Image and Vision Computing 63 (2017) 38–50

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A learning-based variable size part extraction architecture for 6D object pose recovery in depth images☆

Caner Sahin[a,*], Rigas Kouskouridas[b], Tae-Kyun Kim[a]

[a] *Electrical and Electronic Engineering Department, Imperial Computer Vision and Learning Lab (ICVL), Imperial College London, SW72AZ, UK*
[b] *Wirewax, London W1T2RB, UK*

ARTICLE INFO

Article history:
Received 2 September 2016
Received in revised form 27 March 2017
Accepted 13 May 2017
Available online 21 May 2017

Keywords:
Object registration
6 DoF pose estimation
Scale-variant HoCP feature
One class training
Random forest
Iterative refinement

ABSTRACT

State-of-the-art techniques for 6D object pose recovery depend on occlusion-free point clouds to accurately register objects in 3D space. To deal with this shortcoming, we introduce a novel architecture called *Iterative Hough Forest with Histogram of Control Points* that is capable of estimating the 6D pose of an occluded and cluttered object, given a candidate 2D bounding box. Our *Iterative Hough Forest (IHF)* is learnt using parts extracted only from the positive samples. These parts are represented with *Histogram of Control Points (HoCP)*, a "scale-variant" implicit volumetric description, which we derive from recently introduced Implicit B-Splines (IBS). The rich discriminative information provided by the scale-variant HoCP features is leveraged during inference. An automatic variable size part extraction framework iteratively refines the object's roughly aligned initial pose due to the extraction of coarsest parts, the ones occupying the largest area in image pixels. The iterative refinement is accomplished based on finer (smaller) parts, which are represented with more discriminative control point descriptors by using our *Iterative Hough Forest*. Experiments conducted on a publicly available dataset report that our approach shows better registration performance than the state-of-the-art methods.

1. Introduction

Object registration is an important task in computer vision that determines the position and the orientation of an object in camera-centered coordinates [28]. An object of interest that was detected beforehand in a coarse 2D bounding box is fed into a registration system that can superimpose the desired translation and rotation of the object onto the raw camera image. By utilizing such a system, one can propose promising solutions for various problems related to scene understanding, augmented reality [25], control and navigation of robotics [26], *etc.* Recent developments on visual depth sensors and their increasing ubiquity have allowed researchers to make use of the information acquired from these devices to facilitate challenging registration scenarios.

Iterative Closest Point (ICP) algorithm [1], point-to-model based methods [2,3], and point-to-point techniques [4,5] demonstrate good registration results. However, the performance of these approaches is severely degraded in cases of heavy occlusion and clutter, and similar-looking distractors. In order to address these challenges, several learning-based methods formulate occlusion aware features [6], derive patch-based (local) descriptors [15], or encode the contextual information of the objects with simple depth pixels [8], and integrate them into the random forests. Most particularly, iterative random forest algorithms such as Latent-Class Hough forest (LCHF) [15] and iterative Multi-Output Random forest (iMORF) [9] obtain state-of-the-art accuracy on pose estimation. On the other hand, these methods rely on scale-invariant features, while the exploitation of rich discriminative information inherently embedded into the scale-variability is one important point been overlooked.

Unlike the aforementioned learning-based methods, the ones presented by Novatnack et al. [10,11] utilize the detailed information of the scale variation in order to register range images in a coarse-to-fine fashion. Although promising, they extract and match conventional salient 3D key points. However, real depth sensors have several imperfections, such as missing depth values, noisy measurements, and foreground fattening. As a result, salient feature points used in Ref. [10] tend to be located on these deficient parts of the depth images, and hence, they are rather unstable [12]. In such a scenario, 3D reconstruction methods that provide more reliable shape

☆ This paper has been recommended for acceptance by Cornelia M Fermuller
* Corresponding author.
E-mail addresses: c.sahin14@imperial.ac.uk (C. Sahin), rkouskou@gmail.com (R. Kouskouridas), tk.kim@imperial.ac.uk (T.-K. Kim).

http://dx.doi.org/10.1016/j.imavis.2017.05.005
0262-8856/© 2017 Elsevier B.V. All rights reserved.

information can be utilized [6]. Implicit B-Splines (IBS) [7,13] are techniques that can provide shape descriptors through their zero-sets and reconstruct surfaces. They are based on locally controlled functions that when combined with their control points produce a rich part-based object representation.

Our architecture is originated from these observations. We integrate the coarse-to-fine registration approach presented in Ref. [10] into the random forest framework [9,15] using Histogram of Control Points (HoCP) features that we adapt from recently introduced IBSs [13]. We train our forest only from positive samples and learn the detailed information of the scale-variability during training. We normalize every training point cloud into a unit cube and then generate a set of scale-space images, each of which is separated by a constant factor. The parts extracted from the images in this set are represented with the scale-variant HoCP features. During inference, the parts centered on the pixels that belong to the background and foreground clutters are removed iteratively using the most confident hypotheses, and the test image is updated. Since this removal process decreases the standard deviation of the test point cloud, subsequent normalization applied to the updated test image increases the relative scale of the object (foreground pixels) in the unit cube. More discriminative control point descriptors are computed at higher scales, and this ensures the refinement of the object pose. In our prior work [14], we have evaluated the registration performance of the proposed architecture by only using fixed size parts. We extend the work engineering an automatic variable size part extraction framework in such a way that we can further exploit the discriminative information provided by the HoCP features. This framework first roughly aligns the object of interest by extracting coarsest parts, the ones occupying the largest area in image pixels, and then iteratively refines its alignment based on finer (smaller) parts that are represented with more discriminative control point descriptors. Note that, we employ a compositional approach, *i.e.*, we concurrently detect the object in the target region and estimate its pose by aligning the parts in order to increase robustness across clutter. Fig. 1 depicts a sample result of our architecture. To summarize, our main contributions are as follows:

- To the best of our knowledge, this is the first time an implicit object representation, Implicit B-Spline, is adapted into a "scale-variant" part descriptor and is associated with the random forests.
- We introduce a novel iterative algorithm for the Hough forests: it finds out an initial hypothesis and improves its confidence iteratively by extracting more discriminative "scale-variant" descriptors due to the elimination of the background/foreground clutter.
- We engineer an automatic variable size part extraction framework for the random forests: it first roughly aligns the object of interest by extracting coarsest parts and iteratively improves its confidence based on finer (smaller) ones.

The rest of the paper is organized as follows: In Section 2, we present a review on the object registration. Section 3 demonstrates the computation procedure of the HoCP features as a scale-variant part representation, their combination with the Iterative Hough Forest (IHF), and the registration process. Experimental results are provided in Section 4, and finally, the paper is concluded in Section 5 with several remarks, and discussions.

2. Related work

A large number of studies have been proposed for object registration, ranging from point-wise correspondence based methods to learning-based approaches. Iterative Closest Point (ICP) algorithm, originally presented in Ref. [1], requires a good initialization in order not to be trapped in a local minimum during fine tuning. This issue is addressed in Ref. [21] providing globally optimal registration by the integration of a global branch-and-bound (BnB) optimization into the local ICP. The point-wise correspondence problem is converted into a point-to-model registration in Ref. [2]. The object model is represented with implicit polynomials (IP), and the distance between the test point set and the object model is minimized via the Levenberg-Marquardt algorithm (LMA). Zheng et al. [22] propose a 6 DoF pose estimation technique utilizing 3D IPs on ultrasound images. In the off-line phase, object model is represented with 3D IPs, and by utilizing its gradient flow, 2D ultrasound image is registered in the on-line process. In Ref. [23], a coarse-to-fine fast IP-driven registration method is presented. A rough pose estimation is quickly acquired with a coarse IP model (low degree curve fitting), and finer models refine the parameters of this rough estimation (high degree curve fitting). Hinterstoisser et al. [36] extract holistic templates from 3D models of the objects and match to the scene at test time. These

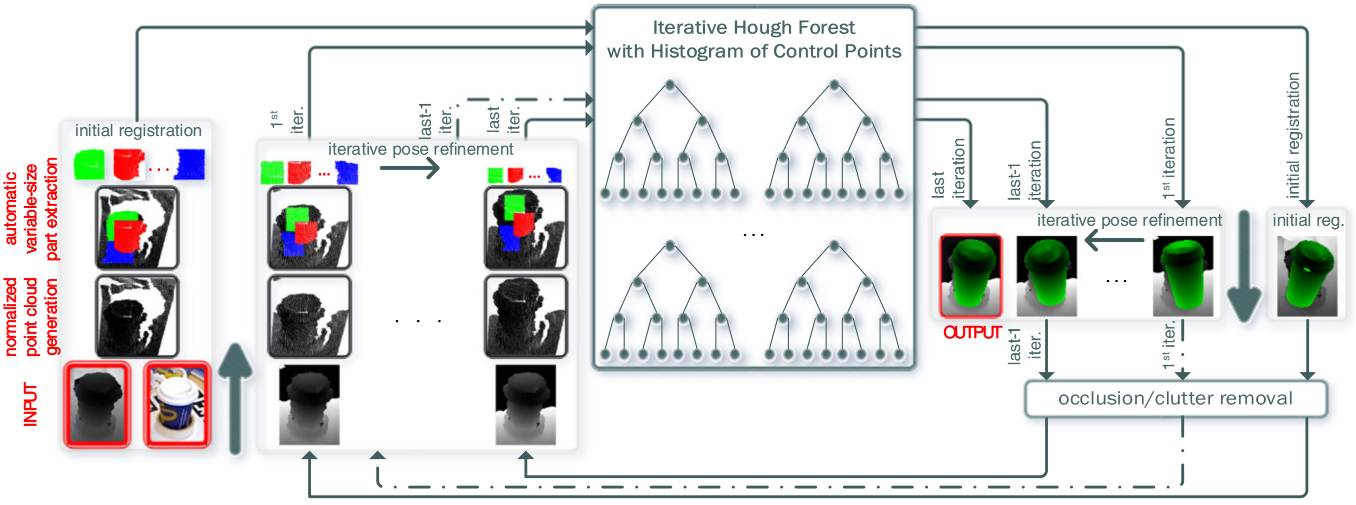

Fig. 1. Sample result of our architecture: the object of interest (lower-left corner) is first roughly aligned by extracting coarsest parts, the ones occupying the largest area in image pixels. This alignment is then iteratively refined based on finer (smaller) parts that are represented with more discriminative descriptors (The RGB image of the object of interest is for visualization purposes, the color-coded parts are centered on the same pixels).

studies have demonstrated good registration results on occlusion-free target point sets, and/or when the points sets are subjected to the artificial Gaussian noises and outliers.

Unlike the abovementioned methods, more realistic registration scenarios have been addressed by the point-to-point techniques that build point-pair features for sparse representations of the test and the model point sets [27,30,31]. Rusu et al. align two noisy point clouds of real scenes by finding correct point-to-point correspondences between the Point Feature Histograms (PFH), and feed this alignment to an ICP algorithm for fine tuning [29]. The cluttered and partially occluded objects' poses are hypothesized by accumulating the votes of the matching features in Ref. [30]. Choi et al. [31] propose point-pair features for both RGB and depth channels. These features are conducted in a voting scheme to hypothesize the rotation and translation parameters of the objects in the cluttered scenes. The features proposed in Ref. [5] make use of the visibility context of the scene to tackle the registration. Despite achieving good registration results, these techniques underperform when the scenes are under heavy occlusion and clutter, and the objects' geometry are indistinguishable from the background.

Learning-based methods have good generalization across severe occlusion and clutter [6,8,24,32,33]. The method presented in Ref. [6] formulates the recognition problem globally and derives occlusion aware features. A set of principal curvature ratios are computed for all pixels in depth images to extract the edgelets. In Ref. [8], the contextual information of the objects is encoded with simple depth and RGB pixels. This technique improves the confidence of a pose hypothesis using a Ransac-like algorithm. Cabrera et al. [33] back project the parts inside the initially found coarse bounding box to the image and pass down the forest again. The parts with the lowest contributions are penalized in such a way that finer registration is produced.

The state-of-the-art accuracy on registration is acquired by the iterative random forest algorithms [16]. The part-based strategy, Latent-Class Hough Forest [15], refines the initially hypothesized object pose by iteratively updating the object class distributions in the leaf nodes during testing. Iterative Multi Output Random Forest (iMORF) [9] jointly predicts head pose, facial expression, and the landmark positions. The relations between these tasks are modelled so that their performances are iteratively improved with the extraction of more informative features. The ideas, iterative pose refinement during testing and iterative extraction of more discriminative features form a basis for our Iterative Hough Forest (IHF) architecture: during training, we encode discriminative shape information of the HoCP features into the forest. Despite that the skeleton of our training procedure is similar to the methodology in Ref. [15], our forest learns the discriminative shape information that will be iteratively exploited at test time. In the course of inference, unlike Ref. [15], we update the test image itself and the hypotheses confidence by a noise removal process that allows us to extract more informative features from the test images. While these approaches [9,15] rely on the scale-invariant features to improve the confidence of a pose hypothesis, Novatnack et al. [10,11] introduce a framework that registers the range images in a coarse-to-fine fashion by utilizing the detailed information provided by the scale variation. The shape descriptors with the coarsest scale are initially matched, and a rough alignment is achieved since fewer features are extracted in coarser scales. The descriptor matching at higher scales produces improved predictions of the pose. Inspired by Ref. [10], we design a "scale-variant" implicit volumetric part description, "Histogram of Control Points (HoCP)", and associate it with the random forest framework.

Selecting the part size is important since larger parts tend to match the disadvantages of a holistic template, while smaller ones are prone to noise [15]. In heavily occluded and cluttered scenes, relatively smaller parts perform well, while the larger ones are more convenient in occlusion/clutter-free scenarios. Discriminative information encoded into small-sized parts might not be fully exploited by larger parts, most particularly when the object representation is scale-variant. Hence, this application-specific/task-dependent part size selection degrades the generalization and it is one of the remaining challenges that should be addressed, apart from occlusion, clutter, and/or similar-looking distractors. Beyond object pose estimation [14,15], there are several part-based solutions proposed for different tasks, such as human pose recognition [17,18], 3D face analysis [19], or hand pose estimation [20] to name a few. They experience different part sizes and select the one that performs best, however, none of these solutions investigate how extracting variable size parts can be utilized in a single framework. In this study, we investigate the effect of this *size variation* and show that the simultaneous utilization of the parts of varying size can improve 6D object pose estimation, especially in heavily occluded and cluttered depth maps, supplying a rich source of discriminative information.

3. Our registration approach

In this section, we detail our registration approach by firstly describing the computation procedure of HoCP features as a scale-variant part representation. We then present how to encode the discriminative information of these scale-variant features into the forest. Finally, we demonstrate how to exploit the learnt shape information in a coarse-to-fine fashion to refine the pose hypotheses. Throughout the paper, we use the terms *part* and *patch* interchangeably.

3.1. Scale-variant part representation: histogram of Control Points

Given a positive depth image, we initially normalize it into a unit cube and then sample new point clouds at different scales as follows:

$$\{\mathbf{X}_N\}_i = \frac{\mathbf{X}_{(n_p\times 3)} - \bar{\mathbf{X}}_{(n_p\times 3)}}{s_i * \boldsymbol{\alpha}} + 0.5, \quad i = 0,1,2, \quad \ldots, m \tag{1}$$

with

$$\boldsymbol{\alpha} = \max \left\{ \begin{array}{l} \max(X) - \min(X) \\ \max(Y) - \min(Y) \\ \max(Z) - \min(Z) \end{array} \right\}.$$

In Eq. (1), $\mathbf{X} = [X, Y, Z]$ denotes the world coordinate vector of the original foreground point cloud, $\bar{\mathbf{X}}$ shows the mean of $\mathbf{X}$. The constant s_i takes real numbers to generate point clouds at different scales, starting from $s_0 = 1$ that corresponds to the initial normalization. α is the scale factor, m is the number of the scales, and n_p is the number of the points of the relevant set. $\mathbf{X}_N = [X_N, Y_N, Z_N]$ is the matrix of normalized foreground pixels. Within this context, we define the scale h of each normalized foreground point cloud i as given below:

$$h_i = \max\left(Z_{N_i}\right) - \min\left(Z_{N_i}\right). \tag{2}$$

Fig. 2 (a) shows a training image and its samples at different scales. Once we generate a set of scale-space point clouds, we represent each of those first globally with the control points of Implicit B-Splines (IBS). IBS is defined through the combination of B-Spline tensor products:

$$f(\mathbf{x}) = \sum_{i,j,k=1}^{N} n_{i,j,k} B_i(x) B_j(y) B_k(z) \tag{3}$$

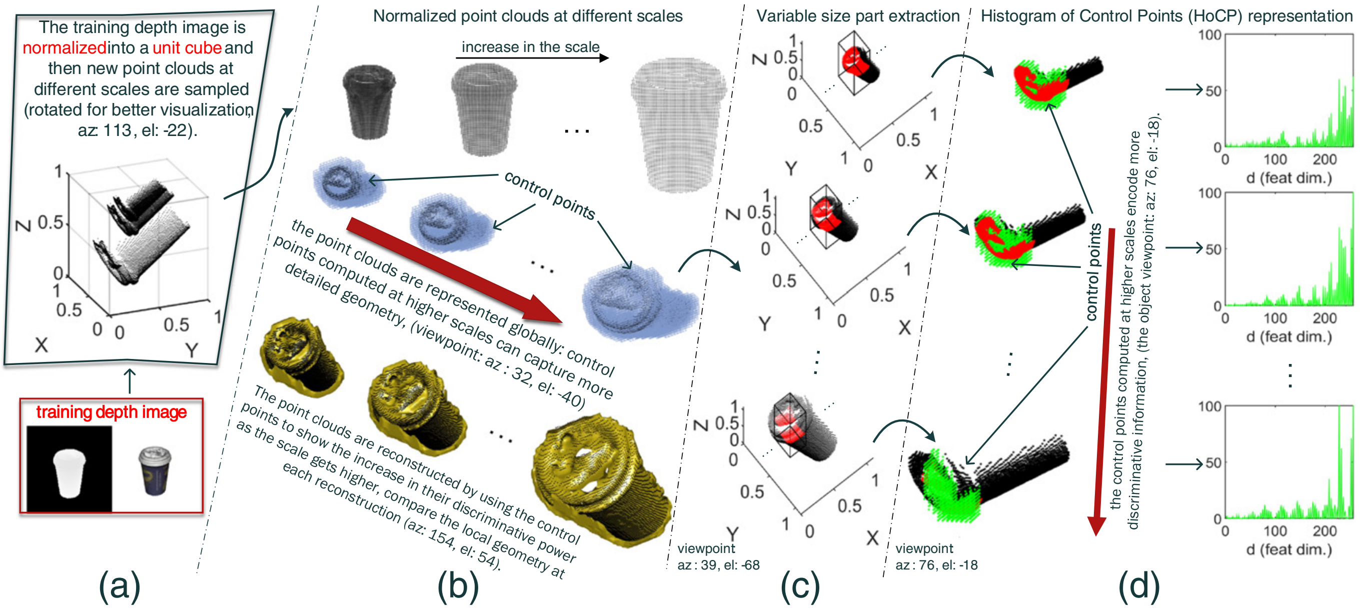
The training depth image is normalizedinto a unit cubeand then new point clouds at different scales are sampled (rotated for better visualization, az: 113, el: -22).
training depth image
Normalized point clouds at different scales
increase in the scale
control points
the point clouds are represented globally: control points computed at higher scales can capture more detailed geometry, (viewpoint: az : 32, el: -40)
The point clouds are reconstructed by using the control points to show the increase in their discriminative power as the scale gets higher, compare the local geometry at each reconstruction (az: 154, el: 54).
Variable size part extraction
viewpoint az : 39, el: -68
viewpoint az : 76, el: -18
Histogram of Control Points (HoCP) representation
control points
the control points computed at higher scales encode more discriminative information, (the object viewpoint: az: 76, el: -18).
d (feat dim.)
X
Y
Z
(a)
(b)
(c)
(d)

Fig. 2. Computation procedure of HoCP features as a scale-variant part representation: (a) initial normalization ($s_0 = 1$) of the training depth image is the outmost point cloud, and the inner ones are sampled by different s_i values. (b) Global representation of the scale-space images. (c) Extraction process of the variable size parts (centers of these parts are the same). (d) HoCP representation of the parts extracted in (c).

where $\{n_{i,j,k}\}$ are the coefficients defining a control lattice of size $N \times N \times N$. The spline bases $B_i(x)$, $B_j(y)$, and $B_k(z)$ are the functions of the given point (x, y, z). This definition can be reformulated as the following inner product:

$$f(\mathbf{x}) = \mathbf{n}^T \mathbf{e}(\mathbf{x}) = \mathbf{e}(\mathbf{x})^T \mathbf{n} \tag{4}$$

where the coefficient vector $\mathbf{n}$ includes the control values $\{n_{i,j,k}\}$, and the basis vector $\mathbf{e}(\mathbf{x})$ depends on the given data points sorting the spline basis function products $\{B_i(x)B_j(y)B_k(z)\}$. The basis vectors in Eq. (4) are computed for the whole point cloud, and the coefficient vector $\mathbf{n}$ is calculated based on the 3L algorithm [37]. Rouhani et al. [13] construct the spline basis functions $B_i(x)$, $B_j(y)$, $B_k(z)$ through the following blending functions:

$$b_0(u) = (1 - u^3)/6, \quad b_1(u) = (3u^3 - 6u^2 + 4)/6$$
$$b_2(u) = (-3u^3 + 3u^2 + 3u + 1)/6, \quad b_3(u) = u^3/6 \tag{5}$$

and reformulate Eq. (3) in order to determine the control point vector $\mathbf{n}$ of the point clouds that are normalized into the unit cube $[0\ 1]^3$:

$$f(\mathbf{x}) = \sum_{l,m,p=0}^{3} n_{i+l,j+m,k+p} b_l(u) b_m(v) b_p(w) \tag{6}$$

where

$$i = \lceil x/\Delta \rceil, \quad j = \lceil y/\Delta \rceil, \quad k = \lceil z/\Delta \rceil$$
$$u = \frac{x}{\Delta} - \left.\frac{x}{\Delta}\right\rfloor, \quad v = \frac{y}{\Delta} - \left.\frac{y}{\Delta}\right\rfloor, \quad w = \frac{z}{\Delta} - \left.\frac{z}{\Delta}\right\rfloor$$
$$\Delta = 1/(N - 3).$$

Thus, the unit cube is split into an $N \times N \times N$ voxel grid where N is the IBS resolution. Each control point in $\mathbf{n}$ is defined with an index-weight pair: the index number indicates the vertex of this grid at which the related control point is located. The weight informs the descriptor significance about the control of the geometry to be represented. The scale-space images in Fig. 2 (a) are globally represented in Fig. 2 (b) with the control point descriptors. We use all control points to represent the structures, but one can sort these descriptors based on their weights and utilize the ones higher than a threshold. In Fig. 2 (b), the point clouds are lastly reconstructed by using the control points to show the increase in their discriminative power as the scale gets higher. Note that, IBS resolution N determines the complexity of the representation (level of detail) in a unit cube, while the scale h indicates the relative size of the object with respect to the unit cube dimensions. In our architecture, despite sampling point clouds at different scales h_i, (*e.g.* $i = 0, 1, \ldots, 8$), we fix the complexity of the representation, (*e.g.* $N = 50$).

IBS is the combination of the locally controlled functions and allows one to propose effective part-based solutions for object registration. We benefit from such a property and partition the globally represented scale-space depth maps into parts. The part size g is expressed in image pixels. It depicts the ratio between the sizes of the extracted part and the bounding box of the global point cloud. In our prior work [14], we have extracted and represented the parts that have the same size, that is, the parts growing around every individual pixel at each scale occupy the same area in image pixels. We now extend the work extracting the parts that are different in size. A 3D bounding box defined in metric coordinates is traversed in the unit cube of each scale-space image, and the parts are extracted around non-zero pixels. The total number of the data points in this 3D bounding box varies for the point clouds at different scales, and consequently, the size of the extracted parts differs. Fig. 2 (c) shows an example of variable size part extraction process in which the red parts are grown around the same data point. Note how the part size decreases when the scale of the normalized object point cloud gets higher, since less number of data points are deployed in the 3D bounding box. Each part has its own implicit volumetric representation, formed by the closest control points to the part center, the ones lying inside the 3D bounding box along depth direction. Such a part description characterizes the locality in a cascaded fashion, growing regions with different characteristics around a point. We encode this information into histograms in spherical coordinates. Each of the part centers is coincided with the center of a sphere. The control points of the part are described by the log of the radius t_r, the cosine of the inclination t_θ, and the azimuth t_ϕ. Then, the sphere is divided into the bins, and the relation between the bin numbers $\nu_r, \nu_\theta, \nu_\phi$ and the histogram coordinates t_r, t_θ, t_ϕ is given as follows [34]:

$$\begin{aligned} t_r &= \frac{\nu_r}{log\left(\frac{r_{\max}}{r_{\min}}\right)} log\left(\frac{r}{r_{\min}}\right) \\ t_\theta &= \nu_\theta \frac{z_d}{r} \\ t_\phi &= \frac{\nu_\phi}{2\pi} \tan^{-1}\left(\frac{y_d}{x_d}\right) \end{aligned} \tag{7}$$

where $r_{\min}$ and $r_{\max}$ are the radii of the nested spheres with the minimum and the maximum volumes, x_d, y_d, z_d are the Cartesian coordinates of each descriptor with radius r. $r_{\max}$ equals to the distance between the patch center and the farthest descriptor of the related patch. The numbers of the control points in each bin are counted and stored in a $d = \nu_r * \nu_\theta * \nu_\phi$ dimensional feature vector $\mathbf{f}$. Fig. 2 (d) illustrates the HoCP representations of the parts extracted in Fig. 2 (c). Note that the control points computed at higher scales capture more detailed part geometry.

3.2. Combination of HoCP and Iterative Hough Forest

The proposed IHF is the combination of randomized binary decision trees. It is trained only on foreground synthetically rendered depth images of the object of interest. We generate a set of scale-space images from each training point cloud. Then, we sample a set of parts $\{P_i\}$ as explained in Section 3.1 and annotate those as follows:

$$\mathcal{P} = \{P_i\} = \left\{(\mathbf{c}_i, \Delta\mathbf{x}_i, \boldsymbol{\theta}_i, \mathbf{f}_i, D_i)\right\} \tag{8}$$

where $\mathbf{c}_i = (c_{x_i}, c_{y_i})$ is the part center in pixels, $\Delta\mathbf{x}_i = (\Delta x_i, \Delta y_i, \Delta z_i)$ is the 3D offset between the centers of the part and the object, $\theta_i = (\theta_{r_i}, \theta_{p_i}, \theta_{y_i})$ is the rotation parameters of the point cloud from which the part P_i is extracted, and D_i is the depth map of the part.

Each tree is constructed by using a subset $\mathcal{S}$ of the annotated training parts $\mathcal{S} \subset \mathcal{P}$. We randomly select a template part T from $\mathcal{S}$ and assign it to the root node. We measure the similarity between T and each part S_i in $\mathcal{S}$ as follows:

- **Depth check:** The depth values of the descriptors $S_i^{\mathbf{n}}$ and $T^{\mathbf{n}}$ that represent the parts S_i and T are checked. The spatially inconsistent descriptors in $S_i^{\mathbf{n}}$ are removed as in Ref. [15], generating Ω that includes the spatially consistent descriptors of the patch S_i.
- **Similarity measure:** Using Ω, the feature vector $\mathbf{f}_\Omega$ is generated, and the $\mathcal{L}_2$ norm between this vector and $\mathbf{f}_T$ is measured:

$$\mathcal{F}(S_i, T) = \left\|\mathbf{f}_\Omega - \mathbf{f}_T\right\|_2 \tag{9}$$

- **Similarity score comparison:** Each patch is passed either to the left or right child node according to the split function that compares the score of the similarity measure $\mathcal{F}(S_i, T)$ and a randomly chosen threshold τ.

The main reason why we apply a depth check to the parts is to remove the structural perturbations, due to occlusion and clutter [15]. These perturbations most likely occur on parts extracted along depth discontinuities, such as the contours of the objects. They force a test part (occluded/cluttered) to diverge from its positive correspondence by changing its representation, $r_{\max}$ of the sphere, and the histogram coordinates consequently.

A group of candidate split functions are produced at each node by using a set of randomly assigned patches $\{T_i\}$ and thresholds $\{\tau_i\}$. The one that best optimize the offset and pose regression entropy [35] is selected as the split function. Each tree is grown by repeating this process recursively until the forest termination criteria are satisfied. When the termination conditions are met, the leaf nodes are formed, and they store votes for both the object center $\Delta\mathbf{x} = (\Delta x, \Delta y, \Delta z)$ and the object rotation $\theta = (\theta_r, \theta_p, \theta_y)$.

Depending on the part extraction approach, all parts in $\mathcal{P}$ (see Eq. (8)) can either be of the same size or of the variable size. From now on, we will refer to the forests trained on variable size parts as the *IHF-variable size*, and to the ones learnt by using fixed size parts as the *IHF-fixed size*.

3.3. 6D object pose estimation

Once we encode the discriminative information of the scale-variant HoCP features into the forest, we next demonstrate 6D pose estimation of objects considering that the learnt forest is IHF-variable size.

The proposed architecture registers objects in two steps: the *initial registration* and the *iterative pose refinement*. The *initial registration* roughly aligns the test object, and this alignment is further improved by the *iterative pose refinement*.

Consider an object that was detected by a coarse bounding box, I_b, as shown in the leftmost image of Fig. 3 (a). At an iteration instant k, the following quantities are defined:

- $\Delta\mathbf{x}^{0:k} = \{\Delta\mathbf{x}^0, \Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^k\} = \{\Delta\mathbf{x}^0, \Delta\mathbf{x}^{1:k}\}$: the history of the object position predictions.
- $\theta^{0:k} = \{\theta^0, \theta^1, \ldots, \theta^k\} = \{\theta^0, \theta^{1:k}\}$: the history of the object rotation estimations.
- $V^{1:k} = \{v^1, v^2, \ldots, v^k\}$: the history of the inputs (noise removals) applied to the test image.
- $\mathcal{M}^{0:k} = \left\{\mathcal{M}^0, \mathcal{M}^1, \ldots, \mathcal{M}^k\right\} = \{\mathcal{M}^0, \mathcal{M}^{1:k}\}$: the history of the set of the feature vectors where $\mathcal{M}^k = \{\mathbf{f}_i\}$.
- h^k: the object scale (the scale of the foreground pixels) in the unit cube at iteration k (see Eq. (2)).
- g^k: the size of the parts extracted at iteration k.

We formulate the *initial registration* as follows:

$$\left(\Delta\mathbf{x}^0, \theta^0\right) = \arg\max_{\Delta\mathbf{x}^0, \theta^0} p\left(\Delta\mathbf{x}^0, \theta^0 | I_b, \mathcal{M}^0, h^0, g^0\right). \tag{10}$$

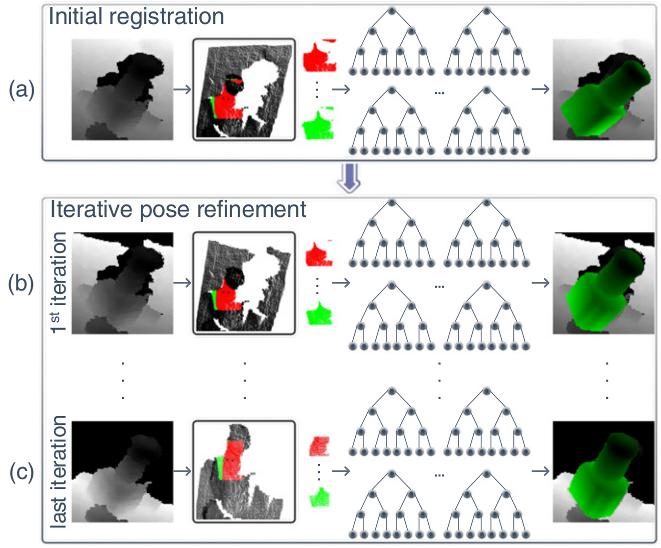

Fig. 3. Object of interest is first roughly aligned by extracting coarsest parts, and this alignment is then iteratively refined based on finer (smaller) ones.

We find the best parameters that maximize the joint posterior density of the initial object position $\Delta\mathbf{x}^0$ and the initial object rotation θ^0. The initial registration process is illustrated in Fig. 3 (a). The test image is firstly normalized into a unit cube. Unlike training, this is a "single" scale normalization that corresponds to $s_0 = 1$ (see Eq. (1)). The parts extracted from the globally represented point cloud are described with the HoCP features and are passed down all the trees. At this stage, we extract the coarsest parts from the test image, *i.e.*, the ones occupying the largest area in image pixels. We determine the effect that all patches have on the object pose by accumulating the votes stored in the leaf nodes as in Ref. [15] and approximate the initial registration given in Eq. (10).

Once the initial hypothesis $\mathbf{x}^0 = (\Delta\mathbf{x}^0, \theta^0)$ is obtained, the set of pixels that belong to the background/foreground clutter are removed from I_b according to the following criterion:

$$v^k = \begin{cases} I_b(p_j) = \mathcal{D}_{I_b}(p_j), & \gamma\psi_1 < \mathcal{D}_{I_b}(p_j) < \beta\psi_2 \\ I_b(p_j) = 0, & \textit{otherwise} \end{cases} \tag{11}$$

with

$$\gamma = \min\left(\mathcal{D}_H^k\right), \quad \beta = \max\left(\mathcal{D}_H^k\right).$$

In Eq. (11), $\mathcal{D}_{I_b}$ depicts the depth map of the test image I_b, and $\mathcal{D}_{I_b}(p_j)$ shows the depth value of the pixel p_j. $\mathcal{D}_H^k$ denotes the depth map of the hypothesis H at iteration k, that is, it is the depth map of the model superimposed onto the image at iteration k. ψ_1 and ψ_2 are the scaling coefficients. γ and β are the minimum and the maximum values of the model's depth map at estimated position (Δx) and rotation (θ). Thus, this criterion defines lower $\gamma\psi_1$ and upper $\gamma\psi_2$ bounds, and then it removes the pixels from the test image I_b if their depth values are out of these bounds.

The efficacy of v^k is illustrated in Fig. 3. In the rightmost image of Fig. 3 (a), the test image and the initial hypothesis H are superimposed. Using this hypothesis, the test image is updated by v^1 as in Eq. (11). The updated image is assigned as the input for the 1st iteration (shown in Fig. 3 (b)). It is normalized and represented globally. The object "scale" (h^1) in the unit cube is relatively increased (compare with the initial registration), and more discriminative control point descriptors $\mathbf{n}$ are computed. This is mainly because of the fact that the standard deviation of the input image decreases, since we removed foreground/background clutter. As a follow up step, we traverse the 3D bounding box in the unit cube during part extraction while the increase in the normalized object scale gives rise to extract parts whose size are smaller (finer) than the ones extracted during the initial registration. The resulted hypothesis H of the 1st iteration is shown on the right. The extraction of finer parts represented with more discriminative control point descriptors along with the noise removal process result in more accurate and confident hypothesis. This pose refinement process is iteratively performed until the maximum iteration is reached (see Fig. 3 (c)):

$$\left(\Delta\mathbf{x}^k, \theta^k\right) = \underset{\Delta\mathbf{x}^k, \theta^k}{\arg\max}\, p(\Delta\mathbf{x}^k, \theta^k \mid \mathcal{M}^{1:k}, V^{1:k}, \mathbf{x}^0, h^k, g^k). \tag{12}$$

We approximate the registration hypothesis at each iteration by using the stored information in the leaf nodes as we did in the initial registration. If we would demonstrate the 6D object pose estimation considering that the learnt forest is the IHF-fixed size, the only difference in the formulation would be the part extraction viewpoint. Instead of traversing 3D bounding box in the unit cube, we would extract the parts with a predefined size in pixels, and at an iteration instant k, g^k would remain the same as g^0. In the next section, we will compare the registration performances of the forests that are separately trained on fixed and variable size parts.

4. Experimental results

There are several publicly available datasets [15,36] to test the performances of the object registration methods. For each object type in these datasets, a set of RGB-D test images are provided with ground truth object pose parameters. We have analysed these images and have found that "Coffee Cup", "Camera", and "Shampoo" (included in the ICVL dataset [15]) are some of the best demonstrable objects to test and to compare our registration architecture with the state-of-the-art methods, since they are located in highly occluded and cluttered scenes. We further process the test images of these objects to generate a new test dataset according to the following criteria:

- Since the HoCP features are scale-variant, the depth values of the training and the test images should be close to each other up to a certain degree. In this study, we train the forests at a single depth value, f_d mm, and test with the images at the range of $[f_d \mp 50]$ mm.
- The test object instances located at the range of $[f_d \mp 50]$ mm are assumed as detected by coarse bounding boxes (see Fig. 4). The image regions included in these bounding boxes are cropped, and the new test dataset is generated.

The generated dataset includes 276 "Coffee Cup", 360 "Camera", and 200 "Shampoo" images each of which is at the range of $[750 \mp 50]$ mm, since we train the forests used in all experiments with the positive samples at $f_d = 750$ mm depth.

Our experiments are two folds: *parameter optimization* and *comparative study*. The architecture parameters have an important impact upon the registration and include the size of the parts extracted during the initial registration g^0, the IBS resolution N, the HoCP feature dimension d (the number of the bins or quantization parameter), and the iteration number. Once the best parameters are acquired, we compare the performance of our architecture with the state-of-the-art methods in the comparative study.

Both experiments use the metric proposed in Ref. [36] to determine whether a registration hypothesis is correct. This metric outputs a score ω that calculates the distance between the ground truth and estimated poses of the test object. The registration hypothesis that ensures the following inequality is considered as correct:

$$\omega \le z_\omega \Phi \tag{13}$$

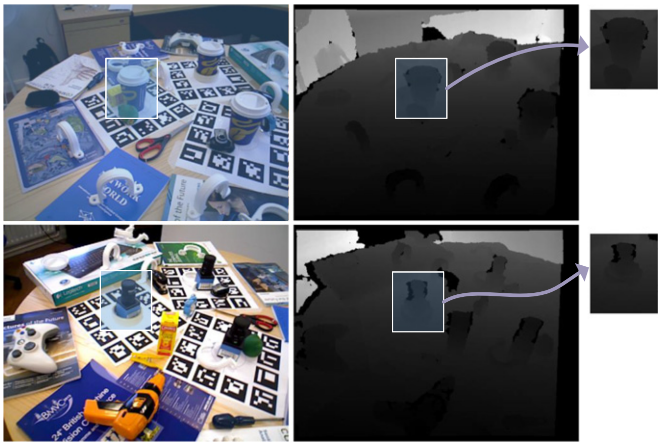

Fig. 4. Dataset generation: input of the proposed architecture is the depth image of a coarsely detected object (RGB correspondence is for better visualization).

where Φ is the diameter of the 3D model of the test object, and z_{ω} is a constant that determines the coarseness of an hypothesis that is assigned as correct. We set z_{ω} to 0.08 when we refine the parameters of the proposed architecture, and the effect of various z_{ω} values is separately examined in the comparative study.

Forest Termination Criteria. The maximum depth of a forest should be less than $log_2(S) + 1$ where S is the number of the data points used during training. Every forest in this study is trained using over 10 M parts extracted from synthetically rendered images. According to this rule, we set the maximum depth to 25 and limit the minimum number of samples at each leaf node to 15 for each tree. Once we set the tree depth, we train 3 different forests each of which is the ensemble of 3, 5, and 10 trees. We test these forests on the validation dataset which is created by selecting a subset of the original dataset used in this paper. The F1 scores produced by each forest is 0.8526, 0.8848, and 0.8867, respectively. We observe that the forests generate higher F1 scores as the number of the trees increases. However, taking into account both training effort and runtime performance, we build every forest using 3 trees. We find that the determined forest termination criteria are coherent with the ones used in Refs. [15,16,24].

4.1. Parameter optimization

The parameters of the proposed architecture are optimized only by training several IHFs based on fixed size parts. These experiments are performed on the "Coffee Cup" dataset.

4.1.1. Size of the parts extracted during initial registration

The initial registration hypothesis is used by the *iterative pose refinement* in order to improve the alignment's confidence (see Eq. (12)), and hence, g^0 is one of the important parameters that have a direct impact on the success of the complete architecture. IHF-fixed size uses the parts that are of the same size during both the *initial registration* and the *iterative pose refinement*. IHF-variable size roughly aligns the object of interest during the *initial registration* extracting coarsest parts, the ones occupying the largest area in image pixels. It iteratively refines this alignment based on the automatically extracted finer (smaller) parts, that is, it works in a size range. Thus, evaluating the performances of the *initial registrations* for different part sizes is a crucial experiment that determines not only the optimum g^0 values, but also the range of the part sizes at which the IHF-variable size works in the most feasible way. The effect of the part size is examined by setting the IBS resolution N to 80, the HoCP feature dimension d to 128 in addition to the previously defined forest parameters. We change the part size g^0: 0.20, 0.25, 0.33, 0.50, 0.66, and 0.75 times of the object bounding box, and for each, we train separate IHF-fixed size. The resultant Precision-Recall (PR) curves of the *initial registrations* are shown in Fig. 5 (a), and the corresponding F1 scores are demonstrated in Table 1. According to this figure and their corresponding F1 scores, we can choose any part size apart from the ones smaller than $\frac{1}{5}$ times of the bounding box. Considering both the computational load and the accuracy, we choose $\frac{1}{2}$ as the optimal part size for the IHF-fixed size. On the other hand, IHF-variable size uses the parts at various sizes, beginning with the coarsest (largest) ones extracted during the *initial registration*, and ending with the finest (smallest) ones extracted at the last iteration of the *iterative pose refinement*. We reanalyse the F1 scores presented in Table 1 taking into account this size variation. When we increase the part size from $\frac{1}{5}$ to $\frac{3}{4}$, the F1 score ranges between 0.6 and 0.7. Despite the significant variation of the part size, the deviation in the F1 scores is negligible. We choose $\frac{3}{4}$ as the size of the parts extracted during the *initial registration* phase of the IHF-variable size. One could suggest to train both IHF-fixed and IHF-variable size separately in order to find the best corresponding g^0 values, however, it does not make sense. Because, the positive impact of the training on variable size parts is much observed during the iterative pose refinement phase. Hence, training only on the fixed size parts and the examination through the resultant F1 scores are reasonable to infer the best g^0 for each approach.

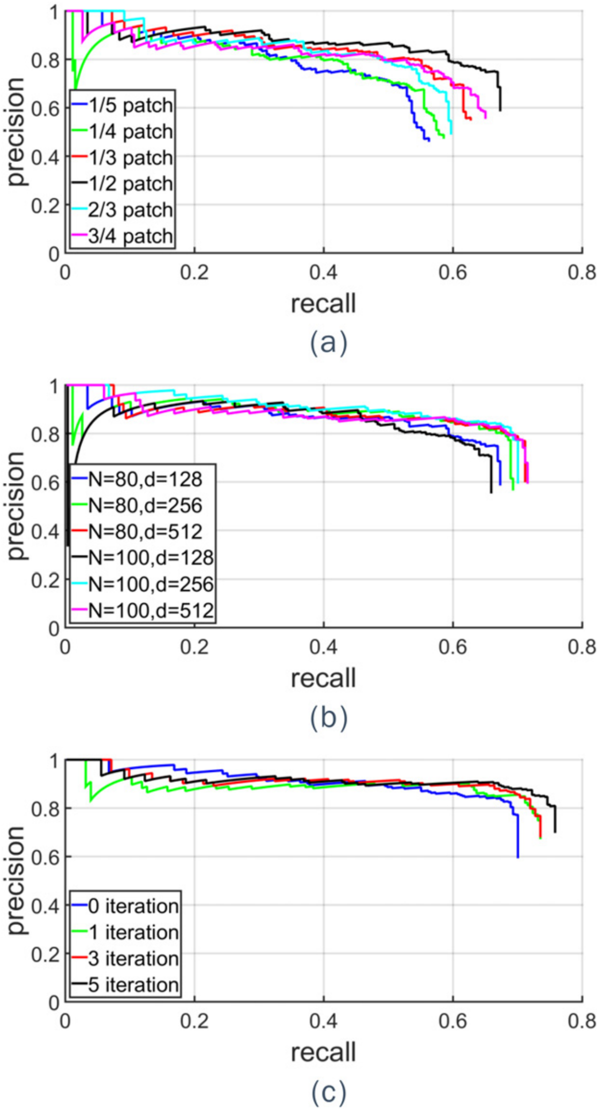

Fig. 5. Precision-Recall curves for parameter optimization: (a) compares the performance of the forests with different patch sizes. (b) illustrates the registration performance for different IBS resolutions N and feature dimensions d. (c) shows the effect of the iteration number. For the corresponding F1 scores see Table 1.

Table 1
F1 scores of the initial registrations determined for different part sizes (g), IBS resolution (N) & feature dimension (d), and number of iteration.

Part size, g	F1 score	N & d	F1 score	# iter	F1 score
$\frac{1}{5}$	0.5966	80 & 128	0.7068	0	0.7510
$\frac{1}{4}$	0.6096	80 & 256	0.7368	1	0.7742
$\frac{1}{3}$	0.6532	80 & 512	0.7425	3	0.7745
$\frac{1}{2}$	**0.7068**	100 & 128	0.6870	**5**	**0.7932**
$\frac{2}{3}$	0.6341	**100** & **256**	**0.7510**		
$\frac{3}{4}$	0.6539	100 & 512	0.7438		

4.1.2. IBS resolution and HoCP feature dimension

We next tune the IBS resolution N and the HoCP feature dimension d by setting the part size to $\frac{1}{2}$. We test the combinations of $N = 80, 100$ and $d = 128, 256, 512$, the ones that are the most applicable N–d pairs to represent $\frac{1}{2}$ patch size. The PR curves of these combinations are depicted in Fig. 5 (b), and the corresponding F1 scores are illustrated in Table 1. According to these results, we infer that the combinations composed by $d = 128$ relatively underperform, while the remaining have approximately the same F1 scores. We take into account both the memory consumption and the accuracy, and agree on the values of $N = 100$ & $d = 256$.

4.1.3. Effect of iteration

The last parameter we optimize is the iteration number. We test several IHFs-fixed size [14] each of which has $k = 0, 1, 3$, and 5 iterations, respectively. Their PR curves are shown in Fig. 5 (c). As expected, the forests that use greater number of iterations show better performances (see Table 1), since more discriminative features are extracted due to the noise removal process.

Fig. 6 demonstrates the registration results of several test objects comparing the IHFs that are trained on both the fixed size and the variable size parts. The RGB correspondences of the test objects are shown at the top, and each "iterative pose refinement" module illustrates the 1st, 3rd, and 5th iteration at its 1st, 2nd, and the 3rd columns, respectively. The sample parts shown in the "part extraction" rows are grown around the same data point. We first discuss the "image id: 650". The object is initially aligned by extracting the parts that are of size $\frac{1}{2}$ for the fixed size and $\frac{3}{4}$ for the variable size approach. By using the initial registration output, background clutter is removed from the test image. The amount of the reduction is approximately the same for both approach. After reduction, the test image is updated and is assigned as the input for the 1st iteration. IHF-fixed size keeps on extracting the parts that are of size $\frac{1}{2}$ till the last iteration, while the IHF-variable size grows finer (smaller) regions in proportion to the removed foreground/background clutter. One can infer that the variable size approach registers the object of interest slightly better than the IHF-fixed size. For the "image id: 979" and the "image id: 1494", the same regions of the test images are removed as the iteration progresses. However, the IHF-variable size demonstrate better results for both objects. This comparison also verifies that we have selected the optimum g^0 value for each approach. As the iteration progresses, we observe smooth transitions between the estimated translation and rotation parameters.

4.2. Comparative study

These experiments are conducted on the "Coffee Cup", "Camera", and "Shampoo" datasets to compare our approach with the state-of-the-art methods including the Latent-Class Hough forests (LCHF) [15] trained separately on the surface normal (LCHF-Depth (D) channel) and the color gradient + the surface normal (LCHF-RGBD channel) features. In order to make a fair comparison between methods, we train and test the LCHFs by using the authors' software. The forest parameters of all approaches are the same. For both datasets, we generate PR curves at various z_ω values, beginning with the value

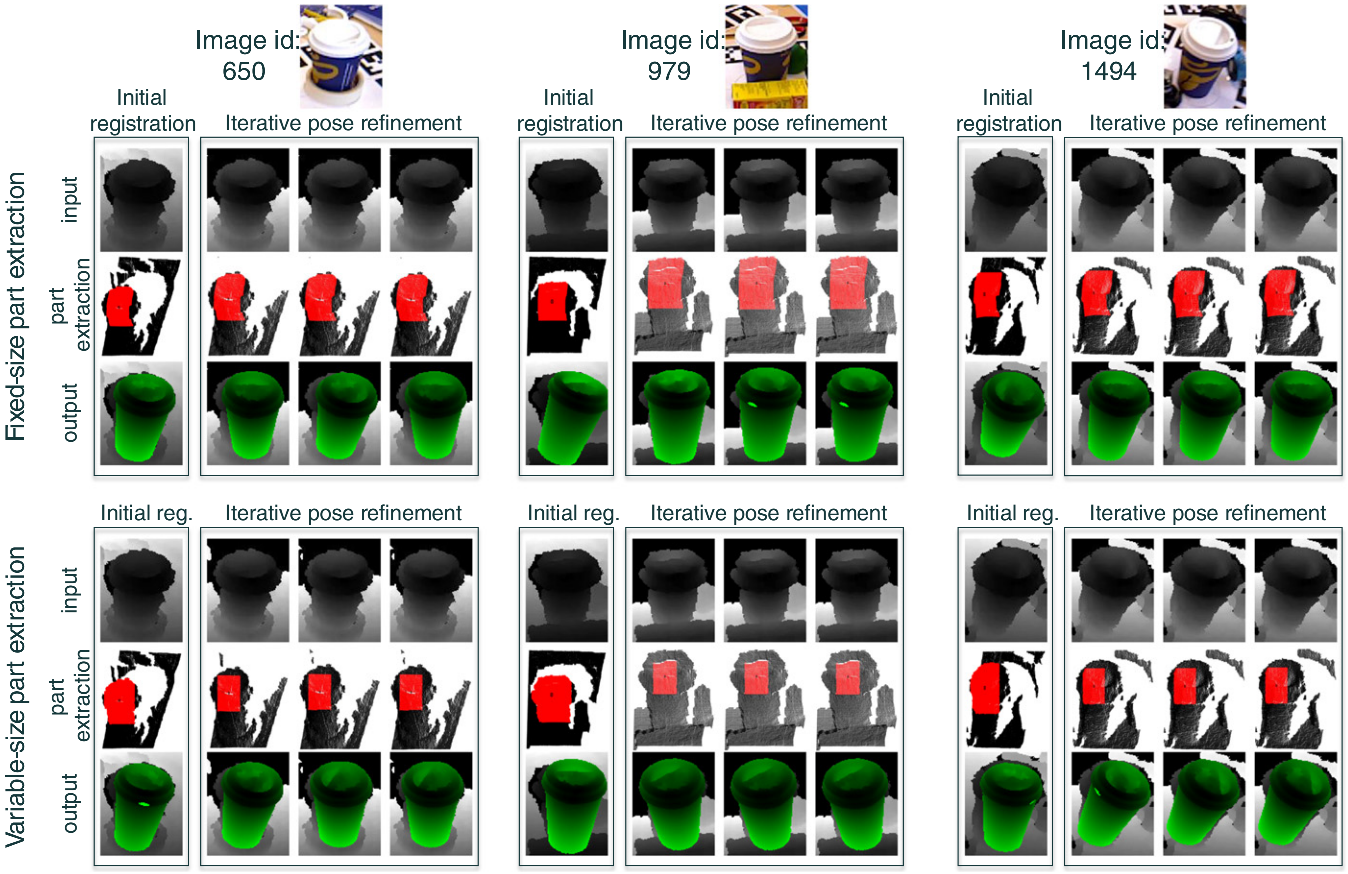
Image id: 650
Image id: 979
Image id: 1494
Initial registration
Iterative pose refinement
Fixed-size part extraction
input
part extraction
output
Initial reg.
Iterative pose refinement
Variable-size part extraction

Fig. 6. Variable size and fixed size part extraction processes are compared, and their effect on the registration results are shown. *Iterative pose refinement* modules illustrate 1st, 3rd, and the 5th iteration from left to right (RGB correspondences of the test objects are for better visualization).

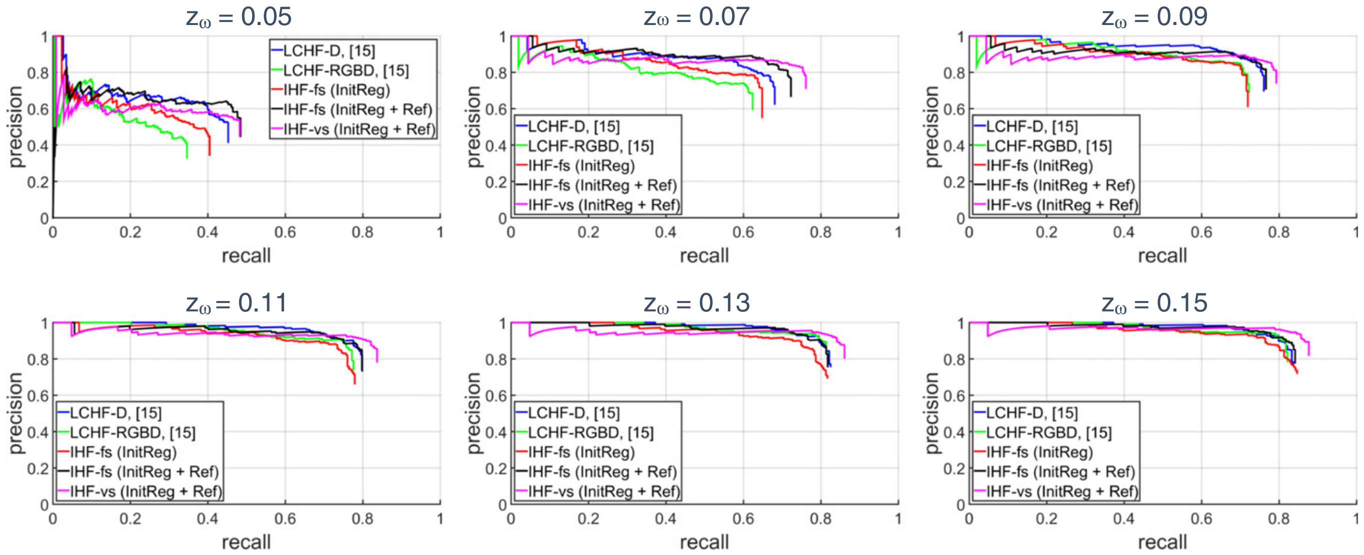

Fig. 7. Precision-Recall curves of the 'Coffee Cup' dataset: each image compares the IHF-fixed size (initial registration), the IHF-fixed size (initial registration + iterative pose refinement), and the IHF-variable size (initial registration + iterative pose refinement) with the LCHFs [15] trained separately on Depth, and on RGB-D channels. Greater values of z_ω result higher precision and recall values. F1 scores are presented in Table 2.

Table 2
F1 scores of the 'Coffee Cup' dataset are shown at different z_ω values.

z_ω value	LCHF-D [15]	LCHF-RGBD [15]	IHF-fixed size (fs) (InitReg)	IHF-fixed size (fs) (InitReg + Ref)	IHF-variable size (vs) (InitReg + Ref)
			F1 scores		
0.05	0.4867	0.3818	0.4375	**0.5297**	0.5095
0.07	0.7202	0.6639	0.6985	0.7595	**0.7891**
0.09	0.7984	0.7683	0.7633	0.7975	**0.8150**
0.11	0.8344	0.8199	0.8000	0.8312	**0.8565**
0.13	0.8548	0.8554	0.8163	0.8481	**0.8773**
0.15	0.8589	0.8595	0.8353	0.8608	**0.8940**

that strictly limits the deviations between the ground truth and the estimated pose parameters, 0.05, going up in 0.01 increments, and ending with the value that accepts relatively rough estimations as correct, 0.15.

The PR curves of the coffee cup dataset are depicted in Fig. 7 for several z_ω values, and their corresponding F1 scores are presented in Table 2. A short analysis on the images of Fig. 7 reveals that the increase in z_ω value generates higher F1 scores for each approach.

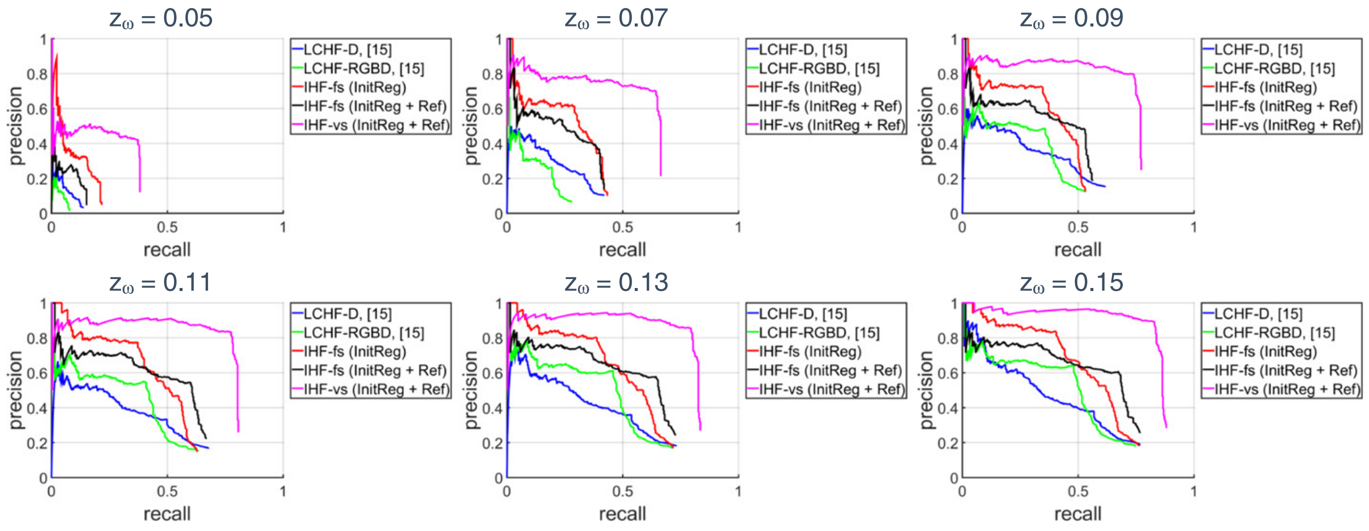

Fig. 8. Precision-Recall curves of the 'Camera' dataset: each image compares the IHF-fixed size (initial registration), the IHF-fixed size (initial registration + iterative pose refinement), and the IHF-variable size (initial registration + iterative pose refinement) with the LCHFs [15] trained separately on Depth, and on RGB-D channels. Greater values of z_ω result higher precision and recall values. F1 scores are presented in Table 3.

Table 3
F1 scores of the ‘Camera’ dataset are shown at different z_ω values.

z_ω value	LCHF-D [15]	LCHF-RGBD [15]	IHF-fixed size (fs) (InitReg)	IHF-fixed size (fs) (InitReg + Ref)	IHF-variable size (vs) (InitReg + Ref)
			F1 scores		
0.05	0.1003	0.0736	0.2071	0.1538	**0.3963**
0.07	0.2696	0.2240	0.3954	0.3878	**0.6706**
0.09	0.3723	0.4121	0.4761	0.5047	**0.7680**
0.11	0.3991	0.4674	0.5140	0.5710	**0.8035**
0.13	0.4304	0.5246	0.5494	0.6091	**0.8242**
0.15	0.4551	0.5500	0.5731	0.6388	**0.8596**

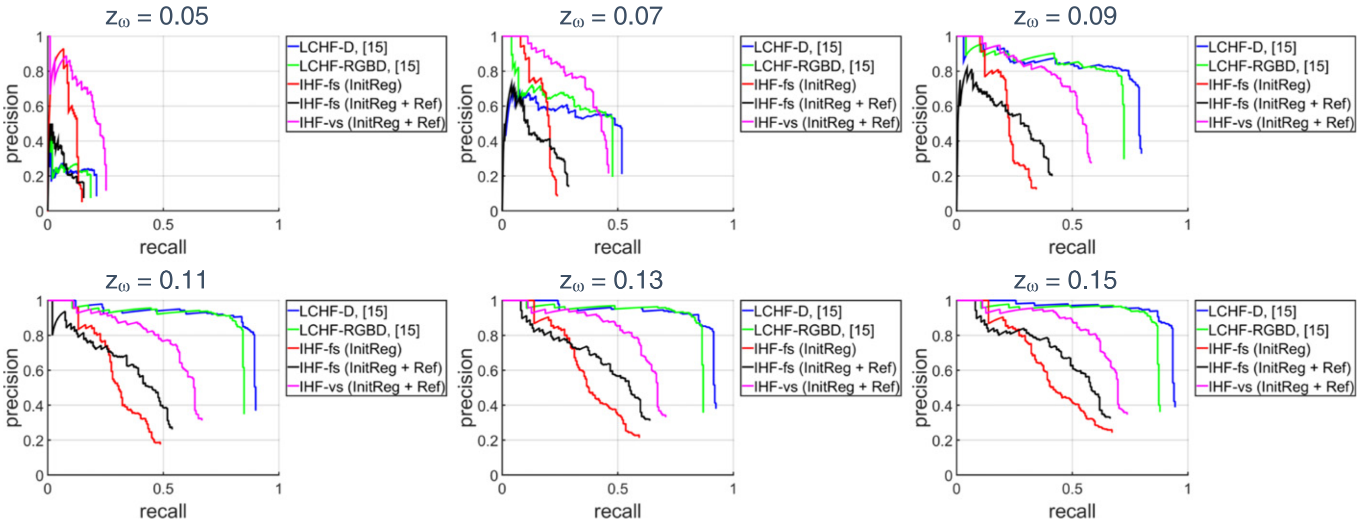

Fig. 9. Precision-Recall curves of the ‘Shampoo’ dataset: each image compares the IHF-fixed size (initial registration), the IHF-fixed size (initial registration + iterative pose refinement), and the IHF-variable size (initial registration + iterative pose refinement) with the LCHFs [15] trained separately on Depth, and on RGB-D channels. Greater values of z_ω result higher precision and recall values. F1 scores are presented in Table 4.

According to Table 2, the LCHF trained on the color gradient + surface normal features underperforms the LCHF trained merely on the surface normals. The main reason of this underperformance is the distortion along the object borders arising from the occlusion and the clutter, that is, the distortion of the color gradient information in the test process. The performance of the initial registration of the IHF-fixed size is similar to the ones LCHFs have. When this initial registration (see the 3rd column of Table 2) acquired from the IHF-fixed size is iteratively refined, more accurate registrations are resulted (see the 4th column of Table 2). Because, the *iterative pose*

Table 4
F1 scores of the ‘Shampoo’ dataset are shown at different z_ω values.

z_ω value	LCHF-D [15]	LCHF-RGBD [15]	IHF-fixed size (fs) (InitReg)	IHF-fixed size (fs) (InitReg + Ref)	IHF-variable size (vs) (InitReg + Ref)
			F1 scores		
0.05	0.2168	0.197	0.2051	0.1597	**0.3125**
0.07	**0.5094**	0.5067	0.2983	0.2811	0.50
0.09	**0.7728**	0.7439	0.3306	0.3819	0.5862
0.11	**0.8720**	0.8463	0.3878	0.4785	0.6379
0.13	**0.8825**	0.8670	0.4437	0.5436	0.6724
0.15	**0.9033**	0.8723	0.4713	0.5692	0.6898

Table 5
F1 scores of the ‘Coffee Cup’, ‘Camera’, and the ‘Shampoo’ datasets are shown. These scores are the average of all z_ω that take each value in the range of [0.05–0.15].

Object	LCHF-D [15]	LCHF-RGBD [15]	IHF-fixed size (fs) (InitReg)	IHF-fixed size (fs) (InitReg + Ref)	IHF-variable size (vs) (InitReg + Ref)
			F1 scores		
Coffee Cup	0.7744	0.7410	0.7419	0.7834	**0.8026**
Camera	0.3441	0.3850	0.4631	0.4881	**0.7323**
Shampoo	**0.7067**	0.6870	0.3577	0.4067	0.5747

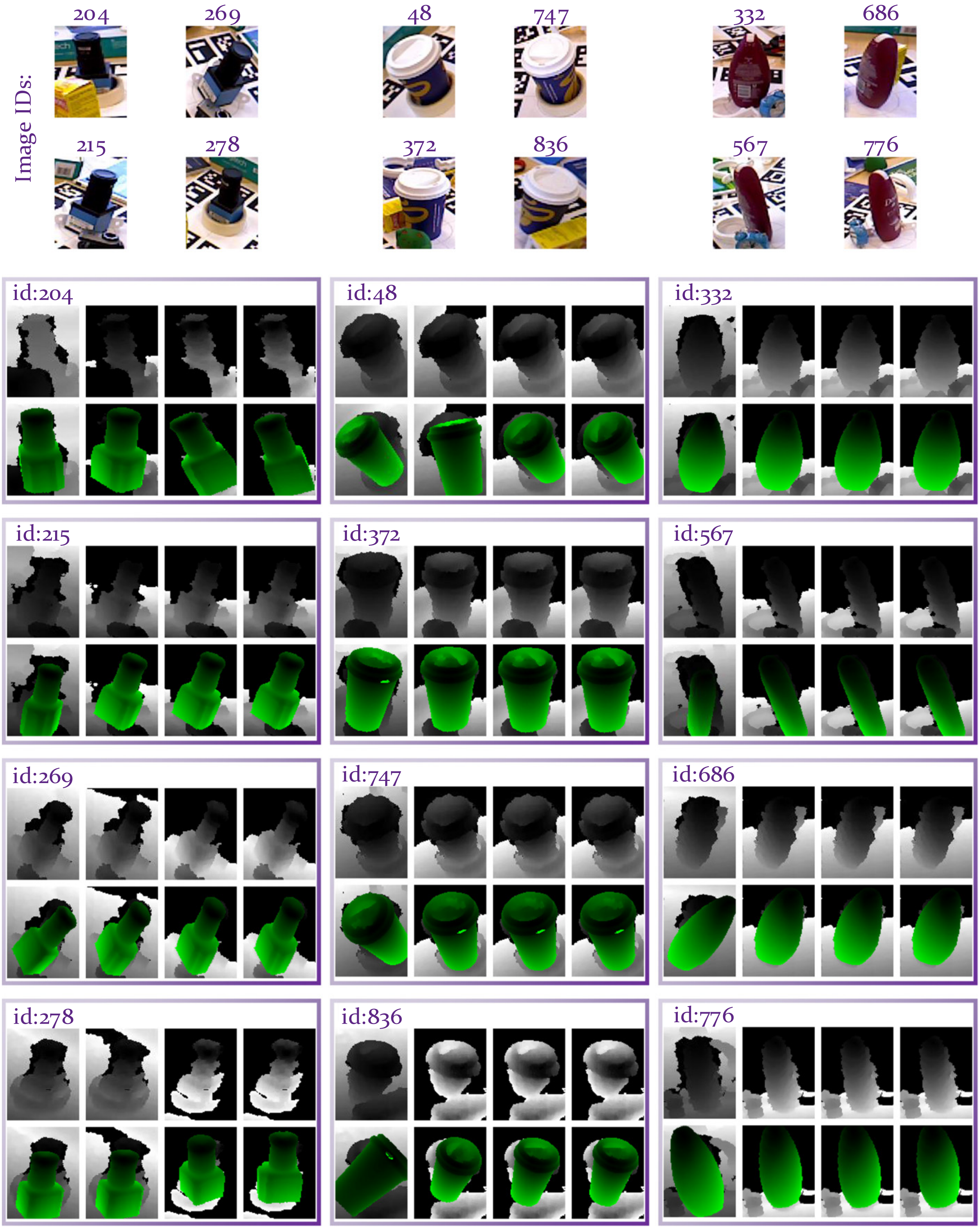

Fig. 10. Some qualitative results. For each octonary: 1st column illustrates the test image and the initial hypothesis (*initial registration*), and the remaining columns demonstrate 1st, 3rd, and the 5th iterations (*iterative pose refinement*). Test images are updated by removing background/foreground clutter.

refinement module of the IHF-fixed size reduces the amount of the noise in the test depth maps removing foreground/background clutter. This removal process also enables IHF-fixed size to compute more discriminative control points for better shape representation. The IHF-variable size with initial registration + iterative pose refinement outperforms other approaches. The main reason of this high performance is the utilization of the parts that are different in size. The cascaded representation of the locality increases the robustness across clutter, occlusion, missing depth pixels, and/or similar-looking distractors. The object of interest is first roughly aligned by extracting the coarsest parts. It is highly possible that these initially extracted parts include the portions belonging to the background/foreground clutter, since they are the coarsest and are close to a holistic template. Despite the fact that we apply a depth check in order to get rid of those noise during testing, it is highly naive. On the other hand, the proposed framework can get rid of those structural perturbations by growing smaller regions as the iteration progresses. Apart from that, the control point descriptors computed at later iterations allows the complete framework to represent the shapes in a more discriminative manner.

Regarding the camera dataset, we show its PR curves in Fig. 8 for several z_ω values and the corresponding F1 scores in Table 3. The approaches under comparison perform worse on this dataset with respect to the coffee cup. Unlike the results obtained from the coffee cup dataset, we see the positive impact of the color gradients when they are utilized along with the surface normals at most of the z_ω values (compare the 1st and the 2nd columns of Table 3). IHF-fixed size registers objects more accurate than any versions of the LCHF thanks to the utilization of the discriminative information embedded into the scale-variant HoCP features and the iterative refinement of the test depth maps. IHF-variable size significantly outperforms other approaches demonstrating the importance of the simultaneous utilization of variable size parts. The HoCP representations of the cascaded regions grown around the same data points allow the algorithm to be aware of occlusion, clutter, and missing depth values. More confident registrations are hypothesized as the iteration progresses based on more discriminative representations of the smaller parts. The registration performance of the proposed architecture is shown in Fig. 9 for the shampoo object, and the corresponding F1 scores are demonstrated in Table 4 for varying z_ω values. In cases of registering at lower z_ω values, our approach shows better performance than the LCHFs, however, when we accept relatively rough estimations as correct, *i.e.*, higher z_ωs, our approach underperforms.

Since we address the registration problem rather than individual object detection or pose estimation, we integrate the effect of the different error ratios into our comparisons. We average the F1 scores that are computed at each z_ω in the range of [0.05–0.15] and report in Table 5. Fig. 10 illustrates several accurate registrations hypothesized by the proposed architecture on the camera, coffee cup, and shampoo objects. We further evaluate the performance of the globally optimized ICP algorithm proposed in Ref. [21] on our test dataset. We use the software kindly provided by the authors and set the default parameters. While accurate registration results are obtained on the clean dataset, it diverges in the case of highly occluded and cluttered point clouds (see Fig. 11).

Run-time Performance. The proposed architecture, Iterative Hough Forest, is capable of achieving real-time execution. It registers the 6D pose of an object processing 34 fps on a i7-3820 CPU @ 3.60 GHz, 16.0 GB computer. Histogram of Control Points (HoCP) features are derived from the source codes of recently proposed Implicit B-Splines and are integrated into the Iterative Hough Forest architecture. Despite the fact that the derivation procedure of HoCP features alleviates the run-time performance, it can be accelerated by GPU processing. Utilization of APIs like OpenMP allows us to parallelise the algorithm for multi-thread, and thus enabling the complete architecture to simultaneously process multiple parts.

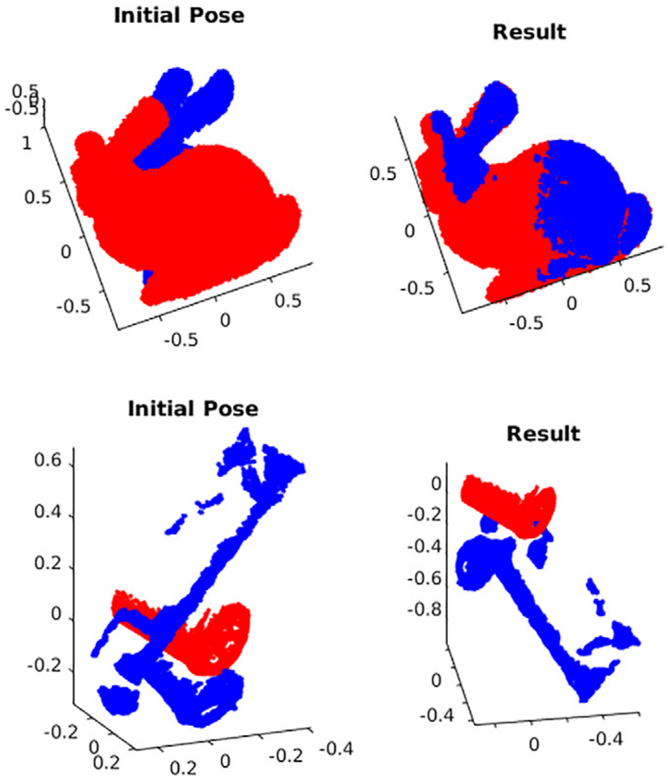

Fig. 11. Performance of the Go-ICP [21] algorithm on our dataset: despite it achieves good registration on the clean point cloud, it diverges on our dataset.

5. Discussion and conclusion

In this study, we have proposed a novel architecture, *Iterative Hough Forest with Histogram of Control Points*, addressing 6D object registration rather than individually estimating either the object's location in a 2D/3D bounding box or the object's orientation (roll, pitch, yaw). Any off-the-shelf detector can accurately provide a coarse 2D or 3D bounding box for the object of interest. Various object orientation predictors are also available, however, they depend on clearly segmented target objects. Our architecture fundamentally targets to eliminate the shortcomings of these individual detectors and orientation predictors estimating occluded and cluttered objects' 6D pose given a candidate 2D bounding box. Our IHF is learnt using parts extracted only from the positive samples. These parts are represented with scale-variant HoCP features, which we derive from recently introduced Implicit B-Splines (IBS).

At test time, we apply two different strategies regarding the parts used to train the forest. The first strategy we apply roughly aligns the object and iteratively refines this alignment based on more discriminative control point descriptors that are computed due to the elimination of background/foreground clutter. The part size is fixed and is empirically predefined. On the other hand, the predefined part size might not be generalizable enough across different objects, degrading the registration performance of the proposed study on one object while working well on another one. Besides, discriminative information encoded into small sized parts might not be fully exploited by larger parts, most particularly when the object representation is scale-variant. Inspired by these observations, we use variable size parts in the second strategy. An automatic variable size part extraction framework iteratively refines the object's initial pose that is roughly aligned due to the extraction of coarsest parts, the ones occupying the largest area in image pixels. The iterative refinement is accomplished based on finer (smaller) parts that are represented with more discriminative control point descriptors by using our *Iterative Hough Forest*. The experimental results report that our approach show better registration performance than the state-of-the-art methods.

Acknowledgements

Caner Sahin is funded by the Turkish Ministry of National Education.

References

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

[2] M. Rouhani, A.D. Sappa, Correspondence free registration through a point-to-model distance minimization, ICCV, 2011.

[3] M. Unel, O. Soldea, E. Ozgur, A. Bassa, 3D object recognition using invariants of 2D projection curves, Pattern. Anal. Applic. (2010)

[4] R.B. Rusu, N. Blodow, M. Beetz, Fast point feature histograms (FPFH) for 3D registration, ICRA, 2009. pp. 3212–3217.

[5] E. Kim, G. Medioni, 3D object recognition in range images using visibility context, IROS, 2011. pp. 3800–3807.

[6] U. Bonde, V. Badrinarayanan, R. Cipolla, Robust instance recognition in presence of occlusion and clutter, ECCV, 2014.

[7] B. Juttler, A. Felis, Least-squares fitting of algebraic spline surfaces, Adv. Comput. Math. (2002) 135–152.

[8] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, C. Rother, Learning 6D object pose estimation using 3D object coordinates, ECCV, 2014. pp. 536–551.

[9] X. Zhao, T.-K. Kim, W. Luo, Unified face analysis by iterative multi-output random forests, CVPR, 2014. pp. 1765–1772.

[10] J. Novatnack, K. Nishino, Scale-dependent/invariant local 3D shape descriptors for fully automatic registration of multiple sets of range images, ECCV, 2008. pp. 440–453.

[11] P. Bariya, K. Nishino, Scale-hierarchical 3D object recognition in cluttered scenes, CVPR, 2010. pp. 1657–1664.

[12] C. Zach, A.P. Sanchez, M.T. Pham, A dynamic programming approach for fast and robust object pose recognition from range images, CVPR, 2015. pp. 196–203.

[13] M. Rouhani, A.D. Sappa, E. Boyer, Implicit B-spline surface reconstruction, IEEE Trans. Image Process. 24 (1) (2015) 22–32.

[14] C. Sahin, R. Kouskouridas, T.K. Kim, Iterative Hough Forest with Histogram of Control Points for 6 DoF Object Registration from Depth Images, IROS, 2016.

[15] R. Kouskouridas, A. Tejani, A. Doumanoglou, D. Tang, T.K. Kim, Latent–class hough forests for 6 dof object pose estimation, 2016.arXiv preprint arXiv:1602.01464.

[16] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, T.-K. Kim, Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd, CVPR, 2016.

[17] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, A. Blake, Real-time human pose recognition in parts from single depth images, Commun. ACM (2013)

[18] M. Budiu, J. Shotton, D. Murray, M. Finocchio, Parallelizing the training of the Kinect body parts labeling algorithm, NIPS, 2011.

[19] G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real time 3D face analysis, Int. J. Comput. Vis. 101 (3). (2013)

[20] G. Rogez, M. Khademi, J.S. Supancic, III, J.M.M. Montiel, D. Ramanan, 3D hand pose detection in egocentric RGB-D images, Computer Vision-ECCV 2014 Workshops, 2014.

[21] J. Yang, L. Hongdong, J. Yunde, Go-ICP: Solving 3D registration efficiently and globally optimally, ICCV, 2013. pp. 1457–1464.

[22] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, K. Ikeuchi, 6-DOF pose estimation from single ultrasound image using 3D IP models, CVPR Workshops, 2008. pp. 1–8.

[23] B. Zheng, R. Ishikawa, J. Takamatsu, T. Oishi, K. Ikeuchi, A coarse-to-fine IP–driven registration for pose estimation from single ultrasound image, Comput. Vis. Image Underst. 117 (12) (2013) 1647–1658.

[24] A. Tejani, D. Tang, R. Kouskouridas, T-K. Kim, Latent-class Hough forests for 3D object detection and pose estimation, ECCV, 2014. pp. 462–477.

[25] K. Pauwels, D. Kragic, Simtrack: a simulation-based framework for scalable real-time object pose detection and tracking, IROS, 2015.

[26] K. Pauwels, D. Kragic, Integrated on-line robot-camera calibration and object pose estimation, ICRA, 2016.

[27] C. Papazov, D. Burschka, An efficient ransac for 3d object recognition in noisy and occluded scenes, Asian Conference on Computer Vision, 2010.

[28] M. Uenohara, T. Kanade, Vision-based object registration for real-time image overlay, Computer Vision, Virtual Reality and Robotics in Medicine, 1995.

[29] R.B. Rusu, N. Blodow, Z.C. Marton, M. Beetz, Aligning point cloud views using persistent feature histograms, IROS, 2008.

[30] B. Drost, M. Ulrich, N. Navab, S. Ilic, Model globally, match locally: efficient and robust 3D object recognition, CVPR, 2010.

[31] C. Choi, H.I. Christensen, 3D pose estimation of daily objects using an RGB-D camera, IROS, 2012. pp. 3342–3349.

[32] K. Hara, R. Chellappa, Growing regression forests by classification: applications to object pose estimation, Computer Vision - ECCV, 2014. pp. 552–567.

[33] C.R. Cabrera, R.L. Sastre, T. Tuytelaars, All together now: simultaneous object detection and continuous pose estimation using a hough forest with probabilistic locally enhanced voting, Proceedings BMVC, 2014. pp. 1–12.

[34] D.J. Kroon, Segmentation of the Mandibular Canal in Cone-Beam CT Data,(thesis). 2011, 69.

[35] G. Fanelli, J. Gall, L.V. Gool, Real time head pose estimation with random regression forests, CVPR, 2011. pp. 617–624.

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

[37] M.M. Blane, Z. Lei, H. Civi, D.B. Cooper, The 3L algorithm for fitting implicit polynomial curves and surfaces to data, PAMI, 2000.