Geometric 3D Comparison – an Application

Marcin Novotni and Reinhard Klein

Institute of Computer Science, University of Bonn, Germany

Abstract

With the advent of powerful 3D visualization, modeling and scanning techniques, there is an increasing need for tools supporting the automatic search for objects in 3D archives. In this paper we adopt our geometric approach to 3D object comparison to an industrial application of finding the best fitting shoe given a 3D foot scan. The geometric similarity estimation is well suited for search among objects with relatively stiff structure, in the presented setting we also have to cope with the reconstruction of 3D models from point clouds, and the definition of location dependent weighting based on anatomic properties of the foot.

1. Introduction and Related Work

In our recent work [10], we have introduced a new method for similarity estimation of 3D objects. The technique presented in the paper is particularly well suited to compare 3D objects having stiff structure or moderate variance, e. g. non-articulated objects like bones, fruit, etc. In this paper we present an application of our algorithm to the problem of retrieval of the best fitting shoes from a given sortiment based on the 3d foot scan of the client. The basic idea of the solution is to compare the lasts which were used to manufacture the shoes and the scanned feet of the clients. Therefore, the problem can be decomposed into two main subproblems:

1. **Reconstruction of the scanned foot.** The input of our algorithm is a set of unorganized samples consisting of about 100.000 points. However, the basic comparison method relies on a representation based on a closed 2-manifold triangle mesh. Therefore, the scanned object has to be fully reconstructed.
2. **Similarity estimation.** After a simple prefiltering processing where most of the unlikely fitting lasts are excluded, the remaining candidates for best matches have to be precisely measured against the input foot under some specific considerations which are inducted by the fact, that we are actually comparing feet with lasts.

The results described in this paper have been achived in cooperation with Tecmath AG company from Kaiserslautern, Germany which already developed a product called *Pedus 3D Foot Scanner* [†]. In the frame of this work, we extended our original 3D object comparison method to incorporate locality by assigning local weighting of the geometric distance function to the surface, in order to be able to differentiate between relevant and less relevant areas of the investigated feet. In the special case of a foot, the area just behind the first joints of toes on the side of the foot for instance has to fit quite accurately into the shoe, whereas one normally wants to have some place left before the tip of the toes in the shoe.

The field of 3D shape analysis and content based retrieval from 3D archives is closely related to the computer vision and image processing, both of which have long tradition. However, unfortunately most of the methods from these fields cannot be applied directly to the 3D problem. One of the well known paradigms is to extract features from 2D or stereo images and search for matches in a 3D database [6, 14].

As for search keys, curvature distribution could principally be used to represent a shape [5, 13], but it proved to be unreliable because of its noise sensitiveness. 3D moments have been successfully used to solve the pose estimation [12, 3, 1, 2], however the higher order moments are again too noise sensitive to be used to characterize a 3D object.

Contributions concerning the problem of general 3D shape comparison and retrieval appeared only in the last few months [10, 11]. An interesting work relevant especially for articulated objects has been very recently presented by Hilaga

[†] http://www.hs.tecmath.de/english/pedus_eng.shtml

et al [8] where the authors use Reeb graphs based on geodesic distances to encode the topology of 3D objects.

Due to the place constraints, in this paper we only outline the methods applied in the algorithm, please refer to the cited original papers for exhaustive descriptions. Section 2 describes our reconstruction method, in Section 3 we review our previous comparison method for the general setting as described in [10], and we tailor this technique to the specific application described above. We present and discuss the results in Section 4. Finally, we describe our ongoing parallel work and draft some future directions in Section 5.

2. Reconstruction

As stated above, the models are represented by unorganized 3D point clouds, so the scanned objects have to be reconstructed before the actual similarity estimation. In our reconstruction method we use the a priori knowledge about the scanning process and object properties. The reconstruction is similar to the algorithm presented by Curless and Levoy [4] with an additional 3D filtering step. The major difference to their method is that while they have a complete information about the respective position of scanner and samples, we know only approximately how the scanner was positioned relatively to the data. Therefore, in order to construct a reliable and robust representation, we have to apply an additional oriented 3D low-pass filtering.

The 3D scanner provides essentially three range images: one from each left and right side and one from bottom of the foot which are subsequently registered. The problem with this representation is, that there are large holes in the surface described by the samples as shown in Figure 2. We cope with these artifacts using a technique called space carving, cf. Figure 3.

Before we can apply this method, the object representation have to be scan converted into a voxel space, which then is traversed along the directions of the scanning rays (the small arrows in Figure 3). During the traversal, the voxels between the scanner and the surface are set to "outside" or 0 (white, non-marked area) and the voxels behind the surface are set to "candidate for inside" or 1 (the green area). After processing the original voxel space from all the scanning directions, we intersect the resulting sets and get a voxel representation of the object. Finally we run the Marching Cubes algorithm to get a triangular mesh from the volume data and smooth the mesh to eliminate the aliasing artifacts.

In our case, the x axis of the coordinate system of the scan is aligned with the length axis of the foot and the client stands on the $x-y$ plane, so that the z axis points upwards. In consequence, we need to traverse the voxel space only in directions $+y$, $-y$ and $+z$ which greatly simplifies and accelerates the computations.

The above algorithm will produce correct results only if

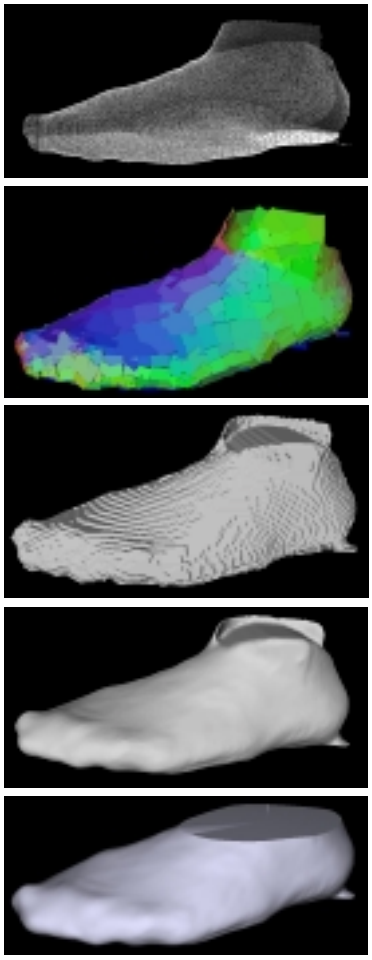


Figure 1: *Stages during the reconstruction: a) input scan, b) the OBBs used to orient the gaussian kernels, c) the closed mesh extracted after space carving d) the smoothed surface, e) the upper part is not represented by the last, therefore it is cut away.*

the traversal rays are guaranteed to hit the object surface. This condition is hard to satisfy due to the point sampled nature of the representation, therefore we apply a 3D low pass filtering with a scaled and oriented 3D gaussian kernel. An important issue in this context is to retain the original shape of the object. Therefore we have to take into account, that using a rotationally symmetric gaussian, the surface would be shifted outwards and thus distorted. We resolve this problem by shrinking the gaussian kernels in a direction normal to the surface at a given sample point. A discrete 3D gaussian kernel $g(\underline{x})$ is defined by:

$$g(\underline{x}) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^t \Sigma^{-1}(\underline{x}-\underline{\mu})},$$

Figure 2: *The back view of the scan roughly triangulated here for better sight. Note the large holes due to missing data.*

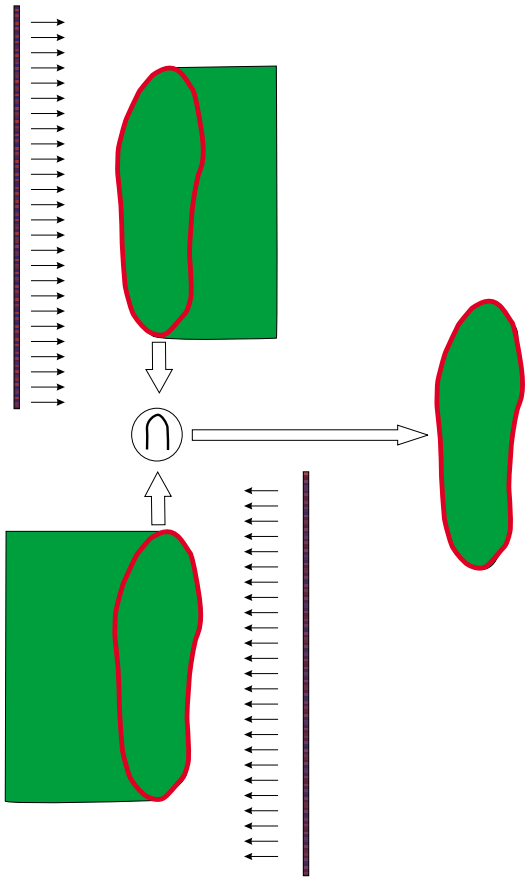


Figure 3: *The process of volumetric reconstruction*

$\underline{\mu}$ being the center of the kernel, $\underline{x}$ is a point in the 3D space. The matrix Σ determines the scaling along axes of the local coordinate system of the kernel with origin in $\underline{\mu}$ and axes aligned with principal axes of the gaussian:

$$\Sigma = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

Thus, we get ellipsoidal filter kernels and the filtering will mainly take place in the tangent plane of the surface. To determine the tangent planes at arbitrary points, we construct a hierarchical representation based on OBBs (Oriented Bounding Boxes), see [7] for reference. After decomposing the input point set using an OBB hierarchy until the boxes are sufficiently thin (the distance between scan samples is an appropriate value for this thickness), the direction of the shortest side of each OBB will approximately be aligned with the corresponding normal. In our implementation, we place an oriented kernel in each sample point, and evaluate the so defined implicit function:

$$f(\underline{p}) = \sum_{i \in S} g(\underline{x}_i),$$

where $\underline{p}$ is an arbitrary point in the 3D space, and S denotes the index set of the sample points. Note that $g(.)$ decreases exponentially, therefore the above sum has to be evaluated only for $\underline{x}_i$ that are sufficiently near to $\underline{p}$. This proximity information in turn is already included in the OBB hierarchy which makes the algorithm very efficient.

3. 3D geometric comparison

In this section we first outline our original algorithm for similarity estimation, and then we show how to modify it to tackle the problem of finding the best fitting shoes.

3.1. Basic technique

The outline of the algorithm is as follows:

1. **Pose Estimation** As already indicated, the foot and the last are already approximately aligned after the scanning process. However, in order to get an optimal pose estimation, we use the principal axes of both objects to align them optimally.
2. **Projection of the weight function** In this step, we project the weight function from the last to the foot as described in the next subsection – this way the areas with larger relevance will be taken into account with larger weights in the final error function.
3. **Computation of geometric similarity** The algorithm for similarity estimation described in [10] is applied. As result we get an error measure for each foot-last pair, which can be used to establish an ordering of best matches.

Our original similarity estimation algorithm is based on the following observation: if one takes two nearly identical

datasets and aligns them so that the surfaces fit one onto another as accurately as possible, every point inside one of the objects will either be contained by the other one or be in a small distance from the other's boundary and vice versa. In contrast to that, in the case of two different objects, there will be volume portions of each object that are in relatively large distance from the surface outside of the other object.

In our algorithm, after the pose estimation, we determine for each object how big portions of its volume are outside of the other object. As an efficient technique to accomplish this task, we use discrete *3D distance fields*. Based on the information already explicitly contained in the distance fields, for each discrete element of the volumes we look up its distance from the boundary of the other object and construct a specific histogram containing these values. We eventually compute a quantitative measure of resemblance based on these histograms. This measure is essentially a weighted sum of subvolumes of object A which are outside of the other object B. These subvolumes are weighted to penalize the voxels of A according to their distance from the surface of B (the weights grow with the distance). Again, please refer to the original paper for detailed discussion.

3.2. Weight function

We assume that each last has an assigned weight function which simply means, that each triangle of the surface has an assigned scalar value $w \in [0,1]$. This describes the already mentioned distribution of areas that determine the comfort feeling if a client wears a particular shoe. The weighting information is extracted on the basis of an extensive statistical study conducted by Tecmath AG.

As stated in the original paper, the geometric distance computation is conducted two times: for each of the objects against the other one. Therefore, we need a weight function both on the foot and the last, where the former has to be computed online. We perform this by determining for each triangle of the foot the closest triangles of the last and projecting the according weights onto the foot triangle. For the closest triangle determination we use color coded discrete distance fields as presented in [9]. The voxels in Voronoi regions of each simplex of the last surface has an assigned unique color value which serves as an index into the simplex array of the last. This way, the projection process can be accomplished in a straightforward and robust way.

4. Results

We have experimented with several foot and last scans, and we found that the cases of most interest are where the foot fits the last based on the basic measures of both, e. g. length or breadth, or simply the a priori known size of the shoe. This rough matching can be accomplished very straightforwardly, for instance using the shoe/foot size, or comparing the breadth and length of the bounding boxes. Having this, the more interesting part of matching is left where we want to retrieve a shoe (or an ordering of several best matches) which gives the client the best comfort feeling. As already mentioned, we attempt to assure this hard-to-define quality by assigning weighting function to the various areas of the last according to the influence of these areas to the overall comfort.

	right_last_1	***right_last_2***
right_foot	616929	552663

	left_last_1	***left_last_2***
left_foot	643223	621594

Table 1: *Errors between foot/last pairs using the weighting. function*

	right_last_1	***right_last_2***
right_foot	1633548	1622735

	left_last_1	***left_last_2***
left_foot	1781129	1768927

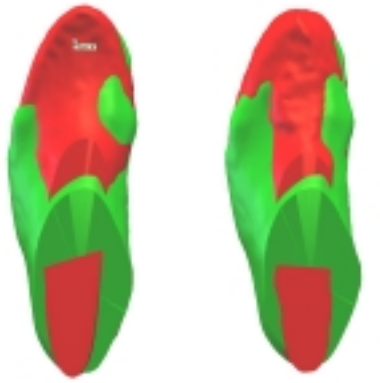
Table 2: *Errors between foot/last pairs without using the weighting function*

Some excerpts from our results are shown in Table 1, results for the same foot/last pairs without application of the weighting function are presented in Table 2, see also Figure 4. Note, that there is virtually no difference between the distances in the latter case, whereas one can unambiguously establish an ordering between the lasts in the former case.

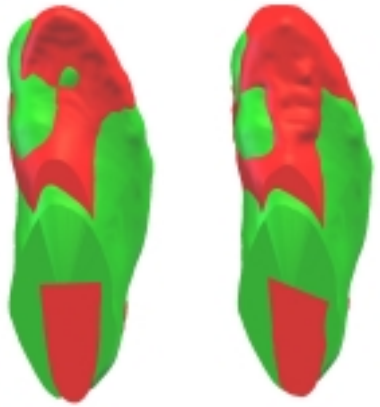
5. Ongoing and future work

There are a few open issues concerning the method presented in this paper. As shown in Section 4, the selection of weight distribution has a significant influence on the algorithm. Therefore a statistic should be built up how the certain parts of the foot and shoe contribute to the overall comfort feeling. Another important point of the procedure is the reconstruction – we believe that the extraction of a mesh could be done directly during the scanning process which would probably accelerate the computations.

We are now in the process of experimentation with new methods similar to the one presented by Hilaga et al [8] This algorithm already provides us with correspondences which are unfortunately not unambiguous. In the near future we plan to extend the method to include geometric features



left_foot/last_1 *left_foot/last_2*



right_foot/last_1 *right_foot/last_2*

Figure 4: *The feet (green) - last (red) pairs laid onto one another after the pose estimation. Note that it is hard to say which last fits better in either cases.*

which will make it even more versatile. We also believe that the approach can be modified in such manner to be suitable for reliable establishment of correspondences and intelligent segmentation of 3D models.

Acknowledgements

We wish to express our gratitude to Deutsche Forschungsgemeinschaft (German Science Foundation) for the financial support and thus making our work possible.

References

1. N. Canterakis. Complete moment invariants and pose determination for orthogonal transformations of 3d objects. Technical report, Technische Universität Hamburg-Harburg, 1996. 1
2. N. Canterakis. 3d zernike moments and zernike affine invariants for 3d image analysis and recognition. In *11th Scandinavian Conf. on Image Analysis*, 1999. 1
3. N. Canterakis and H. Schulz-Mirbach. Algorithms for the construction of invariant features. Technical report, Technische Universität Hamburg-Harburg, 1994. 1
4. B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM SIGGRAPH*, 1996. 2
5. C. Dorai and A. K. Jain. Cosmos - a representation-scheme for 3d free-form objects. *Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997. 1
6. Olivier Faugeras. *Three-Dimensional Computer Vision, a Geometric Viewpoint*. MIT Press, 1993. 1
7. S. Gottschalk, M. Lin, and D. Manocha. Obb-tree: A hierarchical structure for rapid interference detection. In *ACM SIGGRAPH*, 1996. 3
8. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH*, 2001. 2, 4
9. R. Klein, A. Schilling, and W. Strasser. Reconstruction and simplification of surfaces from contours. In *Pacific Graphics*, 1999. 4
10. M. Novotni and R. Klein. A geometric approach to 3d object comaprison. In *International Conference on Shape Modeling and Applications*, 2001. 1, 2, 3
11. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3d models with shape distributions. In *International Conference on Shape Modeling and Applications*, 2001. 1
12. H. Schulz-Mirbach. On the existence of complete invariant feature spaces in pattern recognition. In *IEEE International Conference on Pattern Recognition*, volume 2, pages 178–182, 1992. 1
13. R. Sonthi, G. Kunjur, and R. Gandh. Shape feature representation unsing the curvature region representation. In *Symposium Solid Modeling*, 1997. 1
14. I. Weiss and M. Ray. Model-based recognition of 3d object from single vision. *Transactions on Pattern Analysis and Machine Intelligence*, 14, 2001. 1