ABSTAND: DISTANCE VISUALIZATION FOR GEOMETRIC ANALYSIS

blind version

KEY WORDS: Computer-aided design, mesh processing, geometric errors

ABSTRACT:

The need to analyze and visualize differences of very similar objects arises in many research areas: mesh compression, scan alignment, nominal/actual value comparison, quality management, and surface reconstruction to name a few. Although the problem to visualize some distances may sound simple, the creation of a good scene setup including the geometry, materials, colors, and the representation of distances is challenging.
Our contribution to this problem is an application which optimizes the work-flow to visualize distances. We propose a new classification scheme to group typical scenarios. For each scenario we provide reasonable defaults for color tables, material settings, etc. Completed with predefined file exporters, which are harmonized with commonly used rendering and viewing applications, the presented application is a valuable tool. Based on web technologies it works out-of-the-box and does not need any configuration or installation. All users who have to analyse and document 3D geometry will stand to benefit from our new application.

1. INTRODUCTION

Analyzing differences between surfaces is a necessary task in many fields of research. Measuring the distance between two surfaces is a common way to compare them.
In computer graphics, for example, differences of surfaces are used for analyzing mesh processing algorithms such as mesh compression. They are also used to validate reconstruction and fitting results of laser scanned surfaces. As laser scanning has become very important for the acquisition and preservation of artifacts, scanned representations are used for documentation as well as analysis of ancient objects. Detailed mesh comparisons can reveal smallest changes and damages.
These analysis and documentation tasks are needed not only in the context of cultural heritage but also in engineering and manufacturing. Differences of surfaces are analyzed to check the quality of productions.
A meaningful visualization of surface differences is a challenging task. The goal is a clean representation of facts without overextending the observer. This can be done using still images and also using interactive rendering. Universal 3D files (U3D) embedded in a Portable Document File (PDF) allow to publish and share interactive visualizations on a wide-spread platform.
This paper presents an application which optimizes the workflow to create such visualizations. It uses a classification scheme to group typical scenarios. Reasonable presets of settings are provided for quick output generation. The results can then be imported into common visualization tools like Maya™, 3ds Max™, or Deep Exploration™ to support the overall visualization task.

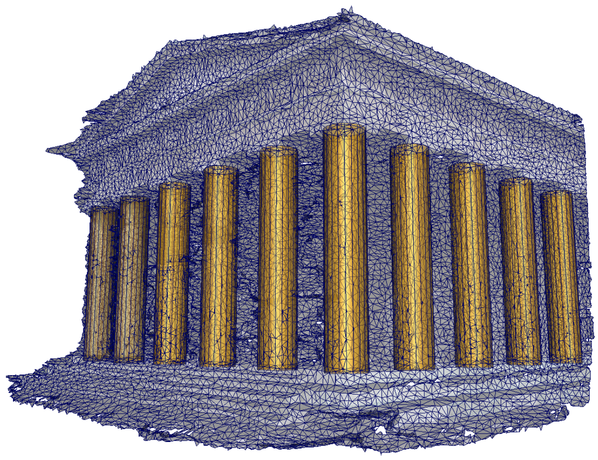

Figure 1: This is an example of a bad visualization. It shows the result of a reconstruction process: a Greek temple and its fitted reconstruction. The surfaces intersect, but based on this image it is not possible to examine the reconstruction's quality. Due to a missing legend no quantifiable error can be determined.

2. RELATED WORK

Our application visualizes distances between geometric objects. It is related to three areas of research. The distance calculation itself is an algorithmic problem. The visualization has to deal with colors and color perception. Last but not least the application is embedded into a context.
Our algorithm calculates distances between dense samplings of geometric objects. Its main idea is based on the method for calculating errors between surfaces presented in *Metro: Measuring error on simplified surfaces* (Cignoni et al., 1998) and *MESH: Measuring Error between Surfaces using the Hausdorff distance* (Aspert et al., 2002). In order to speed up the calculation of a Hausdorff distance, which has a quadratic runtime in a naive implementation, the samples are stored in kd-trees (Gonnet and Baeza-Yates, 1991). The nearest-neighbor-search algorithm we use has an average runtime of $n \cdot log(n)$ and is described in *An introductory tutorial on kd-trees* (Moore, 1991).

Our algorithm to calculate normal distances (i.e. the nearest neighbor search restricted to points inside a double cone) is based on a commonly used grid structure. This technique can be found amongst others in (Farin et al., 2003) and (Hege and Polthier, 2002).

Having calculated the distances our application offers predefined color palettes. The default color settings take the human perception and perceptual ordering into account. An overview on colors and color perception can be found in MAUREEN STONE's field guide to digital color (Stone, 2003).
The set of predefined color maps contains the luminance-based maps with only small variations in the hue value as proposed in (Levkowitz and Herman, 1992) and in (Bergman et al., 1995) as well as the maps proposed in *Rainbow Color Map (Still) Considered Harmful* (Borland and Taylor II, 2007). The often used rainbow color map is available but not used as a default. The predefined color maps also contain neutral color settings. These settings do not have "signal colors" such as red. The selection of the neutral color ranges are based on *How NOT to lie with visualization* by (Rogowitz et al., 1996).
Visualizations of surfaces differences are needed in many fields, e.g. comparison of mesh reduction results (Klein et al., 1996). The main context, in which we use our application, is the field of cultural heritage. Especially for issues on scanning, fitting and reconstruction the application turns out to be a valuable tool. An overview on current research topics in cultural heritage can be found in (Baltsavias et al., 2006). As the program is not limited to this field of application, its context will not be discussed in detail here.

3. THEORY

This section describes the mathematical background for calculating a distance. Without loss of generality it is sufficient to consider point sets. Other geometric primitives can be converted into point sets by dense sampling.

3.1 Metric

A nonnegative function

$$d : X \times X \to \mathscr{R}$$

describing the "distance" between neighboring points for a given set X is called a metric, if it satisfies

$$d(\vec{x},\vec{x}) = 0 \text{ and } d(\vec{x},\vec{y}) = 0 \Rightarrow \vec{x} = \vec{y} \tag{1}$$

as well as the symmetry condition

$$d(\vec{x},\vec{y}) = d(\vec{y},\vec{x}) \tag{2}$$

and the triangle inequality

$$d(\vec{x},\vec{z}) \leq d(\vec{x},\vec{y}) + d(\vec{y},\vec{z}) \tag{3}$$

for all $\vec{x},\vec{y},\vec{z} \in X$.
The most simple example which satisfies all conditions is the discrete metric

$$d(\vec{x},\vec{y}) = \begin{cases} 1, & \vec{x} \neq \vec{y} \\ 0, & \vec{x} = \vec{y} \end{cases}$$

In the field of computer-aided design (CAD) and computer graphics the Euclidean metric is of particular importance. Two points $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ of an n-dimensional space have the Euclidean distance

$$d(\vec{x},\vec{y}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}.$$

In some cases it is convenient to use the maximum metric

$$\mu(\vec{x},\vec{y}) = \max(|x_1 - y_1|, \ldots, |x_n - y_n|)$$

or the absolute value sum metric

$$\sigma(\vec{x},\vec{y}) = |x_1 - y_1| + \cdots + |x_n - y_n|.$$

The relationship between all these metrics in $\mathscr{R}^n$ is given by the inequality

$$\forall \vec{x},\vec{y} \in \mathscr{R}^n : d(\vec{x},\vec{y}) \leq \sqrt{n} \cdot \mu(\vec{x},\vec{y}) \leq \sqrt{n} \cdot \sigma(\vec{x},\vec{y}) \leq n \cdot d(\vec{x},\vec{y}).$$

The special case $n = 1$ leads to

$$d(\vec{x},\vec{y}) = \mu(\vec{x},\vec{y}) = \sigma(\vec{x},\vec{y}) = |\vec{x} - \vec{y}|.$$

3.2 Point Sets

The distance between a single point $\vec{x}$ and a point set Y can be defined using the minimum of all distances between $\vec{x}$ and a point $\vec{y} \in Y$, respectively
For two point sets there are many different ways to define the directed distance. DUBUISSON and JAIN have analyzed the following six distance functions (Dubuisson and Jain, 1994):

$$\begin{aligned} d_1(X,Y) &= \min_{\vec{x} \in X} d(\vec{x},Y) & (4) \\ d_2(X,Y) &= K^{50}_{\vec{x} \in X}\, d(\vec{x},Y) & (5) \\ d_3(X,Y) &= K^{75}_{\vec{x} \in X}\, d(\vec{x},Y) & (6) \\ d_4(X,Y) &= K^{90}_{\vec{x} \in X}\, d(\vec{x},Y) & (7) \\ d_5(X,Y) &= \max_{\vec{x} \in X} d(\vec{x},Y) & (8) \\ d_6(X,Y) &= \frac{1}{||X||} \sum_{\vec{x} \in X} d(\vec{x},Y) & (9) \end{aligned}$$

where $||X||$ is the number of elements in X and $K^{j}_{\vec{x} \in X}$ represents the ranked distance; i.e. $K^{0}_{\vec{x} \in X}$ corresponds to the minimum, $K^{50}_{\vec{x} \in X}$ to the median and $K^{100}_{\vec{x} \in X}$ to the maximum of all distances $d(\vec{x},Y), \forall \vec{x} \in X$.
While it is sensible to use the minimum function for distances between a point and a point set, nested minimum functions (d_1) do not define a meaningful distance between two point sets. All point sets X and Y with non-empty intersection would have a distance of zero. The oriented Hausdorff distance, named after FELIX HAUSDORFF (1868-1942), does a better job. Its definition (d_5) utilizes the maximum function.
Oriented distances are characterized by $d(X,Y) \neq d(Y,X)$ in most cases. Taking the maximum of both oriented distances leads to a *non-oriented* distance; e.g. the non-oriented Hausdorff distance between X to Y takes the maximum of both oriented distances:

$$\begin{aligned} H(X,Y) &= \max\left(d_5(X,Y), d_5(Y,X)\right) & (10) \\ &= \max\left(\max_{\vec{x} \in X} d(\vec{x},Y), \max_{\vec{y} \in Y} d(\vec{y},X)\right) & (11) \end{aligned}$$

An illustrative example on Hausdorff calculations is shown in Figure 2.

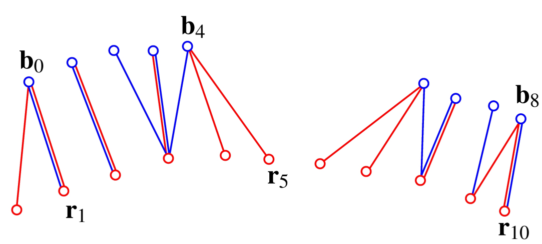

Figure 2: The Hausdorff metric defines the distance between two sets. For illustrative purposes each point of one set is connected in the same color with its nearest neighbor of the other set. The oriented Hausdorff distance from the blue points to the red ones can be found between $\mathbf{b}_0$ and $\mathbf{r}_1$ (longest blue line). The oriented Hausdorff distance from the red points to the blue ones is between $\mathbf{r}_5$ and $\mathbf{b}_4$ (longest red line). The maximum of both distances is the Hausdorff distance between these point sets.

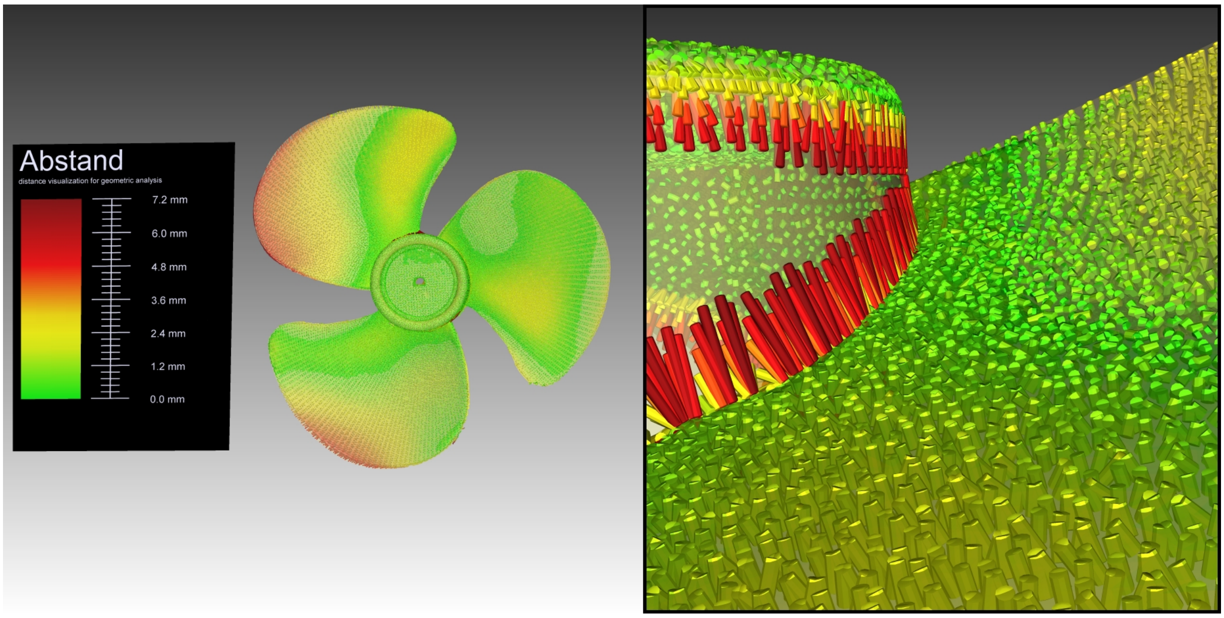

Figure 4: This example shows a laser scan of a propeller and its corresponding CAD model. The reference model has been reengineered based on measurements of the real object. Using this input data, the application ABSTAND calculates an appealing visualization. The scanned surface is colorized according to the distances to the reference model. To have a clear look at the colored distance visuals, the reference model is almost transparent. For this example, only the one-sided distances from the actual model to the reference model are visualized using small, solid cylinders. The user can choose the number of distributed cylinders. The polygon count of the result can be limited to a desired, maximum number. According to these settings the quality/level-of-detail of the cylinders is determined.

The combination of the other distance functions (d_1, ..., d_4, d_6) results in

$$
\begin{aligned}
D_1(X,Y) &= \max(d_1(X,Y), d_1(Y,X)) && (12)\\
D_2(X,Y) &= \max(d_2(X,Y), d_2(Y,X)) && (13)\\
D_3(X,Y) &= \max(d_3(X,Y), d_3(Y,X)) && (14)\\
D_4(X,Y) &= \max(d_4(X,Y), d_4(Y,X)) && (15)\\
D_6(X,Y) &= \max(d_6(X,Y), d_6(Y,X)) && (16)
\end{aligned}
$$

Please note that these functions are not metrics in contrast to the Hausdorff distance. $D_1, \ldots, D_4$ do not fulfill the condition

$$D(X,Y) = 0 \Rightarrow X = Y,$$

which could be a problem for object matching. D_6 violates the triangle inequality. Nevertheless, D_3, D_4, and D_6 have some importance in the field of computer vision.

3.3 Signed distance

For geometric primitives, which form a two-dimensional manifold, it is convenient to indicate the location of a point in the sign of the measured distance.

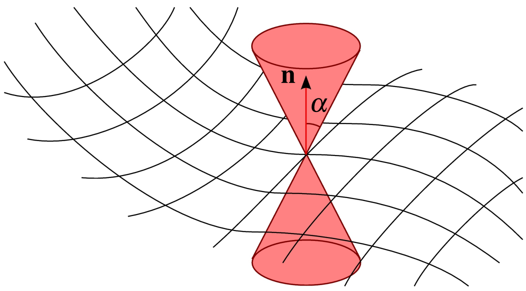

Figure 3: In some cases it is sensible to restrict the nearest-neighbor-search to samples inside a double cone along normal direction. Undesired sample relations at parts, which do not have a corresponding counter part, can be avoided.

A two-dimensional manifold object in $\mathscr{R}^3$ defines an inner and an outer space. By convention points in outer space have positive distance, points in inner space have negative distance.

4. CLASSIFICATION AND ALGORITHMS

The main idea of the application is to classify the distance visualization problem into categories. Each category has a set of default settings which lead to feasible results.

4.1 Variance Analysis

All distance visualization problems belong to one of two distinct groups. The *asymmetric* case analyzes two geometric objects assuming that the first object is the reference / nominal object. The second object is the actual object to be validated. Such a configuration can be found e.g. in the context of quality management using a CAD model as reference to check the resulting product (see cathedral example in Section 6).
The *symmetric* case is characterized by the absence of a reference model. Both objects are on a par. In contrast to the asymmetric case the results of the symmetric one do not change, if the order of the imported objects is swapped. A typical, symmetric situation is the comparison of two range maps of a laser scanning process. If overlapping regions of aligned scans are analyzed, none of them can be considered to be the ground truth (see chess pieces scan examples in Section 6).
These two main groups require different settings.

4.2 Symmetric Distance Visualization

The distance analysis starts by generating samples of both input objects and calculates their normals. Then the one-sided distances are computed and assigned to the samples. The nearest-neighbor-search can be restricted in two ways.

- For some cases it is useful to restrict the search area to a double cone along the normal as illustrated in Figure 3.
- Lower and/or upper bounds are another way to filter the results – for example ignoring all distances smaller than an epsilon.

The remaining distances are sorted and then grouped in a histogram according to their length.

For the two analyzed meshes, a unobtrusive coloring is suggested. The transparency value may vary according to the signed distance. This technique enables to display inner parts which would otherwise be covered by opaque surfaces.

Solid cylinders (or prisms with a lower polygon count) are generated to visualize the distances (see Figure 4). These distance visuals are grouped using the calculated histograms.

4.3 Asymmetric Distance Visualization

Calculations for the asymmetric case start similar to the symmetric one. But the results can also be limited to one-sided distances. As one surface is considered as ground truth, the visualization emphasizes the actual object. The reference object plays a minor role in the visualization. Its main purpose is to provide orientation in 3D – especially if the actual objects (e.g. scanned remains of a vase) are much smaller than the reference object. The actual object may also be colorized according to the assigned distances.

4.4 Colors

The geometric objects / meshes as well as the distance visuals can easily be colorized using preconfigured color scales. These schemes include color maps with good order properties in terms of human perception.

Most of distance visualization schemes use luminance-based scales, for example the black-body radiation spectrum. For surfaces isoluminant color maps with opponent colors are suggested (see Figure 5). These surface colorizations do not compromise the depth perception.

Neutral color tables are also available, if extra highlighting of differences is not desired. If the geometry is shown in a single color (with possibly varying transparency), the application proposes a color which does not belong to the color scale. Furthermore it automatically generates a legend in an appropriate range.

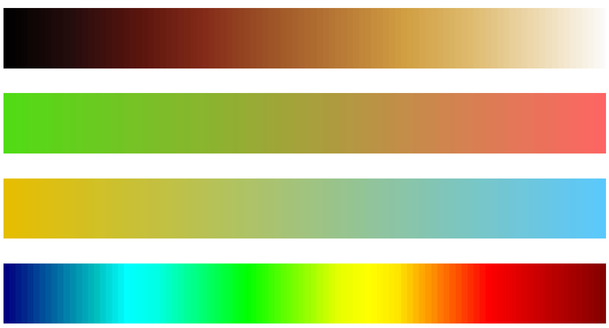

Figure 5: The color settings include color ranges, which take human perception into account; e.g. the Black-Body-Radiation scheme in the first row. According to BORLAND ET AL geometry should be drawn in an isoluminant scheme (second and third rows). The set of predefined color settings also contains some neutral color schemes, proposed by ROGOWITZ ET AL, as illustrated in the third row. It does not use "signal colors" such as red. Deceptive and misleading color ranges such as the rainbow color scheme (last row) are also included, as they are often used in wide-spread visualization systems.

5. IMPLEMENTATION

All of the described algorithms have been implemented into the Java-based application ABSTAND. The stand-alone program can be downloaded or started via a web browser:

`www.double-blind-reviewing.net/Abstand`[1]

The user only needs to provide two meshes to be analyzed.

5.1 File Format

For the input meshes and for the results of our application, a suitable file format is needed. There are too many "standard" formats for 3D data. The practical approach to this problem has been to look for a format that is widely used by common modeling applications and viewers. In this way, the import of geometry and further processing of the results is harmonized.

We use a subset of the 3D Object format (OBJ) introduced by Alias Wavefront. Coloring the results is done by a material file (MTL). In lack of per-vertex-colors, the material file stores a separate material for each color and transparency value. This subset ensures full compatibility to other applications.

The a generated legend can be included into the scene. Special care has been taken to ensure a correct rendering of the legend. Therefore, the captions are put on a texture template whereas the color scale itself is not part of the texture. It is built out of quads using the same material file and settings as used by the distance visuals. This approach ensures that the appearance of the colors is coherent. Different texture environments, which specify how texture values are interpreted when a fragment is textured, may otherwise lead to color discrepancies.

5.2 Usability

The visualization of distances can be done in various ways with many parameters. To have a useful and supportive tool, it is very important to have a good set of predefined parameters working out-of-the-box. Only by choosing a scenario based on the classification in 4.1 the application is able to automatically generate an appealing visualization. But tweaking of all the parameters is also possible in the advanced settings.

5.3 Manifold Approximation using Normal Vectors

The normal vectors of the triangles are used to define half-spaces. With these half-spaces the inside and outside of the difference space in-between is defined. To work properly, the normals have to point outwards of tessellated objects. There is no way to determine correct inside and outside spaces fully automatically in all cases, especially for surfaces with holes.

The distance calculation provides signed distances to each sample. Then it is possible to set the transparency according to the depth. The inside parts of a surface can be opaque and the outside parts transparent.

The distances can be visualized using simple cylindrical solids (or prisms with a lower polygon count). Using solids gives a more volumetric look than with lines. Furthermore, this representation is supported by almost all applications capable of importing OBJ files, while importing lines is only available in a small number of viewers.

5.4 Transparency

In most cases the visualization contains two surface layers. To show both objects it is necessary to use advanced techniques

[1] A video, which demonstrates the usage of the application, can be found in the "additional material" section on the conference server.

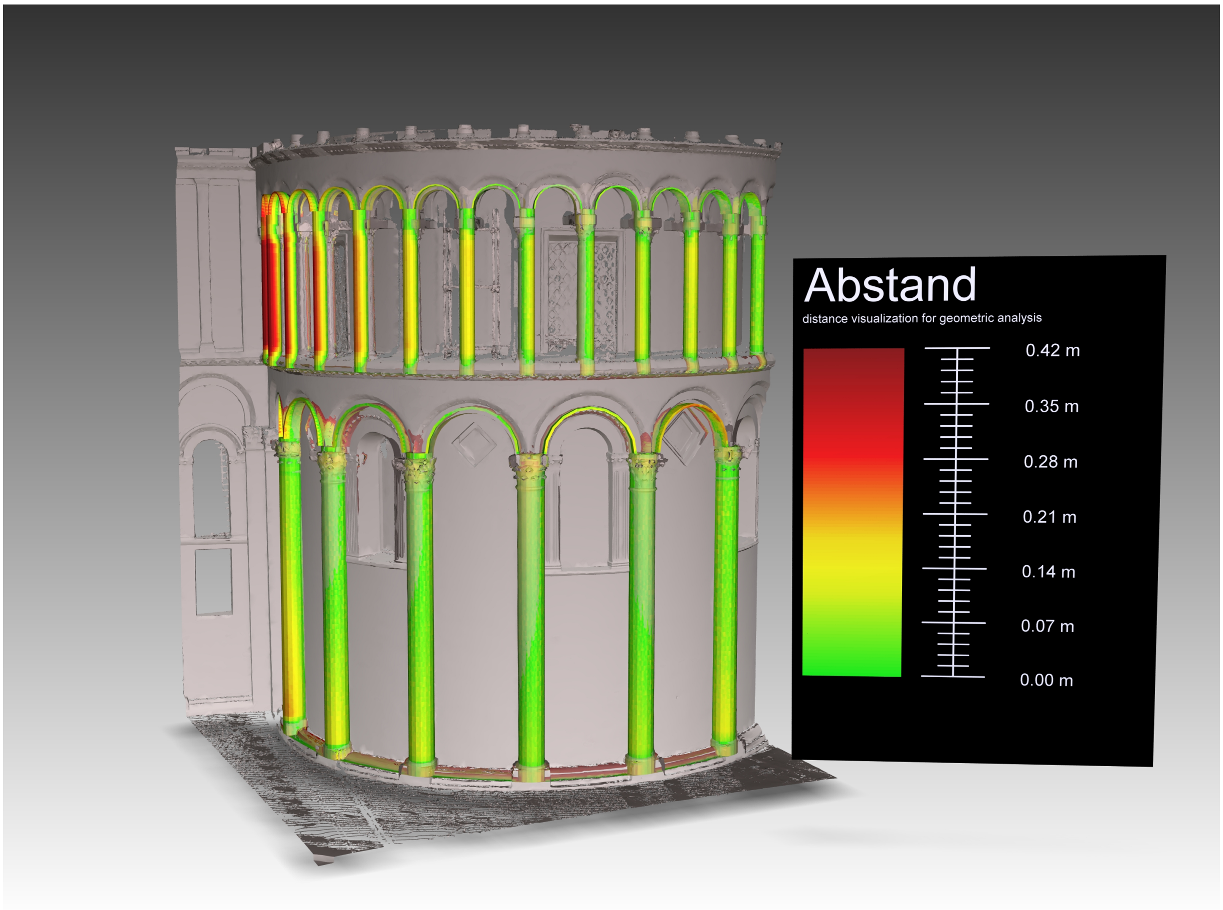

Figure 7: In an asymmetric configuration an actual/nominal comparison is visualized. The reference object is a laser scan of the Pisa Cathedral. Two fitted arcades have been determined by an algorithm. The difference between the scan and the automatically calculated surfaces describe the quality of the fit. Using ABSTAND these differences can be analysed and visualized easily.

such as cut-away illustrations, which can hardly be made automatically and need special viewers, or adequately chosen transparency. In order to use conventional tools/viewers we chose an automatic approach based on transparencies.

While in general it is no problem to render still images with transparency, interactive display may not always render transparencies correctly. Correct rendering of transparent objects needs a back-to-front sorting of all surfaces. Special care has to be taken for interpenetrating objects.

To have an appealing visualization for the interactive rendering, we offer a transparency simulation by a varying wire frame representation. Each transparent triangle, for example, is replaced by three quads as shown in Figure 6.

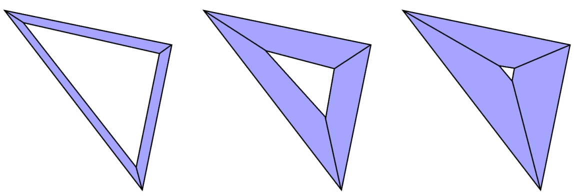

Figure 6: The overall transparency of a rendered object can be simulated by a varying wire frame representation. The area of the quads is inversely proportional to the triangle's transparency. This technique offers a comparable illustration in viewers, which do not render transparency correctly, at the expense of the polygon count.

The area of the quads is inversely proportional to the triangle's transparency. For high transparency values, the quads almost form lines along the borders of the triangle. For low values, the triangle appears rather opaque leaving only a small hole in the middle of the face. This technique offers a comparable illustration in viewers, which do not render transparency correctly, at the expense of the polygon count.

6. EXAMPLES

The first example shows a result of the asymmetric preset of ABSTAND. It shows a data set of the Pisa Cathedral, which has been generated by the Visual Computing Laboratory at the Institute of Information Science and Technologies (ISTI) of the Italian National Research Council (CNR).

The scan of the Duomo Pisa is compared to the results of an algorithm which identifies arcades automatically. The visualization – shown in Figure 7 – helps to verify the quality of the algorithm's output. Missmatching areas can be identified easily.

The second set of test objects demonstrates the symmetric case. It consists of two range maps from a laser scanner. The range maps have only a small area of overlap. Having set the upper bound of visualized distances slightly above the scanner's accuracy allows to concentrate on the alignment fit. Both objects have been acquired using a NextEngine^{TM} laser scanner.

Figure 8 shows a high-quality rendering of the resulting visualization. Furthermore a 3D representation is embedded in U3D format. Transparent faces are transformed according to the method described in 5.4 to ensure an appealing result independent of the capabilities of the U3D rendering plug-in.
The Acrobat™ U3D plug-in allows to inspect the surfaces as well as the histogram structure of the distances. The predefined groups can be selected, marked and hidden using the plug-in's tree view.

7. CONCLUSION

We presented an out-of-the-box application to visualize distances between surfaces in an easy to use way. The proposed classification scheme is suitable to assign all surface comparison tasks to two main sets of default settings. Matching color tables are suggested automatically on a scientific basis.
By using a well established file format, the output files are harmonized with common 3D rendering tools and viewers. On the one hand the resulting files can be used to produce still images in the classic ray tracing fashion. For interactive illustrations on the other hand the application offers a technique to optimize the export settings to get appealing results. These optimized settings avoid irritating misinterpretations and visualization errors in different viewers.
The overall work-flow to produce high-quality visualizations of distances between surfaces has been reduced significantly. Furthermore the application takes the latest results in human perception and visualization techniques into account.
Although the presented application offers many possibilities, sensible defaults allow an easy handling. The application is organized in a check list-like manner. Processing each point (choose distance function, select color scheme, etc.) step-by-step reduces the probability of many commonly made errors in diagrams (e.g. no legend included).
The application uses state-of-the-art algorithms to calculate the distances very fast, but in contrast to many other distance visualization programs it also concentrates on the resulting visualization: a clean representation of facts without overextending the observer.
Last but not least, the application is based on web technologies and works out-of-the-box and does not need any configuration or installation:

`www.double-blind-reviewing.net/Abstand`[2]

[2] A video, which demonstrates the usage of the application, can be found in the "additional material" section on the conference server.

REFERENCES

Aspert, N., Santa-Cruz, D., and Ebrahimi, T. 2002. MESH: Measuring Error between Surfaces using the Hausdorff distance. *Proceedings of the IEEE International Conference on Multimedia and Expo 2002 (ICME)*, 1, pp. 705–708.

Baltsavias, M., Gruen, A., Van Gool, L., and Pateraki, M. 2006. *Recording, Modeling and Visualization of Cultural Heritage*. Taylor & Francis.

Bergman, L. D., Rogowitz, B. E., and Treinish, L. A. 1995. A rule-based tool for assisting colormap selection. *Proceedings of the 6th Conference on Visualization*, 6, pp. 118–125.

Borland, D. and Taylor II, R. M. 2007. Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications*, 27(2), pp. 14–17.

Cignoni, P., Rocchini, C., and Scopigno, R. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2), pp. 167–174.

Dubuisson, M.-P. and Jain, A. K. 1994. A Modified Hausdorff Distance for Object Matching. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1, pp. 566–568.

Farin, G., Hamann, B., and Hagen, H. 2003. *Hierarchical and Geometrical Methods in Scientific Visualization*. Springer.

Gonnet, G. H. and Baeza-Yates, R. 1991. *Handbook of Algorithms and Data Structures*. Addison-Wesley Publishing.

Hege, H.-C. and Polthier, K. 2002. *Mathematical Visualization: Algorithms, Applications and Numerics*. Springer.

Klein, R., Liebich, G., and Strasser, W. 1996. Mesh Reduction with Error Control. *Proceedings of IEEE Visualization*, 1, pp. 311–318.

Levkowitz, H. and Herman, G. T. 1992. Color Scales for Image Data. *IEEE Computer Graphics and Applications*, 12(1), pp. 72–80.

Moore, A. W. 1991. An introductory tutorial on kd-trees. *Technical Report, Computer Laboratory, University of Cambridge*, 209, pp. 1–20.

Rogowitz, B. E., Treinish, L. A., and Bryson, S. 1996. How NOT to lie with visualization. *Computers in Physics*, 10, pp. 268 – 273.

Stone, M. 2003. *A Field Guide to Digital Color*. AK Peters, Ltd.

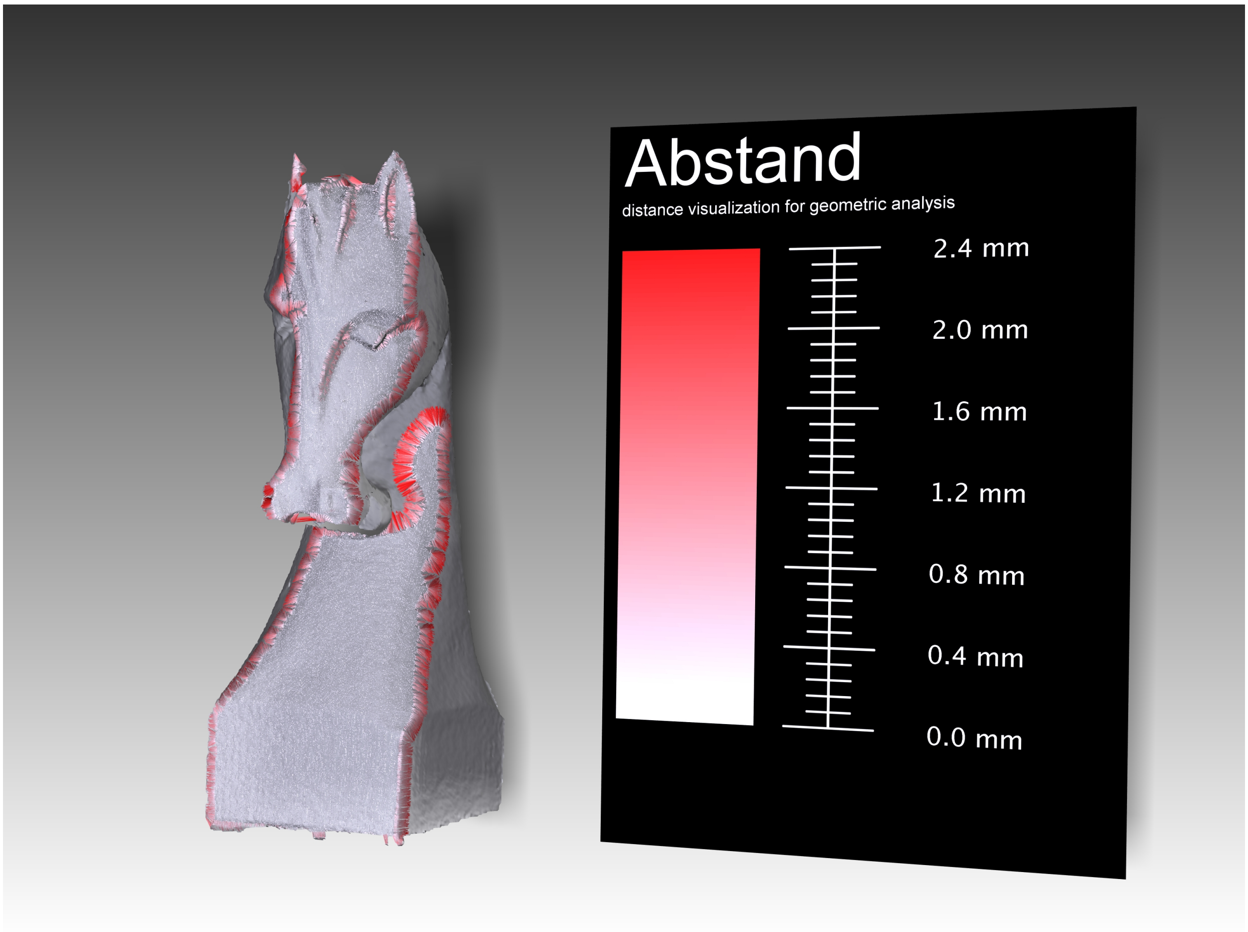

Figure 8: During the scanning process several range maps of a scanned object have to be aligned to each other. The presented application ABSTAND is able to create the visualizations that highlight the alignment fit. Having selected the class of settings only the range of distances to visualize has been adjusted to the accuracy of the scanner. The export routine has been harmonized with diverse tools to allow easy conversion – for example to U3D for PDF embedding.

Please note:

This interactive rendering requires at least Acrobat Reader 8.

The 3D rendering results may vary depending on your display settings, graphics hardware, or plug-in configuration.