box aligned with the tangent plane that has half the height than width and depth (we use 8cm × 8cm × 4cm boxes). This means, that building the descriptors from the given depth images or training meshes is very fast.

3.2. Matching

At test time descriptors are computed for each pixel with valid depth and estimated surface normal in the (subsampled) depth image, and the task is to efficiently determine the set of object coordinates with similar local shape appearance. The natural choice to quantify similarity of binary strings is the Hamming distance. We experimented with approximated nearest neighbours implementation for binary data in FLANN [14] and with a hashing based indexing data structure using orthonormal projections [7].[1] Since in our experience the performance is roughly similar for both acceleration strategies, we only report the results using FLANN below.

3.3. Pairwise Compatibility

The matching step returns a list of object coordinate candidates for each pixel with attached descriptors. Even without generating a pose hypothesis it is possible to assess the quality of pairs of putative correspondences by exploiting the information contained in the range image. If p and q are two pixels in the query range image, and $\hat{X}_p$ and $\hat{X}_q$ are the respective back-projected 3D points induced by the observed depth, and X_p and X_q are putative correspondences reported at p and q, then a necessary condition for $\hat{X}_p \leftrightarrow X_p$, $\hat{X}_q \leftrightarrow X_q$ being inlier correspondences is that

$$\big\|\hat{X}_p - \hat{X}_q\big\| \approx \big\|X_p - X_q\big\|. \tag{1}$$

If the Euclidean distance between $\hat{X}_p$ and $\hat{X}_q$ deviates substantially from the one between X_p and X_q, then X_p and X_q cannot be part of an inlier set. The exact quantification of "sufficiently large" deviations depends on the depth sensor characteristics. Note that this criterion is invariant to any hypothesized pose. It can be made stronger (more discriminative) by adding the compatibility of normal estimates as e.g. considered in [4]. In order not to introduce extra tuning parameters of how to weight the distance and normal compatibility terms, we focus on the distance based compatibility of predicted object coordinates in the following. We believe that the loss of discrimination power by excluding normal compatibility has minimal impact on the results, since the final compatibility scores are based on triplets of correspondences as described below. Thus, our scoring function to assess the compatibility between correspondences $X_p \leftrightarrow \hat{X}_p$ and $X_q \leftrightarrow \hat{X}_q$ (which will play the role of pairwise potentials in the following) is given by

$$\psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \overset{\text{def}}{=} \tag{2}$$
$$\begin{cases} \Delta^2(X_p, X_q; \hat{X}_p, \hat{X}_q) & \text{if } |\Delta(X_p, X_q; \hat{X}_p, \hat{X}_q)| \le \sigma \\ \infty & \text{otherwise.} \end{cases}$$

with $\Delta(X_p, X_q; \hat{X}_p, \hat{X}_q) \overset{\text{def}}{=} \|\hat{X}_p - \hat{X}_q\| - \|X_p - X_q\|$. σ is the maximum noise or uncertainty level expected from the depth sensor and matching procedure. Since we densely sample the training data, the value of σ does not need to reflect the surface sampling density of training meshes. We set $\sigma = 3$mm in our experiments.

3.4. Minimal Sample Set Generation

Rigid pose estimation requires at least three (non-degenerate) point-to-point correspondences. Given three such correspondences, e.g. $\{\hat{X}_p \leftrightarrow X_p, \hat{X}_q \leftrightarrow X_q, \hat{X}_r \leftrightarrow X_r\}$, a Euclidean transformation and therefore pose estimate can be computed via the Kabsch algorithm or Horn's method [9]. The task at hand is to generate a promising set of three correspondences from the candidate object coordinates determined for each pixel.

Randomly sampling three putative correspondences will be inefficient, since the inlier ratio is very small as illustrated in the following example: if the object of interest is seen in about 5% of the image pixels, and 10 putative correspondences are maintained per pixel (and contain a true positive for each pixel covered by the object), the inlier ratio

[1] Since the input of the orthogonal transformation is a binary string, faster hashing can be achieved by using respective lookup tables.

is 0.5%, and naive RANSAC sampling at a 95% confidence level will require more than 20 million iterations. This value is only a coarse estimate, since it is too pessimistic (e.g. by assuming a naive sampling over the full image instead of a more sophisticated sampling strategy) and too optimistic (by assuming pixels seeing the object have always a true positive correspondence) at the same time. Nevertheless, almost all random minimal sample sets will contain at least one outlier, and the pairwise compatibility criterion described in Section 3.3 will be crucial to efficiently determine promising sample sets.

To this end we propose to compute min-marginals via dynamic programming on a tree[2] to quickly discard outlier contaminated sample sets. Let $\{p, q, r\}$ be a set of (non-collinear) pixels in the query image, let X_s, $s \in \{p, q, r\}$ range over the putative object coordinates, and $\phi_s(X_s)$ be a unary potential (usually based on the descriptor similary), then the negative log-likelihood (energy) of states (X_p, X_q, X_r) according to our graphical model is

$$\begin{aligned} E_{pqr}(X_p, X_q, X_r) &\overset{\text{def}}{=} \phi_p(X_p) + \phi_q(X_q) + \phi_r(X_r) \\ &+ \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) + \psi(X_p, X_r; \hat{X}_p, \hat{X}_r). \end{aligned} \quad (3)$$

We use the Hamming distance between the descriptor extracted at pixel s and the ones returned by the (approximate) nearest neighbor search for X_s as unary potential $\phi_s(X_s)$.

Note that min-marginals, i.e. the quantities $\mu_{pqr}(X_p) \overset{\text{def}}{=} \min_{X_q, X_r} E_{pqr}(X_p, X_q, X_r)$ for each X_p can be computed via the bottom up pass of belief propagation on a tree rooted at p. In our case we only need 3 correspondences to determine a pose estimate, and therefore the tree degenerates to a chain. If the minimum sample size is larger—e.g. when computing the pose of an object subject to low-parametric and (approximately) isometric deformations—the obvious generalization of the underlying graph is a star graph.

The relevant values computed during BP are the upward messages

$$m_{q \to q}(X_p) = \min_{X_q} \left\{ \phi_q(X_q) + \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \right\} \quad (4)$$

sent from a leaf q to the root p. Note that the min-marginals can be expressed as

$$\begin{aligned} \mu_{pqr}(X_p) &= \min_{X_q, X_r} E_{pqr}(X_p, X_q, X_r) \\ &= \phi_p(X_p) + m_{q \to p}(X_p) + m_{r \to p}(X_p). \end{aligned} \quad (5)$$

Further, observe that the vector of messages $m_{q \to p} \overset{\text{def}}{=} (m_{q \to p}(X_p))_{X_p}$ can be reused in all trees containing the (directed) edge $q \to p$, leading to substantial computational savings. For certain pairwise potentials ψ the message vector computation is sub-quadratic in the number of states (i.e. putative object coordinates in our setting, see e.g. [5]), which would lead to further computational benefits. Unfortunately our choice of the pairwise potential given in Eq. 2 does not allow an obvious faster algorithm for message computation. Message computation does not only yield the value of the messages, $m_{q \to q}(X_p)$, but also the minimizing state

$$X^*_{q \to p}(X_p) \overset{\text{def}}{=} \arg\min_{X_q} \left\{ \phi_q(X_q) + \psi(X_p, X_q; \hat{X}_p, \hat{X}_q) \right\},$$

which is used to quickly determine the optimal object coordinate predictions at pixels q and r given a prediction X_p at pixel p. Computation of the min-marginals $\mu_{pqr}(X_r)$ does not take into account the third edge potential between pixel q and r, $\psi(X_q, X_r; \hat{X}_q, \hat{X}_r)$. Adding this edge to the energy Eq. 3 would require dynamic programming for triple cliques, which we considered to be computationally too costly at this point.[3]

We densely compute the min-marginals for each pixel in the query image (i.e. every pixel is the root), and compute messages $m_{p+\delta_k \to p}$ from pixel located at an offset δ_k, $k \in \{1, \dots, K\}$, from p. Our choice of the set $\{\delta_k\}$ contains the 16 offsets of axis aligned and diagonal offsets at 8 and 16 pixels distance (which aims to trade off locality of predictions and numerical stability of pose estimation). For two edges $q \to p$ and $r \to p$ the predictions $(X_p, X^*_{q \to p}(X_p), X^*_{r \to p}(X_p))$ form a minimal sample set for estimating the rigid pose, and min-marginals are for all $K(K-1)/2$ such triplets are used to rank these minimal sample sets. The method proceed with estimating and evaluating the pose for the top ranked ones (we use 2000) as described in the next section.

3.5. Pose Hypotheses Evaluation

Assessing the quality of a pose hypothesis by aligning the 3D model with the range image appears to be straightforward—if the poses are affected by no or minimal noise. We do expect a substantial noise level in our pose hypotheses, and a sensible scoring function to rank the poses needs to take this into account. To this end a scoring function needs to be invariant to pose uncertainties. Since the true pose is effectively a latent variable, we can either marginalize (i.e. average) over nearby poses[4] or maximize over the latent pose. We choose the latter option. Since we do not expect or assume to obtain many pose hypotheses near the true pose, we refrain from using pose clustering or averaging approaches e.g. employed in [4, 16]. In contrast to works such as [20, 3], which refine a pose entirely based on correspondences between predicted object coordinates and observed depth geometry, we utilize a "classical"

[2] Understood as an instance of min-sum belief propagation.

[3] DP would be cubic in the number of states in such setting.

[4] Which essentially amounts to smoothing the input, see [12] for an extensive discussion of building invariance with respect to (geometric) transformation.

geometric approach by determining an optimal alignment between the given 3D model points and the depth map.

A proper way to assess the quality of a hypothesized pose (or any latent variable in general) is to "explain" the data given the assumptions on the sensor noise, i.e. to formulate a respective cost function that sums (integrates) over the image domain. Unfortunately, this more principled formulation is expensive to optimize. Thus, we employ—like most of the respective literature—the reverse direction of "explaining" the model for computational reasons (recall that up to 2000 pose hypotheses are considered at this stage). We implemented several methods to robustly refine the pose of a point set with respect to a depth map, including pose refinement via (robust) non-linear least squares. In our experience the following simple alternation algorithm proves to be efficient and effective:

1. Perform "projective data association" (i.e. establish the correspondence between a model point X_j and the back-projected depth $\hat{X}_j$ with both $\hat{X}_j$ and $RX_j + T$ being on the same line-of-sight), and

2. update R and T using a weighted extension of the Kabsch algorithm (also known as Wahba's problem). The weights w_j are derived from the smooth approximation of the robust truncated quadratic kernel (see e.g. [25, 24] for a discussion of this kernel)

$$\rho_\tau(e) \stackrel{\text{def}}{=} \begin{cases} \frac{e^2}{4}\left(2 - \frac{e^2}{\tau^2}\right) & \text{if } e^2 \leq \tau^2 \\ \frac{\tau^2}{4} & \text{otherwise,} \end{cases} \tag{6}$$

$$\omega_\tau(e) \stackrel{\text{def}}{=} \rho'_\tau(e)/e = \max\{0, 1 - e^2/\tau^2\}, \tag{7}$$

and given by

$$w_j = \omega_\tau\left(\left(RX_j + T - \hat{X}_j\right)_3\right). \tag{8}$$

The weights given in Eq. 8 are based on depth deviation between the transformed model point and the corresponding value in the depth map. If a deph value is missing for the projected model point, that correspondence is considered an outliers and has zero weight. τ is the inlier noise level and we use the same value as for σ (which is 3mm, recall Sec. 3.3). Please observe that this algorithm does not optimize a single energy (a property shared with most ICP variants using projective data association). We iterate these two steps 10 times on a (random) subset of 1000 model points. The final score of the pose hypothesis is evaluated on a larger subset of 10000 model points by using a robust fitting cost,

$$\sum_j \rho_\tau\left(\left(RX_j + T - \hat{X}_j\right)_3\right). \tag{9}$$

The pose with the lowest cost is reported and visualized.

3.6. Implementation Notes

Training phase: The core data used in the training stage are depth images of the object(s) of interest together with the respective pose data. These depth maps can be generated synthetically from e.g. CAD models or captured by a depth sensor. If CAD models are rendered, the camera poses are generated randomly looking towards the object's center of gravity. At this point we do not aim to simulate the real depth sensor characteristic (e.g. noise or quantization effects), which in some cases led to missed correspondences in parts of the object (e.g. the top of the pipe in Fig. 1 has a substantially different appearance in rendered and real depth maps). From these depth maps we extract a target number of descriptors (typically 32k in our experiments) by selecting a random subset of (valid) pixels in the depth map. Random sampling is slightly biased towards pixels in the depth map with close to fronto-parallel surface patches. Thus, about 600k descriptors (32k × 18 for the sampled tangent-plane rotations) are generated and stored. No further processing takes part at training time. Consequently, the training phase is completed within seconds.

Parallel implementation: Most steps in our approach can be trivially parallelized (including descriptor extraction, matching against the database, message passing, and pose evaluation). While we did not implement any part of the algorithm on the GPU, we made straightforward use of OpenMP-based multi-processing whenever possible. The input depth maps are 640×480 pixels, but we compute predicted object coordinates on either 320×240 or 160×120 images (the latter one for to achieve interactive frame rates). On a dual Xeon E5-2690 system we achieve between 2 frames per second (320×240 resolution) or up to 10 Hz (160×120). Nearest-neighbor descriptor matching is usually the most time consuming part (see also Fig. 4). We anticipate real-time performance of a GPU implementation.

4. Experiments

We show results on the Mian dataset [13], since it is the de facto baseline benchmark dataset for 3D object detection algorithms. We also show our own datasets recorded with the ASUS Xtion camera in order to demonstrate our algorithms ability to cope with noisy inputs. Since our 3D object detection algorithm takes depth maps as input, we converted the given meshes to range images by rendering into 640×480 depth maps using approximate parameters for the camera intrinsics (since exact calibration parameters of the range scanner are not available). Consequently, the amount of occlusions in our depth maps may be slightly higher than in the provided meshes. We show as baseline methods the following approaches: Spin images [13], Tensor matching [13], Drost et al. [4], SVS [15] and Tuzel et al. [22].

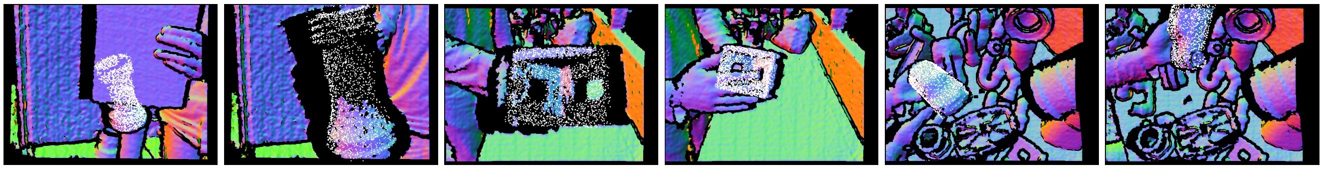

Figure 2. Sample frames from the ASUS Xtion sequences. The respective model point cloud is superimposed on the normal-map rendered input. Correct detections and poses can be seen despite large occlusions, missing depth data, and strong viewpoint changes. The full sequences are provided in the supplementary material.

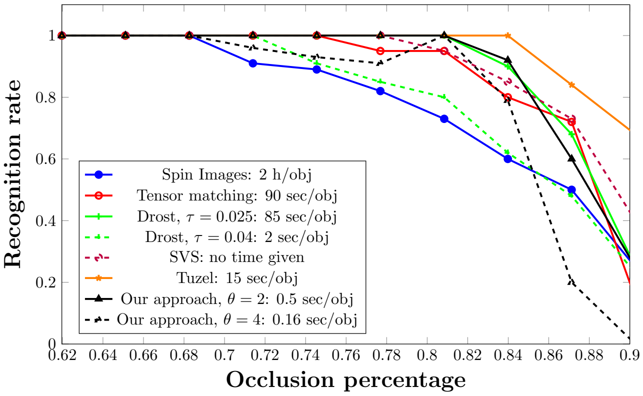

Figure 3. Results obtained on the Mian dataset [13]. It can be seen that our method is capable to handle occlusions of up to 81% and still give 100% of detection rates. It is also significant that the time required to detect a single object compared to the only other approaches that obtain similar or better detection rates [4, 22], is of up to 30 times less for our approach when compared with Tuzel and up to 170 times less compared to Drost.

Experimental setup: The Mian dataset contains 50 scenes with 4 models on which to perform detection. Ground truth pose is provided for all instances of all objects. Apart from those 4 models another model exists that was excluded in Mian's experiments [13]; hence our approach and all baselines do not include this object. To validate a detection as valid we use the same thresholds as used in [4], we also define occlusion values in the same manner. We provide results for two different resolutions for the prediction image, 320×240 (downsampling factor $\theta = 2$), and 160×120 ($\theta = 4$). A smaller resolution of the predicted object coordinate image means faster computation, but also a lower probability of finding a inlier sample set (and consequently returning a successful detection).

Experimental results: The quantitative evaluation using the same evaluation methodology as in [13] is shown in Fig. 3. In general, our method has state-of-the-art performance at the "high quality setting" ($\theta = 2$), and the choice of $\theta = 4$ to achieve interactive frame rates outperforms other fast methods at most occlusion percentages. Due to the impact of downsampling our method performs worse for highly occluded objects. Note that according to the evaluation methodology the curves in Fig. 3 are not necessarily monotonically decreasing with respect to occlusion percentages.

Commodity depth sensor data: The results on the Mian dataset give us a clear understanding of how our approach compares against previous work, but at the same time the data is much cleaner than depth maps obtained by current commodity sensors. Consequently, we recorded our own data using an ASUS Xtion depth sensor and ran our method for objects with available CAD models (either obtained from a 3D model database, such as the toy car and the bracket, or by approximate manual 3D modeling of pipe-like structures). When creating the descriptors for the objects of interest we do not simulate any of the depth sensor characteristics (such as boundary fattening and depth quantization). Thus, the 3D model to detect and the actual range images may be significantly different in their depth appearance. Fig. 2 depicts sample frames with the model point cloud superimposed on the input depth (rendered via its normal map). The full sequences are provided in the supplementary material. These sequences differ in several aspects from the benchmark dataset [13]: the depth sensor characteristics at training time and test time do not match, the depth maps at test time are limited in quality (compared to a more expensive scanning setup), and objects themselves are less discriminative in shape.

Computation time: We present results with a CPU implementation of the approach, although a GPU implementation for most steps in the algorithm is straightforward and is expected to yield real-time performance (20Hz). In Fig. 4 we break down the individual contributions of the various stages of our method (descriptor computation, Hamming distance based descriptor matching using FLANN, message

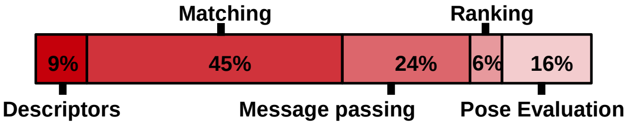

Figure 4. Percentage of the total time employed in each of the stages of the algorithm. We can see that by far the most expensive step is the feature matching step.

passing for min-marginal computation, ranking/sorting according to Eq. 5, and final pose evaluation including ICP). The exact values vary depending on the input frame and the object of interest, but in general feature matching (i.e. nearest neighbor search) consumes a dominant fraction of the overall frame time. The matching time is typically faster for object with a highly distinctive local shape appearances than for object with redundant surface structures, since in the former case the search trees tend to be more balanced.

5. Conclusions

We have addressed the problem of 3D object detection and corresponding pose estimation, and we discussed a more efficient paradigm to solve this task while still obtaining state of the art detection rates. We believe that this work creates a new and robust framework from which to build new 3D object detection approaches. In this current work we left out basically any learning-based technique to boost the detection performance or run-time behavior. While we argue that computationally expensive learning techniques will limit the general applicability of 3D object recognition (since adding new objects requires time-consuming retraining), we foresee that more sophisticated processing of training objects than our current one will lead to more discriminative descriptors, and therefore will be highly beneficial for this task.

Ackowledgement: We are very grateful to Frank Perbet for his help with creating CAD models used for training.

References

[1] P. Bariya and K. Nishino. Scale-hierarchical 3d object recognition in cluttered scenes. In *Proc. CVPR*, pages 1657–1664, 2010. 2

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. 3

[3] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proc. ECCV*, volume 8690, pages 536–551, 2014. 1, 2, 5

[4] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Proc. CVPR*, pages 998–1005, 2010. 2, 3, 4, 5, 6, 7

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006. 5

[6] F. Fleuret. Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research*, 5:1531–1555, 2004. 4

[7] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2916–2929, 2013. 4

[8] 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 *Proc. ICCV*, 2011. 1, 2

[9] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987. 4

[10] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999. 2, 3

[11] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool. Hough transform and 3D SURF for robust three dimensional classification. In *Proc. ECCV*, pages 589–602, 2010. 2, 3

[12] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 5

[13] A. S. Mian, M. Bennamoun, and R. Owens. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1584–1601, 2006. 2, 3, 6, 7

[14] M. Muja and D. G. Lowe. Fast matching of binary features. In *Computer and Robot Vision*, pages 404–410, 2012. 4

[15] H. V. Nguyen and F. Porikli. Support vector shape: A classifier-based shape representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):970–982, 2013. 6

[16] M.-T. Pham, O. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla. A new distance for scale-invariant 3d shape recognition and registration. In *Proc. ICCV*, pages 145–152, 2011. 2, 5

[17] R. Rios-Cabrera and T. Tuytelaars. Discriminatively trained templates for 3d object detection: A real time scalable approach. In *Proc. ICCV*, December 2013. 1, 2

[18] E. Rodolà, A. Albarelli, F. Bergamasco, and A. Torsello. A scale independent selection process for 3d object recognition in cluttered scenes. *IJCV*, 102(1-3):129–145, 2013. 2

[19] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA'09*, pages 1848–1853, 2009. 2

[20] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proc. CVPR*, pages 2930–2937, 2013. 1, 5

[21] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *Proc. ECCV*, volume 8694, pages 462–477, 2014. 1, 2

[22] O. Tuzel, M.-Y. Liu, Y. Taguchi, and A. Raghunathan. Learning to rank 3d features. In *Proc. ECCV*, pages 520–535, 2014. 2, 3, 6, 7

[23] O. Woodford, M.-T. Pham, A. Maki, F. Perbet, and B. Stenger. Demisting the hough transform for 3d shape recognition and registration. In *Proc. BMVC*, pages 32.1–32.11, 2011. 2

[24] C. Zach. Robust bundle adjustment revisited. In *Proc. ECCV*, pages 772–787, 2014. 6

[25] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, and C. Theobalt. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics, TOG*, 2014. 6