Continuous Surface-point Distributions for 3D Object Pose Estimation and Recognition

Renaud Detry and Justus Piater

University of Liège, Belgium
Renaud.Detry@ULg.ac.be Justus.Piater@ULg.ac.be

Abstract. We present a 3D, probabilistic object-surface model, along with mechanisms for probabilistically integrating unregistered 2.5D views into the model, and for segmenting model instances in cluttered scenes. The object representation is a probabilistic expression of object parts through smooth surface-point distributions obtained by kernel density estimation on 3D point clouds. A multi-part, viewpoint-invariant model is learned incrementally from a set of roughly segmented, unregistered views, by sequentially registering and fusing the views with the incremental model. Registration is conducted by nonparametric inference of maximum-likelihood model parameters, using Metropolis–Hastings MCMC with simulated annealing. The learning of viewpoint-invariant models and the applicability of our method to pose estimation, object detection, and object recognition is demonstrated on 3D-scan data, providing qualitative, quantitative and comparative evaluations.

1 Introduction

Autonomous systems need to acquire object models for detection, recognition and manipulation. Models should be acquired autonomously, which implies a method that does not require precisely controlled environmental conditions, exact ground-truth pose, or full 360° viewpoint covering. Furthermore, partial models should be directly usable, and allow for incremental completion.

We present a 3D, probabilistic object-surface model, along with mechanisms for probabilistically integrating unregistered 2.5D views (range images) into the model, and for segmenting model instances in cluttered scenes.

Our model encodes object structure through continuous probability density functions representing the distribution of object-surface points. This allows us to achieve detection by probabilistic inference, effectively avoiding explicit model-to-scene correspondences. Our method learns an initial model from a single view of an object; the model can then be used to detect and estimate the pose of the object in novel scenes, provided that the view is sufficiently similar. If a new view provides more information, the model can be extended, in principle until the entire surface is completely modeled. Model learning and model exploitation are thus seamlessly integrated. We demonstrate that our approach is competitive with state-of-the-art methods. While performing at least as well as state-of-the-art algorithms on public datasets, our approach shows advantages

in the paradigmatic and technical rigor of the techniques it builds on. Instead of defining e.g. probabilistic metrics on top of ad-hoc likelihood functions, our sensor model is intrinsically probabilistic. This approach allows for theoretical abstraction and flexibility. In particular, familiar building blocks from statistical learning are applied in appropriate places, such as kernel density estimation, Monte Carlo integration and inference, and expectation-maximization (see following sections).

Model learning is demonstrated on 3D-scan data from Biegelbauer and Vincze [1], and on the popular CAD dataset of Hetzel et al. [2]. The applicability of our method to pose estimation in cluttered point clouds is demonstrated on the data of Biegelbauer and Vincze, and object recognition rates are presented for the dataset of Hetzel et al.

2 Related Work

The modeling of objects from point-cloud data has been achieved through a variety of approaches [3]. The idea is generally to break down the object surface into a number of primitives; an object is then described by describing each primitive, and possibly by also describing their relative spatial configuration. Primitives correspond e.g. to complex parametric shapes such as superquadrics [1, 4], local surface descriptor [2, 5–9], or local edge descriptors [10]. In this work, primitives correspond directly to 3D points, with each point further parametrized by a local surface orientation computed from the distribution of the k nearest neighbors. In the following, we simply refer to these position-orientation pairs as *(5D) points*.

Depending on the application, recording the geometric structure of surface primitives, i.e. their relative spatial configuration, may or may not be necessary. Object recognition motivates discriminative models. Methods aiming at object recognition (without segmentation/pose estimation) [2, 5, 8] may completely ignore the spatial configuration of primitives, or encode it implicitly. They may also match a model by matching each view contained in the model separately, therefore also avoiding view registration. Conversely, pose estimation and segmentation require the modeling of the global shape of the object through the encoding of relative primitive configurations [6, 7, 9, 10]. This generally leads to a generative model. Although it may not necessarily be their primary aim, generative models often provide recognition, too [7].

When building a generative 3D model from multiple views, it becomes necessary to derive an exhaustive registration of individual 2.5D views. Our method learns an initial model from a single view of an object; the model can then be used to detect and estimate the pose of the object in novel scenes, provided that the view is sufficiently similar. If a new view provides more information, the model can be extended, in principle until the entire surface is completely modeled. Thus, model learning and model exploitation are seamlessly integrated.

Detection and alignment of generative models is typically achieved through the matching of model descriptors to scene descriptors, possibly followed by

geometrically-constrained optimization [7, 9]. We follow a different approach: We encode object structure through a continuous probability density function representing the distribution of object-surface 5D points. This allows us to achieve detection through probabilistic inference, which in turn avoids explicit model-to-scene correspondences. Our model is inferred by a Markov-chain Monte-Carlo (MCMC) algorithm which yields the maximum-likelihood pose of a model in an arbitrary scene.

Within a point-cloud reconstruction, the quantity of information conveyed by a point from a large and uniform surface is arguably smaller than the information conveyed by a point on a smaller, distinctive surface. In other words, the contribution of a surface segment to the identity of an object is generally not proportional to the number of points supporting the segment in a point-cloud reconstruction. Many 3D modeling techniques acknowledge this observation, e.g. through the detection of salient points [11], or the use of surface primitives of varying size [7]. We proceed by splitting object points into groups that represent object *parts* of different spatial size, and give each part the same weight in the detection process.

We note that the problem of 3D pose estimation (and recognition) has also been addressed for 2D images [12, 13]. Image-based methods often rely on the matching of 2D patch descriptors; they work best on highly textured objects. Although these methods can be fast and convenient to deploy, their 3D estimates are generally less accurate than those obtained on range data.

This work is inspired by the work of Detry et al. [10], from which we borrow the idea of representing low-level sensor data with probability density functions. This paper goes beyond the work of Detry et al. in multiple ways. Inference is approached differently: we present a maximum likelihood MCMC algorithm, while Detry et al. compute a posterior density through belief propagation and importance sampling. Our learning method autonomously registers independent views, and autonomously identifies parts from spatial structure. Finally, our paper demonstrates the application of our system to range data, which is structurally different from the sparse-stereo edge data used by Detry et al., with two important results: (1) Our method is applicable to a much wider range of input data and does not depend on the heavy-weight ECV (Early Cognitive Vision) system. (2) Contrary to Detry et al., we can – and do – compare our results to competing approaches.

3 Object Model

As mentioned above, we consider point-cloud reconstructions in which each point is composed of a position and a local surface normal. The surface normal is computed at each point of the cloud from the covariance matrix C of its k nearest neighbors [14]. Let us denote by e_1, e_2, e_3 the eigenvalues of C, with $e_1 \geq e_2 \geq e_3$; depending on whether $e_1 - e_2$ is smaller or greater than $e_2 - e_3$, the local orientation is set to the eigenvector associated to e_3 or e_1 respectively, allowing for stable orientations on both surface and line configurations. Denoting

by S^2 the 2-sphere (the space of unit 3D vectors), computing local orientations yields a point-cloud $O = \left\{\left(\lambda^\ell, \theta^\ell\right)\right\}_{\ell\in[1,n]}$ where $\lambda^\ell \in \mathbb{R}^3$ and $\theta^\ell \in S^2$. We note that surface normals correspond to axial information; in other words, θ is equivalent to $-\theta$.

Our pose estimation method relies on the modeling of 3D surfaces with *surface-point distributions*. A surface-point distribution is a probability density function which models the spatial configuration of 5D points sampled from an object's surface. The function has a high value in regions surrounding object surfaces, and a low value elsewhere.

Surface-point distributions are represented with kernel density estimation. Kernel density estimation (KDE) allows one to model a continuous density function from a set of *observations* drawn from it, by assigning a local kernel function to each observation [15]; the density is estimated by summing all kernels. KDE allows us to define a continuous distribution from the point-cloud reconstruction of an object. The surface observations we are dealing with are points which belong to $\mathbb{R}^3 \times S^2$; denoting the separation of point components and kernel parameters into positions and orientations by $x = (\lambda, \theta)$, $\mu = (\mu_t, \mu_r)$, we define our kernel as

$$\mathbf{K}_{\mu,\sigma}(x) = \mathbf{N}_{\mu_t,\sigma_t}(\lambda)\, \mathbf{\Theta}_{\mu_r,\sigma_r}(\theta)\,, \tag{1}$$

where μ is the kernel mean point, $\sigma = (\sigma_t, \sigma_r)$ denotes the isotropic bandwidths in position and orientation, $\mathbf{N}$ is a trivariate isotropic Gaussian kernel, and $\mathbf{\Theta}$ corresponds to a pair of antipodal S^2 von-Mises Fisher distributions. The S^2 von-Mises Fisher distribution corresponds to a Gaussian-like distribution on 3D unit vectors [16]. Formally, the value of $\mathbf{\Theta}$ is given by

$$\mathbf{\Theta}_{\mu_r,\sigma_r}(\theta) = C_3(\sigma_r)\frac{e^{\sigma_r\, \mu_r^T\theta} + e^{-\sigma_r\, \mu_r^T\theta}}{2}, \tag{2}$$

where $C_3(\sigma_r)$ is a normalizing constant. The pair of antipodal von-Mises Fisher kernels in $\mathbf{\Theta}$ models the lack of direction in surface-normal orientations (see above); hence $\mathbf{K}_{(\lambda,\theta),\sigma}(x) = \mathbf{K}_{(\lambda,-\theta),\sigma}(x)$. The bandwidths σ_t and σ_r are computed using a k-nearest neighbor technique [15] on point positions. A surface observed through a point-cloud $\{x^\ell\}_{\ell\in[1,n]}$ is modeled with $\psi(x) = \sum_{\ell=1}^{n} \mathbf{K}_{x^\ell,\sigma}(x)$. An illustration is provided in Fig. 1.

We model an object composed of q parts with a set of surface-point distributions $\{\psi_i(x)\}_{i\in[1,q]}$; each surface distribution $\psi_i(x)$ models the distributions of points belonging to part i. All parts are defined in a common reference frame, so that $\sum_{i=1}^{q} \psi_i(x)$ yields a reconstruction of the whole object.

4 Inference

Model detection relies on the observation that a surface-point distribution can readily be used as a "3D template" that provides an *object pose likelihood* when convolved with the surface-point distribution of a scene. Let us consider an object model $\{\psi_i(x)\}_{i\in[1,q]}$. Also, let us denote by $SE(3) = \mathbb{R}^3 \times SO(3)$ the group of 3D

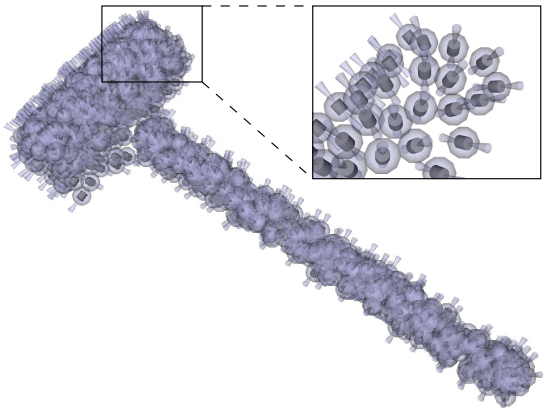


Fig. 1: Surface-point distribution computed from the 3D scan of a mallet. Surface-point observations are rendered with cylinders. The axis of a cylinder represents the orientation of the associated observation. Kernels are illustrated with translucent shapes: spheres and cones show one standard deviation in position and orientation respectively.

poses, and by $\phi(x)$ the surface-point distribution of an arbitrary scene in which the object appears. We model the pose of the object with a random variable $W \in SE(3)$; the distribution of the object's pose in the scene modeled by $\phi(x)$ is given by

$$p(w) \propto \prod_{i=1}^{q} m_i(w), \tag{3}$$

with

$$m_i(w) = \int \psi_i(x)\, \phi(t_w(x))\, \mathrm{d}x, \tag{4}$$

where $t_w(\cdot)$ denotes a rigid transformation by w. Each integral $m_i(w)$ corresponds to the evaluation at w of the cross-correlation of part i with the scene.

Pose estimation is achieved by searching for the maximum of $p(w)$. Furthermore, the value of $p(w)$ at its peak may be used as a matching score, hence yielding object detection and recognition (see Section 6.2).

As computing the maximum-likelihood (ML) object pose $\arg\max_w p(w)$ is analytically intractable, we approximate it with Monte Carlo methods. The integrals $m_i(w)$ are approximated as:

$$m_i(w) \simeq \frac{1}{n} \sum_{\ell=1}^{n} \phi(t_w(x_\ell)) \quad \text{where} \quad x_\ell \sim \psi_i(x). \tag{5}$$

Simulating $p(w)$ directly is not possible, although simulating a random variate w_* from one integral $m_i(w)$ can be achieved as follows:

1. Generate $x_\phi \sim \phi(x)$,

2. Generate $x_\psi \sim \psi_i(x)$,
3. Generate $w_* \sim f(w)$, where $f(w) = \phi(t_w(x_\psi))$, by selecting w_* from a uniform distribution on the transformations which map x_ψ onto x_ϕ.

The ML pose $\arg\max_w p(w)$ is computed via simulated annealing on a Markov chain whose invariant distribution at iteration j is proportional to $p^{1/T_j}(w)$ [17, 18], where T_j is a decreasing cooling schedule such that $\lim_{j\to\infty} T_j = 0$. The chain is defined with a mixture of two local- and global-proposal Metropolis–Hastings transition kernels, which are detailed below. Our choosing of the standard Metropolis–Hastings algorithm is motivated by the complexity of $\mathbb{R}^3 \times S^2$, which renders the calculation of local derivatives difficult. Also, $p(w)$ is likely to present a large number of narrow modes. A mixture of global and local proposals will compromise between distributed exploration of the pose space and fine tuning of promising regions. The independence-chain component of our transition kernel requires a global proposal function which we can simulate, and which should ideally resemble $p(w)$. In this paper, the global proposal corresponds to $s(w) = \sum_i m_i(w)$, which can be simulated by selecting $i \sim U_{[1,\dots,q]}$, and sampling from $m_i(w)$. The local proposal for the random-walk component of the transition kernel is given by the $SE(3)$ kernel

$$\mathbf{K}^\star_{\mu,\sigma}(x) = \mathbf{N}_{\mu_t,\sigma_t}(\lambda)\, \mathbf{\Theta}^\star_{\mu_r,\sigma_r}(\theta)\,, \tag{6}$$

where $\mathbf{\Theta}^\star$ is a pair of antipodal S^3 von-Mises Fisher distributions, and rotations θ and μ_r are expressed as quaternions [19]. The location bandwidth σ_t of this kernel is set to a fraction of the size of the object, which in turn is computed as the standard deviation of input object points to their center of gravity. Its orientation bandwidth is set to a constant allowing for 5° of deviation. The complete algorithm is listed in Fig. 2. For our purposes, the mixture weight ν is typically set to 0.75; T_0 and T_N are set to 0.5 and 0.05 respectively; N is of the order of 5000. Simulated annealing does not guarantee convergence to the global maximum of $p(w)$. Hence, we run several chains in parallel and eventually select the best estimate.

The model presented above is intrinsically sensible to relative spatial resolution within the input point cloud: the cross-correlation of parts with scene evidence (3) will be proportional to the global value scale of $\phi(x)$ in the region covered by the model. Unfortunately, the spatial resolution of 3D scans is generally not uniform. For example, objects closer to the sensor will generate more return than further ones. Hence, the maximum of the pose likelihood (3) may not correspond to the pose that best explains *surface shape*. In the experiments presented below, we largely mitigate this effect by evaluating scene densities $\phi(x)$ as the *maximum* of underlying kernel evaluations at x. We note that model distributions $\psi_i(x)$ are not concerned by this issue since they are integrated over multiple views.

Finally, we note that the expression of $p(w)$ can be identified to an application of the Belief Propagation algorithm to a Markov random tree. The tree root W models the object pose. All the other nodes of the tree are leaves, which we denote by X_i. The network compatibility potential linking W to X_i is defined

Initialize w_0 arbitrarily
Initialize σ_t and σ_r as explained in the text
For $j = 0$ to N :

$$T_j = \max\left\{T_0\left(\frac{T_N}{T_0}\right)^{j/N}, T_N\right\}$$

Sample $u \sim U_{[0,1]}$
If $u < \nu$:
Sample $w_* \sim s(w)$ *(global proposal)*
Sample $v \sim U_{[0,1]}$

$$\text{If } v < \min\left\{1, \left(\frac{p(w_*)}{p(w_j)}\right)^{1/T_j} \frac{s(w_j)}{s(w_*)}\right\} : w_{j+1} = w_*$$

Else : $w_{j+1} = w_j$
Else :
Sample $w_* \sim \mathbf{K}^{\star}_{w_j,(\sigma_t,\sigma_r)}(w)$ *(local proposal)*
Sample $v \sim U_{[0,1]}$

$$\text{If } v < \min\left\{1, \left(\frac{p(w_*)}{p(w_j)}\right)^{1/T_j}\right\} : w_{j+1} = w_*$$

Else : $w_{j+1} = w_j$

Fig. 2: Simulated annealing algorithm

by $\psi_i(t_w^{-1}(x))$, where $t_w^{-1}(\cdot)$ denotes the inverse of a transformation by w, such that $(t_w \circ t_w^{-1})(x) = x$ for all x in $\mathbb{R}^3 \times S^2$. Observation potentials are given by $\phi(x)$. Each integral $m_i(w)$ corresponds to the message sent from X_i to W in a belief-propagation inference of the marginal distribution $p(w)$.

5 Learning

The generation of a model from a single point-cloud reconstruction of an object is described in Section 5.1. Section 5.2 explains how a model is learned from multiple views.

5.1 Modeling A Point-cloud Reconstruction

Learning a model from a point-cloud reconstruction amounts to identifying the number and shape of object parts. Object parts are computed by clustering object points in $\mathbb{R}^3$; they are identified through the mixture of k trivariate Gaussians that best explains object point positions. K mixtures of $q = 1, ..., K$ Gaussians are fit using the Expectation-Maximization (EM) algorithm, and the most appropriate mixture is selected in a way inspired by the Bayesian information criterion [20]: the selected mixture is the one that minimizes

$$-2 \log L + Cq \log n, \tag{7}$$

where L is the maximized value of the data likelihood, n is the number of points, and C is a numerical constant; the difference between the Bayesian information

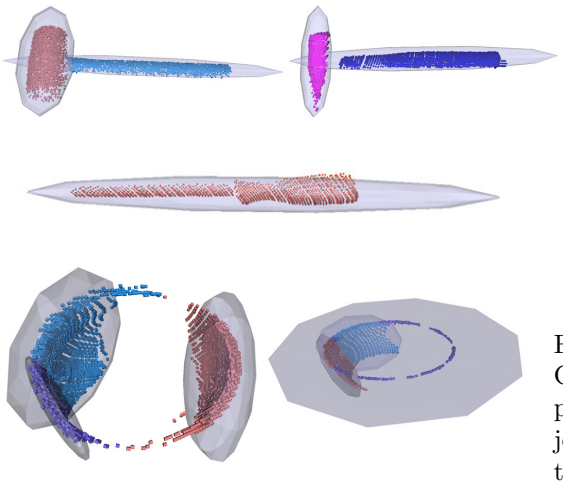


Fig. 3: Objects and their parts. Color indicates to which part a point belongs. Ellipsoids illustrate the mixture components that identify object parts.

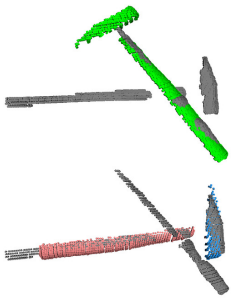


Fig. 4: Example pose estimates. Grey dots correspond to scene points; color dots show object points aligned to the scene through pose estimation. The single-part model fails to produce a correct estimate (top), whereas a two-part model succeeds (bottom).

criterion and Eq. 7 is in the additional factor C which allows us to strongly penalize large mixtures, hence keeping the number of parts reasonably low. The object model M is eventually composed of q surface-point distributions $\psi_i(x)$ built through KDE on the points that belong to cluster i.

Clustering is responsible for the identification of characteristic object parts (Fig. 3), drawing attention to smaller areas that would otherwise be overwhelmed by larger surfaces. In Fig. 4, the top image shows a pose estimate computed from a single-part model of a hammer. Inference finds a best match of the handle of the hammer on a screwdriver, and ignores the unmatched head of the hammer. In the bottom image, the two-part model of Fig. 3 draws inference towards a correct estimate. The part identification method described above is similar to the procedure of Bouchard and Triggs [21], except that their work eventually expresses parts as cluster centers. Here, clustering is exclusively used to *identify* parts. Parts are represented by fine-grained surface-point densities (Section 3), which hold significantly more information than a single Gaussian.

5.2 Learning From Multiple Views

The construction of a model that expresses the full 3D geometry of an object requires pairwise registration of multiple views. Naturally, only pairs of sufficiently overlapping views can be registered. Finding overlapping views through an exhaustive registration of all pairs is unfortunately rather inefficient. Therefore, a meta-process should ideally detect strongly correlated views, which are good candidates for registration [7]. In this section, we present a somewhat simpler

method, which iteratively integrates views into a model, expecting each additional view to overlap with at least one of the previous views. Let us assume that each view contains n points, and let us denote by M_ℓ a model made up of ℓ views, and denote by O_ℓ the set of points used to construct M_ℓ. The first model M_1 is built, following the procedure of the previous section, from the points produced by the first available view v_1. Let us then assume that we have a model M_ℓ constructed from ℓ views, and the set O_ℓ from which it was built. Adding $v_{\ell+1}$ to the model works as follows. The pose of M_ℓ is estimated in $v_{\ell+1}$ (Section 4), which allows us to transform the points of $v_{\ell+1}$ into the object reference frame, yielding an object-registered point set T. A set of points $O_{\ell+1}$ that spans $\ell+1$ views is then formed by selecting $n/(\ell+1)$ points at random from T and $n\ell/(\ell+1)$ points from O_ℓ. $M_{\ell+1}$ is constructed by applying the procedure of Section 5.1 to $O_{\ell+1}$.

6 Evaluation

In the following experiments, models typically contain 1 to 4 parts. Scene surface-point distributions are computed from 5000 scan points. In order to limit the computational cost of inference, the total number of surface-point observations within object parts is limited to 500. The number of parallel chains in MCMC inference is typically set to 16. Our implementation estimates the pose of a model in a scene in about 5–10s on an 8-core desktop computer, and its memory footprint is always below 50MB. The cost of detecting multiple objects is linear in the number of objects.

6.1 Cluttered-Scene Pose Estimation

The suitability of our model for pose estimation in cluttered scenes is demonstrated on 3D-scan data from Biegelbauer and Vincze [1]. We learned a model of a mallet, a hammer, a screwdriver, and two bowls, using between 1 and 4 segmented range views of each object. The objects and their parts are illustrated in Fig. 3. The pose of these objects was estimated in 4 range scenes. Because points from the ground plane represent approximately 85% of each scene, we removed these prior to detecting the objects, by isolating them through RANSAC plane fitting. Although this step is not necessary, it significantly lowers inference time; also, through this process, we put our system in the same experimental conditions as Biegelbauer and Vincze. As illustrated in Fig. 5, all 11 pose estimates were correct. We followed the scenario of Biegelbauer and Vincze and reproduced the experiment several times using different software random seeds, and every run lead to the same correct estimates. When using models made of a single part, instead of the multi-part models of Fig. 3, only 7 out of 11 poses were correct, for reasons identical to these explained in Section 5.1. Despite its simplicity, the part-learning process is instrumental in discriminating between objects of similar shapes.

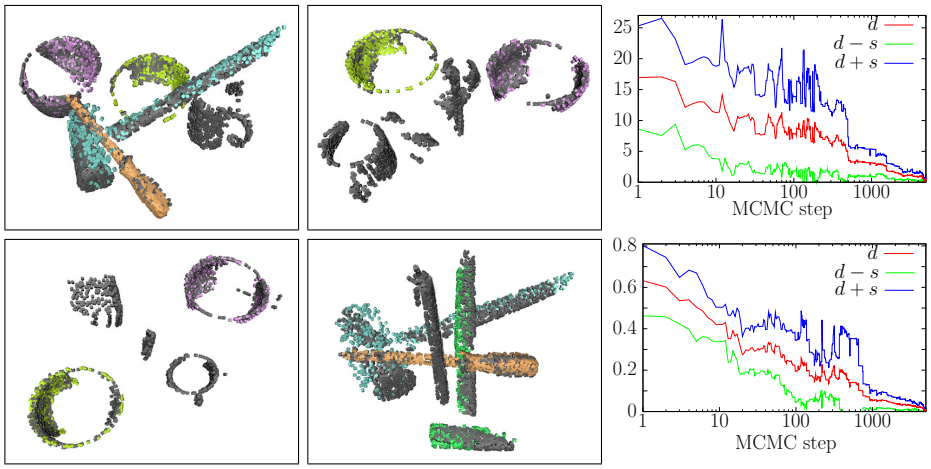


Fig. 5: Left side: cluttered scenes with pose estimates. Grey dots correspond to scene points. The rest of the dots illustrate pose estimates: they correspond to object points (Fig. 3) aligned to the poses of the objects in the scenes. There are 11 pose estimates (4 in the top-left scene, etc.). Right side: convergence of the MCMC process, as the mean position distance to optimum (top-right, distance in mm) and mean orientation distance to optimum (bottom-right, distance in radians) (see text for details).

The two graphs of Fig. 5 illustrate MCMC convergence. Pose estimation consists in running multiple Markov chains in parallel. The convergence of a pose estimation process e can be illustrated by tracking, at each MCMC step i, the distance d_i^e between the position component of the ground truth pose and the position component of the chain which is, at step i, the closest to the ground truth pose. The graph in the top-right corner of Fig. 5 (red curve) shows, as a function of the MCMC step i, the average $d_i = \frac{1}{11}\sum_e d_i^e$ across the eleven pose estimates of Fig. 5 (we note that the i axis is in log scale). The green and blue curves indicate an interval of one standard deviation s_i, with $s_i = \sqrt{\frac{1}{11}\sum_e (d_i^e - d_i)^2}$. The bottom-right graph illustrates orientation convergence. We note that as we do not have ground truth poses for the scenes of Fig. 5; the eleven ML estimates shown in the figure were used as ground truth. For both position and orientation, the error and error variance rapidly decrease between steps 1 and 1000, then smoothly converge to zero.

We note that the scenes of Fig. 5 contain more objects than those shown in Fig. 3. These objects are not part of the experiment simply because we do not have segmented views of them. A segmented view of the deeper bowl (purple in Fig. 5) was also missing. Its model was built from data extracted from the top-left scene of Fig. 5. It seemed relevant to include that object because of its similarity with the second (yellow) bowl.

6.2 Object Detection And Recognition

As mentioned above, the value of the pose-likelihood expression (3) at its peak may be used as a matching score, hence yielding object detection and recognition.

Object detection was evaluated on the online-available CAD dataset of Hetzel et al. [2]. This dataset contains 258 simulated range images for each of its 30 objects (Fig. 6). It is divided into a training and a testing set, containing respectively 66 and 192 views of each object. We learned a view-invariant model of each object using its 66 training views, providing them to the multi-view learning algorithm of Section 5.2 in the order in which they appear on the dataset website. Even though this order is not always ideal, it allowed for the construction of a good model of most objects. Fig. 7 shows eight examples. The bunny and the dinosaur are correctly reconstructed. The deodorant bottle is missing a side; this is explained by the symmetry of the object, which causes all views to be registered to the same side of the model.

Object detection determines whether a given model is present in a view. Detecting an object in a view amounts to estimating the object's ML pose w in that view (Section 4), reading the value of the pose distribution at w, and comparing the "score" $p(w)$ to a threshold (the whole process taking 5–10s). Thresholds were learned as follows:

1. We instantiated all 30 object models in 300 views of the training set – 10 views from each object – providing 9000 training scores. Fig. 8 (dashed curves) shows the distribution of the resulting scores for two objects.
2. For each object o, we trained a binary naive Bayes classifier on the 300 training scores produced by o, providing means of distinguishing o from all other objects.

Object detection rates were obtained by instantiating (Section 4) the 30 models in 300 images from the testing set, yielding 9000 testing scores. Fig. 8 (solid curves) shows the distribution of the resulting scores for two objects. Confronting the testing scores to the 30 detection classifiers yielded a 98% detection rate, i.e. out of the 9000 binary classifications, there are 298 true positives, 8580 true negatives, 2 false negatives and 120 false positives.

By contrast to object detection, object recognition determines, given one view, which object this view is most likely to show. For this purpose, we trained a single SVM classifier on the 9000 training scores obtained above. This classifier allows us to determine which object an arbitrary image corresponds to, by matching (Section 4) all 30 object models to that image, and submitting the 30 resulting scores to the classifier. On a set of 300 images from the testing set of the database, we obtained a 99% recognition rate, i.e. 297 true positives and 3 false positives. This result is directly comparable and competitive with recent discriminative approaches on the same dataset which yield 98% [8] and 93% [2]. It is also comparable to the 95% presented in Section 8.1 of the article from Mian et al. [7], although the object library used by Mian et al. is a superset of the one we are using. Recognizing which object a view belongs to requires the inference of all known object models; recognition is linear in the number of object models.

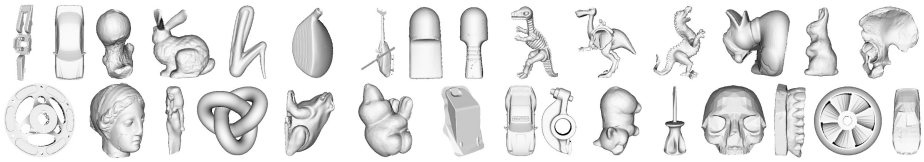


Fig. 6: Object library from Hetzel et al. [2]. Illustration kindly provided by Li and Guskov [8].

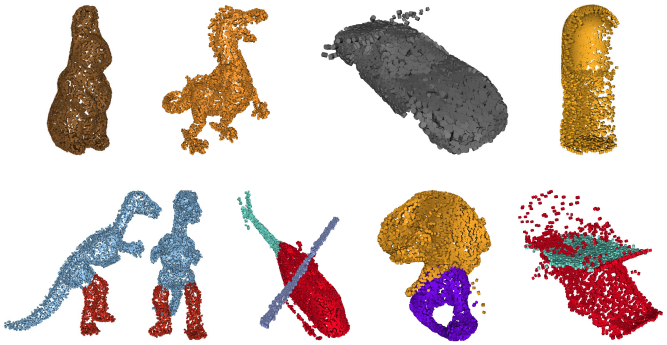
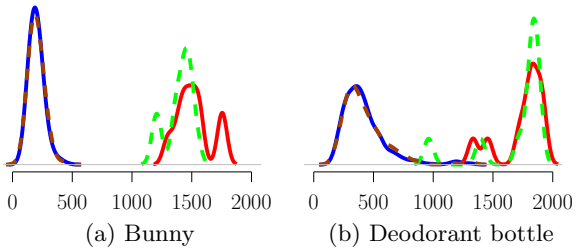


Fig. 7: Object points obtained from the registration of sequences of 66 views. Color indicates learned parts; objects on the first row are made up of a single part, whereas those on the second row yield two- or three-part models. The models of the deodorant bottle (top-right) and of the pedal (bottom-right) were not correctly constructed, because of symmetries and similarities within the objects.



(a) Bunny (b) Deodorant bottle

Fig. 8: Distribution of detection scores for the bunny and deodorant bottle (see Fig. 7). The dashed green line shows the distribution of the scores resulting from the instantiation of the object model in all training images of that object. The dashed brown line (almost overlapping with the blue line) corresponds to the instantiation of the model in the rest of the training set. The red line corresponds to testing images of the object; the blue line corresponds to testing images of the other objects. The scores provided by the bunny model are clearly separable. The deodorant bottle is less robustly detected, largely because of a second, similar bottle in the object set.

We emphasize that the classifiers above are only applied to find appropriate score-separating thresholds or planes. The underlying inference mechanism is not discriminative, and goes further than object recognition by providing an $SE(3)$ object pose.

7 Conclusion

We presented the definition, inference and construction of a 3D object model. The model consists of a set of parts represented with smooth surface-point densities. Object pose likelihood is defined through the cross-correlation of parts with scene evidence. The ML pose is computed through simulated annealing on a Markov chain whose invariant distribution is proportional to an increasing power of the pose likelihood, yielding an effective balance between exploration and convergence. The learning procedure probabilistically registers and fuses partly overlapping object views and identifies object parts through expectation-maximization. The suitability of our model for pose estimation was demonstrated on cluttered range scenes, using a set of objects of similar shapes; object recognition results competitive with recent generative and discriminative methods were obtained on a publicly available dataset.

While performing at least as well as state-of-the-art algorithms on public datasets, our approach shows advantages in paradigmatic and technical rigor of the techniques it builds on. Instead of defining e.g. probabilistic metrics on top of ad-hoc likelihood functions, our sensor model is intrinsically probabilistic. This approach allows for theoretical abstraction and flexibility. In particular, familiar building blocks from statistical learning are applied in appropriate places, such as kernel density estimation, Monte Carlo integration and inference, and expectation-maximization. Using rigorous, formal building blocks also facilitates the adaptation of the system to different situations. For instance, for problems where local derivatives of the pose density are available, using hybrid Monte Carlo instead of Metropolis-Hastings would improve inference performances.

Acknowledgments

The authors warmly thank Dr. Georg Biegelbauer and Pr. Markus Vincze for providing us with 3D-scan data. This work was supported by the Belgian National Fund for Scientific Research (FNRS) and the EU Cognitive Systems project PACO-PLUS (IST-FP6-IP-027657).

References

1. Biegelbauer, G., Vincze, M.: Efficient 3D object detection by fitting superquadrics to range image data for robot's object manipulation. In: IEEE International Conference on Robotics and Automation. (2007)

2. Hetzel, G., Leibe, B., Levi, P., Schiele, B.: 3D object recognition from range images using local feature histograms. In: Computer Vision and Pattern Recognition. (2001) 394–399
3. Campbell, R.J., Flynn, P.J.: A survey of free-form object representation and recognition techniques. Comput. Vis. Image Underst. **81** (2001) 166–210
4. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE Trans. Pattern Anal. Mach. Intell. **12** (1990) 131–147
5. Frome, A., Huber, D., Kolluri, R., Bulow, T., Malik, J.: Recognizing objects in range data using regional point descriptors. In: European Conference on Computer Vision. (2004)
6. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. **21** (1999) 433–449
7. Mian, A.S., Bennamoun, M., Owens, R.A.: Three-dimensional model-based object recognition and segmentation in cluttered scenes. IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 1584–1601
8. Li, X., Guskov, I.: 3D object recognition from range images using pyramid matching. In: International Conference on Computer Vision. (2007) 1–6
9. Rusu, R., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: IEEE International Conference on Robotics and Automation. (2009)
10. Detry, R., Pugeault, N., Piater, J.: A probabilistic framework for 3D visual object representation. IEEE Trans. Pattern Anal. Mach. Intell. **31** (2009) 1790–1803
11. Li, X., Guskov, I.: Multi-scale features for approximate alignment of point-based surfaces. In: Eurographics symposium on Geometry processing. (2005)
12. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. Int. J. Comput. Vision **66** (2006) 231–259
13. Collet, A., Berenson, D., Srinivasa, S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: IEEE International Conference on Robotics and Automation. (2009)
14. Liang, P., Todhunter, J.S.: Representation and recognition of surface shapes in range images: A differential geometry approach. Computer Vision, Graphics, and Image Processing **52** (1990) 78–109
15. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC (1986)
16. Fisher, R.A.: Dispersion on a sphere. In: Proc. Roy. Soc. London Ser. A. (1953)
17. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220** (1983) 671–680
18. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Machine Learning **50** (2003) 5–43
19. Sudderth, E.B.: Graphical models for visual object recognition and tracking. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (2006)
20. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics **6** (1978) 461–464
21. Bouchard, G., Triggs, B.: Hierarchical part-based visual object categorization. In: Computer Vision and Pattern Recognition. Volume 1. (2005) 710–715