3D Object Pose Refinement in Range Images

Xenophon Zabulis(✉), Manolis Lourakis, and Panagiotis Koutlemanis

Institute of Computer Science, Foundation for Research and Technology - Hellas, Nikolaou Plastira 100, 70013 Heraklion, Greece
{zabulis,lourakis,koutle}@ics.forth.gr

Abstract. Estimating the pose of objects from range data is a problem of considerable practical importance for many vision applications. This paper presents an approach for accurate and efficient 3D pose estimation from 2.5D range images. Initialized with an approximate pose estimate, the proposed approach refines it so that it accurately accounts for an acquired range image. This is achieved by using a hypothesize-and-test scheme that combines Particle Swarm Optimization (PSO) and graphics-based rendering to minimize a cost function of object pose that quantifies the misalignment between the acquired and a hypothesized, rendered range image. Extensive experimental results demonstrate the superior performance of the approach compared to the Iterative Closest Point (ICP) algorithm that is commonly used for pose refinement.

1 Introduction

Human-made environments abound with textureless objects and several applications demand knowledge of their position and orientation (i.e., pose) in 3D space. Examples include household object manipulation in service robotics or bin picking and intelligent assembly in industrial settings. In such scenarios, the pose of textureless objects cannot be estimated with state of the art techniques like [5,14] that capture object appearance via photometric local patch detectors and descriptors.

When visual texture is not available, depth measurements provided by range sensors become a natural choice of input for determining object pose. The recent proliferation of low-cost RGB-D sensors such as the Kinect has renewed interest in depth-based object recognition and localization. Pose estimation in range images is often addressed as a final step in model-based object detection and recognition pipelines [15]. Such approaches typically estimate pose in a geometric verification step that is aimed to confirm the agreement of a certain model with the depth data and thus eliminate false positives. Since an approximate pose is often provided as a byproduct of object recognition, the ICP algorithm whose initialization is requires such a pose is a highly popular choice for pose refinement with 3D data.

This work deals with accurately estimating the pose of arbitrary rigid objects for which a 3D model and an initial pose are available. Its contributions include (a) an algorithm for model-based pose refinement, that is based on the simultaneous evaluation of many pose hypotheses and is shown to perform better than

L. Nalpantidis et al. (Eds.): ICVS 2015, LNCS 9163, pp. 263–274, 2015.
DOI: 10.1007/978-3-319-20904-3_25

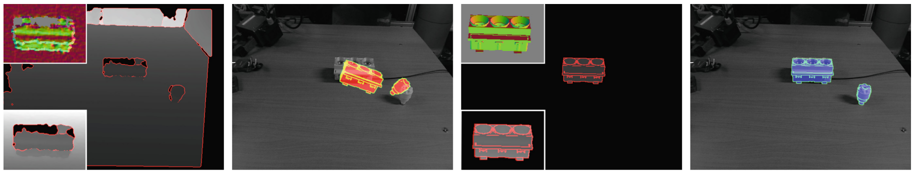


Fig. 1. Method overview; left to right: *(a)* input depth image and detected edges superimposed. Thumbnails show in magnification the depth values and edges (bottom) and color-coded surface normals (top, see text). *(b)* visualization of the initial poses for an electrical fuse and fusebox. *(c)* rendering of the fusebox, generated during hypothesis formulation, and detected edges superimposed; the illustrated pose is also the refined, output pose; thumbnails as in (a). *(d)* visualization of the refined poses (Color figure online).

ICP, (b) a novel objective function for comparing the alignment of two range images and (c) an efficient parallel implementation of the algorithm on the GPU. The proposed approach employs a range image and an initial, possibly crude, pose estimate which is assumed to originate from an object recognition algorithm. The accuracy of this pose is improved using exclusively depth information and making no assumptions regarding the presence of texture. Apart from the provision of an initial pose, the approach is oblivious to the object recognition algorithm, details of which shall not concern us in this paper.

An overview of the proposed approach is illustrated in Fig. 1. It employs a hypothesize-and-test scheme that forms pose hypotheses and evaluates them by generating synthetic depth images and measuring their affinity to the acquired image. This measurement involves the depth values, the associated surface normals, and the edges of the two depth images. The evaluated pose that best accounts for the observed data is kept. Candidate and final resulting poses are visualized in Fig. 1 by overlaying upon intensity images renderings of the object models at these poses, along with the depth edges detected at the corresponding depth images. Intensity images are used solely for illustration.

Accurate pose estimation for small objects with consumer RGB-D sensors is challenging due to the wide FOV and medium imaging detail of the latter, resulting in noisy depth measurements of low precision and resolution [13]. The proposed approach strives to maximize the number of pixels involved in a comparison. No correspondences need to be established between the depth image and the 3D model. Furthermore, no assumptions are made regarding the richness of surface normal variations or surface texture, hence the approach has no issues with large planar faces or densely textured objects. Missing data and moderate occlusions are tolerated and the implementation greatly benefits from the availability of modern GPU hardware. The approach employs a fully projective imaging formulation and can readily accommodate new objects of arbitrary resolution without any parameter tuning.

2 Previous Work

Pose estimation from 3D data is often addressed in the broader context of object detection and recognition. Existing approaches use either object-centered or view-centered representations. In the first case, distinctive features of an object's shape are extracted and their properties are encoded in an object-centered coordinate system to form a model that is independent of an observation viewpoint. Given a depth image with an unknown object, the latter is recognized by matching its features with those of a model. The pose of the object can be estimated from the underlying geometries of corresponding features using standard absolute orientation techniques [11]. Most approaches in the literature fall in this category and a far from exhaustive list includes spin images [12], 3D shape contexts [9], point histograms [19] and oriented point pairs [6]. It should be noted that these approaches perform better on clean, high-resolution and complete 3D scans rather than on noisy, low-resolution 2.5D range images. In recent years, local point feature descriptors that combine texture and shape cues on a per pixel basis have appeared, e.g. [8,23]. Nevertheless, such works focus primarily on the aspect of description and do not address the issue of feature detection.

View-centered representations capture the appearance of an object using multiple descriptions, each dependent on the vantage point from which the object is observed. Such approaches are trained with several images of an object at different poses and distances from the viewer. Descriptions extracted from these images are stored in an associative index along with the pose of the object. Recognizing an object in an image then amounts to finding its best match among the descriptions stored in the index. A rough estimate of object pose is readily provided by the pose associated with the best matching description. For example, Hinterstoisser et al. devised an efficient template matching technique capable of detecting textureless 3D objects [10]. They rely on depth edges and surface normals to represent an object with a set of binary training templates. The pose retrieved from the best matching template is used as a starting point for subsequent refinement with ICP. Tejani et al. [22] converted an early version of [10] to a scale-invariant patch descriptor and combined it with a regression forest. Park et al. [16] use rendering to construct a database of reference range images from multiple viewpoints, identify in parallel the reference image among them that best aligns with the input one and finally refine its pose with ICP. Wang et al. [24] develop a descriptor which combines color and shape features. They employ surface patches and describe them using color histograms and shape features that depend upon the geometric relationship of patch points with the patch centroid and the sensor viewpoint. Descriptors extracted from images of unknown objects are used in a nearest neighbor search to determine the identity of objects and their approximate 3D pose. Choi and Christensen [4] augment oriented point pairs with color information and use them in a voting scheme to generate clusters of pose hypotheses the most voted of which are refined with ICP. Sun et al. [21] represent objects as collections of patch parts and propose a generalized Hough voting scheme for object detection, succeeded by pose estimation with ICP.

In view-centered approaches, the pose retrieved via the best matching template is approximate. This is due to the limited resolution of the pose sampling process in training and potentially slightly wrong matches during recognition. Therefore, the retrieved pose should be refined with an optimization step that typically employs ICP [1,18]. It is well-known that ICP is sensitive to initialization, converging correctly only when sufficient overlap exists between two point sets and no gross outliers are present. Otherwise, it can easily get stuck in local minima. This has motivated the recent trend to regularize the ICP error metric by using sparsity-inducing $\mathcal{L}_p$ norms for $p \leq 1$. In this work, we propose to substitute ICP with the optimization scheme described in Sect. 3. This scheme is compared in Sect. 4 with a modern ICP variant based on $\mathcal{L}_p$ norms [2].

3 Proposed Method

The proposed method estimates the pose of an object whose position and orientation in space are assumed to be approximately known. Its input is a mesh model of the object, an initial object pose $\{\mathbf{R}_0, \mathbf{t}_0\}$, the acquired depth image and the depth sensor's intrinsic parameters. The result is a refined pose $\{\mathbf{R}, \mathbf{t}\}$. Both the initial and the refined poses refer to the sensor coordinate frame. Objects are represented with arbitrary 3D triangle meshes, which can originate from CAD drawings or digital scans. A mesh $\mathcal{M}$ is comprised of an ordered set of 3D vertex points V and an ordered set G of triplet indices upon V that define the mesh triangles. The 3D oriented bounding box B of each model is precomputed using the eigenvectors of V's covariance matrix.

The method generates candidate poses $\{\mathbf{R}_i, \mathbf{t}_i\}$ and evaluates them with the aid of depth images S_i synthesized by rendering $\mathcal{M}$ at each $\{\mathbf{R}_i, \mathbf{t}_i\}$. An objective function yields score $o(i)$, which quantifies the similarity between the acquired depth image and each S_i. The proposed objective function effectively compares two depth images using depth, edge and orientation cues and is amenable to a parallel implementation. Derivative-free particle swarm optimization is used to efficiently explore the pose space and optimize the objective function. Finally, the pose whose rendering is found to be the most similar to the acquired depth image is selected. Details on the method follow next.

Initialization. The initial pose is used to bootstrap pose estimation and can be quite crude. The projection of the model at the initial pose determines a 2D, axis-aligned bounding box b. To mitigate possible initial pose inaccuracies, b is inflated proportionally to the initial camera-object distance. To suppress sensor noise, the acquired depth image is median filtered (with a 5×5 kernel) and the result is retained as image D. Depth shadows and other shortcomings of consumer depth sensors manifest as invalid pixels in D. Surface normals for valid depth pixels are estimated by local plane fitting and stored in N. A binary image E is computed by Sobel edge detection on D. The distance transform T of E [7] is also computed for later use. These operations are parallelized on the GPU at pixel level, while T uses the parallel formulation of [3]. Only D is uploaded to the GPU, which uses it to compute N and T.

Pose Hypotheses Rendering. A rendering process simulates depth images of the target object at a hypothesized pose against a blank background. The simulated depth sensor shares the same intrinsic and extrinsic parameters with the real sensor. A rendered image simulates the image that the latter would acquire if it imaged the target object in isolation, at the hypothesized pose. Pose rendering is formulated as follows. Transform $\{\mathbf{R}_0, \mathbf{t}_0\}$ brings the model to an approximate location and orientation, in the sensor's reference frame. Candidate poses are parametrized relative to this initial pose, using a relative translation $\mathbf{t}_i$ and an "in place" rotation $\mathbf{R}_i$. This rotation is with respect to the centroid $\mathbf{c}$ of points in V. Specifically, the model is first translated by $-\mathbf{c}$ to center it on the coordinate axes origin, then rotated by $\mathbf{R}_i$, and finally translated back in place by $\mathbf{c}$. Rotation $\mathbf{R}_i$ is thus the product of primitive rotations about the 3 axes: $\mathbf{R}_i = \mathbf{R}_x(\theta_i) \cdot \mathbf{R}_y(\phi_i) \cdot \mathbf{R}_z(\omega_i)$. The transformation model point $\mathbf{x}$ undergoes is, thus, $\mathbf{R}_i \cdot (\mathbf{x} - \mathbf{c}) + \mathbf{c} + \mathbf{t}_i$. To avoid repeated calculations, the initial and candidate poses are combined into the following overall transformation:

$$\mathbf{R}_i \cdot \mathbf{R}_0 \cdot \mathbf{x} + \mathbf{R}_i \cdot (\mathbf{t}_0 - \mathbf{c}) + \mathbf{c} + \mathbf{t}_i. \tag{1}$$

The model transformed according to Eq. (1) is rendered in depth image S_i. Depth edges and surface normals of S_i are computed and stored in binary image E_i and data structure N_i, respectively, for subsequent use.

Computation and storage of S_i, E_i, and N_i is delegated to the GPU. The process employs Z-buffering to respect visibility and realistically deal with self-occlusions. Parallelization is performed at two levels of granularity. At a fine level, rendering is parallelized upon the triangles of the rendered mesh. At a coarser level, multiple hypotheses are rendered together, with a composite image gathering all renderings (see Fig. 2). In this manner, multiple hypotheses are evaluated in a single batch, resulting in better utilization of GPU resources and reduced CPU communication. Edge detection is applied once, directly upon the composite image.

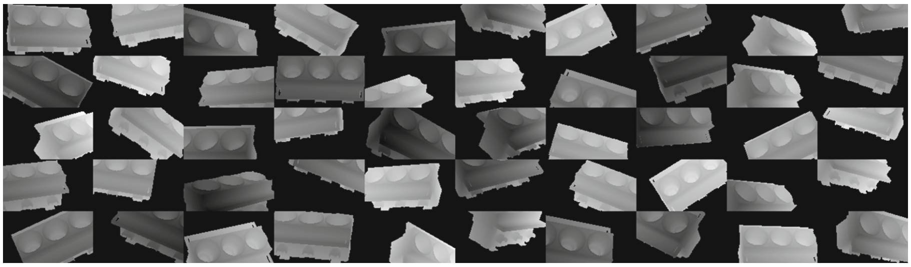


Fig. 2. A composite depth image for 50 pose hypotheses, all rendered in one batch. Each hypothesis is bounded by a rectangle and cropped when lying outside it, to avoid interference with its neighbors. Brighter pixels are further away from the sensor.

Pose Hypotheses Evaluation. A candidate pose is evaluated with respect to the extent to which it explains the acquired depth image. Ideally, rendering the object model at its true pose would produce an image identical to the acquired one. Function $o(\cdot)$ avails a score $o(i)$ by examining the similarity of depth values, surface normals, as well as edges between D and S_i. Two range images are corresponded in terms of their coordinates and are compared as follows. Depth values from D and S_i are compared pixelwise. For n pixel pairs, depth differences δ_k are computed and the cumulative depth cost term is given by Eq. (2), in which $|\delta_k|$ is set to ∞ if greater than a threshold d_T ($20\,mm$ in our implementation) to avoid comparing with background surfaces. For the same n pairs of pixels, the cost due to surface normal differences is as in Eq. (3), where γ_k equals the dot product of the two corresponding surface normals. Edge differences are aggregated in an edge cost using E and E_i. Let m be the number of edgels of E_i within bounding box b. For each such edgel j, let ϵ_j be the distance from its closest edgel in D which is looked up from T. The corresponding edge cost term is given by Eq. (4).

$$d_i = \sum_{k=1}^{n} \frac{1}{|\delta_k| + 1} \quad (2) \qquad u_i = \sum_{k=1}^{n} \frac{1}{|\gamma_k| + 1} \quad (3) \qquad e_i = \sum_{j=1}^{m} \frac{1}{\epsilon_j + 1} \quad (4)$$

Each of the cost terms in Eqs. (2)–(4) involves two ordered pixel sets, one from each image D and S_i, that contain the pixel locations to be compared. As d_i, u_i, and e_i have different numeric ranges, the combined cost is defined by their cubed geometric mean, i.e. $o(i) = -d_i \cdot e_i \cdot u_i$. The minus sign is used to ensure that optimal values correspond to minima, since d_i, e_i and u_i are non-negative. Summing the reciprocals of partial differences $|\delta_k|$, $|\gamma_k|$ and ϵ_j rewards poses that maximize the support (i.e., spatial overlap) between the compared regions of D and S_i: these partial sums are positive, thus the more pixels, the lower (better) the overall score becomes. The larger a difference, the smaller its induced decrease in the score. Hence, the objective function improves when more pixels in the rendered depth map overlap with the surfaces captured in the acquired image, whereas pose hypotheses based on little support do not yield good scores. Computation of $o(i)$ is parallelized per pixel on the GPU with minimal CPU communication. The scores of multiple pose hypotheses, evaluated together, are grouped and sent to the host CPU in a single message.

As no segmentation procedure is employed, during the evaluation of inaccurate pose hypotheses, the rendered object may be compared against pixels that correspond to background or occluding surfaces. To counter this effect, only pixels located within b are considered. Hypotheses that correspond to renderings partially outside b obtain a poor similarity score and thus the solution does not drift towards an irrelevant surface. Moreover, during the evaluation of each hypothesis, the oriented bounding box B_i that corresponds to hypothesis i is computed by transforming B according to Eq. (1). By so doing, depth pixels within b but corresponding to 3D points outside B_i are not considered, as they are irrelevant to the evaluated hypothesis.

Pose Optimization. The search space for pose estimation is constrained in a 6D neighborhood of the initial pose estimate $\{\mathbf{R}_0, \mathbf{t}_0\}$. As the cost of an exhaustive grid-based search is prohibitive, a numerical optimization approach is adopted to optimize objective function $o(\cdot)$ and find the pose that yields the highest similarity between the rendered and acquired image. Minimization of $o(\cdot)$ is based on PSO [17], which has proven to be an effective and efficient computational method for solving other vision optimization problems [25]. PSO stochastically evolves a population of candidate solutions dubbed particles that explore the parameter space in runs called generations. PSO does not require knowledge of the derivatives of the objective function, depends on very few parameters and requires a relatively small number of objective function evaluations until convergence. Compared to gradient-based optimization methods, PSO has a wider basin of convergence, exhibiting better robustness to local minima. Furthermore, as particles evolve independently at each generation, it is amenable to an efficient parallel implementation.

The rotational component of candidate poses is parameterized using Euler angles while translation is parameterized with Euclidean coordinates. Each dimension of the pose search space is bounded, defining a search hyperrectangle centered on the initial pose estimate. Particles are initialized with the aid of a 6D Sobol sequence [20], which ensures a good spatial distribution. PSO updates the state of each particle after the completion of each generation, whereas particles evolve independently within a generation. This observation suggests that the rendering and evaluation of particles representing pose hypotheses can be parallelized at every generation. As confirmed experimentally (see Sect. 4), such an optimization results into considerable computational savings.

4 Experiments

The proposed method was implemented in C++ and CUDA on a PC with an NVIDIA $GTX580$ programmable GPU. The Kinect 1 RGB-D sensor was employed for data acquisition in the experiments, thus D's resolution was 480×640 pixels and its field of view $57° \times 43°$. All experiments employed a PSO with parallelized hypotheses rendering. This PSO with 50 particles and 50 generations requires ≈ 0.1 s per frame to estimate the pose of a model with ≈ 7 K triangles. Compared to the 1.2 s required when hypotheses are evaluated sequentially, this time constitutes a 12-fold speedup.

Experiments employed data from the public dataset[1] of [10]. This dataset consists of 15 RGB-D sequences of textureless objects captured with a Kinect sensor that scanned a mostly static scene from various viewpoints. Occlusions, background clutter and sensor noise contribute to the emergence of significant amounts of outliers in the depth data. A subset of 11 sequences were used in the experiments reported here, which were selected depending on whether a triangle mesh model for the corresponding object was included in the dataset. Thus, the selected sequences are the 'duck', 'cat', 'ape', 'benchviseblue', 'can', 'driller', 'glue', 'holepuncher', 'iron', 'lamp' and 'phone', comprised of approximately 1200 RGB-D frames each. The corresponding models have between 11.6 K

[1] http://campar.in.tum.de/Main/StefanHinterstoisser.

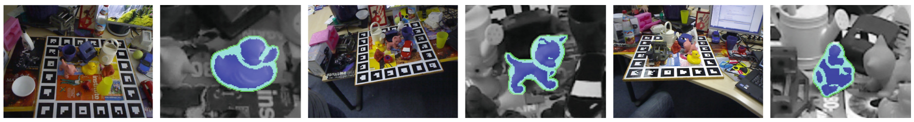


Fig. 3. Sample results from the experiments, for the 'duck', 'cat', and 'ape' sequences (left to right). For each result, shown is the RGB image on the left and a zoomed-in visualization of the estimated pose to its right (Color figure online).

and 467.5 K triangle faces. Objects measure from 5 to 15 cm in each dimension and are between 0.8 to 1.2 m away from the sensor. Figure 3 shows indicative pose estimation results with the proposed method.

Ground truth sensor poses obtained via marker-based extrinsic calibration from the RGB images are included in the dataset. To make the experiments independent of a particular object recognition method, the initial pose estimates required by our method were obtained by Monte Carlo-type simulations as random perturbations of the ground truth poses. More specifically, the ground truth pose for each frame was additively perturbed by a random vector following a multivariate uniform distribution. The marginal distributions of this vector were uniform in the interval $[-\tau\,\mathrm{mm}, \tau\,\mathrm{mm}]$ for the translational components and $[-\alpha°, \alpha°]$ for the rotational ones; this perturbation is henceforth denoted as $U\{\tau\,\mathrm{mm}, \alpha°\}$. Perturbations were applied independently to each pose parameter. This was repeated 10 times for each frame, estimating the pose for each perturbation and averaging the obtained pose estimation errors. Pose error was quantified by the misalignment between the true and estimated pose, as in [10]: for a ground truth pose $\{\mathbf{R}_g, \mathbf{t}_g\}$ and an estimated one $\{\mathbf{R}_e, \mathbf{t}_e\}$, the error is $e = (\sum_i |\mathbf{g}_i - \mathbf{e}_i|)/\nu$, where $\mathbf{g}_i = \mathbf{R}_g x_i + \mathbf{t}_g$, $\mathbf{e}_i = \mathbf{R}_e x_i + \mathbf{t}_e$, and i enumerates the ν vertices of V.

Impact of PSO Parameters. These experiments investigate the impact of key PSO parameters on the accuracy and speed performance of the proposed method. Its computational cost is determined by the number of objective function evaluations, which relates directly to the number of model renderings. PSO involves p particles that evolve for g generations, amounting to a total of $p \cdot g$ renderings. This product represents a tradeoff between accuracy and speed of execution and we refer to it as the "budget" of an optimization. A small budget will lead to premature termination and poor pose estimate, while a large budget will prolong execution without noticeable improvements in accuracy. An application of PSO with p particles and g generations is referred to as *configuration* p/g.

A set of exponentially decreasing budgets, namely configurations 200/200, 100/100, 50/50 and 25/25, examine the performance of the method with different settings. For each frame and configuration, the true pose was perturbed 10 times with $U\{20\,\mathrm{mm}, 20°\}$, then used to bootstrap the proposed method and finally the obtained pose estimation errors were averaged. Figure 4(left) summarizes the average execution time, whereas Fig. 4(right) illustrates the corresponding mean errors for the estimated poses. Sequences are enumerated in the order referenced in Sect. 4. As rendering time depends on the number of model triangles, pose estimation time was different for each object. From those figures, it is observed that configuration 50/50, amounting in 2500 renderings, constitutes a good tradeoff between accuracy and execution time.

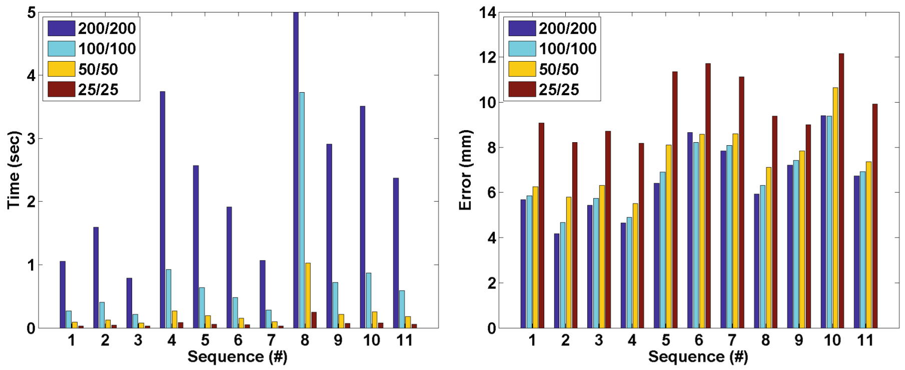
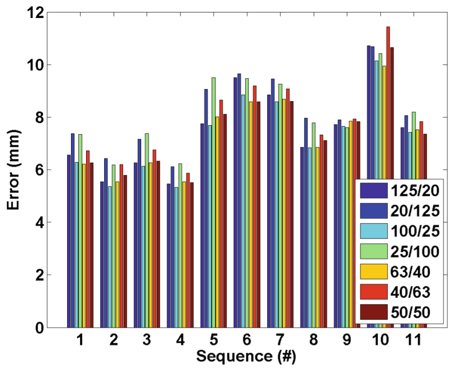


Fig. 4. Results from the decreasing budgets experiment from 533880 pose estimations. *Left:* execution times for the 4 configurations and all 11 sequences used. The vertical axis is limited to 5 s. The execution time for configuration #1 for sequence 8 is 15 s (its model has 467.5 K faces) and is off the chart. *Right:* accuracy obtained with the different configurations, shown by the mean error for each sequence.

Given a suitable budget, its allocation between particles and generations is investigated next. Towards this end, 7 configurations of particles and generations were considered, all with approximately the same budget of 2500 renderings: 125/20, 20/125, 100/25, 25/100, 63/40, 40/63, and 50/50. The same random perturbation $U\{20\,\text{mm}, 20°\}$ of the true pose was employed for the initial poses. Figure 5 shows the mean pose errors for the 7 configurations and all models. It is observed that for a given budget, a large number of particles is more beneficial than a large number of generations. This is because more particles avail a better coverage of pose space, reducing the possibility that the global minimum is missed for a local one. Besides, more particles entail increased parallelism among rendering calculations. For the configurations tested, it was also observed that PSO typically converged to a minimum in fewer than 50 generations, yielding more generations redundant. The most accurate configuration was 100/25 and, since it is also one of the most efficient, it is adopted when execution time is of primary concern.

Comparison with ICP. The accuracy of the proposed method was compared against a state-of-the-art ICP variant, chosen for its increased robustness to outliers. Specifically, the sparse ICP of [2] with the point-to-plane metric, referred to as "SICP" in the following, was employed for the comparison. The input to SICP was oriented 3D points produced by rendering models at the initial poses. Compared with the alternative of directly matching the range sensor data against



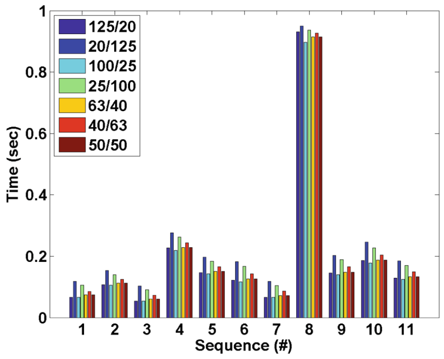


Fig. 5. Results from the experiment with the fixed budget, summarizing 934290 pose estimations. *Left:* execution times for the 7 configurations and all 11 sequences. *Right:* mean error for 7 configurations, grouped per each of the 11 employed sequences.

the complete 3D models, this choice enables a more fair comparison with the proposed method. The reason is that it prevents SICP from matching the input with self-occluded parts of the model and, at the same time, ensures that the point clouds being matched are of similar densities. PSO was applied with the configuration 100/100 as this was found in the previous experiment to yield accurate results within a moderate amount of time.

For each frame, ground truth poses were perturbed 10 times using $U\{20\,\text{mm}, 20°\}$, $U\{30\,\text{mm}, 30°\}$ and $U\{40\,\text{mm}, 40°\}$. For the larger of these perturbations, the corresponding initial poses differ considerably from the true ones. The pose for each perturbation was estimated with both the proposed method and SICP and the means of the pose estimate errors were computed. The results in Fig. 6, illustrate that the proposed method is more accurate than SICP. It is also more robust as indicated by the standard deviations of the errors (not included due to lack of space), which are much smaller for the proposed method.

We further report a wider basin of convergence for the proposed method. As the inaccuracy of the initial poses increases, SICP yields large errors, but this has only a marginal effect on the magnitude of the estimation errors for the proposed method which exhibits a graceful degradation: perturbations rising up to $U\{40\,\text{mm}, 40°\}$ still yield small errors for the proposed method, while SICP yields considerably larger errors that are comparable to the size of the objects and, therefore, can be regarded as pose estimation failures. In a view-centered pipeline (cf. Sect. 2), this facilitates a coarser sampling of training poses for the recognition step, while preserving pose estimation accuracy.

A direct comparison with [10] is not possible as its authors did not publish similarly detailed pose errors for these data. This is because they were only interested in comparing the pose errors against a threshold defined relative to the diameter of a model in order to determine whether a detected object corresponded to a true positive. It is noted that the performance of SICP constitutes

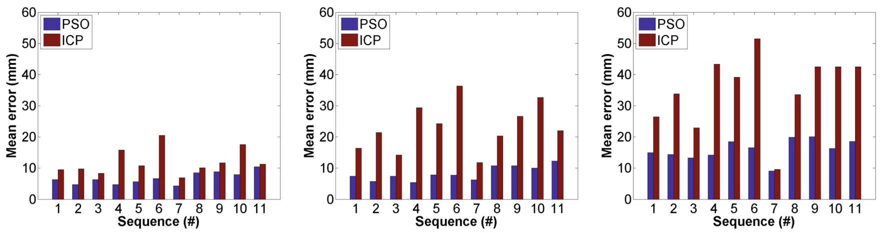


Fig. 6. Mean errors in mm for the PSO and SICP methods for perturbations $U\{20\,\text{mm}, 20^\circ\}$ (left), $U\{30\,\text{mm}, 30^\circ\}$ (middle), and $U\{40\,\text{mm}, 40^\circ\}$ (right). Each pair of vertical bars summarizes errors from 800160 pose estimations.

an upper bound on the performance of standard ICP variants, which would perform considerably worse due to their sensitivity to outliers, clutter and occlusions. As a result, the performance of SICP is a good indication of what can at best be achieved with ICP-based approaches.

5 Conclusions

This paper has presented an approach for accurately localizing 3D objects in range images. Bootstrapped with an initial pose estimate, the proposed approach refines it using a scheme that combines population-based randomized optimization with rendering-based synthesis. Extensive experiments on publicly available data have demonstrated that it can successfully cope with the limitations of current consumer RGB-D sensors, while delivering superior accuracy, robustness and speed compared to the commonly used ICP algorithm.

Acknowledgements. This work has received funding from the EC FP7 programme under grant no. 270138 DARWIN and by FORTH-ICS internal RTD Programme “Ambient Intelligence and Smart Environments”.

References

1. Besl, P., McKay, N.: A method for registration of 3-D shapes. PAMI **14**(2), 239–256 (1992)
2. Bouaziz, S., Tagliasacchi, A., Pauly, M.: Sparse iterative closest point. Comput. Graph. Forum **32**(5), 113–123 (2013)
3. Cao, T.-T., Tang, K., Mohamed, A., Tan, T.-S.: Parallel banding algorithm to compute exact distance transform with the GPU. In: I3D, pp. 83–90 (2010)
4. Choi, C., Christensen, H.: 3D pose estimation of daily objects using an RGB-D camera. In: IROS, pp. 3342–3349 (2012)
5. Collet, A., Martinez, M., Srinivasa, S.: The MOPED framework: object recognition and pose estimation for manipulation. Int. J. Robot. Res. **30**(10), 1284–1306 (2011)
6. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: CVPR, pp. 998–1005, June 2010

7. Felzenszwalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Theory Comput. **8**(19), 415–428 (2012)
8. Fischer, J., Bormann, R., Arbeiter, G., Verl, A.: A feature descriptor for texture-less object representation using 2D and 3D cues from RGB-D data. In: ICRA, pp. 2112–2117 (2013)
9. Frome, A., Huber, D., Kolluri, R., Bülow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: Pajdla, T., Matas, J.G. (eds.) ECCV 2004. LNCS, vol. 3023, pp. 224–237. Springer, Heidelberg (2004)
10. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part I. LNCS, vol. 7724, pp. 548–562. Springer, Heidelberg (2013)
11. Horn, B.: Closed-form solution of absolute orientation using unit quaternions. J. Optical Soc. Am. A **4**(4), 629–642 (1987)
12. Johnson, A., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. PAMI **21**(5), 433–449 (1999)
13. Khoshelham, K., Elberink, S.: Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors **12**(2), 1437–1454 (2012)
14. Lourakis, M., Zabulis, X.: Model-based pose estimation for rigid objects. In: Chen, M., Leibe, B., Neumann, B. (eds.) ICVS 2013. LNCS, vol. 7963, pp. 83–92. Springer, Heidelberg (2013)
15. Mian, A., Bennamoun, M., Owens, R.: Automatic correspondence for 3D modeling: an extensive review. Int. J. Shape Model. **11**(02), 253–291 (2005)
16. Park, I., Germann, M., Breitenstein, M., Pfister, H.: Fast and automatic object pose estimation for range images on the GPU. Mach. Vis. Appl. **21**(5), 749–766 (2010)
17. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. Swarm Intell. **1**(1), 33–57 (2007)
18. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3DIM, pp. 145–152 (2001)
19. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: ICRA, pp. 3212–3217 (2009)
20. Sobol, I.: Distribution of points in a cube and approximate evaluation of integrals. U.S.S.R. Comput. Maths. Math. Phys. **7**, 86–112 (1967)
21. Sun, M., Bradski, G., Xu, B.-X., Savarese, S.: Depth-encoded hough voting for joint object detection and shape recovery. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 658–671. Springer, Heidelberg (2010)
22. Tejani, A., Tang, D., Kouskouridas, R., Kim, T.-K.: Latent-class hough forests for 3D object detection and pose estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VI. LNCS, vol. 8694, pp. 462–477. Springer, Heidelberg (2014)
23. Tombari, F., Salti, S., di Stefano, L.: A combined texture-shape descriptor for enhanced 3D feature matching. In: ICIP, pp. 809–812 (2011)
24. Wang, W., Chen, L., Liu, Z., Khnlenz, K., Burschka, D.: Textured/textureless object recognition and pose estimation using RGB-D image. J. Real-Time Image Process. 1–16 (2013)
25. Zhang, X., Hu, W., Maybank, S., Li, X., Zhu, M.: Sequential particle swarm optimization for visual tracking. In: CVPR, pp. 1–8 (2008)