Vision, Modeling, and Visualization (2012)
M. Goesele, T. Grosch, B. Preim, H. Theisel, and K. Toennies (Eds.)

Optimized Canonical Coordinate Frames for 3D Object Normalization

Michael Martinek[1], Roberto Grosso [1] and Günther Greiner[1]

[1]Univerity of Erlangen-Nuremberg, Germany

Abstract

In this paper, we describe a method to optimize an orthogonal system of axes for 3D objects in order to perform normalization with respect to orientation and scale. An energy function evaluates the quality of a system by considering symmetry, rectilinearity and the origin of the system within the current axis aligned bounding box. Starting with the PCA-axes as initial system, we find a canonical coordinate frame by minimizing the energy in an efficient and elaborate optimization process. We provide a fully automatic normalization pipeline with the possibility to manually set various intuitive parameters in order to influence the outcome. The symmetry part of our energy function uses a combination of plane reflective and rotational symmetries. In this context, we introduce a novel continuous symmetry measure which is entirely implemented on the GPU. The high efficiency of the implementation enables us to find an optimal alignment for 3D objects interactively, making our method suitable even for large 3D databases. We also demonstrate the applicability of our framework for 3D shape matching by approximating the Hausdorff distance for 3D models.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling —Geometric algorithms, languages, and systems

1. Introduction

Normalizing 3D objects with respect to translation, rotation and scaling is an important pre-processing step in many different tasks such as shape analysis, visualization and robotics. Many shape matching and retrieval methods, for example, have to rely on a robust alignment of similar objects before extracting shape features or measuring similarity [PPPT07, TV04, NK01].

An essential task in order to normalize 3D objects is to find a canonical coordinate frame for each object (see Figure 1) and the most common method to do this is the principle component analysis (PCA). It uses the barycenter as center and computes the system of orthogonal axes along which the variance of the vertices is maximal. Variants of this method take the triangle areas (CPCA) or the surface normals (NPCA) into account. However, the simplicity of the computation opposes the lack of robustness. PCA-axes are known to be very sensible to small object perturbations as it is illustrated for example in [FMK*03, BKS*05].

In this approach, we introduce an elaborate energy function to evaluate the quality of an object's coordinate frame. The major parts of the energy function are a measure of plane reflective and rotational symmetries and a measure of rectilinearity which evaluates how well the faces of the object are aligned with the current axis aligned bounding box (AABB). A third term, the center energy, is used to control the origin of the othogonal system. An optimized system is

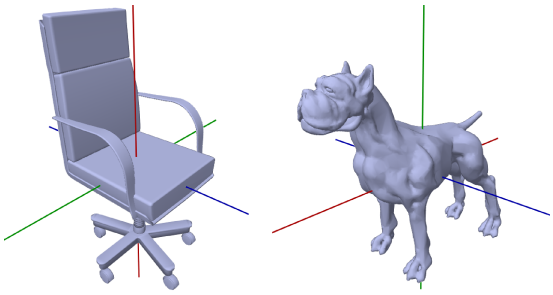

Figure 1: *Canonical coordinate frames as returned by our method.*

DOI: 10.2312/PE/VMV/VMV12/167-174

then found by minimizing the energy function in the space of rotations and translations. This optimization process can be done very efficiently due to the following properties of the energy function: Firstly, it behaves smooth when the current system is translated or rotated, allowing the application of gradient-based optimization methods. Secondly, a single evaluation of the energy function takes less than a millisecond due to an efficient GPU-implementation of the symmetry measure. The contributions of our work are:

- We introduce a novel continuous symmetry measure for plane reflective and rotational symmetries which can be implemented entirely on the GPU.
- We propose a hybrid energy function which measures the quality of a orthogonal coordinate frame for 3D objects by considering symmetric properties and rectilinearity.
- We present an efficient method to find optimized coordinate frames for 3D objects by minimizing the energy function in the space of rotations and translations.

The rest of the paper is organized as follows: in Section 2, we give an insight into previous methods. Section 3 proposes our continuous symmetry measure and its implementation. In Section 4, we describe the individual contents of our energy function while Section 5 deals with the optimization of this function to determine a canonical coordinate frame. Finally, Section 6 provides some results and discussion with respect to 3D object alignment and shows the applicability of our framework for shape matching and Section 7 concludes this paper.

2. Related Work

As already mentioned in the introduction, PCA is the most commonly used method to perform object normalization. The remainder of this Section summarizes existing methods which use more elaborate techniques.

2.1. Symmetry-based alignment

In [Fer00], Ferguson found out that symmetry plays an important role in orientation effects, leading many researchers to develop alignment algorithms based on symmetry. Tedjokusumo and Leow [TL06] use bilateral symmetry planes (BSP) to normalize the pose of 3D models. They find the plane with the largest amount of reflective symmetry and define the first BSP axis to be the direction on this plane along which the variance of the vertices is greatest. Chaouch and Verroust-Blondet [CVB09] exploit the properties of PCA axes to detect plane reflection symmetries and select axes based on the amount of symmetries. If the object has at most one reflectional symmetry, they introduce a measure of local translational invariance in order to find the remaining axes. While the methods above only deal with the orientation of the objects, Podolak et al. [PSG*06] also propose a center of the canonical system by introducing the center of symmetry. They define the three orthogonal alignment axes to be the normals of the three orthogonal planes with maximal reflectional symmetry and the center of symmetry to be the interSection point of these planes.

2.2. Projection-based alignment

The methods in this class find canonical coordinate frames by considering the orthographic projection areas of the object. Johan et al. [JLWI11] take the axis which renders the minimum projection area as first principal axis and then search for the minimum projection area along the directions perpendicular to the first axis. Napoleon and Sahbi [NS10] define the visual hull as the sum of the projection areas of three orthographic projections along the axes of an orthogonal system and pick the system out of different PCA-variants which renders the minimum visual hull for normalization. Lian et al. [LRS08] define a measure of rectilinearity as the ratio of the visual hull and the surface area. The maximum of this measure under all possible orthogonal systems is found by means of an evolutionary algorithm and the resulting axes are taken for object alignment.

2.3. Hybrid methods

A method which combines the benefits of different attributes was proposed by Fu et al. [FCODS08]. The goal of this work is not to find an entire canonical coordinate frame of the model, but to detect the plane on which the object would naturally stand upright. To achieve this, they consider attributes such as static stability, symmetry, parallelism and visibility.

The method we propose in this paper also falls in the category of hybrid methods. The focus thereby lies on symmetry and rectilinearity. While only reflective symmetry was considered in previous symmetry-based approaches, we also take rotational symmetries into account and show that they contribute valuable additional information to improve the normalization system. However, our objective function is not restricted to symmetry, but also comprises the rectilinearity. In this context we use an alternative way to get a measure of rectilinearity than the one in [LRS08], which is not based on projection areas but on the distribution of normal vectors.

3. Continuous Symmetry Measure

Measuring global symmetries of an object is a well studied topic and the principle behind it is to measure the correlation between the object and an accordingly transformed version of it. Existing approaches use the extended Gaussian image [SS97] or volumetric functions [KCD*02] to measure these correlations. We propose an efficient method using a precomputed distance field and a sample point representation of the object.

3.1. Pre-Processing Steps

In order to measure the distance between an object f and a transformed version f^*, the distance between each transformed sample of f^* and the closest point on f should be considered. As a distance metric, we use an integer metric such as the Manhattan distance instead of exact euclidean distances, which are exhaustive to compute.

The distance field we use is the one proposed in [MGG11]. An efficient boundary voxelization of the object is performed and the surrounding voxels are filled with the discrete distance values in a wavefront scheme using the Manhattan distance. Figure 2 illustrates a cut through a resulting distance field of a chair model. The dimension of the voxel-domain is 128^3 and the distance field is stored in a 3D texture, making it accessible form a shader. The final part of the pre-processing is to generate uniformly distributed sample points on the object. If the object is initially represented as a point cloud, for example from 3D scanner data, we can skip this step. The distance field computation from [MGG11] is also capable of processing point data. Furthermore, it is entirely GPU-implemented and the time consume of the process is in the magnitude of 10^{-3} seconds, including the voxelization.

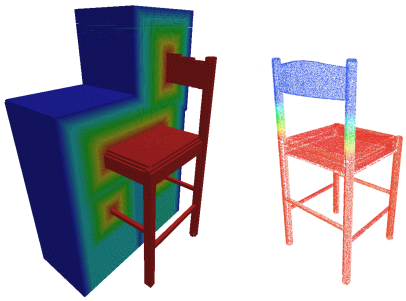

Figure 2: *Left: insight of a voxelized distance field around a chair object. The right image shows the sample points of a rotated version of the object rendered into the distance field.*

3.2. Symmetry distance

After the pre-process, we have a voxelized distance field around the object in the reference pose featuring M different distance layers and a representation of f in terms of N sample points. An approximate distance of a sample point in f^* to the closest point on f can now be obtained by looking up in which of the M distance layers the points lies. The right image of Figure 2 shows an example. The object was rotated by 180° and the sample points are color-coded to indicate the distance to the unrotated version of the object. Let m_i be the number of sample points located in distance layer i , $i \in [0,M]$ while a point outside the distance field is counted to m_M. We now define the distance $d(T)$ between f and its transformed version f^* as follows:

$$d(f,f^*) = \sum_{i=1}^{M} \frac{i}{MN} m_i \tag{1}$$

If f and f^* perfectly match each other, all sample points are located in distance layer 0 and equation 1 correctly returns the value 0.0. The other extreme occurs when all points of f^* are outside the distance field, returning a distance of 1.0. In between, the function gives a continuous measure of how much f and f^* match. We can re-formulate equation 1 as a sum over all 3D sample points $\mathbf{p_i}$ by introducing a function $l(\mathbf{p}) \in [0..M]$, which returns the distance layer of a point $\mathbf{p}$. We also parameterize the distance by the Transformation Matrix T, which maps f onto f^*.

$$d(T) = \sum_{i=0}^{N} \frac{l(T\mathbf{p_i})}{MN} \tag{2}$$

In order to measure plane reflective symmetry (PRS), we set T to the according mirror matrix. In the following, we will identify a plane through the origin with its normal vector. Thus, if we say PRS with respect to an axis $\mathbf{a}$, we mean a reflection at the plane to which $\mathbf{a}$ is a normal vector. The matrix for this reflection is denoted as $P^{\mathbf{a}}$ and the measure for PRS $S_{PRS}(\mathbf{a})$ is defined as follows:

$$S_{PRS}(\mathbf{a}) = \sum_{i=0}^{N} \frac{l(P^{\mathbf{a}}\mathbf{p_i})}{MN} \tag{3}$$

In contrast to PRS, rotational symmetry (RS) is ambiguous since there is an additional parameter in terms of the order of RS. An object is rotationally symmetric of order n with respect to an axis $\mathbf{a}$, if it looks the same after a rotation about an angle $2\pi/n$. However, we introduce a function which considers the best order for each individual sample point to get a measure which captures different orders of RS in a single value. The remaining question is which orders are to be considered. Obviously, every even-ordered RS implies RS of order two, so it would not make sense to take the orders $2n$ for $n > 1$ into account. In case of odd-ordered RS, we only consider the first two orders (i.e 3 and 5) since higher orders would imply nearly cyclic RS which is anyway covered by the other orders. The rotation angles corresponding to the three considered orders 2,3 and 5 are 180°,120° and 72° respectively. However, we measure RS of fifth order with an angle of 144° in our implementation since a rotation of the object with an angle of 72° would be too close to the original pose. Let $R^{\mathbf{a}}_j$ be the rotation matrix which performs a rotation around axis $\mathbf{a}$ with an angle of $2\pi/j$. We now introduce the symmetry measure for RS $S_{RS}(\mathbf{a})$ with respect to an axis $\mathbf{a}$ as the following equation:

$$S_{RS}(\mathbf{a}) = \sum_{i=0}^{N} \frac{1}{MN} \min_{j=2,3,5} l(R^{\mathbf{a}}_j \mathbf{p_i}) \tag{4}$$

The measure allows us to detect axes along which the object has rotational symmetries of different orders. An example is

provided in Figure 3. While the object does not have a global RS of a fixed order along the displayed axis, we find at least one perfect RS for each individual sample point, which we assigned a color according to the order of RS. The axis is correctly identified as perfect RS axis by our measure.

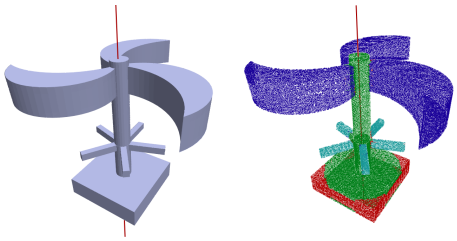

Figure 3: *Object with different local roational symmetries. The colors on the right indicate the order of the RS for each sample point. Red means perfect RS of 2nd order, blue of 3rd order, cyan of 5th order and green means cyclic RS.*

3.3. GPU Implementation

The design of the two symmetry functions from equations 4 and 3 are perfectly suitable for an evaluation on the GPU. In fact, given an orthogonal system of three axes $\mathbf{a_i}$ with $i = 1,2,3$, we can evaluate all of the six symmetries (one RS and one PRS for each of the three axes) in a single pass using the geometry shader. Since we eventually need six output values, we set the render target to a FBO of size 1×6. Having pre-computed the voxel-based distance field into a 3D texture, we can access it in the geometry shader so that the inner parts of the sums in equations 4 and 3 can be entirely evaluated there. This is done six times for each incoming sample point and a vertex, carrying the computed value, is emitted such that it is rendered into the respective fragment of the 1×6 render target. The blend stage adds the computed value for each measure to the value already in the FBO, eventually producing the required sums.
In our implementation, we use $N = 10,000$ and $M = 10$ and the evaluation of the all six symmetry measures takes less than 1 millisecond.

4. Energy of an Orthogonal System

We express an orthogonal system as tuple $\Omega = (\mathbf{c}, A)$ where $\mathbf{c}$ is a 3D Point denoting the origin of the system and A describes the three orthogonal directions $\mathbf{a_1}$, $\mathbf{a_2}$ and $\mathbf{a_3}$. The entire energy function $E(\Omega)$ consists of three parts, the symmetry energy E_{sym}, the rectilinearity energy E_R and the center energy E_C:

$$E(\Omega) = \alpha E_{sym}(\Omega) + \beta E_R(A) + \gamma E_C(\mathbf{c}) \qquad (5)$$

In the following, we describe the individual parts of this function.

4.1. Symmetry Energy E_{sym}

As can be seen in equation 5, the symmetry part E_{sym} depends on the entire System Ω since rotation axes as well as reflection planes are not fully defined by an axis alone, but also by a reference point. Thus, we have to perform a translation of the object such that $\mathbf{c}$ coincides with the origin before computing the 6 symmetry measures $S_{RS}(\mathbf{a_i})$ and $S_{PRS}(\mathbf{a_i})$ for $i = 1,2,3$. The remaining question is how to combine these individual measures to a single symmetric energy value in terms of avoiding potential redundancies. We therefore consider the fact that a 180°-rotation around an axis $\mathbf{a}$ is identical to a series of two plane reflections. The respective planes are orthogonal and contain the axis $\mathbf{a}$. This implies the following property:
Given an orthogonal system Ω and an object with perfect plane reflective symmetry with respect to the axis $\mathbf{a_1}$, the 180°-rotation around axis $\mathbf{a_2}$ is identical to the plane reflection with respect to $\mathbf{a_3}$ and the plane reflection with respect to $\mathbf{a_2}$ is identical to the 180°-rotation around $\mathbf{a_3}$. Another fact is that most man-made objects have at least one good place reflective symmetry which was empirically shown in [CVB09]. In order to avoid the redundancy of two identical pairs of measures in our energy function, we use the following strategy, which implicitly provides a robust ordering of the axes.
Let $\mathbf{a_r}$ be the axis rendering the minimum value for S_{PRS} over all three axes and let $\mathbf{a_s}$ be the axis with the minimum value of S_{RS} over the remaining two axes. The symmetric energy E_{sym} is defined as:

$$E_{sym} = \frac{\lambda(S_{PRS}(\mathbf{a_r}) + S_{RS}(\mathbf{a_r})) + (1-\lambda)(S_{RS}(\mathbf{a_s}) + S_{RS}(\mathbf{a_s}))}{2} \qquad (6)$$

This strategy avoids the redundancy by only considering one of the two axes which would have the same sum of S_{RS} and S_{PRS} in most cases. The parameter λ is introduced to gain more weight on the significant measures, i.e. the best RS measure and the best PRS measure of the system Ω. We use $\lambda = 0.75$ in our implementation.

4.2. Rectilinearity Energy E_R

The rectilinearity measure for 3D meshes was proposed by Lian et al. [LRS08] and describes the extent to which a 3D object is rectilinear. Given three coordinate axes, Lian computes the measure as the ratio between the surface area of the entire mesh and the sum of the three projected areas according to the three coordinate planes. We introduce an alternative and more efficient computation of the amount of rectilinearity which not only works on polygonal meshes but also on point clouds. The only demand is that each 3D point has a normal vector. If no normals are associated to the points, they can be computed by the method described in [HDD*92], for example.

According to [LRS08], a 3D mesh M is rectilinear, if and only if there is a coordinate system for which the surface

area of M coincides with the sum of the three projected areas. We reformulate this definition with respect to the surface normals: A 3D mesh or a 3D point cloud is rectilinear if and only if each normal vector is aligned with one of the coordinate axes. As a continuous measure, we take the average of the dot product of each normal vector and the best coordinate axis.

$$R(A) = \frac{1}{N}\sum_{i=0}^{N} \max_{j=1,2,3} \mathbf{n_i} \cdot \mathbf{a_j} \tag{7}$$

If the object is a triangle mesh, we weight each dot product with the area of the respective triangle and normalize the sum by the total area of the mesh rather than by N. Clearly, a perfectly rectilinear object returns a value of 1.0. The largest angle which can occur between a vector and the best of three orthogonal vectors is $\tan^{-1}\sqrt{2} \approx 54.74°$ (i.e. the angle between the diagonal of a unit cube and one of the edges). In the worst case, the measure from equation 7 returns $\cos(54.74) \approx 0.57$. Our rectilinearity energy, just as the symmetry energy, should contribute 0 in the best case and 1.0 in the worst case which is achieved by the following equation:

$$E_R(A) = \frac{1 - R(A)}{1 - \cos(\tan^{-1}\sqrt{2})} \tag{8}$$

4.3. Center Energy E_C

The rectilinearity energy only depends on A, i.e the directions of the orthogonal axes of the system Ω. We need additional constraints for the center $\mathbf{c}$ particularly for cases when E_R is the superior part in the total energy function from equation 5. But also the symmetric energy can sometimes push $\mathbf{c}$ far off the canonical center as can be seen in Figure 6a).
We thus introduce the center energy as the distance between the system's center $\mathbf{c}$ and the center $\mathbf{c_{BB}}$ of the axis aligned bounding box (AABB), which refers to the three orthogonal axes of A. We also normalize this value to the range $[0,1]$ by dividing it by the largest possible distance a point inside a box can have from it's center, the half of the diagonal's length d.

$$E_C(\mathbf{c}) = \frac{2||\mathbf{c} - \mathbf{c_{BB}}||_2}{d} \tag{9}$$

if $\mathbf{c}$ is even further away from $\mathbf{c_{BB}}$ than $d/2$, the value is clamped to 1.0.

5. Optimization of the System Ω

The previous section described how we can measure the quality of a current orthogonal system Ω. We now aim at finding the system with the minimum energy.

5.1. Local Minima

The objective function is the energy function from equation 5 and the parameter space for the global optimization is the 6D-space of allowed transformations which include a translation T and a rotation R. The objective function is continuous and smooth in the space of these transformations and thus, we can apply gradient-based optimization methods in order to find local minima. Figure 4 shows the smooth behavior of the energy function during a translation and a rotation of the system Ω.
We found, that instead of performing gradient computations and line searches in the entire 6D domain it is much more efficient to split the process into two 3D optimizations, one translation for the center optimization and one rotation for the optimization of the orientation of the system. The entire local optimization process then employs a flip-flop optimization between both sub-processes. Before we deal with the strategy to find the global minimum, let us describe how this process can be significantly simplified by reducing the complexity of the problem during the iterative process.

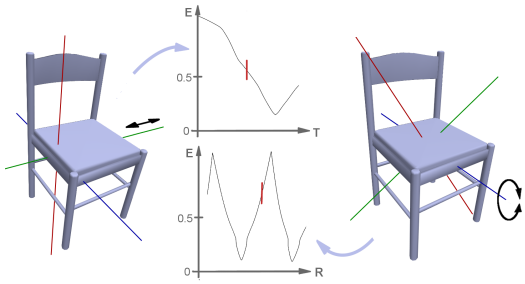

Figure 4: *Plot of the energy function of a translation T along the green axis (left) and a rotation R around the blue axis (right). The respective value of the current system is marked red in the plots.*

5.2. Dimension Reduction due to Symmetry

We can reduce the dimension of the optimization process on-the-fly by considering the individual parts of the symmetric energies. As soon as one of the three axes reveals (almost) perfect rotational or plane reflective symmetries, we can mark it as fixed and reduce the subsequent searching domain. The exact reduction depends on the kind of symmetry which is shown in Table 1.
If a PRS was detected, we can reduce the space of transformations to a rotation around the respective axis and a translation along the plane spanned by the other two axes, thus reducing the dimensions from six to three. In case of RS, the dimensions can even be reduced to two. The choice of the PCA axes as initial system has a big advantage in terms of immediate dimension reduction. If an object has a good PRS, the normal to this plane is an eigenvector of the covariance matrix and thus, an axis of the PCA-system. This property

	Translation	Rotation
$P_{PRS}(\mathbf{a_k}) < \varepsilon$	$\mathbf{a_l}, \mathbf{a_m}$	$\mathbf{a_k}$
$P_{RS}(\mathbf{a_k}) < \varepsilon$	$\mathbf{a_k}$	$\mathbf{a_k}$

Table 1: *Remaining Translations and Rotations axes after a symmetry with respect to axis* $\mathbf{a_k}$ *was detected in a system of axes* $\mathbf{a_k}, \mathbf{a_l}, \mathbf{a_m}$.

was proven in the appendix of [CVB09]. Therefore, the optimization will be a 3D problem from the beginning for objects with a PRS. For objects with no PRS but a RS, such as the fan from Figure 3, the scenario for dimension reduction will be reached after the first local minimization.

5.3. Determination of the Canonical System

We now describe the automatic method which finds a canonical coordinate frame by minimizing the energy function. The optimization process starts by initializing the system Ω with the PCA-system, so the initial center Point $\mathbf{c}$ is the barycenter and the axes from A are the PCA-axes. Note that after each step of the algorithm the axes are sorted such that $\mathbf{a_2}$ is always the best PRS-axis and $\mathbf{a_1}$ is the best RS-axis out of the remaining axes.

The first step of the algorithm is a local optimization of the initial PCA-system with all three parameters $\alpha = 1.0$, $\beta = 1.0$ and $\gamma = 0.5$. The value for γ is set to a lower value since the bounding box center is sensitive to outliers.

The subsequent step depends on the detected symmetries.

a) If more than one axis reveal RS or PRS, the system is already optimal.
b) If $\mathbf{a_1}$ reveals RS, we find the global minimum of the system Ω by two line-searches, one translation along $\mathbf{a_1}$ and one rotation around $\mathbf{a_1}$.
c) If $\mathbf{a_2}$ reveals PRS, the search domain can be reduced to three degrees of freedom according to Table 1. The optimal center point $\mathbf{c}$ lies somewhere on the plane spanned by $\mathbf{a_1}$ and $\mathbf{a_3}$ and the optimal orientation can be at most a rotation around $\mathbf{a_2}$ away. We sample this remaining rotation as well as the translation along the remaining axis with the least amount of RS and start a local optimization from the best sample point. This helps us out of a possible local minimum in which we might have fallen due to imperfect local symmetries.
d) If no significant symmetry was detected after the first local optimization, we automatically decrease the parameter α in the energy function, giving more weight to the rectilinearity measure, and start another local optimization with the new parameter set.

6. Results and Discussion

In this section we present some normalization results and analyze the importance of the individual energies.

6.1. Object Alignment

Our energy function is a combination of different measures which provides more robustness and can be applied to a wider range of objects than the use of only one of these measures alone. In [LRS08] for example, a rectilinearity measure is used for object alignment (only orientation), which can be emulated in our framework by setting α and γ in our energy function (equation 5) to 0. In [PSG*06], principal symmetry axes are defined with respect to plane reflections. These axes can also be computed by our method by setting β and γ to zero while considering only the PRS for each axis.

Generally, all existing alignment methods based on symmetric properties have only applied plane reflective symmetries. We found that the additional consideration of rotational symmetries has a positive effect on the canonical system of objects with local rotational symmetries. A good example is the office chair from Figure 5.

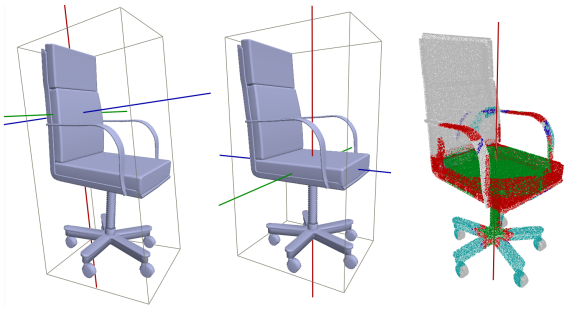

Figure 5: *Left: Use of PRS only. Middle: Additional consideration of our RS measure. Right: Display of the local RS. green means cyclic RS, red 2nd order RS and magenta 5th order.*

The object has a global PRS with respect to the green axis. Since the rectangular-shaped backrest is significantly larger than the seat, the other mirror planes are aligned such that they capture the local PRS at the backrest. However, the object has a strong local rotational symmetry due to the square-shaped seat and the 5-fold rotational symmetric legs. Our method is able to detect this major RS axis since it is capable of determining hybrid RS as we have described in Section 3.2. In the following, we describe the benefits of each component of our energy function and fortify them by examples.

The importance of the center energy S_C was already mentioned in Section 4.3. Since the rectilinearity measure is independent of the center $\mathbf{c}$, only the symmetric energy would be responsible to optimize $\mathbf{c}$ if there was no additional center energy. In that case, the resulting center might be highly unsatisfying if strong local symmetries are located at the outer regions of an object. Part a) of Figure 6 shows two exam-

ples as well as the corresponding improved alignments with enabled center energy S_C.

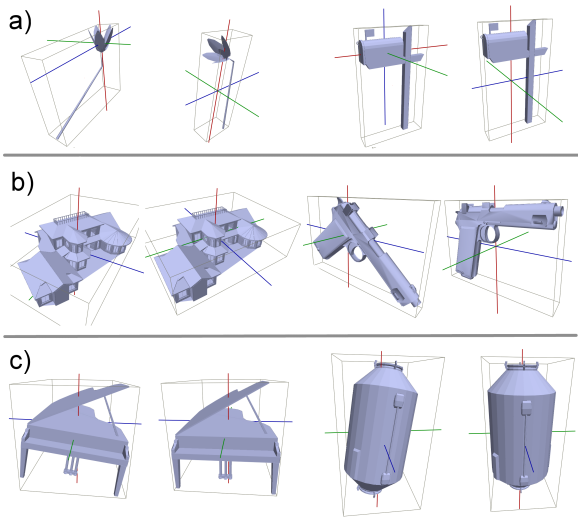

Figure 6: *Benefits of the individual parts of the energy function. The respective object on the right shows the canonical system of our hybrid method. The one on the left respectively shows the system with a) no center energy b) no rectilinearity energy c) no symmetry energy*

Figure 6 b) shows examples which demonstrate the importance of the rectilinearity measure. Sometimes, local symmetries can lead to unnatural alignment axes which can be detected by a high rectilinearity energy E_R. In case of the gun from Figure 6, for example, the handle partly maps onto the barrel under a plane reflection with respect to the blue axis in the left version. However, due to the high amount of E_R (and E_C) of this system, our method proposes the axes shown in the version to the right.

In Figure 6c), we see examples where the rectilinearity measure alone fails to determine good coordinate frames. In case of the piano, the system is tilted due to the opened cover which contributes a large amount of normals pointing to a rather unnatural alignment axis. Generally, the energy E_R is not suitable for round objects since their normals are distributed rather uniformly along the radial directions. However, Such objects have a high amount of rotational symmetry so the symmetric energy E_{sym} robustly leads the coordinate frame to a good canonical system in these cases. Figure 7 shows further alignment results of our method in direct comparison to the PCA method. We particularly chose objects for which PCA produces poor results and also demonstrate the applicability of our alignment scheme for 3D point data.

The time for an optimization is depending on the amount of symmetries of the object since significant symmetries lead to an early reduction of the degrees of freedom as described in Section 5.3. In the average, our method took 350ms to optimize the coordinate frame of an object of the Princeton Shape Benchmark on an Intel Core i7 with an Nvidia GTX 485. The maximum time we measured for a single object was 1.1 seconds.

6.2. Shape Matching

Our framework is capable of performing shape matching by approximating the Hausdorff-distance of two objects in a very efficient way, using the distance-field-based compare function from Section 3.2. While we applied this function on a model and a translated version of it in order to measure symmetry during the optimization process of the orthogonal system Ω, we can as well apply it on two different models after they have been perfectly aligned with our normalization method. In that case, however, the function is not commutative. We have to compute the distance field for one object and render a point cloud version of the other into it, and vice versa. The resulting distance between the two objects is the maximum of both measures, which is nothing else but the Hausdorff distance, approximated by the Manahattan metric instead of using exact euclidean distances. Due to the fact that both, the distance field computation and the actual distance evaluation for each sample point are implemented on the GPU, the evaluation of the approximated Hausdorff distance can be performed in real time. Figure 8 shows examples of the Hausdorff measure for similar objects.

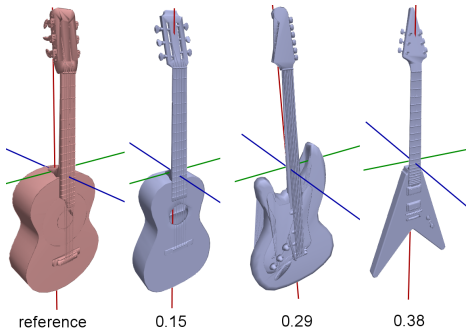

Figure 8: *Hausdorff distance of different guitars-objects using our method. The relative similarities are well reflected by the computed distances.*

7. Conclusion

We presented a method to determine a canonical coordinate frame of 3D polygonal meshes and point clouds which is based on the minimization of a hybrid energy function. This function combines symmetric properties and rectilinearity and we have shown, that the combination provides more robustness and flexibility than one measure alone. We also introduced a novel continuous symmetry measure which is ca-

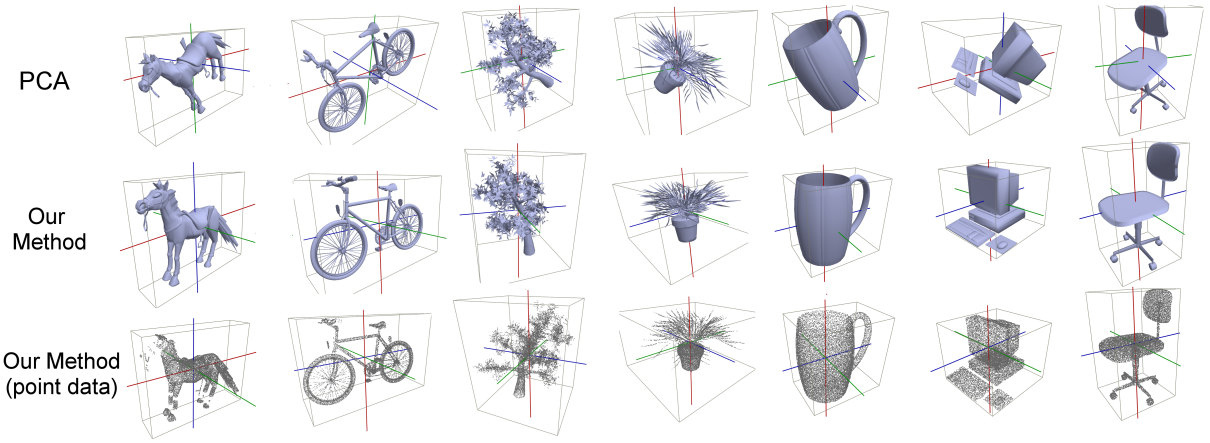

Figure 7: *Alignment results of our method with polygonal data and point data compared to the corresponding PCA-alignment.*

pable of measuring rotational symmetries of different orders along a specific axis. An efficient GPU-implementation of this measure allows us to measure symmetries in real-time. The minimization process of the energy function starts by initializing the coordinate frame with the PCA axes and then performs a series of elaborate steps to find the system with the lowest energy, exploiting symmetric properties to significantly reduce the complexity of the process on-the-fly. The resulting global minimum of the energy function provides a natural and intuitive coordinate frame for the majority of objects. It not only delivers canonical orthogonal axes, but also a consistent object center and can thus be applied to perform normalization with respect to translation, rotation and scale. We further showed how our normalization in combination with our efficient pose-variant compare function can be applied to efficiently approximating the Hausdorff distance between 3D objects.

References

[BKS*05] BUSTOS B., KEIM D. A., SAUPE D., SCHRECK T., VRANIĆ D. V.: Feature-based similarity search in 3d object databases. *ACM Comput. Surv. 37*, 4 (Dec. 2005), 345–387.

[CVB09] CHAOUCH M., VERROUST-BLONDET A.: Alignment of 3d models. *Graph. Models 71*, 2 (Mar. 2009), 63–76.

[FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 42:1–42:7.

[Fer00] FERGUSON R. F.: Modeling orientation effects in symmetry detection: The role of visual structure, 2000.

[FMK*03] FUNKHOUSER T., MIN P., KAZHDAN M., CHEN J., HALDERMAN A., DOBKIN D., JACOBS D.: A search engine for 3d models. *ACM Trans. Graph. 22*, 1 (Jan. 2003), 83–105.

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. In *COMPUTER GRAPHICS (SIGGRAPH âĂŹ92 PROCEEDINGS)* (1992), pp. 71–78.

[JLWI11] JOHAN H., LI B., WEI Y., ISKANDARSYAH Y. W.: 3d model alignment based on minimum projection area. *Vis. Comput. 27*, 6-8 (June 2011), 565–574.

[KCD*02] KAZHDAN M., CHAZELLE B., DOBKIN D., FINKELSTEIN A., FUNKHOUSER T.: A reflective symmetry descriptor. In *7th European Conference on Computer Vision (ECCV 2002)* (May 2002), pp. 642–656.

[LRS08] LIAN Z., ROSIN P. L., SUN X.: A rectilinearity measurement for 3d meshes. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval* (2008), MIR '08, pp. 395–402.

[MGG11] MARTINEK M., GROSSO R., GREINER G.: Fast and efficient 3d chamfer distance transform for polygonal meshes. In *Proceedings of Vision, Modeling, and Visualization (VMV)* (2011), Eurographics Association, pp. 121–128.

[NK01] NOVOTNI M., KLEIN R.: A geometric approach to 3d object comparison. In *Shape Modeling and Applications, SMI 2001 International Conference on.* (may 2001), pp. 167 –175.

[NS10] NAPOLÉON T., SAHBI H.: From 2d silhouettes to 3d object retrieval: contributions and benchmarking. *J. Image Video Process. 2010* (Jan. 2010), 1:1–1:22.

[PPPT07] PAPADAKIS P., PRATIKAKIS I., PERANTONIS S., THEOHARIS T.: Efficient 3d shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recogn. 40*, 9 (Sept. 2007), 2437–2452.

[PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3d shapes. *ACM Trans. Graph. 25*, 3 (July 2006), 549–559.

[SS97] SUN C., SHERRAH J.: 3-d symmetry detection using the extended gaussian image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19* (1997), 164–168.

[TL06] TEDJOKUSUMO J., LEOW W. K.: Normalization and alignment of 3d objects based on bilateral symmetry planes. In *Proceedings of the 13th international conference on Multimedia Modeling - Volume Part I* (2006), pp. 74–85.

[TV04] TANGELDER J. W. H., VELTKAMP R. C.: A survey of content based 3d shape retrieval methods. pp. 145–156.