EUROGRAPHICS 2013 / I. Navazo, P. Poulin
(Guest Editors)

Volume 32 (2013), Number 2

A Correlated Parts Model for Object Detection in Large 3D Scans

M. Sunkel[1], S. Jansen[1], M. Wand[1,2], H.-P. Seidel[1]

[1]MPI Informatik [2]Saarland University

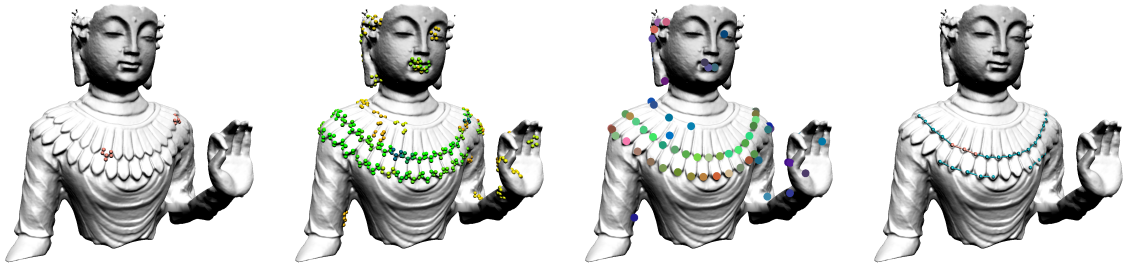


Figure 1: *Based on sparse user annotations a shape model is learned. The detected instances are transformed into descriptors for the second hierarchy level. Hierarchical detections shown on the right are obtained using only the example marked red.*

Abstract

This paper addresses the problem of detecting objects in 3D scans according to object classes learned from sparse user annotation. We model objects belonging to a class by a set of fully correlated parts, encoding dependencies between local shapes of different parts as well as their relative spatial arrangement. For an efficient and comprehensive retrieval of instances belonging to a class of interest, we introduce a new approximate inference scheme and a corresponding planning procedure. We extend our technique to hierarchical composite structures, reducing training effort and modeling spatial relations between detected instances. We evaluate our method on a number of real-world 3D scans and demonstrate its benefits as well as the performance of the new inference algorithm.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Object hierarchies; Image Processing and Computer Vision [I.4.8]: Scene Analysis—Object recognition; Artificial Intelligence [I.2.10]: Vision and Scene Understanding—Shape

1. Introduction

3D scanning technology has matured to a point where very large scale acquisition of high resolution geometry has become feasible. Using mobile LIDAR scanners, point clouds at centimeter resolution of complete countries can be captured at economically viable costs (such as in the well-known projects by companies like Google or Navteq). Cost efficient approaches such as structure-from-motion reconstruction from community photo collections [ASS*09, FGG*10, GAF*10] complement these efforts.

Having, at some point, accurate 3D models of our entire planet offers enormous opportunities, but it also poses new technical challenges. A key problem is semantic understanding: Almost any application beyond simple 3D rendering, such as mobile navigation, maintenance of public infrastructure, or planning for disaster preparedness, requires an understanding of the semantics of acquired geometry, such as finding roads, cars, street lights, entrances to buildings, and the similar. This information is not acquired by any 3D scanner, and human annotation of large scale data is obviously infeasible. Hence, the development of automatic techniques for semantic labeling and correspondence computation has become a very important research topic.

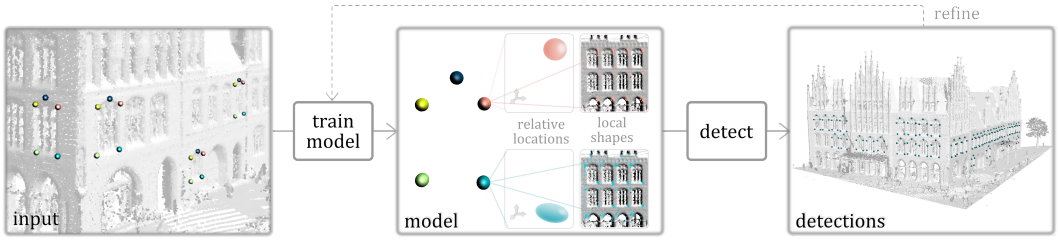


Figure 2: *Outline of our method. Given sparse user annotations defining some shapes and their correlations, a shape model characterizing these shapes is constructed. Each node in the model is associated with corresponding distributions for the local shapes (descriptors) and the relative location within the shape. The efficient detection allows for refining the model interactively.*

In this paper, we address the problem of detecting object instances (*shapes*) according to semantic object classes. Shapes are defined as a set of distinctive parts describing local geometry (*local shape*) as well as the spatial layout (*constellation*) of these parts. From a small number of hand-annotated examples, a part-based shape model is derived to retrieve large quantities of further instances (see Fig. 2). This problem is closely related to "scene understanding" approaches for 2D images: In addition to sophisticated local descriptors [DT05], and bags of descriptors [LSP06, GD05], two main strategies have emerged: (1) Conditional random fields (CRFs) are used to model local appearance and consistency between neighboring pixels [KH03, HZRCP04], aiming at labeling of amorphic categories such as "vegetation" or "buildings". (2) Using part-based models objects are recognized by detecting constellations of parts [FPZ03, FH05, LLS04, FMR08], excelling in detecting individual objects, such as "cars" or "bikes". In the context of 3D scanner data, several methods have been proposed based on local descriptors [GKF09], bags-of-words [LG07, BBGO11], and conditional random fields [ATC*05, ZLZ*10, KHS10]. However, object detection beyond traditional symmetry detection [MPWC12] has been explored only briefly [SJW*11]. Filling in this gap is the main objective of our paper.

Detecting constellations of parts in 3D geometry poses some unique challenges and opportunities that have not yet been addressed in previous work:

Correlations: The local shape of object parts is typically strongly correlated (e.g. corners of a window or tires of a car). Unlike previous work our model captures these general correlations. We demonstrate empirically that this leads to a significant improvement in performance.

Frame invariance: Detection should be invariant under translations and rotations. In contrast to previous work, our method provides full rotational invariance.

Correspondences: In 3D computer graphics, detecting semantic correspondences can be an intermediate step towards building generative models such as morphable shape spaces. Therefore, obtaining accurate part correspondences is desirable. Our method improves correspondence accuracy over previous approaches.

Semantical symmetry: In a large 3D scan, a large number of instances of the same object class such as cars or windows show up simultaneously. Our algorithm detects all instances of an object class in a single pass.

Structuring scenes: As an extension, we describe a hierarchical version of our method that can not only model simple objects, but also learn compound models, i.e. higher order co-occurrence patterns to structure the input.

Efficiency: Finally, training models interactively on large 3D scans requires fast, high-throughput detection algorithms. Our method provides interactive training and very efficient detection.

We evaluate our method empirically on a number of benchmarks, evaluating the detection performance as well as the runtime costs. This includes a large city scan with 4GB of input data, for which training is interactive and detecting all instances of an object class takes less than 2 min, using single-threaded, unoptimized C++ code.

2. Related Work

Image understanding: Our method is related to the idea of constellation models in image understanding as described in Fergus *et al.* [FPZ03]. However, their model does not consider pairwise relations of part appearances. The utility of appearance correlations has been shown in the context of bags-of-words models: Wang *et al.* [WZFF06] demonstrate that explicitly modeling the inter-dependencies of local patches yields more discriminative models. In the context of part-based models, pairwise geometric relations of lines have also proven to be helpful for recognition [LHS07, SGS09]. Leordeanu *et al.* [LHS07] use a set of angles and distances to represent the geometric relations between parts. Stark *et al.* [SGS09] enrich constellation models by pairwise symmetry relations between contour segments. In the 3D domain, these correlation can be expected to be even more pronounced, as variability due to lighting, texture, and occlusion is not present.

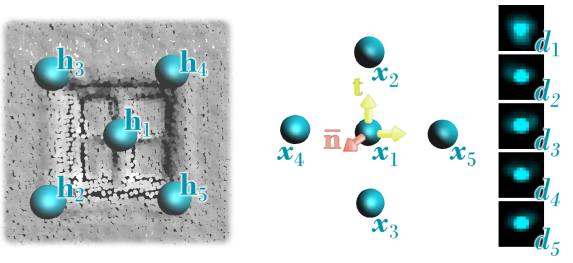


Figure 3: *Each shape H is associated with a spatial layout X and local shapes $\mathbf{d}_i$. The layout is given by relative coordinates $\mathbf{x}_i$ of the individual parts $\mathbf{h}_i$ in a coordinate frame centered at the first part. The local shape is a collection of the local shape descriptors of the shape parts.*

3D Object Recognition: Segmentation and labeling of geometry according to semantic categories has been tackled by using CRF models [ATC*05,KHS10,ZLZ*10]. The application to large scenes is limited because all labels have to be estimated in a global optimization problem. Another alternative is learning the geometry of segmented shapes [GKF09]. The objective of these approaches is to consistently segment and classify geometry, not to detect objects.

A large amount of work has been devoted to the recognition of single objects (see for example [MSS*06], and [DGD*11] for a survey), not detecting instances within a larger scene. Furthermore, fixed models of deformations have also been studied for matching template shapes to data (typically isolated objects) under isometry or different types of elastic deformation (see [vKZHCO11] for a survey). In contrast, our method aims at learning the variability from training data.

Most related to our method is the work of Sunkel *et al.* [SJW*11]: They present a part-based CRF model based on a Markovian chain of features. In contrast to their work, we employ a full pairwise correlation model and an according inference algorithm, and our model provides full rotational invariance. We demonstrate empirically that the model improvements lead to significantly better results in practice.

3. Shape Model

Each shape model characterizes an object class by encoding similarities of shapes from this class. As illustrated in Fig. 3, we define shapes as a set of correlated parts. Each part i consists of a relative position $\mathbf{x}_i$ and the local shape description $\mathbf{d}_i$. The individual parts are subsumed into local shape D and their overall spatial layout X. The shape model $\theta = (\theta_D, \theta_X)$ is then defined by Gaussian models $\theta_D \sim \mathcal{N}(\mu_D, \Sigma_D)$ and $\theta_X \sim \mathcal{N}(\mu_X, \Sigma_X)$ over D and X, thus encoding dependencies (*correlations*) of the individual parts.

3.1. Probabilistic Model

In the following, let $\mathcal{S} \subset \mathbb{R}^3$ be a smooth manifold embedded in three-space. We use $\mathbf{n}(\mathbf{x})$ to denote the surface normal at point $\mathbf{x} \in \mathcal{S}$. Typically $\mathcal{S}$ is represented by a sampled approximation (*point cloud*) $S = \{\mathbf{s}_1, ..., \mathbf{s}_n\}, \mathbf{s}_i \in \mathbb{R}^3$, acquired by 3D scanners and thus subject to noise artifacts.

Given a shape model θ with k parts, our goal is to find reasonable assignments of the k parts to points in S: $H = (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_k)$, where $\mathbf{h}_i \in S$ denotes the position of part i. The detection problem can be formulated as a maximum a posteriori hypothesis search over the joint posterior distribution of H and manifold evidence S. Our objective is to maximize the following probability:

$$p(D, X, H|\theta) = \underbrace{p(D|H, \theta_D)}_{\textit{Local Shape}} \cdot \underbrace{p(X|H, \theta_X)}_{\textit{Layout}} \cdot \underbrace{p(H|\theta)}_{\textit{Prior}} \quad (1)$$

It consists of a term accounting for the descriptors (*local shape*) and their constellation in 3D (*layout*). In addition, a prior distribution $p(H|\theta)$ can be used to model additional constraints on the detection. In our experiments, it is assumed to be uniform over $\mathcal{S}$.

3.1.1. Spatial Layout

Since Gaussian models cannot represent rotations well, we encode the spatial layout X relative to a local coordinate frame $\mathbf{T}$. The reference frame $\mathbf{T}$ is spanned by the first two parts $\mathbf{h}_1, \mathbf{h}_2$ and the smoothed surface normal $\bar{\mathbf{n}}$ at $\mathbf{h}_1$, as illustrated in Fig. 3. We compute a tangent vector $\mathbf{t} = (\mathbf{h}_2 - \mathbf{h}_1) \times \bar{\mathbf{n}}$ and set $\mathbf{T} = (\bar{\mathbf{n}} \times \mathbf{t}\ ,\ \mathbf{t}\ ,\ \bar{\mathbf{n}})$. The $3(k-1)$-dimensional layout vector $X(H)$ is then given by:

$$X(H) = (\mathbf{T}(\mathbf{h}_2 - \mathbf{h}_1), \mathbf{T}(\mathbf{h}_3 - \mathbf{h}_1), ..., \mathbf{T}(\mathbf{h}_k - \mathbf{h}_1)) \quad (2)$$

After this normalization, we model the spatial layout of model parts as a joint Gaussian distribution over the relative coordinates of $X(H)$, with mean μ_X and covariance Σ_X:

$$p(X|H, \theta_X) \sim \exp(-\frac{1}{2}(X(H) - \mu_X)^T \Sigma_X^{-1} (X(H) - \mu_X)) \quad (3)$$

3.1.2. Local Shape

Local shape D is modeled by a joint Gaussian density on $(d \cdot k)$-dimensional vectors $D(H)$ composed of all k d-dimensional local shape descriptors:

$$p(D|H, \theta_D) \sim \exp(-\frac{1}{2}(D(H) - \mu_D)^T \Sigma_D^{-1} (D(H) - \mu_D)) \quad (4)$$

Again, μ_D represents the (learned) mean and Σ_D the covariance of the descriptors, again including all cross-correlations between the local shapes captured by descriptors of all of the different parts.

Our framework is independent of the actually chosen shape descriptor. Any function $S \to \mathbb{R}^d$ can be used. The descriptors used in our experiments are described in Section 5.1.

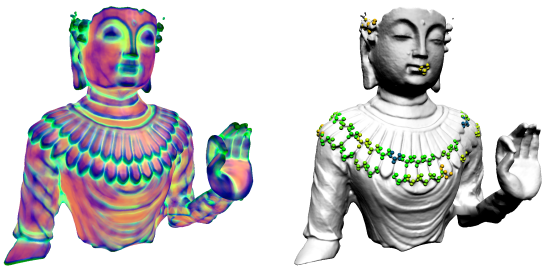


Figure 4: *Shapes under arbitrary rotations. The first three principle components of the descriptor are illustrated on the left. The shape model was constructed using the four exemplars shown in Fig. 1, however allowing for less variations. Detection results are shown on the right. Colors indicate the quality of the match (blue: perfect; red: bad).*

3.2. Learning

In our scenario the model is manually defined: For a number of shapes the user labels all correspondences between the shapes – thus defining the parts. The ability to manually select meaningful parts is important for many applications that can use these coarse correspondences as input, like replacing found objects with a template. Whereas, parts automatically chosen by some objective function might not be the ones desired by the user.

The model parameters $\theta = (\mu_D, \Sigma_D, \mu_X, \Sigma_X)$ are learned by supervised training. This is done by specifying a sparse set of corresponding points on objects of interests. Given the training instances, mean and covariance are estimated for both local shape and spatial layout.

For small sets of training instances, the learned covariance matrices are rank deficient. For example, a principal component analysis (*PCA*) extracts at most a 3-dimensional space from 4 examples. Hence, the variability is underestimated, and whole subspaces are falsely assumed to be noise-free in such cases. Even if in theory the actual class is described sufficiently by a few dimensions, inaccuracies such as scanner noise typically make a covariance of full rank inevitable in practice. We model these unmeasured effects by a uniform Gaussian noise model, i.e. by adding $\lambda\mathbf{I}$ to the covariance matrix, where λ is a user-controllable parameter. In certain scenarios it might also be necessary to amplify (or attenuate) the variations learned from the training data. For instance, if we want to detect objects of different sizes but have only observed two sizes so far. This is implemented by scaling the observed covariances.

The final covariances are given by:

$$\Sigma_X = \gamma_X \Sigma_X^{obs.} + \lambda_X \mathbf{I} \quad \text{and} \quad \Sigma_D = \gamma_D \Sigma_D^{obs.} + \lambda_D \mathbf{I}.$$

3.3. Hierarchical Shape Models

We propose a simple extension of our model, which allows the detection to be performed hierarchically, using detected constellations as parts of higher level constellation models. The motivation is two-fold: First, many complex shapes can be described by constellations of simpler base shapes. By training part models separately, fewer training examples are required for estimating good model parameters. Secondly, finding constellations of constellations allows us to recognize structural relations between parts (such as windows being arranged on a regular grid), which permits the extraction of additional information.

As shown in Fig. 1, the hierarchical extension is straightforward: First, several different base models are constructed as described in the preceding sections. Then, further instances are detected in the model and offered to the user as additional feature points to be selected for composite models. The position is set to the centroid of the found instance and the local shape is obtained by concatenating all local shape descriptors of the base instance (followed by a reduction to the original d dimensions of a local shape via principle component analysis). The class of the base model is also used to discriminate feature points – the class used during detection must match the trained one.

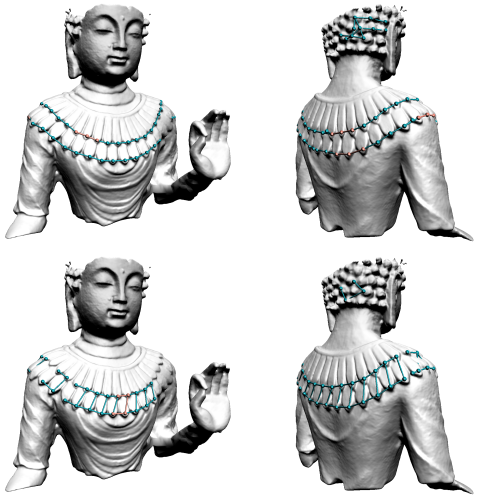


Figure 5: *More hierarchical detections for the statue. The hierarchical descriptors shown in Fig. 1 were used. Instances used to train the model are marked red.*

4. Shape Inference

We now need to find instances of the model defined in the previous section. Given an input point cloud S, we want to retrieve all local maxima with significant density of $p(H|D,X,\theta)$. Obviously, the log-likelihood of this density is non-convex in any non-trivial case.

Traditionally, inference in constellation models is done by expectation maximization (EM) [FPZ03] or Markov-chain Monte-Carlo sampling (MCMC) [SGS09], but these techniques are slow and can only compute one solution at a time. Another option is a restriction to tree-structured models [FH05], allowing for exact inference, or even further restrictions to star [LLS04] or chain models [SJW*11].

We use a greedy dynamic programming scheme for approximately retrieving the local maxima of $p(H|D,X,\theta)$. Algorithmically our inference scheme is related to belief propagation for chain structured models [SJW*11]. However, instead of only considering direct predecessors, we incorporate dependencies to all predecessors. Omitting these dependencies in the model would allow for solving for local optima exactly, but at the cost of a weaker model. We examine the effect of including these global dependencies in Section 5.

Like for chain-structured models, as used in [SJW*11], we successively compute sets of partial assignments. Since complete enumeration is not feasible (i.e., of exponential effort), the key idea is to restrict evaluation to those assignments which are likely to be part of an actual instance. Given a set of candidate assignments for the first i parts of the model, denoted by $\mathcal{H}_i \subset S^i$, we form augmented assignments for the first $i+1$ model parts. For each possible value $\mathbf{h}_{i+1}$ of part $i+1$, we search for the best partial assignment (in terms of maximizing Eq. 1) to be combined with $\mathbf{h}_{i+1}$:

$$\mathcal{H}_i(\mathbf{h}_{i+1}) = \underset{(\mathbf{h}'_1,\ldots,\mathbf{h}'_i)\in\mathcal{H}_i}{\arg\max}\; p(\mathbf{h}'_1,\ldots,\mathbf{h}'_i,\mathbf{h}_{i+1}) \qquad (5)$$

We form the set of candidate assignments $\mathcal{H}_{i+1}$ by combing all $\mathbf{h}_{i+1} \in S$ with their corresponding best matching assignment $\mathcal{H}_i(\mathbf{h}_{i+1})$:

$$\mathcal{H}_{i+1} = \left\{ (\underbrace{\mathbf{h}_1,\ldots,\mathbf{h}_i}_{=\mathcal{H}_i(\mathbf{h}_{i+1})},\mathbf{h}_{i+1}) \middle| \mathbf{h}_{i+1} \in S \right\}; \quad \mathcal{H}_1 = S \qquad (6)$$

Once the candidate assignments for part k have been computed, we perform a local maxima search to retrieve the final detections.

By just keeping track of the current best estimates, we might lose track of a desired instance in favor of a seemingly better but wrong match. Luckily, real-world data is typically benign, as demonstrated in Section 5, since fixing the first few parts imposes substantial restrictions on the remaining ones. Accordingly, we will optimize for the order in which the parts are processed, as detailed in Section 4.2.

4.1. Efficiency

Even though we have reduced the complexity from exponential (for the naïve but exact evaluation) to quadratic costs in $|S|$, our algorithm is still too slow for large, real-world scenes with several million points.

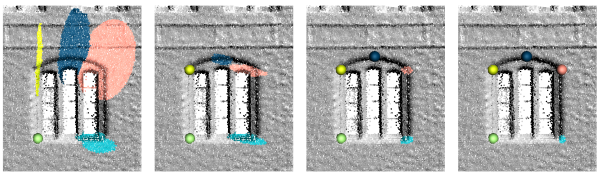


Figure 6: *Effect of covariance updates. Potential locations for the remaining parts after fixing preceding parts (for purpose of illustration the orientation of the shape was fixed). Fixing the second part of the shape already decreases the horizontal variance drastically.*

Our goal is to retrieve instances with significant probabilities; accordingly we can discard all values $\mathbf{h}_i$ for part i if their local shape or relative position does not match the model at all. We regard $\mathbf{h}_i$ as a potential value for part i if its local shape has a Mahalanobis distance of at most 2 to the local mean shape of part i, thus reducing the set of initial candidates for each part drastically, see Fig. 7. Similarly, $(\mathbf{h}_1,\ldots,\mathbf{h}_i)$ is only included in the set of candidates $\mathcal{H}_i$ during detection, if $\mathbf{h}_i$ adds at most 2 to the overall Mahalanobis distance to each model mean.

Another improvement concerns the evaluation of Eq. 1: The incremental algorithm requires repetitive evaluation for partial assignments $(\mathbf{h}_1,\ldots,\mathbf{h}_i)$. We can reduce computational effort from $O(i^2)$ to $O(1)$ if we update the model using the Schur complement:

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right)$$

$$Y_2|Y_1 \sim \mathcal{N}\left(\tilde{\mu},\tilde{\Sigma}\right)$$

with $\tilde{\mu} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(Y_1 - \mu_1)$ and $\tilde{\Sigma} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$.

The update incorporates restrictions caused by the previously assigned parts to the updated model. Further, the updated covariance matrices are independent of the ongoing inference and thus can be precomputed. Since the dependencies on previous parts are encoded in the updated models, they can be used for a more efficient pruning. The effect of these model updates (for the spatial layout) is illustrated in Fig. 6, giving a hint of the effectiveness of these updates.

4.2. Planning Inference Order

In order to improve the performance of the inference, both in terms of accuracy and speed, we use a planning step that determines the order in which part hypothesis are tested. The goal of the planning is to first search for parts for which the location has the least uncertainty. This has two benefits:

- The search space is reduced, due to the pruning of unlikely hypothesis (Section 4.1), thereby improving the run-time.

Figure 7: *Effect of planning. The set of initial candidates for each part, shown in the order obtained by planning.*

- By fixing the parameters for the part with the least uncertainty first, the risk of propagating wrong information to later stages of the search is reduced. This is important since our heuristic algorithm does not use backtracking or backward propagation of information to part hypothesis tested earlier.

We will demonstrate empirically (see Section 5) that planning has a significant impact on both of these aspects, improving both on run-time costs and accuracy.

Planning itself is easy: We pick those parts first for which the uncertainty in localization is minimal. Again, we use a greedy optimization algorithm to make the choices: The first part is selected by descriptor uniqueness: We match the descriptor model against the whole training set and choose the part with the lowest matching frequency, i.e., whose descriptor matches the fewest other points (again, using a threshold of a Mahalanobis distance of 2). For choosing subsequent parts, we now need to model the influence of the previous choices. Specifically, both the ambiguity in terms of descriptor match as well as variation in spatial localization should be minimized. For the locality, we compute the marginal of the Gaussian model: we estimate a marginal covariance in position, given we fix the previous plan-points. For the descriptor, the improvement depends both on how frequent the descriptors are expected in the input as well as on the correlations with previously detected part descriptors. We therefore multiply the frequency of the part descriptor by the relative change in volume in descriptor space due to correlations. This shrinking of descriptor volume is modeled by the ratio of the determinants of the unconstrained marginal descriptor covariance and the marginal descriptor covariance obtained after fixing the already selected plan-points. A typical result obtained from the planning step is show in Fig. 7.

5. Results and Evaluation

We evaluate the performance of our shape model and inference by using LIDAR range-scans from the Hannover city scan collection (available at http://www.ikg.uni-hannover.de, courtesy of C. Brenner, IKG, University of Hannover). In addition, we also use 3D scans of figurines to demonstrate rotational invariance as well as the hierarchical variant of our method. All experiments are performed using an unoptimized single-core C++ implementation on an dual socket workstation with two Intel Xenons X5650 (2.6GHz) and 48GB of RAM.

5.1. Local Shape Descriptors

The local shape descriptor is used to characterize the geometry within the r-neighborhood $N_r(\mathbf{x}) = \{\mathbf{y} \in S | \|\mathbf{x}-\mathbf{y}\| \leq r\}$ of a point $\mathbf{x}$ in S.

In order to effectively assess the correlated parts model, we do our experiments without the use of sophisticated shape descriptors such as in [FHK*04], [CSM*06], [KPW*10]. Our experiments on large point cloud data require descriptors which come with low computation costs and yield a robust descriptor for small, almost planar, surface regions (which does not hold for spin images [JH99]). We have implemented the following shape descriptors:

- *Normal histograms*: For every point $\mathbf{y} \in N_r(\mathbf{x})$, we express the normal direction $\mathbf{n}(\mathbf{y})$ in polar coordinates with respect to a coordinate frame defined by the smoothed normal $\overline{\mathbf{n}}(\mathbf{x})$ and $\mathbf{y}-\mathbf{x}$. We then build a joint histogram of the two angles using 15×15 bins.
- *Oriented normal histograms:* Here, we augment the normal histograms by using a fixed reference frame given by $\overline{\mathbf{n}}$ and a fixed global upward direction $\mathbf{u}$.

For efficiency reasons and in order to smooth out noisy data, the set of high-dimensional descriptors in an input scene is projected onto a low-dimensional subspace using principal component analysis.

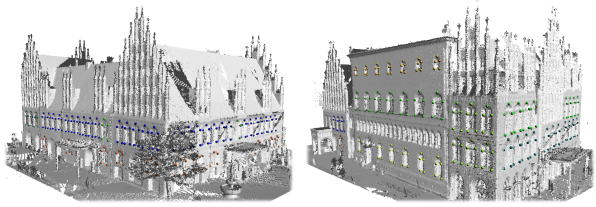


Figure 8: *Manual annotation of the old town hall used for evaluation. All windows are used for the general class, while the specific class only subsumes the windows marked in blue.*

5.2. Quantitative Evaluation

For a quantitative evaluation of different aspects of our method we manually annotated two different test sets on the old town hall, as shown in Fig. 8, ranging from a very specific class, capturing the most prominent window type, to a general class, comprising all types of windows present in the building. We count the number of false positives and negatives by a coarse criterion that measures the distance of the centroids of the detection hypotheses to the centroids of ground truth data. A detected instance only counts as a true positive if this distance to the nearest ground truth data is smaller than the descriptor radius r employed to compute the local shapes. We use cross-validation for measuring the performance; we always use 3 examples per type and average over 8-10 stratified random samples; we precompute a random partition that guarantees to cover all examples at least once. Curves are measured by varying one of the model parameters while keeping the others fixed.

5.3. Shape Model Experiments

Rotation invariance and hierarchical shape model: We demonstrate the rotation invariance of our approach in Fig. 4. The statue model features a deformed, regular pattern in different orientation, not captured by a common "upward direction", which is need for the method of [SJW*11] to work. We also examine the use of hierarchical models (see Figs. 1, 5, 15), improving the recognition accuracy and yield a structuring of the instances in terms of a regularly repeating grid.

Effect of correlations between parts: We first evaluate the effect of including all pairwise correlations of the parts' local shapes into our model (which are not included in the traditional constellation model [FPZ03]). Curves are measured by varying descriptor noise parameter λ_D, describing the tolerance of a fit to noise and unmodeled effects. The results are shown in Fig. 9: When learning a complex class for different windows in the old town hall, the detection performance improves by including these correlations. For the class of rigidly similar windows, this effect is less pronounced.

Secondly, we compare to different underlying graph structures, i.e. dependencies of relative locations of parts: Both, our full model as well as constellation style model include all pairwise correlations of parts locations. Star shaped models, as used in [FMR08, LLS04], only consider spatial relations to one center part. Chain structured models, as employed by [SJW*11], merely consider pairwise correlations between a part and its predecessor. In order to compare the different model structures, we restrict our more general model appropriately by removing accordant pairwise interactions. The results are shown in Fig. 10. We varied covariance scale γ_D, to observe the behavior under shape models of different flexibility. The underlying structure of the shape is captured better the more dependencies are included.

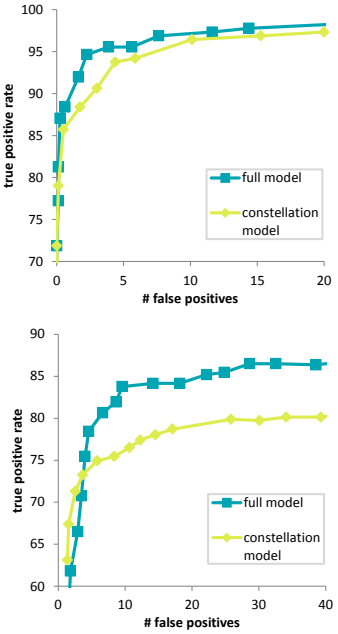


Figure 9: *The detection rate improves by including correlations of descriptors. The upper diagram shows results for the specific class of rigidly similar windows, the lower for the more general class containing more varying geometry.*

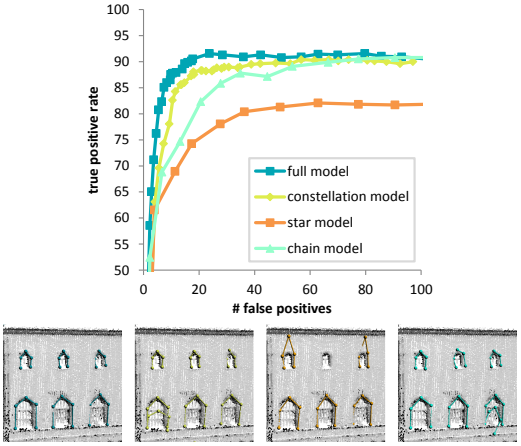
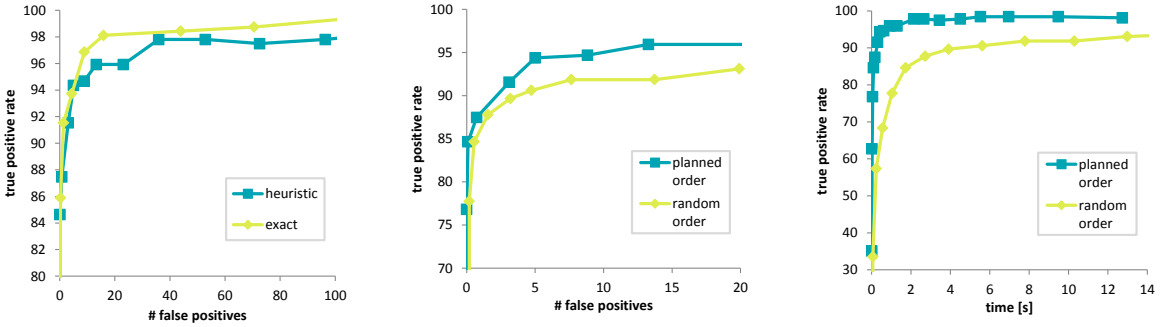


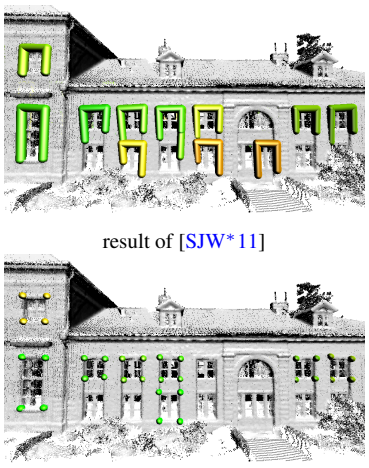
Figure 10: *Evaluation of different model structures. The excerpts show typical results for the different model types. Results are shown for similar numbers of false positives.*

We also compare our method qualitatively to Sunkel *et al.* [SJW*11] (Fig. 11). Our method yields more accurate correspondences. The global correlations avoid drift over the course of the chain. For applications in computer graphics, beyond pure detection, this is an additional benefit.



exact vs heuristic inference | planning vs random order | effect of planning on timings

Figure 12: *Evaluation of the inference scheme. For a very narrow class (where exact inference is feasible) exact inference only performs slightly better than our approximate inference (left). Curves are measured by varying the noise parameter λ_D, describing the tolerance of a fit to noise and unmodeled effects. Effect of planning: the recognition performance improves over the average of a random order (middle). The runtime improves significantly (right). Curves are measured by varying λ_D.*



result of [SJW*11]

our result

Figure 11: *Comparison to previous work by Sunkel et al. [SJW*11]. Because of the lacking global correlations, lower boundaries of the windows do not match up, which our model avoids. For both examples identical training instances are used. Edges encode the chain used in [SJW*11].*

5.4. Inference Experiments

Effect of approximate inference: In order to quantify the error imposed by the approximate inference, we have implemented an exact version of our algorithm. To make exact inference feasible, a very specific model (small variances) is required since this allows for efficient pruning: We use the specific class (Fig. 8) and assume very little noise. The results are show in Fig. 12 (middle). As expected, the exact version yields slightly better results, but the gain of the exponential algorithm is below 3% (please note the scale!).

Effect of Planning: We also study the effect of planning (see Fig. 12). We compare to the average of a random order for inference, again for finding windows in the "old town hall". The recognition rate improves consistently by up to 4%. The effect on the run-time is more dramatic: As shown in Figure 12 (right), we obtain high detection rates much more rapidly than without planning.

Scalability: We apply our method to all facades from the Hannover data set (126 million sample points, 4GB binary data). Since we do not want to compute descriptors for the complete set, we reduce the point cloud to a set of interest points. This is done by extracting points of high curvature, using the technique of Gumhold *et al.* [GWM01], i.e., using the smallest eigenvalue of a PCA-analysis of local neighborhoods as curvature measure. Only these points are considered as candidates for points in H, the rest of the method remains unchanged. We denote the reduced point set by $\tilde{S}$.

Fig. 13 shows an example where 35 windows are used for training, retrieving a large subset of the actual windows in the rest of the town with few false positives. Our inference algorithm runs in less than 2 minutes (on the reduced set), statistics are shown in Figure 14. We also vary the scene size by cutting out excerpts: the run-time scales almost perfectly linear with scene size (Figure 14).

6. Conclusion and Future Work

We have presented an object detection technique for 3D scans that is based on correlating layout and local shape of object parts. We have also introduced a new algorithm for simultaneous detection of many object instances. Empirically, the new model of including all correlations leads to a significant improvement. In addition, we also improve over previous work in terms of frame invariance and by providing

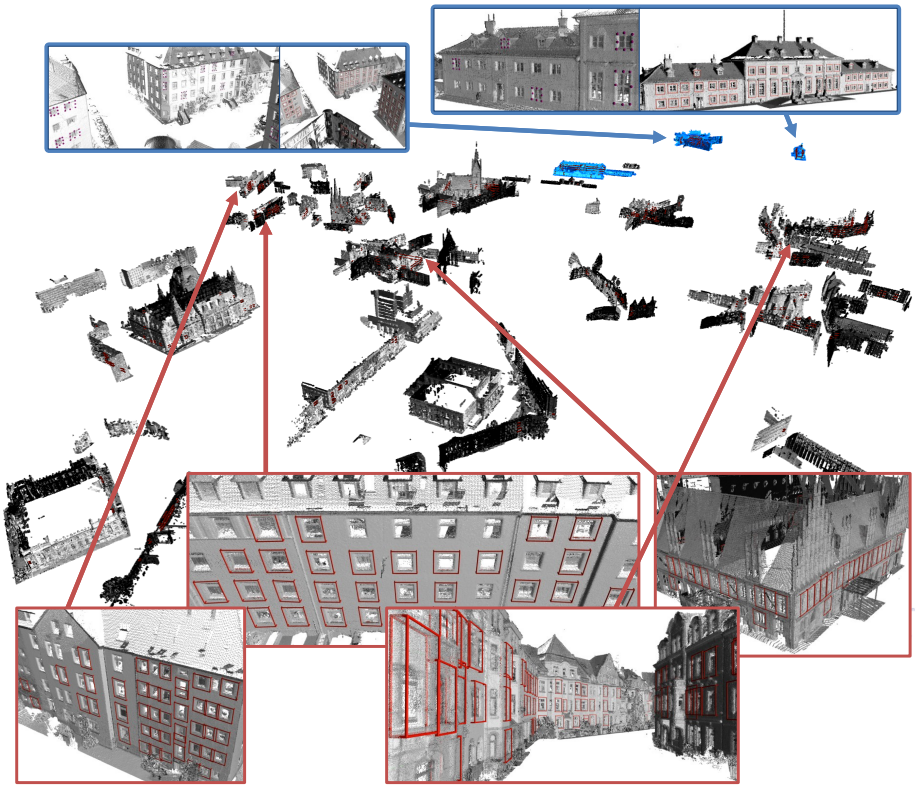


Figure 13: *Large scale result: The complete set of Hannover scans (126 million points, 4GB of raw data). We train 35 examples of windows, all taken from the buildings marked in blue (16 million points). We are able to detect a substantial fraction of further instance of the "window" class on the rest of the data set. The computation time is approx. 2 minutes.*

Figure 15: *Hierarchical detections for the crocodile.*

a hierarchical matching model to reduce combinatorial redundancy which can in addition be used to obtain a natural structuring of the scene. Nonetheless, our algorithm is fast and scalable; a single-threaded implementation can retrieve sets of object instances in large 3D scans with more than 100 million points within 2 min, a figure that has also not yet been shown in literature.

In our experiments, we have deliberately chosen to employ simple, basic descriptors to focus our study on the effect of the improved correlated parts model and the approximate inference. Even then, we already obtain remarkable results, such as discovering a large number of windows in a city scene with few false positives from a rather small training set, that have not been demonstrated previously. First experiments with integrating various different descriptors as in [KHS10] indicate potential for improvement.

Acknowledgments

This work has been supported by the Cluster of Excellence "M2CI". The authors wish to thank the anonymous reviewers for their valuable comments, Claus Brenner and the IKG Hanover for making their data available.

	$\|S\|$	$\|\tilde{S}\|$	# found
train	16,193,592	1,013,301	249
test	110,763,979	8,645,128	672
all	126,957,571	9,658,429	921

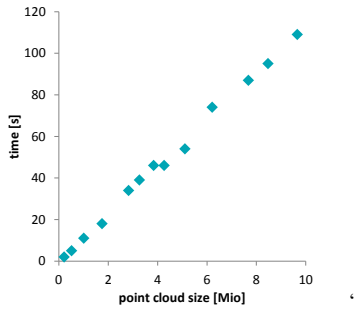


Figure 14: *Statistics for the complete Hannover scan.*

References

[ASS*09] AGARWAL S., SNAVELY N., SIMON I., SEITZ S. M., SZELISKI R.: Building rome in a day. In *ICCV* (2009). 1

[ATC*05] ANGUELOV D., TASKAR B., CHATALBASHEV V., KOLLER D., GUPTA D., HEITZ G., NG A.: Discriminative learning of markov random fields for segmentation of 3d scan data. In *CVPR* (2005). 2, 3

[BBGO11] BRONSTEIN A. M., BRONSTEIN M. M., GUIBAS L. J., OVSJANIKOV M.: Shape Google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30* (February 2011). 2

[CSM*06] CORREA S. R., SHAPIRO L., MEILA M., BERSON G., CUNNINGHAM M., SZE R.: Symbolic signatures for deformable shapes. In *PAMI* (2006). 6

[DGD*11] DUTAGACI H., GODIL A., DARAS P., AXENOPOULOS A., LITOS G. C., MANOLOPOULOU S., GOTO K., YANAGIMACHI T., KURITA Y., KAWAMURA S., FURUYA T., OHBUCHI R.: Shrec '11 track: Generic shape retrieval. In *EG 3DOR Workshop* (2011). 3

[DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *CVPR* (2005). 2

[FGG*10] FRAHM J.-M., GEORGEL P., GALLUP D., JOHNSON T., RAGURAM R., WU C., JEN Y.-H., DUNN E., CLIPP B., LAZEBNIK S., POLLEFEYS M.: Building rome on a cloudless day. In *ECCV* (2010). 1

[FH05] FELZENSZWALB P., HUTTENLOCHER D.: Pictorial structures for object recognition. *IJCV 61*, 1 (2005). 2, 5

[FHK*04] FROME A., HUBER D., KOLLURI R., BULOW T., MALIK J.: Recognizing objects in range data using regional point descriptors. In *ECCV* (2004). 6

[FMR08] FELZENSZWALB P., MCALLESTER D., RAMANAN D.: A discriminatively trained, multiscale, deformable part model. In *CVPR* (2008). 2, 7

[FPZ03] FERGUS R., PERONA P., ZISSERMAN A.: Object class recognition by unsupervised scale-invariant learning. In *CVPR* (2003), pp. 264–271. 2, 5, 7

[GAF*10] GOESELE M., ACKERMANN J., FUHRMANN S., KLOWSKY R., LANGGUTH F., MUECKE P., RITZ M.: Scene reconstruction from community photo collections. *IEEE Computer 43*, 6 (2010). 1

[GD05] GRAUMAN K., DARRELL T.: The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV* (2005). 2

[GKF09] GOLOVINSKIY A., KIM V. G., FUNKHOUSER T.: Shape-based recognition of 3D point clouds in urban environments. *ICCV* (Sept. 2009). 2, 3

[GWM01] GUMHOLD S., WANG X., MACLEOD R.: Feature extraction from point clouds. In *Proc. Meshing Roundtable* (2001). 8

[HZRCP04] HE X., ZEMEL, R.S., CARREIRA-PERPINAN M.: Multiscale conditional random fields for image labeling. In *CVPR* (2004). 2

[JH99] JOHNSON A., HEBERT M.: Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE PAMI 21* (1999), 433 – 449. 6

[KH03] KUMAR S., HEBERT M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In *ICCV* (2003). 2

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3d mesh segmentation and labeling. *ACM Trans. Graph. 29*, 3 (2010). 2, 3, 9

[KPW*10] KNOPP J., PRASAD M., WILLEMS G., R.TIMOFTE, GOOL L. V.: Hough transform and 3d surf for robust three dimensional classification. In *ECCV* (2010). 6

[LG07] LI X., GUSKOV I.: 3d object recognition from range images using pyramid matching. In *ICCV, Workshop on 3D Representation for Recognition* (2007). 2

[LHS07] LEORDEANU M., HEBERT M., SUKTHANKAR R.: Beyond local appearance: Category recognition from pairwise interactions of simple features. In *CVPR* (2007). 2

[LLS04] LEIBE B., LEONARDIS A., SCHIELE B.: Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV* (2004). 2, 5, 7

[LSP06] LAZEBNIK S., SCHMID C., PONCE J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR* (2006). 2

[MPWC12] MITRA N. J., PAULY M., WAND M., CEYLAN D.: Symmetry in 3d geometry: Extraction and applications. In *EUROGRAPHICS State-of-the-art Report* (2012). 2

[MSS*06] MATEI B., SHAN Y., SAWHNEY H., TAN Y., KUMAR R., HUBER D., HEBERT M.: Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation. *PAMI 28* (2006), 1111 – 1126. 3

[SGS09] STARK M., GOESELE M., SCHIELE B.: A shape-based object class model for knowledge transfer. In *ICCV* (2009). 2, 5

[SJW*11] SUNKEL M., JANSEN S., WAND M., EISEMANN E., SEIDEL H.-P.: Learning line features in 3d geometry. In *Proc. Eurographics* (April 2011), vol. 30. 2, 3, 5, 7, 8

[vKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 3

[WZFF06] WANG G., ZHANG Y., FEI-FEI L.: Using dependent regions for object categorization in a generative framework. In *CVPR* (2006). 2

[ZLZ*10] ZHAO H., LIU Y., ZHU X., ZHAO Y., ZHA H.: Scene understanding in a large dynamic environment through a laser-based sensing. In *ICRA* (2010). 2, 3