Optimal Alignment of 3D Data for Spatial Discretization

Benjamin Huhle, Timo Schairer, Andreas Schilling, Wolfgang Straßer
Department of Graphical Interactive Systems (GRIS)
University of Tübingen
Tübingen, Germany
{*huhle, schairer*}*@gris.uni-tuebingen.de*

***Abstract*—We present an algorithm that finds the rotation which best aligns a given configuration of directions on an unsorted set of directions. Using a cost function that we derive in the paper, we show that this algorithm efficiently finds the optimal coordinate frame for a given 3D data set in order to maximize the performance of axis-aligned bounding volumes such as octrees or occupancy grids with regard to the average storage or query cost. The optimization is independent from the initial orientation, since the whole space of rotations is explored. The evaluation becomes computationally feasible by using the Fourier transforms of spherical functions that describe the data as well as the cost function.**

***Keywords*-orientation estimation; bounding volumes; octrees; spherical harmonics**

I. Introduction

For a robot to interact with its environment, it is essential to have a map of the environment at its disposal. Such maps are ideally acquired without human intervention and stored in an efficient representation. A variety of such representations exist in the literature. In this paper we consider the simple, but very basic, first step of aligning such maps or any other 3D data acquired from the environment of the robot, prior to any discretization and further processing. We present an algorithm that finds the best orientation of the data, such that the quantization volume of 3D objects in axis-aligned data structures is minimized. This way the performance of such data structures is improved with regard to storage and query costs.

The proposed method allows us to fit an orthogonal coordinate frame to the data such that the surface normals coincide with the new axes as well as possible. We show that, in this new basis, various discrete data structures are more effective than when they are built from arbitrarily rotated data (cf. Figure 1). Apart from minimized storage and query costs, the reduced quantization volume can help in higher level tasks such as path-planning due to an increased amount of free space. Furthermore, the resulting frame can be considered the inherent orientation of the model.

The performance gain of the proposed method, of course, depends on the data itself. The more predominate orthogonal structures in the model are, such as in models of man-made objects, the more it benefits from this alignment.

The *Principal Component Analysis* (PCA) is often used to align data, yet it does not solve the same problem. PCA aligns the data such that the variance along the axes is maximized. This way, simple bounding boxes are perfectly aligned to the data. For regular grids or hierarchical data structures, however, this alignment does not fit well in general.

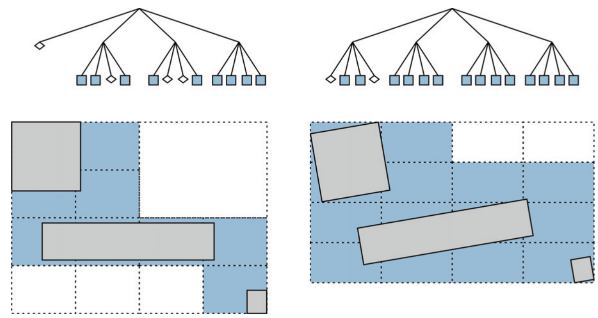


Figure 1. 2D Toy Example: Consider a map consisting of three rectangles. An optimized orientation (left) leads to minimal quantization volume (number of cells) and a more compact tree representation (top left), compared to an arbitrary orientation (right).

Similarly, methods that solve the *(Extended) Orthogonal Procrustes Problem* ([1], [2], [3]) which are used in *Iterated Closest Point* (ICP) algorithms, are not applicable in our case. Since correspondences between data and template are not given a priori, we effectively have to solve the additional combinatoric data association problem, assigning each data vector to its best fitting axis. This way, we can also understand the method as a clustering algorithm, where the structure of the cluster (the configuration of the directions) is known, and only the orientation of the clusters as well as the affiliation of the data is sought-after.

Note that the resulting orientations are unique up to different labelings of the axes, *i.e.*, up to arbitrary rotations of $\pi/2$ around any of the coordinate axes. This is due to the fact that no semantic information such as gravity, or any specific knowledge about the objects is used. Obviously, the quality of the resulting bounding volumes is invariant to these rotations.

In the next section we give a short overview of related work. In Section III we will describe the general alignment algorithm and in Section IV the application to the axis-alignment of 3D data is investigated with regard to the performance gain of different hierarchical data structures.

II. Related Work

Podolak et al. [4] introduced the *Principal Symmetry Axes* as an alignment feature of a model. The alignment is determined by large parts of an object that show significant symmetry, and it often matches the inherent coordinate frame that a human would select. Chaouch and Verroust-Blondet [5] propose a method that searches for an orientation of a 3D model such that directions of reflective and translational symmetry are aligned with the coordinate axes.

Related to the alignment of directional data is also the work by Daniilidis [6] or Kamgar-Parsi et al. [7] who consider the alignment of sets of 3d line segments. However, they share the same starting point, where correspondences are known a priori. Alternatively, a common approach is to randomly assign potential correspondences and select the hypothesis that produces the best match (*e.g.* [8], [9]).

In contrast, we do not assign correspondences, but test for a large set of potential orientations. This becomes computationally feasible due to the very efficient evaluation in the Fourier domain, introduced by Kostelec and Rockmore [10]. Based on the same technique, different algorithms for 3D model retrieval [11] and for *pairwise* registration of 3D models were developed [12], [13].

III. Alignment of Directional Data

In order to find the optimal orientation of a fixed configuration of directions (the *template*), we first map the data and the template to the unit sphere. We use the technique of kernel density estimation to model the distribution of the data based on a set of measurements D. The template has to be chosen according to the application-dependent cost function. Both modeling steps are described in detail in the following two subsections. Further, we minimize the correlation of the template and the density function with regard to their relative orientation. This part of the algorithm is described in the third subsection.

A. *Density Estimation on the Sphere (*The Data*)*

To model the data as well as the template, we use the spherical harmonics representation of functions on the 2-sphere. Driscoll and Healy [14] showed that the spherical harmonic functions Y_l^m form a complete orthonormal basis over the unit sphere and that any square-integrable function $f \in L^2(S^2)$ can be expanded as a linear combination of spherical harmonic functions (Spherical Fourier Transform, SFT)

$$f(\omega) = \sum_{l \in \mathbb{N}} \sum_{m \in \mathbb{Z}, |m| \leq l} \hat{f}_l^m Y_l^m(\omega), \tag{1}$$

with expansion coefficients $\hat{f}_l^m \in \mathbb{C}$. We refer to $\omega = (\theta, \phi)$ as a point on the sphere with colatitude $\theta \in [0, \pi]$ and azimuth $\phi \in [0, 2\pi)$. The spherical harmonic function Y_l^m

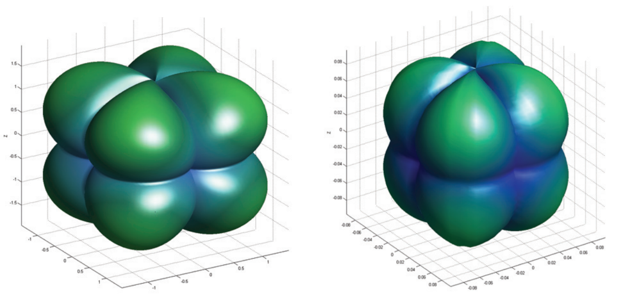


Figure 2. Spherical plot of the template function (left) and of the total volume (right, cf. IV-A1) of a model consisting of three orthogonal squares (depicted for verification). The function values are mapped to distances from the origin.

of degree l and order m is given by

$$Y_l^m(\theta, \phi) = \sqrt{\frac{(2l+1)(l-m)!}{4\pi(l+m)!}} P_l^m(\cos\theta) \exp(im\phi), \tag{2}$$

with P_l^m denoting the associated Legendre polynomials.

To perform kernel density estimation on the sphere, we use the spherical equivalent of a Gaussian kernel (see [15]; we will informally refer to it as a *spherical Gaussian* for simplicity) and accumulate rotated versions that are centered at the data points. The unrotated spherical harmonics representation of this rotationally symmetric kernel (w.r.t. the z-axis) of width σ, can be expressed as a *zonal harmonic*

$$\hat{z}_l^0 = \sqrt{\frac{2l+1}{4\pi}} \exp\left(\frac{-l(l+1)\sigma^2}{2}\right), \tag{3}$$

$$\hat{z}_l^m = 0 \quad \textit{for } m \neq 0. \tag{4}$$

A zonal harmonic can be rotated very efficiently [16] by multiplying it with a spherical harmonic

$$\hat{f}_l^m = \sqrt{\frac{4\pi}{2l+1}} \hat{z}_l^0 Y_l^m(d), \tag{5}$$

such that the new z-axis points to the direction $d = (\theta, \phi) \in D$.

B. *The Spherical Cost Function (*The Template*)*

For two sets of directional data D and C, the function

$$F(R) = \sum_{d_j \in D} \sum_{v_i \in C} k_{d_j}(R\, v_i), \tag{6}$$

where k_d denotes a spherical Gaussian centered at d, measures how well rotation R aligns both sets. This function can be evaluated by correlating the data density function (derived in the preceding subsection) with an appropriate sampling function, which we will call *template* in the following.

When we are interested in the inherent coordinate system of a model, it is reasonable to construct the template in the same way as the data density function using the axis directions, *i.e.* the set

$$\left\{ \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}-1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\-1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\0\\-1\end{bmatrix} \right\} \tag{7}$$

as input. This solves a smoothed version of Equation 6.

Since we search for the coordinate system that specifically minimizes the quantization volume, we derive a cost function that is explicitly tailored to this problem. In the following we consider angles in $[0, \pi/4]$ only, which however is generalized easily. In 2D, the quantized representation of a line of length ℓ with angle α in average requires a number of

$$n(\alpha) = \frac{\ell\ (cos\alpha + sin\alpha)}{\tau} \tag{8}$$

cells with side length τ. The extremal values are $n(0) = \frac{\ell}{\tau}$ and $n(\pi/4) = \sqrt{2}\,n(0)$. The extension to 3D is obtained as the integral of such lines over the third dimension. The projection onto this direction is computed analogously, resulting in a size of cells

$$n(\phi, \theta) = n(\phi)\ n(\theta) \tag{9}$$

to represent an $\ell \times \ell$-square with angles ϕ and θ to the reference coordinate system. Thus, the cost factor with regard to the alignment is

$$t(\phi, \theta) = (\cos\phi + \sin\phi)(\cos\theta + \sin\theta), \tag{10}$$

which varies between a maximum value of 2 at multiples of $\pi/4$ for ϕ and θ, and a minimum value of 1 at multiples of $\pi/2$. The resulting template which we used in all experiments, *i.e.*, the projection of t to the sphere is depicted in Figure 2.

Depending on the application, different templates could be chosen, *e.g.*, for grasping or packing problems.

C. Minimizing the Correlation

The constrained maximization of Equation 6 with regard to the relative orientation is done by minimizing the correlation of the data function and the template function.

Kostelec and Rockmore [10] presented an effective method to evaluate the cross-correlation of two spherical functions for a whole grid of relative rotations. They showed that the correlation

$$C(R) = \int_{S^2} g(\omega)\, \Lambda(R) h(\omega)\, d\omega \tag{11}$$

of two functions g and h can efficiently be evaluated in the Fourier domain. Here, Λ denotes the rotation operator corresponding to the rotation $R = R(\alpha, \beta, \gamma)$ where α, β, γ are the Euler angles (in *ZYZ* representation) defining the rotation. Further, the spherical harmonic functions Y_l^m form an orthonormal basis for the representations of $SO(3)$ and the $SO(3)$ Fourier transform (*SOFT*) coefficients of the correlation of two spherical functions can be obtained directly

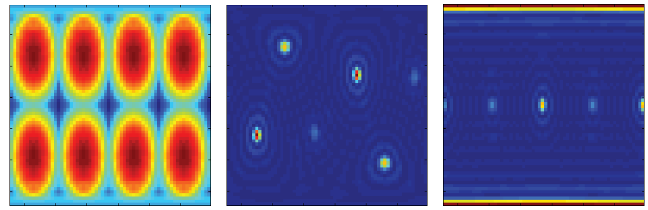


Figure 3. Latitude/longitude-mappings (equirectangular projection) of PSB model # 765 (desk), showing the template (left) and the data functions with random (center) and optimized (right) orientation.

by calculating the bandwise outer product (denoted by $\diamond$) of their individual SFT coefficients. Taking the inverse *SOFT*,

$$C(R) = \text{SOFT}^{-1}\left(\hat{g} \diamond (\hat{h})^{\star}\right), \tag{12}$$

where $(\hat{h})^{\star}$ denotes the complex conjugate of $\hat{h}$, yields the correlation $C(R)$ evaluated on the $2B \times 2B \times 2B$ grid of Euler angles G.

Using this approach, we can perform a global search for the extremum. The accuracy of the rotation estimate

$$\tilde{R} = \arg \min_{(\alpha,\beta,\gamma)\in G} C(R(\alpha, \beta, \gamma)), \tag{13}$$

however, is limited by the resolution of the grid which in turn is specified by the number of bands used in the SFT. Given spherical harmonics of bandwidth B, the resolution of the grid implicates an inaccuracy of up to $\pm(\frac{180}{2B})^{\circ}$ in α and γ, and $\pm(\frac{90}{2B})^{\circ}$ in β. The cubic computational cost when evaluating the grid, in practice, restricts this method to bandwidths up to $B = 256$. For all results in this paper, we used $B = 32$.

To refine the estimate indicated by the minimum grid value, we perform a direct non-linear optimization on the correlation function, evaluated in the SFT domain, as proposed in [17]. Following an idea by Makadia et al. [18], for the optimization, the rotation can be divided into two parts:

$$R(\alpha, \beta, \gamma) = R_2(\beta + \pi, \frac{\pi}{2}, \gamma + \frac{\pi}{2}) R_1(\alpha + \frac{\pi}{2}, \frac{\pi}{2}, 0). \tag{14}$$

This way, the computational cost is significantly reduced since the argument of the Wigner-d matrix $d_{km}^l(\beta)$ is constant and the matrices have to be evaluated for $\beta = \frac{\pi}{2}$ only:

$$\begin{aligned} C(R) \;=\; & \sum_l \sum_{m=-l}^{l} \sum_{p=-l}^{l} \sum_{k=-l}^{l} \hat{g}_l^m\ (\hat{h}_{pl})^{\star}\ d_{pk}^l\left(\frac{\pi}{2}\right)\ d_{km}^l\left(\frac{\pi}{2}\right) \\ & \exp\left(-i(p\gamma' + k\beta' + m\alpha')\right). \end{aligned} \tag{15}$$

Minimizing this function using a steepest gradient method with line search, we obtain the refined angles $\alpha' = \alpha + \frac{\pi}{2}$, $\beta' = \beta + \pi$ and $\gamma' = \gamma + \frac{\pi}{2}$.

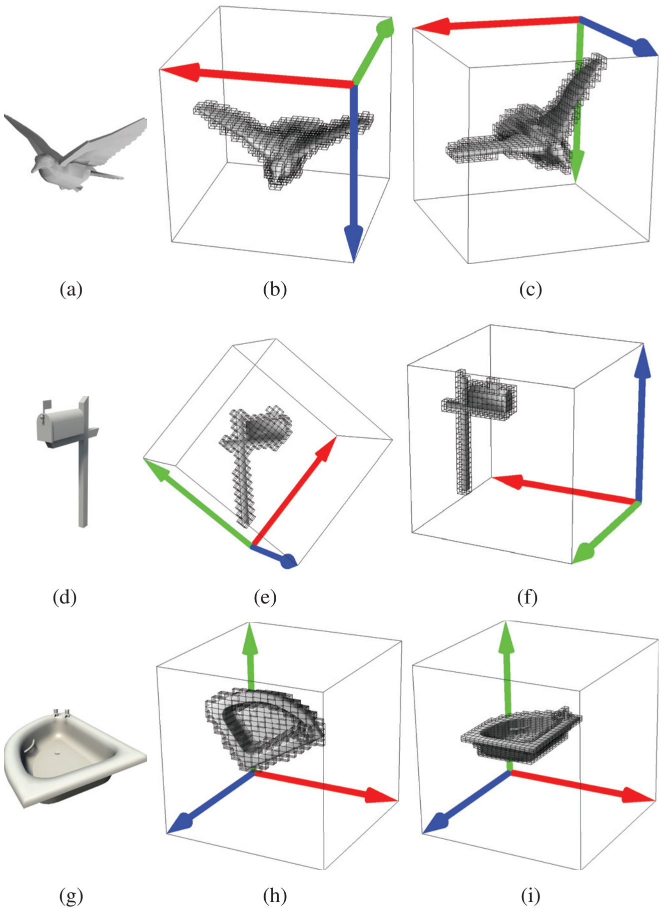


Figure 4. Exemplary octrees for three models of the PSB, resulting from the PCA orientation (middle) and from the proposed alignment (right). The resolution of the octrees is lowered for better visualization.

	initial	PCA	ours
total volume	100	92.1	**87.3**
query cost	100	90.7	**86.9**
storage cost	100	92.3	**87.9**

Table I
AVERAGE OCTREE QUALITY FOR DIFFERENT ORIENTATIONS OF THE MODELS OF THE PRINCETON SHAPE BENCHMARK. NORMALIZED W.R.T. THE INITIAL ROTATION.

		worst	PCA	ours
Bremen City	# cells (percentage)	9477 100	8911 94.0	**8724** **92.0**
D. Borrmann, J. Elseberg (Jacobs University, Bremen)	run length code (percentage)	3693 100	3338 90.4	**3151** **85.3**
Bremen Sea Front	# cells (percentage)	9660 100	9445 97.7	**9307** **96.3**
D. Borrmann, J. Elseberg, A. Nüchter (Jacobs University, Bremen)	run length code (percentage)	3353 100	3014 89.9	**2567** **76.6**
AASS Department	# cells (percentage)	5758 100	5132 89.1	**5077** **88.2**
M. Magnusson (University of Örebro)	run length code (percentage)	1691 100	1210 71.6	**1200** **70.9**
Freiburg Campus	# cells (percentage)	4278 100	**3785** **88.5**	3804 88.9
B. Steder (University of Freiburg)	run length code (percentage)	1556 100	1066 68.5	**1046** **67.2**
Rescue Arena	# cells (percentage)	27183 100	23320 85.8	**22819** **83.9**
D. Borrmann, J. Elseberg, A. Nüchter (Jacobs University, Bremen)	run length code (percentage)	9219 100	5247 56.9	**4884** **53.0**
Dagstuhl Castle	# cells (percentage)	5379 100	4654 86.5	**4622** **85.9**
A. Nüchter, K. Lingemann (University of Osnabrück)	run length code (percentage)	1551 100	**1218** **78.5**	1226 79.1

Table II
GRID QUALITY W.R.T. ORIENTATION FOR DIFFERENT REAL-WORLD POINT CLOUDS.

IV. EXPERIMENTS

We tested the alignment algorithm on two types of data. We chose (1) the *Princeton Shape Benchmark* (PSB, [19]) consisting of a set of 1814 synthetic 3D models represented as triangular meshes and (2) a selection of real-world point clouds from the *Robotic 3D Scan Repository* [20], supplemented by the *Freiburg Campus* dataset of Steder [21]. The data functions are composed using surface normals as input. In the case of the mesh-models, the normals are computed from the triangular faces and we weighted their contributions by the area of the respective triangle, *i.e.*, approximating each triangle by a square of the same size. In the case of the scanned data, we use PCA on the set of 20 nearest neighbors in order to estimate the normals in each point. Here, no weights are assigned which corresponds to the assumption that the surfaces are homogeneously sampled.

Every dataset was quantized into a regular grid, similar to a discrete occupancy grid, yet without the distinction of *unknown* versus *empty* cells. In order to allow for a fair comparison, octrees are subdivided up to a constant maximum depth d ($d = 7$ in all experiments), and for the octree as well as for the regular grid, we chose cubical (leaf) cells of constant size. To compensate for random discretization effects, all results are averaged over eight tests, where for each test we translate the data slightly (randomly, in the range of one grid cell).

A. Criteria

To quantify the results, we compute different measures of the data structures which determine the performance of many applications based on the data:

1) The total volume: measures the summed volume of the occupied cells. Since we are dealing with representations of surfaces, the volume should ideally be close to zero.

2) The query cost: for the mesh-based models, counts the average number of primitive tests that have to be performed when an arbitrary position in the bounding box of the model is chosen, *i.e.*, we sum over the products of the volume of

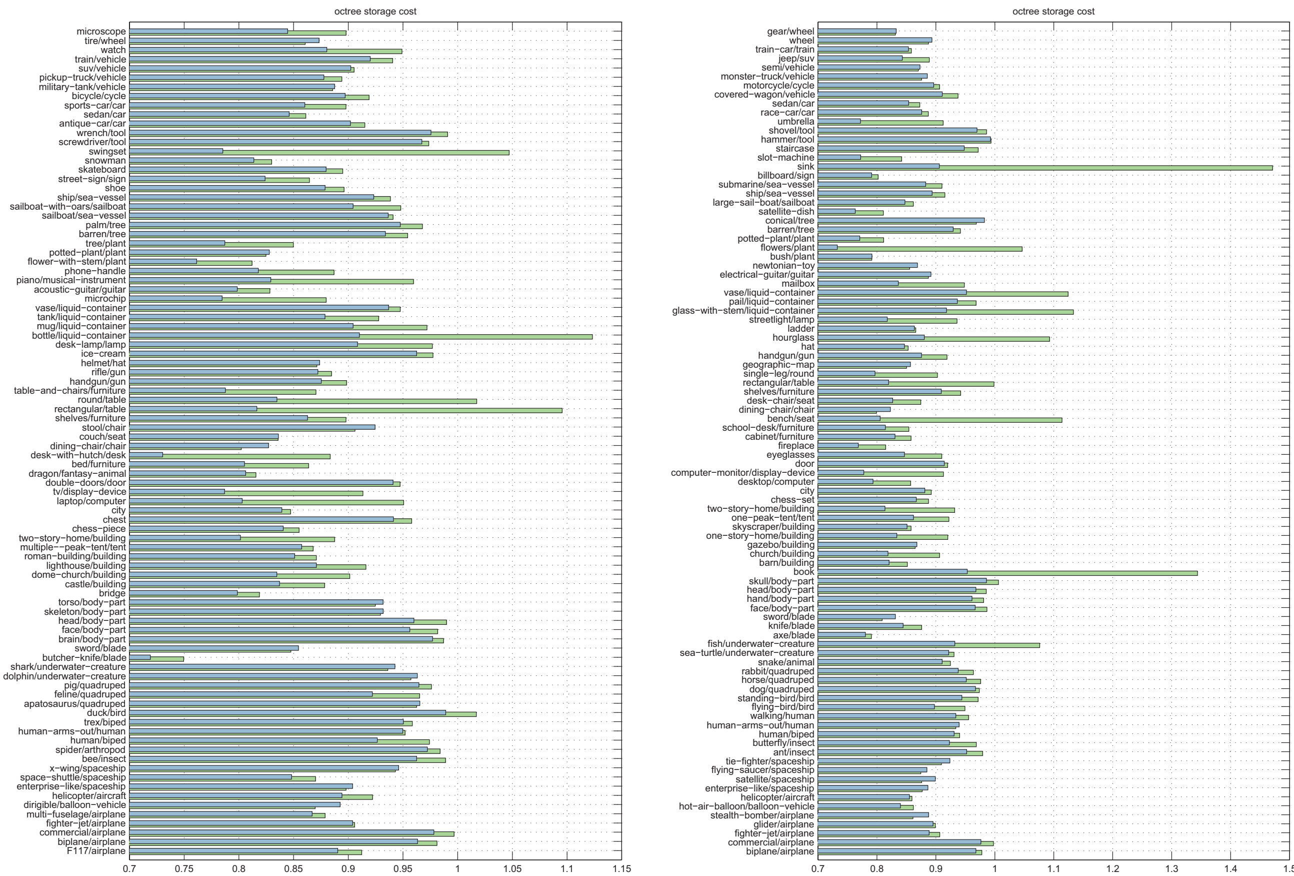


Figure 5. Average octree storage cost for the Princeton Shape Benchmark, according to the training (left) and test (right) class definitions. The values are normalized to a random initial orientation. We compare our algorithm (blue) with an alignment based on PCA (green), using a constant size for the leaf cells and for the maximum depth of the tree.

each cell with the number of faces that intersect this cell. The cost of traversing the tree is neglected.

3) The storage cost: using the lossless compression scheme of Wurm et al. [22] for the mesh models is given in bit. The point clouds are run length encoded and we output the minimum number of runs of the three canonical linearizations of the grid. Both code lengths are computed without further processing such as, *e.g.*, entropy encoding.

B. Results

Figure 3 exemplarily depicts the alignment process of a mesh-based model. The equirectangular projection of the data function exhibits accumulation points of certain directions. After the optimization of the orientation of the model, the positions of these clusters are rotated onto the minima of the template function and therefore are co-aligned with the coordinate axes.

The models of the PSB were rotated to a random initial orientation. Figure 4 shows the resulting alignment of the model with the cell boundaries of the octrees for three of the models. The overall alignment results are summarized in Table I: Our alignment method improves the performance of the octree representation by more than twelve percent and outperforms PCA. A detailed evaluation is presented in Figures 5 and 6. As expected, the more a model exhibits planar structures and orthogonal relations predominate, the more the alignment improves the performance. This equally holds true for the total volume measure which is omitted here for brevity.

The results of the evaluation of our method using the real-world point cloud data is presented in Table II. The gain in total volume and storage cost is lower than for the mesh-based models. This is due to the inhomogeneous sampling of the surfaces in the scenes and the error-prone estimation of the normals. We believe that a more sophisticated algorithm for normal estimation could help to further improve the alignment.

In Figure 7, for each measure, we plot the actual performance gains with regard to the orientation of the data. The sampled rotations are ordered by the values of the cost

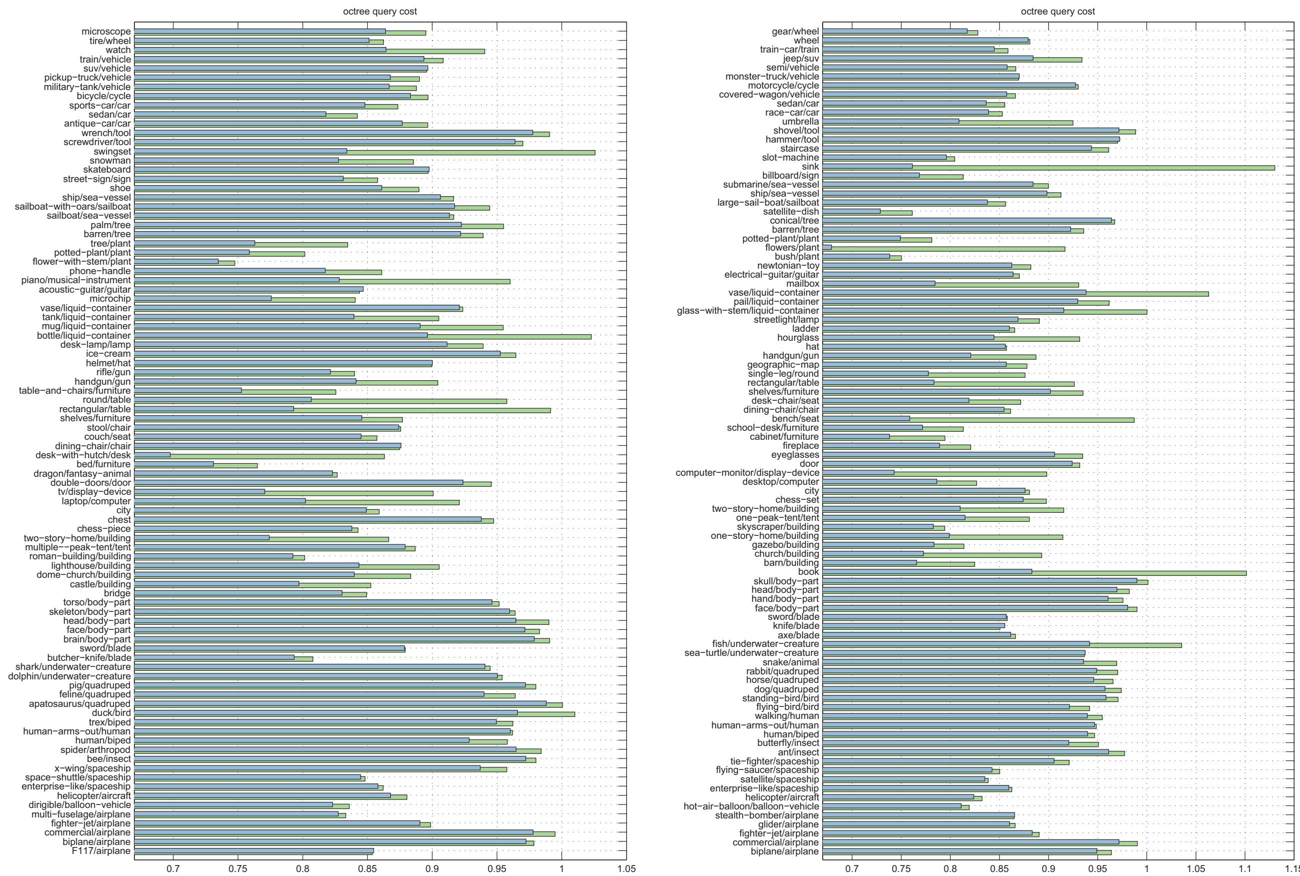


Figure 6. Average query cost – cf. Fig. 5.

function, showing the dependence of the correlation values and the quantization quality. It can be seen that the maximum benefit from a good alignment depends on the dataset. For example, the Freiburg Campus data is very challenging for our algorithm due to the numerous trees present in the scans.

In general, the alignment algorithm not only improves the performance of different bounding volumes, but can also be considered a method to find the inherent reference system of a 3D model. We provide the same selection of models from the PSB for comparison with Chaouch et al. [5] in Figure 8.

V. Conclusion

We presented a generally applicable algorithm for the alignment of directional data. We derived a cost function that allows us to optimize the quantization of a given 3D dataset by selecting an optimal orientation. Due to the evaluation of the whole rotation group in its Fourier domain, the method robustly finds the global optimum and the additional non-linear optimization further improves the precision of the result. A thorough evaluation on diverse publicly available datasets validates this approach.

Our algorithm is easily included into the processing pipeline of various applications and helps to improve the performance of data structures that are based on axis-aligned bounding volumes. Furthermore, the alignment also improves the handling of the data in general, *e.g.*, for postprocessing or human interaction.

Acknowledgment

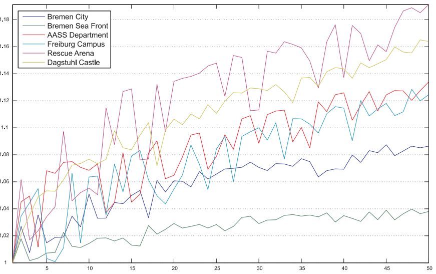
We thank Sebastian Herholz for his contribution to the implementation. The authors gratefully acknowledge financial support by the German Research Foundation (DFG).

References

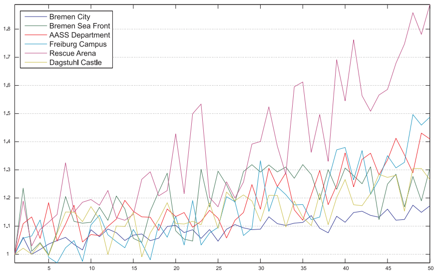
[1] P. H. Schönemann, “A Generalized Solution of the Orthogonal Procrustes Problem,” *Psychometrika*, vol. 31, 1966.

[2] B. K. P. Horn, “Closed-form Solution of Absolute Orientation using Unit Quaternions,” *J. of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

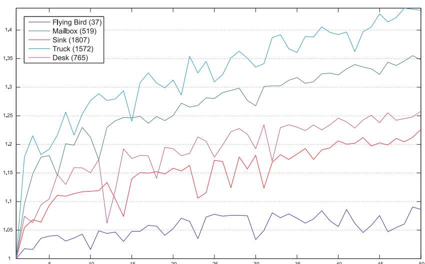
[3] K. Arun, T. Huang, and S. Blostein, “Least Squares Fitting of two 3-D Point Sets,” *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI)*, vol. 9, pp. 698–700, 1987.



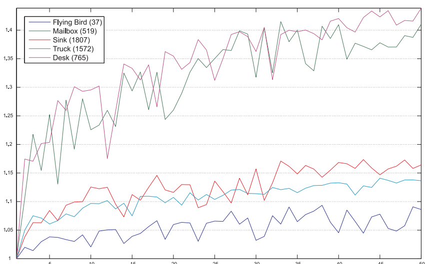
(a) total volume



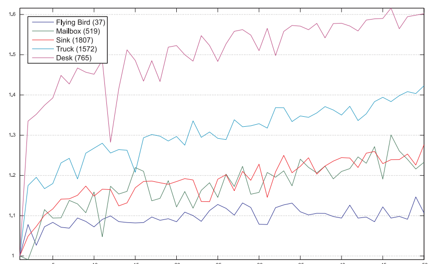
(b) storage cost (run length encoding)



(c) total volume

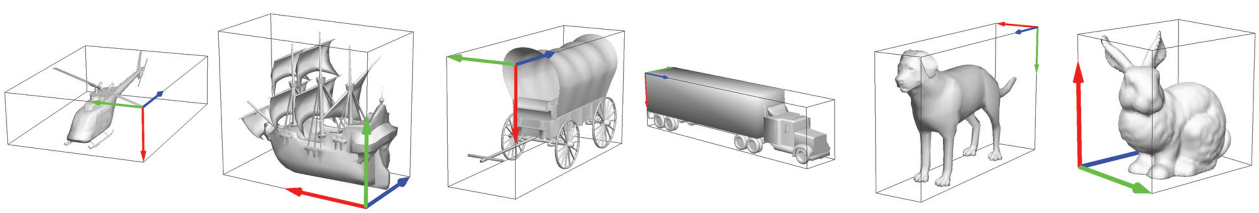


(d) octree query cost

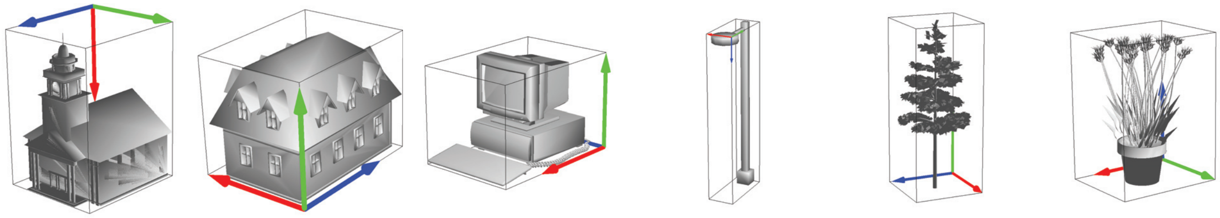


(e) octree storage cost

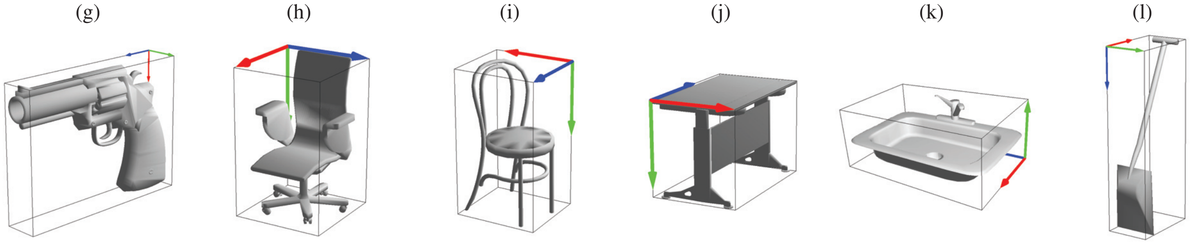
Figure 7. The performance measures for the 3D scans (top) and the PSB models (bottom) are plotted w.r.t. different orientations, each normalized to its minimum. We took a sample of 50 arbitrary rotations and sorted them according to the respective values of the cost function.



(a) (b) (c) (d) (e) (f)



(g) (h) (i) (j) (k) (l)



(m) (n) (o) (p) (q) (r)

Figure 8. A selection of models from the Princeton Shape Benchmark together with the estimated coordinate frames (for comparison with [5]).

[4] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser, "A Planar-Reflective Symmetry Transform for 3D Shapes," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 25, pp. 549–559, July 2006.

[5] M. Chaouch and A. Verroust-Blondet, "Alignment of 3D Models," *Graphical Models*, vol. 71, pp. 63–76, 2009.

[6] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," *Int. J. of Robotics Research*, vol. 18, pp. 286–298, 1999.

[7] B. Kamgar-Parsi and B. Kamgar-Parsi, "Algorithms for Matching 3D Line Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 582–593, 2004.

[8] C. Guerra and V. Pascucci, "On Matching Sets of 3D Segments," in *Proc. Conf. Vision Geometry*, 1999.

[9] H. Chen and T. Huang, "Matching 3-D Line Segments with Applications to Multiple-Object Motion Estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 1002–1008, 1990.

[10] P. Kostelec and D. Rockmore, "Ffts on the rotation group," Fe Institutes Working Paper Series, Tech. Rep., 2003.

[11] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Symposium on Geometry Processing (SGP)*, 2003.

[12] M. Kazhdan, "An Approximate and Efficient Method for Optimal Rotation Alignment of 3D Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1221–1229, 2007.

[13] J. Fehr, M. Reisert, and H. Burkhardt, "Fast and accurate rotation estimation on the 2-sphere without correspondences," in *European Conference on Computer Vision (ECCV)*, ser. LNCS, vol. 5303, 2008, pp. 239–251.

[14] J. Driscoll and D. M. Healy, Jr., "Computing fourier transforms and convolutions on the 2-sphere," *Adv. Appl. Math.*, vol. 15, no. 2, pp. 202–250, 1994.

[15] T. Bülow, "Spherical Diffusion," University of Pennsylvania, Philadelphia, USA, Tech. Rep. MS-CIS-01-38, 2001.

[16] P.-P. Sloan, "Stupid Spherical Harmonics (SH) Tricks," in *Game Developers Conference*, 2008.

[17] T. Schairer, B. Huhle, and W. Straßer, "Application of particle filters to vision-based orientation estimation using harmonic analysis," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.

[18] A. Makadia and K. Daniilidis, "Rotation recovery from spherical images without correspondences," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1170–1175, 2006.

[19] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The Princeton Shape Benchmark," in *Shape Modeling International, Genova, Italy*, 2004.

[20] A. Nüchter and K. Lingemann, "Robotic 3D Scan Repository," February 2011. [Online]. Available: http://kos.informatik.uni-osnabrueck.de/3Dscans/

[21] B. Steder, "Freiburg Campus," February 2011. [Online]. Available: http://www.informatik.uni-freiburg.de/ steder/datasets.html

[22] K. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems," in *Proc. ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.