Robustifying Correspondence Based 6D Object Pose Estimation

Antti Hietanen[1], Jussi Halme[2], Anders Glent Buch[3], Jyrki Latokartano[2] and J.-K. Kämäräinen[1]

***Abstract*— We propose two methods to robustify point correspondence based 6D object pose estimation. The first method, curvature filtering, is based on the assumption that low curvature regions provide false matches, and removing points in these regions improves robustness. The second method, region pruning, is more general by making no assumptions about local surface properties. Our region pruning segments a model point cloud into cluster regions and searches good region combinations using a validation set. The robustifying methods are general and can be used with any correspondence based method. For the experiments, we evaluated three correspondence selection methods, Geometric Consistency (GC) [1], Hough Grouping (HG) [2] and Search of Inliers (SI) [3] and report systematic improvements for their robustified versions with two distinct datasets.**

I. INTRODUCTION

6-DoF object pose estimation from 3D data (point cloud/colored point cloud) is an active yet challenging problem in robotics, e.g., for vision based manipulation [4], [5]. A popular approach is to find correspondence point between the captured scene and stored models which are both represented by 3D point clouds [1], [2], [3]. It turns out that in practice these methods can easily fail if an object is observed from a difficult view point, or if other objects occlude a large part of the object. In this work, we assume that not all points can be treated with equal importance, e.g. large solid areas and sharp object corners, but robustness can be improved by selecting a good sub-set of points that guarantees more robust pose estimation (see Fig. 1). Our research problem is to identify a good sub-set of the model points.

In this work, we assume availability of a set of validation images that represent typical scene captures and propose two methods to robustify correspondence based pose estimation. The first method is based on our findings of failures cases in automated heavy outdoor robot tool changing, where many tools contain large planar areas that provide false correspondences, consequently leading to poor pose estimates. To remove planar areas we exploit computational curvature estimates and filter out low curvature regions. The second method does not make assumptions about the shape properties around surface points, but divides the model point cloud into local regions by clustering. Then a randomized procedure is executed to find a good combination of these regions. Our experiments with two distinct datasets verify that our robustifying procedures consistently achieve better accuracy and decrease the number of wrong or inaccurate pose estimates.

[1]Signal Processing Laboratory, [2]Laboratory of Mechanical Engineering and Industrial Systems, Tampere University of Technology
[3] SDU Robotics, University of Southern Denmark

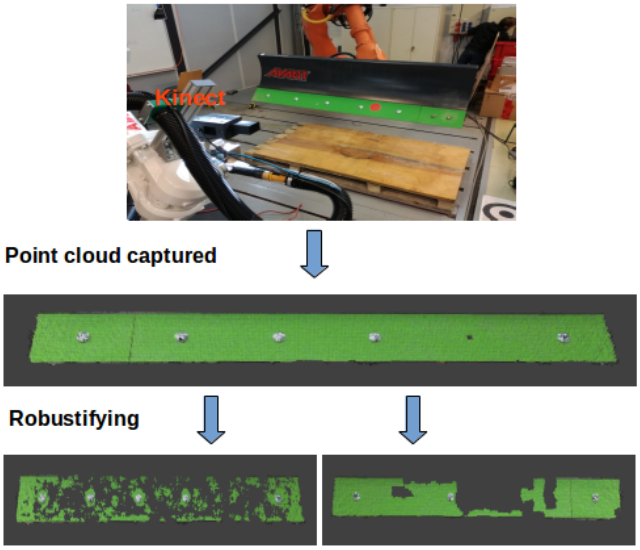

Fig. 1. Robot setup of a fixed RGB-D sensor (Kinect) and a snow blade attached to a manipulator (top). 3D colored point cloud of the green part of the blade containing attaching bolts (middle). Robust sub-sets of the points by Curvature Filtering (bottom left) and Region Pruning (bottom right).

II. RELATED WORK

In the following we focus on 3D-to-3D pose estimation methods and, in particular, methods that store object models and capture test scenes as 3D point clouds. Many proposed methods are developed for object recognition, but since they are also suitable for pose estimation we include them here.

3D Object Recognition – Local region detectors and descriptors have been successful in 2D vision problems and have therefore been extended to 3D surfaces and point clouds, e.g., 3D SURF [6], 3D HOG and DoG [7]. A recent survey and evalutaions of the detectors and descriptors can be found in Guo et al. [8], [9]. The descriptors provide reliable object recognition, but for accurate pose estimation the best result can be achieved by registering model points to corresponding scene points. For this registration process, obtaining correct point correspondences becomes a crucial task.

Point cloud based methods have been proposed by Papazov and Burschka [10] and Drost et al. [11]. Papazov and Burschka utilise a random sample consensus (RANSAC) matching and Drost et al. use Hough-like voting. More recently, a mesh-based local descriptor was used for achieving good results for a series of 3D recognition tasks [12]. In another work [13], local descriptors were integrated into a sophisticated global hypothesis verification framework. Re-

cently, attention has also been paid to 3D point selection [14], [3], and in this work we adopt three recent methods with distinctive approaches: Geometric Consistency (GC) [1], Hough Grouping (HG) [2] and combined local and global Search of Inliers (SI) [3]. We describe these three selected methods in more details in Section III.

6D Pose Estimation – A 3D point cloud is the typical modality used for object pose estimation in robotics [4], [5]. It is noteworthy that the best RGB-D SLAM methods are also based on point clouds [15], [16], but in their case the previous frame provides a good initial estimate of the pose and can be refined by dense gradient or Iterative Closest Point (ICP) matching. In the case of object pose estimation, the initial estimate for the ICP must be provided by robust correspondence-based estimation (RANSAC, Hough Voting) [10], [11] that cope with occlusion and clutter. In the most recent works, more complex correspondence search algorithms have been proposed that utilize the neighborhoods of the surface points [4], [3], [5].

Contributions – We propose two methods to robustify correspondence based 6D object pose estimation

- *Curvature Filtering* that removes points within low curvature areas of model point clouds and
- *Region Pruning* that processes the model point cloud as local regions for which a good combination is sought using a trial-and-error procedure with validation data.

Effectiveness of the proposed robustifying methods is verified using two distinct datasets where the first one is used in many related works and the second one is generated by ourselves using tools for our outdoor robot for land moving and snow clearance. Our dataset, ground truth and code will be made publicly available.

III. 3D CORRESPONDENCE METHODS

For baseline methods in this work, we selected tree recent methods available: *Geometric Consistency* (GC) [1], *Hough Grouping* (HG) [2] and *Search of Inliers* (SI) [3]. In our experiments, for GC and HG we use the available implementations in the Point Cloud Library [17] and for SI we use the implementation by the original authors. In the following, we briefly explain these methods and their most important parameters.

All three methods start processing initial correspondence candidates and refine the model using various correspondence verification procedures that remove poor matches between two point clouds (a model and query scene). The initial correspondence are created by using the SHOT features [18] that performed well in the recent comparison [9] and provide good balance between computational complexity and performance. Fast nearest neighbor search for initial correspondence is done using the FLANN library (Fast Library for Approximate Nearest Neighbors [19]).

A. Search of Inliers (SI) [3]

The SI method is based on two consequtive processing stages, *local voting* and *global voting*. At the end the votes are accumulated to form a quantitative indicator (number of votes) to denote the degree of trust for each correspondence.

The initial corresponses are refined by Lowe's test of ratio between the best and the second best matches with the threshold set to $\geq \tau_{Lowe} = 0.2$. The ratio test refines the original set of correspondences $\mathcal{C}$ to a sub-set $\mathcal{C}_{Lowe}$ such that $|\mathcal{C}_{Lowe}| \leq |\mathcal{C}|$. The first voting step performs local voting, where locally selected correspondence pairs are selected from the model and a scene, and the score is computed using their pair-wise similarity score $s_{local}(\vec{p})$ for each 3D point $\vec{p}$. The second global voting stage samples point correspondences, estimates a transformation and gives a global score to the points correctly aligned outside the estimation point set: $s_G(\vec{p})$. The final score $s(\vec{p})$ is computed by integrating both local and global scores, and an adaptive threshold between inliers and outliers is automatically found by Otsu's bimodal distribution thresholding. The inlier set is used for final pose estimation. The two following methods alter the input correspondence set $\mathcal{C}$ by selecting only robust sub-regions such that $\mathcal{C}' \subseteq \mathcal{C}$.

B. Geometric Consistency (GC) [1]

The geometric consistency (GC) incrementally builds regions (clusters) of correspondence that are geometrically consistent. In our work we will use the implementation by Chen and Bhanu [1] which is a modified version of the original method by Johnson and Hebert [20], [21] and available in Point Cloud Library [17].

The method clusters correspondence pairs of similar accuracy by imposing an absolute pairwise distance constrained equal to the Euclidean distance between the feature points:

$$\left| \|p_{i,m} - p_{j,m}\|^2 - \|p_{i,s} - p_{j,s}\|^2 \right| < \tau_\epsilon, \qquad (1)$$

where $p_{:,m}$:s are model points and $p_{:,s}$ captured scene points. We initialize the algorithm with $|\mathcal{C}|$ clusters each having a seed correspondence. Then for each cluster we search the set of correspondence whose pairwise distances are less than a predefined threshold value τ_ϵ. The clustered correspondences are marked as visited, and the seed growing repeats until all correspondences have been visited. As the final step, RANSAC can be computed to refine each cluster set [22].

In principle, GC can return more than one correspondence cluster and for pose estimation we have to rank them using a suitable metric. In our evaluations, we found that the cluster which has the largest number of correspondence lead to best average pose estimation.

C. Hough Grouping (HG) [2]

The key idea of the Hough 3D correspondence grouping (HG) [2] is to iteratively cast votes for object location and pose bins in the Hough parameter space and at the end of the process the highest accumulated bins represent the most likely pose candidates and correspondence contributed to the bins are accepted.

The method requires an unique reference point in the model, typically the model centroid, and each bin represents

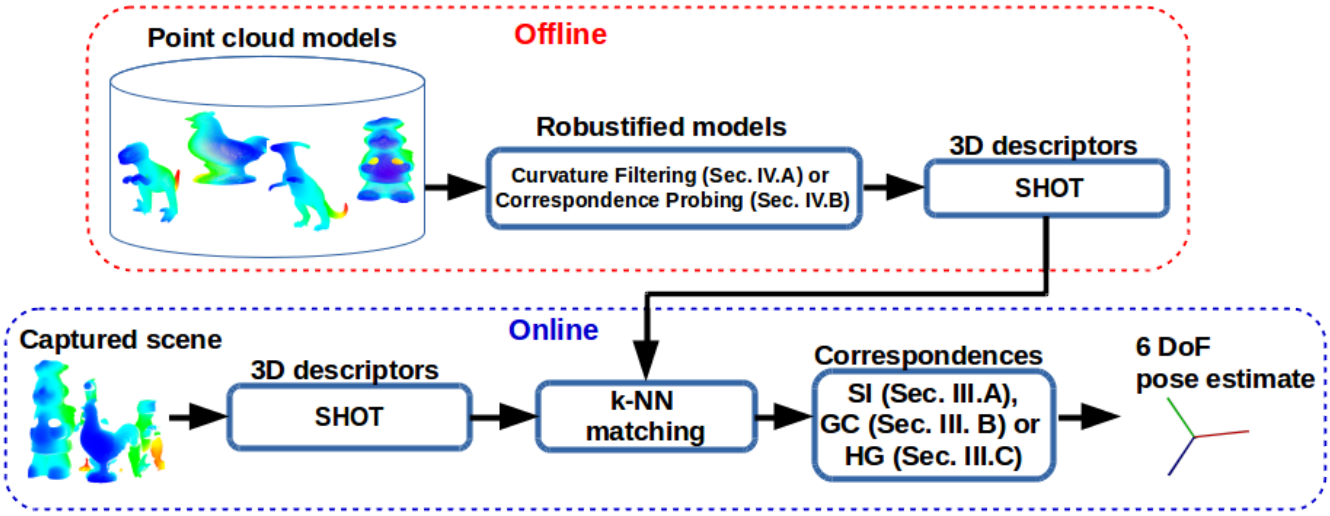

Fig. 2. Used model for correspondence based 6D object pose estimation.

a single pose instance candidate. Therefore correct correspondence vote a same bin which gets quickly accumulated. To make correspondence points invariant to rotation and translation between the model and scene, every point is associated with local Reference Frame (RF) [18]. In the voting stage each correspondence between a capture scene and a model cast a single vote to a single or multiple bins in the 3D translation Hough accumulator space and pose is stored in the local reference frame. Finally, correspondence contributing bins having votes more than a set threshold which is adaptively set as it depends on the number of available points and the most important parameter is the Hough accumulator bin size. In addition, in [23] two different weighting methods for the voters were proposed, but in our experiments they did not improve performance.

IV. ROBUSTIFYING METHODS

In the following, we explain the two proposed methods to robustify pose estimation with more reliable correspondences.

A. Curvature Filtering

Curvature is surface property that may affect to 3D object detection, tracking and pose estimation. For example, tracking does not converge on large planar areas where matches can be equally good everywhere. On the other hand, sudden surface normal changes in high curvature areas, such as corners and edges, provide strong cues for tracking and pose estimation. There also exist a number of studies on perceptual experiments that demonstrate the importance of curvature in the human visual system [24], [25].

We compute the curvature value of a point as the *surface variation* defined in [26]:

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \ , \tag{2}$$

where the λ:s are the eigenvalues of the corresponding eigen vectors $\vec{v}_i$ (λ_0 is the largest) of the covariance matrix C:

$$C = \frac{1}{N_{curv}} \sum_{i=1}^{N_{curv}} (\vec{p}_i - \vec{\mu}) \cdot (\vec{p}_i - \vec{\mu})^T \ , \tag{3}$$

where N_{curv} is the number of points considered in the neighbourhood of $\vec{p}_i$, and $\vec{\mu}$ represents the 3D centroid (mean) of the points.

The number of neighbours is a free parameter of the method but it should be set large enough to tolerate noise. The second free parameter of the method is the actual curvature threshold, which we denote τ_{curv}. Points having lower curvature value than τ_{curv} will be removed from the point cloud. Figure 1 illustrates the model after curvature based selection.

B. Region Pruning

First we segment the model point cloud to supervoxels (Figure 3) using the algorithm described in [27]. The grouping starts by dividing the 3D space of the model into a voxelized grid with resolution R_{seed}. Expansion of the supervoxels is then done by local k-means clustering controlled by the feature distance measure:

$$D = \sqrt{w_c D_c^2 + \frac{w_s D_s^2}{3R_{seed}} + w_n D_n^2}, \tag{4}$$

where D_s is the spatial distance by the seeding resolution, D_c is the Euclidean color distance in normalized RGB space, and the normal distance D_n measures the angle between surface normal vectors. Weights w_c, w_s and w_n control the influence of color, spatial and normal features respectively. Finally we end up with n supervoxels each having a central point $p_n(x, y, z)$.

Now the main task is to "prune" the generated regions (see Figure 3) and select a good sub-set that provides the best performance using a validation set. It is apparent that for a

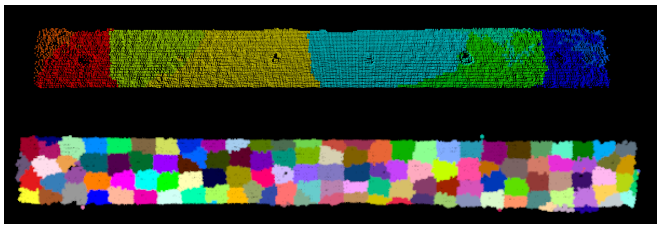

Fig. 3. Top: snow blade point cloud divided to 10 supervoxels; Bottom: 128 supervoxels.

large k the exhaustive search of the best combination quickly becomes computationally intractable. Exhaustive search with k regions requires

$$\frac{n!}{(n-k)!k!} \tag{5}$$

experiments with all validation set scenes. The total number of tests is

$$\sum_{k=1}^{n} \frac{n!}{(n-k)!k!} \tag{6}$$

combinations which is infeasible except for a very small n (10-15). The solution used in this work is to perform random pruning of 1-10% of the regions with a fixed R_{seed} and k. This procedure is experimentally evaluated in Section V-D.

V. EXPERIMENTS

In this section, we report results for the experiments with the three correspondence methods (GC, HG and SI - see Section III) combined with the proposed robustifying methods in Section IV: *Curvature Filtering* (curv) and *Region Pruning* (regp). We also provide results for the correspondence methods with validation set optimized parameters (GC-opt, HG-opt and SI-opt) and using RANSAC as the standard robustifying procedure.

A. Data and Performance Measure

Laser Scanner Dataset – As a benchmark to compare to other works we use the Laser Scanner Dataset[1], which has been used to evaluate 3D object recognition and 6D pose estimation methods [28], [2], [3]. The dataset contains four difficult models: *T-rex*, *Chicken*, *Parasaurolophus* and *Cheff*. Objects are occluded in the test scenes and in average $71\% - 77\%$ of the points are missing. The dataset contains also ground truth transformation matrices to align each model to the test scenes.

Outdoor Robot Tool Dataset – The tool dataset was collected using our robot setup (Figure 4) where a ABB IRB6640 manipulator was used to systematically move the selected tools (a snow blade and a container box) to different locations and pose angles. Each configuration was captured by a Kinect v2 sensor. One of the views was selected as the canonical view and for all other views we provide $4x4$ homogeneous transformation matrices that align them to the canonical view. The groundtruth transformation matrices were generated by manually selecting corresponding points in all point clouds and using the direct linear method for initial estimation and the Iterative Closest Point (ICP) algorithm to refin estimation [29]. RGB-D images, ground truth transformations and our evaluation code will be made publicly available.

[1]http://staffhome.ecm.uwa.edu.au/ 00053650/recognition.html

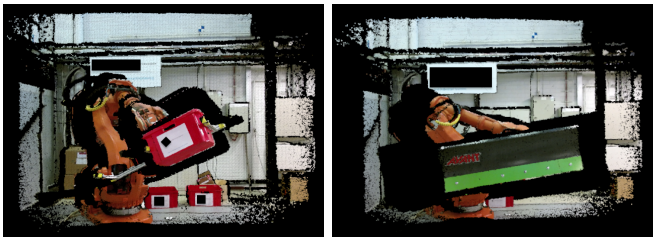

Fig. 4. The two outdoor robot tools used in our dataset: a container box (left) and a snow blade (180 kg). An ABB manipulator was used to systematically change the view point and RGB-D data was recorded using a Kinect V2 on a tripod.

Error measure – We adopt the error measure proposed in [3], which measures the mean squared error (MSE) of all model points $\vec{p} \in M$ using the ground truth transformation $\mathcal{T}_{gt}$ and the estimated transformation $\hat{\mathcal{T}}$:

$$\epsilon_{MSE} = \frac{1}{|M|} \sum_{\vec{p} \in M} \|\hat{\mathcal{T}}(\vec{p}) - \mathcal{T}_{gt}(\vec{p})\|^2 \ . \tag{7}$$

Since some methods may completely fail for certain test scenes, we also report top-50% and top-25% MSE values, which are less affected by estimation failures providing large errors.

B. Method Comparison

The results for the selected three methods and their variants are shown in Table I for the Laser Scanner Dataset and in Table II for our Outdoor Robot Tool dataset. From the results we can make the following observations: the Geometric Consistency (GC) based correspondence provide the most accurate and robust pose estimation. For the Laser Scanner Dataset objects GC variants are the best for 10/12 cases and SI-opt (curv) wins 2/12.

In general, parameter optimization with a validation dataset always improves accuracy; this is particularly evident for top-50% and top-25% MSEs indicating that fewer poses are falsely detected (far from the true pose). For GC, RANSAC post-processing sometimes improves the results, but for HG most of the time it does not. The curvature based filtering to robustify the methods does not improve HG and GC, but consistently improves SI making it comparable or even better than HG and GC. Region pruning consistently improves both GC and SI often achieving the best accuracy (8 out of 12 cases).

For our own dataset in Table II the results are very similar, although the objects are very different from those in the Laser Scanner dataset - our objects contain many large planar areas which supposedly should benefit from curvature filtering. Again Geometric Consistency (GC) variant is the winning

TABLE I

METHOD PERFORMANCE FOR THE LASER SCANNER DATASET. **Note:** *RANSAC IS PART OF THE METHOD.

$\times 10^{-3}$	Cheff			T-rex			Chicken			Parasaurolophus		
	MSE	top-50%	top-25%	MSE	top-50%	top-25%	MSE	top-50%	top-25%	MSE	top-50%	top-25%
Original with default parameters												
GC [1]	6.235	0.026	0.002	24.111	9.437	0.479	12.997	2.188	0.070	50.493	3.091	0.012
HG [2]	45.719	26.425	22.599	62.805	34.913	21.927	13.633	3.233	0.227	51.331	9.464	1.907
SI [3]	15.622	0.049	0.034	24.906	12.904	4.858	16.552	3.360	0.111	46.051	3.588	0.012
Optimized parameters												
GC-opt	5.108	**0.002**	**0.001**	**17.321**	8.015	0.197	11.175	1.727	0.042	46.253	2.697	0.009
HG-opt	7.586	0.520	0.003	17.557	12.496	7.334	10.592	2.829	0.227	46.772	8.679	0.624
SI-opt	5.300	0.021	0.010	27.700	11.600	3.900	13.000	1.678	0.012	46.000	3.503	0.005
Optimized & RANSAC												
GC-opt-RANSAC	3.100	0.006	0.001	19.464	7.150	0.102	10.936	1.554	0.089	48.000	2.575	0.016
HG-opt-RANSAC	35.501	21.260	15.000	45.900	21.100	15.200	14.396	2.484	0.204	46.981	7.572	0.049
SI-opt*	5.300	0.021	0.010	27.700	11.600	3.900	13.000	1.678	0.012	46.000	3.503	0.005
Our robustifying procedures												
GC-opt-RANSAC (curv)	6.900	0.036	0.010	21.440	13.838	7.779	10.537	1.581	0.100	45.600	2.720	0.024
HG-opt-RANSAC (curv)	13.930	5.207	2.937	21.881	9.711	3.198	12.374	1.989	0.179	50.663	3.852	0.141
SI-opt (curv)	3.900	0.017	0.007	21.900	8.385	0.384	10.017	1.252	**0.007**	45.900	2.963	**0.002**
GC-opt (regp)	**2.332**	0.004	**0.001**	18.326	**5.320**	**0.050**	**9.301**	**0.779**	0.016	**45.198**	**1.952**	0.006
HG-opt (regp)	14.670	9.718	9.134	32.136	21.249	15.897	29.902	15.622	11.512	60.179	22.303	16.045
SI-opt (regp)	4.600	0.018	0.007	22.600	8.300	1.300	16.734	3.496	0.120	46.695	3.131	0.082

TABLE II

METHOD PERFORMANCE FOR THE OUTDOOR ROBOT TOOL DATASET.

	Blade			Box		
	MSE	top-50%	top-25%	MSE	top-50%	top-25%
GC [1]	6.2730	1.0320	0.1890	6.7880	2.6190	0.0002
HG [2]	6.0620	0.6960	0.1240	8.7000	5.4800	1.5330
SI [3]	2.4080	0.0200	0.0005	9.7780	4.9950	0.0389
GC-opt	0.8671	**0.0003**	**0.0001**	5.7827	1.6394	**0.0001**
HG-opt	4.6077	0.2024	0.0510	6.7606	3.1600	0.0002
SI-opt	2.1690	0.0022	0.0005	6.3588	3.0081	0.0005
GC-opt-RANSAC	0.4184	0.0004	0.0002	4.1384	0.0463	0.0002
HG-opt-RANSAC	0.7333	0.2010	0.0330	6.0916	1.7224	**0.0001**
SI-opt	2.1690	0.0022	0.0005	6.3588	3.0081	0.0005
GC-opt-RANSAC (curv)	**0.2280**	0.0004	0.0002	**2.9893**	**0.0113**	**0.0001**
HG-opt-RANSAC (curv)	0.2595	0.1288	0.0334	6.0283	0.0249	0.0002
SI-opt (curv)	2.1734	0.0020	0.0004	6.2161	1.4291	0.0004
GC (regp)	0.2744	**0.0003**	0.0002	5.1058	0.1475	0.0002
HG (regp)	2.2614	0.2367	0.0805	7.4540	2.8303	0.0006
SI-opt (regp)	2.2948	0.0014	0.0007	6.0578	2.0800	0.0014

method in all cases (6/6). Clearly, the method of choice is GC with optimized parameters and curvature filtering as the GC-opt-RANSAC (curv) wins 4/6 cases.

C. Curvature Filtering

In the method comparison experiments (results in Tables I and II) robustifying was performed by optimizing the curvature filtering parameters with a validation set (example scenes). As described in Section IV-A the two important parameters are the *number of neighbour points* N_{curv} to estimate the curvature value and the *curvature threshold* τ_{curv}, which is used to remove low curvature points (below the threshold). MSEs using varying values of the curvature parameters for the snow blade images are shown in Figure 5 and Figure 6. We can make two observations: the neighbourhood size must be large enough to compute a robust curvature estimate (≥ 5). However, finding a suitable value for the curvature threshold is essential and is likely to depend on each model's properties. One should also note that although curvature filtering does not significantly improve GC or HG, we can still remove insignificant points while maintaining the same or an even better pose estimation rate.

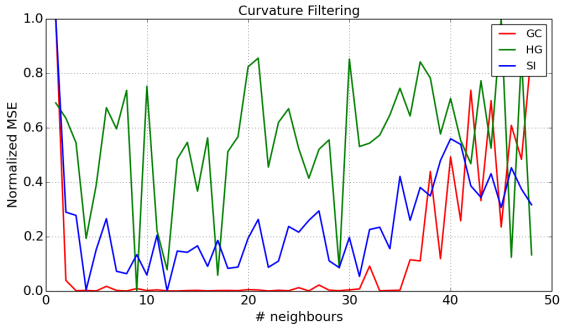

Fig. 5. Effect of the neighborhood size N_{curv} parameter to the performance of curvature filtering (snow blade).

D. Region Pruning

The good property of the Region Pruning method (Section IV-B) is that it does not make assumption on what kind of point cloud regions are good for robust pose estimation. The main parameter of the region pruning is the number of regions N_{reg} which also defines the computational time and it turns out that exhaustive search is doable only for $N_{reg} \leq 10$, but for good results we typically need $N_{reg} \geq 100$. In our case this was solved by randomly removing 10% of the regions and executing this random procedure 1,000 times.

E. Optimizing Method Parameters

From Table I and Table II it is clear that each method's parameters affect to the performance and robustify the method if individually set for each object.

Search of Inliers (SI) – The main parameters of the SI method are related to its two voting stages: local voting and global voting. The parameters that strongly influence the performance are the size of local voting neighbourhood

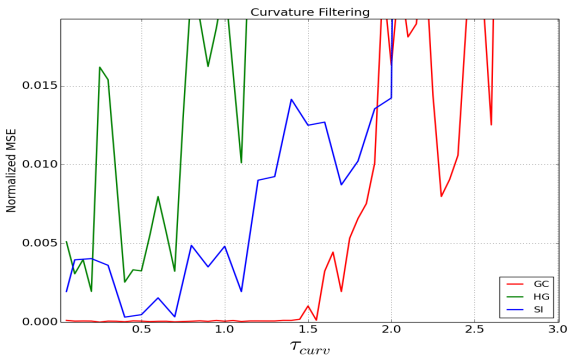

Fig. 6. Effect of the curvature threshold τ_{curv} parameter to the performance of curvature filtering (snow blade).

(the default value is 250) and the correspondence distance of global voting (the default value ≥ 0.9). The pose estimation errors as functions of the two parameters are shown in Figure 7. The default values perform reasonably well for the neighbourhood size, but the effect of the correspondence distance is important for robustness. For the snow blade, the optimal values are far from the default settings and there are optimal points for a low values 0.1 and high value ≥ 0.92 which indicates "alternative" point regions for robust pose estimation and these settings can only be found using cross-validation.

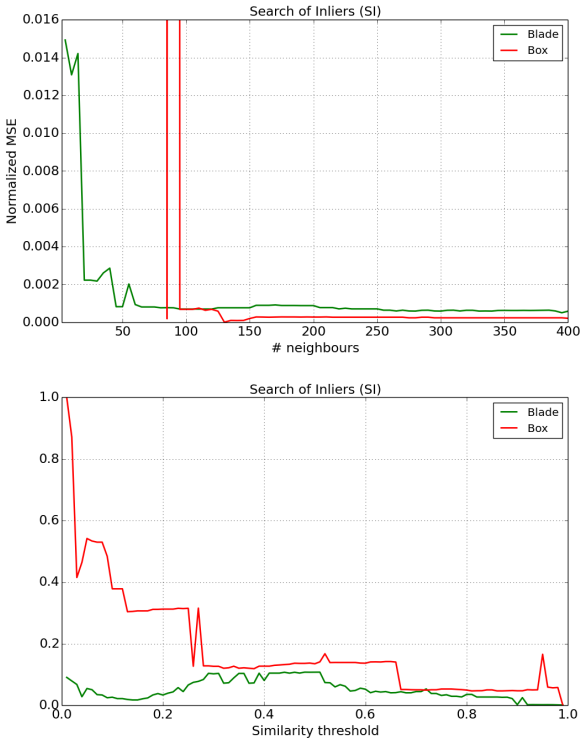

Fig. 7. Snow blade estimation error as the function of the SI parameters: neighborhood size (top) and distance threshold (bottom)

Geometric Consistency (GC) – The main parameter for GC is the geometrical consistency threshold and the results from the parameter optimization are shown in Figure 8. We can see that the optimal pairwise distance between correspondence points is 7 mm for the snow blade. The value is approximately $2\times$ higher than the threshold value used with the laser data. This is understandable due to Kinect's noisy sensor data.

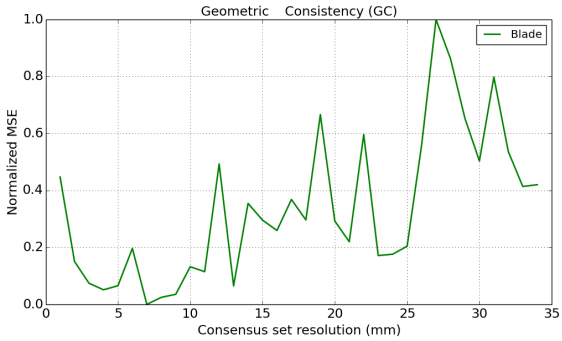

Fig. 8. Snow blade estimation error as the function of the GC consistency threshold.

Hough Grouping (HG) – The main parameter for HG is the Hough accumulation space bin size and the results from the parameter optimization are shown in Figure 9. A good value for the bin size is approx. 4 mm for the snow blade object.

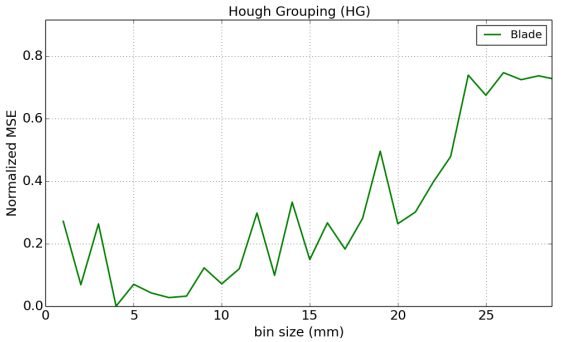

Fig. 9. Snow blade estimation error as the function of the Hough space's bin size.

VI. CONCLUSIONS

We proposed two alternative methods to improve 3D point correspondence based object pose estimation. Our methods, Curvature Filtering (Section IV-A) and Region Pruning (Section IV-B), were used to select a robust sub-set of correspondence against estimation failures. In our experiments (Table I and Table II), the robustifying methods consistently improved the three correspondence based methods: Geometric Consistency (GC) [1], Hough Grouping (HG) [2] and Search of Inliers (SI) [3]. Surprisingly, in all experiments Geometric Consistency (GC) outperformed the other two as combined with our robustifying using Region Pruning (Laser Scanner Dataset) or Curvature Filtering (Outdoor Robot

Tool Dataset). There was no clear winner between the two robustifying methods and more work is required to find the most suitable one. Our future work will address combining the two robustifying methods, branch-and-bound search for faster region pruning and cross-validation without validation images, i.e. using rendered views of the model point cloud itself. Robustified GC will be used in an autonomous robot service station where outdoor robot can change its tool.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Tampere University of Technology Robotics and Intelligent Machines Flagship project and the Academy of Finland (the ROSE project under the grant 292980).

REFERENCES

[1] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," *Pattern Recogn. Lett.*, vol. 28, pp. 1252–1262, July 2007.

[2] F. Tombari and L. Di Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *PSIVT*, pp. 349–355, IEEE, 2010.

[3] A. Buch, Y. Yang, N. Krüger, and H. Petersen, "In search of inliers: 3d correspondence by local and global voting," in *CVPR*, 2014.

[4] A. Buch, D. Kraft, J.-K. Kamarainen, H. Petersen, and N. Krüger, "Pose estimation using local structure-specific shape and appearance context," in *ICRA*, 2013.

[5] C. Li, J. Bohren, E. Carlson, and G. Hager, "Hierarchical semantic parsing for object pose estimation in densely cluttered scenes," in *ICRA*, 2016.

[6] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *ECCV*, 2010.

[7] A. Zaharescu, E. Boyer, and R. Horaud, "Keypoints and local descriptors of scalar functions on 2d manifolds," *IJCV*, vol. 100, pp. 78–98, 2012.

[8] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, "3d object recognition in cluttered scenes with local surface features: A survey," *PAMI*, vol. 36, no. 11, 2014.

[9] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. Kwok, "A comprehensive performance evaluation of 3D local feature descriptors," *IJCV*, vol. 116, pp. 66–89, 2016.

[10] C. Papazov and D. Burschka, "An efficient RANSAC for 3D object recognition in noisy and occluded scenes," in *ACCV*, 2010.

[11] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *CVPR*, 2010.

[12] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, "Rotational projection statistics for 3d local surface description and object recognition," *IJCV*, vol. 105, no. 1, pp. 63–86, 2013.

[13] A. Aldoma, F. Tombari, L. Di Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition," in *ECCV*, pp. 511–524, 2012.

[14] E. Rodola, A. Albarelli, F. Bergamasco, and A. Torsello, "A scale independent selection process for 3d object recognition in cluttered scenes," *IJCV*, vol. 102, no. 1, pp. 129–145, 2013.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *ISMAR*, pp. 127–136, 2011.

[16] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, (Sydney, Australia), Jul 2012.

[17] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *ICRA*, (Shanghai, China), May 9-13 2011.

[18] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, (Berlin, Heidelberg), pp. 356–369, Springer-Verlag, 2010.

[19] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *PAMI*, vol. 36, 2014.

[20] A. E. Johnson and M. Hebert, "Surface matching for object recognition in complex 3-d scenes," *Image and Vision Computing*, vol. 16, pp. 635–651, 1998.

[21] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *TPAMI*, vol. 21, pp. 433–449, May 1999.

[22] A. Aldoma, F. Tombari, L. di Stefano, and M. Vincze, "A global hypotheses verification method for 3d object recognition.," in *ECCV* (A. W. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), vol. 7574 of *Lecture Notes in Computer Science*, pp. 511–524, Springer, 2012.

[23] S. Salti, F. Tombari, and L. di Stefano, "On the use of implicit shape models for recognition of object categories in 3d data.," in *ACCV* (R. Kimmel, R. Klette, and A. Sugimoto, eds.), vol. 6494 of *Lecture Notes in Computer Science*, pp. 653–666, Springer, 2010.

[24] F. Attneave, "Some informational aspects of visual perception.," *Psychol Rev*, vol. 61, no. 3, pp. 183–193, 1954.

[25] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, pp. 115–147, 1987.

[26] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the conference on Visualization'02*, pp. 163–170, IEEE Computer Society, 2002.

[27] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *CVPR*, (Portland, Oregon), June 22-27 2013.

[28] A. Mian, M. Bennamoun, and R. Owens, "Three-dimensional model-based object recognition and segmentation in cluttered scenes," *PAMI*, vol. 28, no. 10, 2006.

[29] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image Vision Comput.*, vol. 10, pp. 145–155, Apr. 1992.