ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2008, Vol.37, No.3

MEASURING ERROR BETWEEN SIMILAR SURFACES

Kęstutis Jankauskas, Algirdas Noreika

Multimedia Engineering Department, Kaunas University of Technology
Studentų St. 50, LT–51368 Kaunas, Lithuania

Abstract. This paper suggests that the measurement of error between mesh surfaces that are similar can be performed more time-efficiently than the existing methods offer. We presume that evaluation of error between similar surfaces does not require so many sample points and two-way distance measurement. Also we apply a different technique for sample point picking that involves barycentric coordinates. Tests versus other methods show that our method performs faster in cases of identical and similar surfaces, but also produces a less precise result. Thus, it is considered to be effective in applications that do not need precise measurements, like fast detection of flaws, deformation or overlapping areas.

Keywords: Error measurement, similar surfaces, surface difference, mes.

1. Introduction

The measurement of error between surfaces gives us information how similar or different they are. It is a convenient way to tell if one of presumably identical surfaces is flawed (not identical). This is one of geometrical analysis problems that lose a great deal of practical potential, because of relatively slow performance when processing large amount of data. Most of up-to-date applications, involving such analysis, process a large amount of data. While memory requirements can be satisfied easily, processor upgrades are usually expensive and are not always available. That is why time efficiency is the most considerable parameter of the analysis method. Thus we invested some effort to develop a method that exceeds the performance of the existing similar algorithms.

The existing error measurement tools, like *Mesh* 0 and *Metro* 0, were primarily developed to evaluate error between surface and its simplified copy, but also work in general case as well. Both tools evaluate Hausdorff distance between certain points (samples), lying on compared surfaces. The mean value, acquired from distances, represents the level of difference. The precision and performance of such algorithms is determined by the quantity of sample points, the point projection method and also the distance evaluation technique. For example, *Mesh* suggests a sampling step, smaller than 0.5% of the diagonal length of the bounding box. Also it employs a triangle region check to ensure that the sample point is projected within the bounds of a face.

In this work we introduce a new constraint to the problem and also expect a substantial increment of time efficiency at low cost of precision. Actually, we consider the compared surfaces to be similar. It allows us to use less samples (one sample per vertex) and simple Euclidean distance instead of Hausdorff distance. We also apply less time-consuming sample projection technique that involves *barycentric* coordinates [2, 5, 6, 7]. The concept and the tests of time efficiency increment versus precision loss are stated in further sections.

2. Evaluation of Error between Similar Surfaces

The mesh representation of the surface S can be compared to the other mesh surface S' by measuring distances $d(p_i, p_i')$ between surfaces at certain points, called samples (sample points). Sample points p_i lie on the surface S and p_i' lie on S', where $i = 1, 2.. N$; N denotes the number of samples on one surface. In our case, the surface S is called a *base surface* (pivot mesh 0 – a pattern to which other surfaces are compared) and S' is called a *destination surface*. Consequently, distances $d(p_i, p_i')$ represent error values between base and destination surfaces. High error values mean a significant difference between S and S', while considerably low error values suggest S and S' to be very similar. Presumption that surfaces are similar allows setting an upper bound r for $d(p_i, p_i')$, called a *search radius*. All areas of the surface S' containing p_i', such that $d(p_i, p_i') > r$, are interpreted as dissimilar and are excluded from further calculations. The constraint of similarity is the key concept in formulating the problem of evaluation of error between two similar surfaces; it is covered in the following section.

2.1. Surface Similarity

There are a few requirements that must be met before measuring error. Whereas evaluation is performed between two surfaces represented as meshes, similarity is discussed in mesh–specific terms. Since the distance $d(p_i, p_i')$ is measured between sample points p_i and p_i' lying on meshes, each sample is either a specific vertex point or a certain point on a face. Obviously, to obtain the smallest distance values, base and destination mesh elements (vertices and faces) must be as close to each other as possible. That is, if both surfaces are identical (all distances expected to equal zero), but oriented differently, most of distances will be greater than zero. Thus the position, orientation and scale of both surfaces are expected to be adequate. These are called *calculation space integrity requirements*, while the *calculation space* is a three-dimensional space defined by a single box bounding both meshes.

Apart from calculation space integrity requirements, meshes must meet *discretization step requirements* that also have a great impact on the definition of similarity. Two meshes, acquired from the same surface, using a different discretization step, are considered to be less similar than meshes acquired using the same step. Also, there are *topology requirements*, involving vertex indexation. The same surfaces, meshed using different methods to generate topology, despite different enumeration and usage of different vertices to form edges as long as they are correct (no crossing edges, no duplicated vertices, etc.), are treated as slightly dissimilar. Therefore topology requirements can be omitted, as long as faces are formed using the same number of vertices. In this paper, faces are considered to be triangles, formed by three vertices.

We need a more formal definition of the *factor of similarity*, involving distances $d(p_i, p_i')$ and the requirements stated before. Let a vertex v belong to the set of vertices V of the surface S and v_j' belong to the set V' of the surface S'. The index j is such that the distance $d(v, v_j')$ is a minimum possible Euclidean distance between v and vertices in V':

$$d_v = \min_{v_j' \in V'} d(v, v_j') \ . \qquad (1)$$

The sum of such distances d_v per area of the surface S divided by the maximum available distance d_{max} gives a normalized value that defines how different two surfaces are. Thus the factor of similarity can be calculated from the expression:

$$\varsigma = 1 - \frac{1}{d_{\max}|S|} \sum_{i=1}^{N_F} \frac{d_{v1i} + d_{v2i} + d_{v3i}}{3} |F_i| \ , \qquad (2)$$

where d_{v1i}, d_{v2i}, d_{v3i} are distances from the vertices of a face F_i and $|F_i|$ denotes its area, also N_F stands for the number of the faces of the surface S. The notation $|S|$ represents the overall area of the surface S. Since surfaces must meet calculation space integrity requirements, d_{max} is calculated by taking a half length of the bounding box diagonal. In fact, switching base and destination surfaces results in a different factor of similarity (a non-symmetrical case – see Section 2.2.2), therefore the minimum value is chosen: $\varsigma = \min(\varsigma_{SS'}, \varsigma_{S'S})$, where $\varsigma_{SS'}$ is acquired when S is a base surface and $\varsigma_{S'S}$ is acquired when S' is a base surface. Obviously, $\varsigma = 1.0$ means an ideal similarity when the surface is compared to itself. The situation $\varsigma = 0.0$ will never occur, unless one of surfaces is concentrated in one point. It is important to note that the distance d_v is usually not the smallest Euclidean distance between surfaces (as mentioned before, the smallest distance represents error value), but it encapsulates information about discretization.

2.2. Error Measurement

Once the compared surfaces are considered to be similar, the error measurement algorithm can be defined by several steps: (1) the search of sample points, (2) the calculation of the distance between certain sample points, (3) the generalization of results. The search of sample points is the most complicated step, involving sample picking techniques. The calculation of the distance includes a simple mathematical processing regarding to either symmetrical or non-symmetrical case. The generalization consists of the extraction of statistical values from raw data.

2.2.1. Sample Point Search

The search of sample points is divided into finding samples p_i, lying on the base surface S, and finding an appropriate sample point p_i', lying on the destination surface S', for each point p_i. Since the step of discretization can not be infinitely small, the mesh representation can not offer complete accuracy for smooth surfaces. Thus base surface mesh is taken as it is, presuming it provides 100% correct information about geometry. All spatial information (peaks) is presumed to be concentrated in vertices, while points along faces represent only intermediate values. Therefore all vertices of a base surface are treated as required sample points $p_i = v_i \in V \subset S$, where i = 1, 2. N_V; N_V represents the number of vertices of base surface. Points $p' \in S'$ are found according to two cases of the minimum distance between surfaces (see Figure 1).

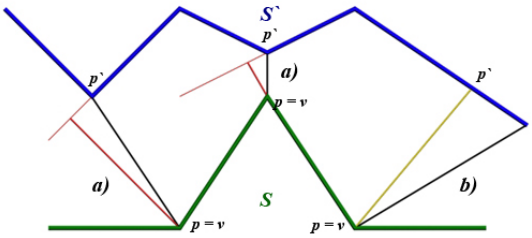


Figure 1. Two cases of the minimum distance: a) vertex-vertex; b) vertex-face

In Figure 1 the blue line illustrates a 2D projection of the faces of the surface S', the green line illustrates the faces of the surface S. An appropriate p_i' for each sample p_i is found either by picking the closest vertex (a) vertex-vertex case), or by calculating the closest point on a certain face $F_j' \in S'$ (b) vertex-face case). In vertex-face case, the point p_i' is the projection of the sample p_i to the face F_j' and the line p_ip_i' is parallel to the normal of the face F_j', where F_j' is the closest face to the point p_i. As faces are triangles bounded by specific vertices, a sample point p_i' can be outside the triangle (represented by red lines in Figure 1 – case a)). Thus the closest vertex of the destination surface is picked as p_i' instead. Generally, the sample p_i' is the closest point of all vertices V' and all projection points (p_i to F_j'), positioned inside the triangle of the face F_j' where $j = 1, 2.. N_{F'}$ and $N_{F'}$ denotes the number of the faces of the destination surface S'. The length of p_ip_i' must not be greater than the search radius.

Since every point p_i is to be projected onto a face F_j' (to acquire a sample point p_i') and p_i' must be inside the triangle of the face F_j', two problems must be solved: (1) the projection of a point and (2) the determination if a point is inside a triangle. Alternatively, a single task can be formulated instead: the line-triangle intersection, where the line is parallel to the normal of F_j' and includes the point p_i. This operation is performed many times during the search of sample points on a destination surface, thus a time efficient method is vital for the entire error measurement process. The evaluation of three different approaches was performed to determine the most efficient one: (1) the calculation of a line and a face plane intersection point, forming three angles with vertices of the same face, that sum equals 360 degrees 0; (2) the conversion of point coordinates to barycentric coordinates of a triangle 0, 0, 0 and 0; (3) the calculation of a line and a face plane intersection point, passing a triangle region check 0. The barycentric coordinate method involves no cross product operations and requires no pre-calculated projection point. Therefore it proves to be the most effective in our case.

The concept of barycentric coordinates enables us to convert 3D space coordinates of a point to the parametric coordinates of the specific triangle 0. Barycentric coordinates (u_1, u_2) define if the projected point p_i' falls within the bounds of the triangle of the face F_i'. Furthermore, it is simple to convert barycentric coordinates back to 3D coordinates, which gives the position of the projection point (not the original point):

$$\begin{cases} p_i' \notin F_j', \text{if } u_1 < 0 \text{ or } u_2 < 0 \text{ or } u_1 + u_2 > 1 \\ p_i' = v_1' + u_1(v_3' - v_1') + u_2(v_2' - v_1'), \text{ otherwise} \end{cases}, \quad (3)$$

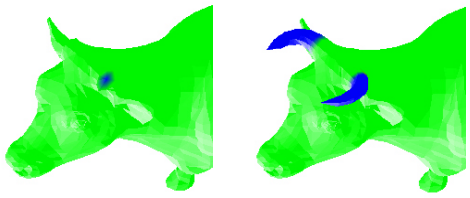
where v_1', v_2' and v_3' are three vertices of the face F_j'. If p_i' is located outside F_j', then the position of p_i' is considered invalid and a more distant sample point (i.e. a vertex or other projection point) is taken as p_i'. It is possible that there is no valid sample point within the search radius. Such distances $d(p_i, p_i')$ are not regarded in the step of the generalization of results.

2.2.2. Distance Calculation

Once every vertex v_i of a base surface S is a sample point p_i and has a closest point p_i' on a destination surface S', the distance $d(p_i, p_i')$ can be calculated for every v_i, where $i = 1, 2.. N_V$; N_V is the number of vertices in $V \subset S$. The distance $d(p_i, p_i')$ represents the error value and shows how far the vertex v_i is from the destination surface (this is not applicable to vertices, which distance from S' exceeds the search radius). The distance between two points $p = (x, y, z)$ and $p' = (x', y', z')$ is calculated from the expression:

$$d(p, p') = \sqrt{(x'-x)^2 + (y'-y)^2 + (z'-z)^2}\ . \quad (4)$$

Let the function $cl(v_i)$ refer to a point $p_i' \in S'$, the closest point from the vertex v_i to the surface S'. Since $p_i = v_i$, the distance between points p_i and p_i' can be expressed: $d(v_i, cl(v_i))$. Let S' stand for the base surface instead of S and S stand for the destination surface. In that case the number of sample points $p_i' = v_i'$ equals the count of vertices v_i' and the distance for every v_i' equals $d(v_i', cl(v_i'))$. Whereas $v_i \neq v_i'$, $v_i \neq cl(v_i')$, $cl(v_i) \neq v_i'$ and $cl(v_i) \neq cl(v_i')$, obviously, $d(v_i, cl(v_i)) \neq d(v_i', cl(v_i'))$, except when surfaces are identical and $v_i = v_i'$. This gives us a general case of a non-symmetrical distance, when error values obtained from calculations for S, standing as the base surface (forward distance measurement) and S, standing as the destination surface (backward distance measurement), are different 0, 0 (see Figure 2).



a) b)

Figure 2. A non-symmetrical distance case, when surfaces are not similar: a) a cow with no horns; b) a cow with horns

In general case, compared surfaces can be very dissimilar. Results from forward and backward distance measuring may differ greatly. In Figure 2, two surfaces are compared: a cow with no horns and a cow with horns. The green color denotes similar areas, and the blue color denotes dissimilar areas. The cow with horns treated as the base surface produces more dissimilar area than the cow with no horns. This is called a non-symmetrical problem. Asymmetry arises because of the different number of mesh elements and the different orientation of faces. Thus the Hausdorff distance is employed in order to make calculation results symmetrical. It requires forward and backward calculations, and returns the maximum value as the

result. In case, when surfaces are similar, we presume that forward and backward distances are very small and nearly the same, so only one way calculation is needed.

2.2.3. Result Generalization

Since the distance is measured for every vertex of the base surface, the comparison of high resolution surfaces produces a large amount of data. Thus result generalization is necessary to provide a few values, representing the overall error value. The generalization is performed by calculating the average error value for the area unit. *Mean error* (d_{ME}) and *root mean square* (d_{RMS}) values are sufficient for this task:

$$d_{ME} = \frac{1}{|S|}\sum_{i=1}^{N_F}\frac{d_{v1i} + d_{v2i} + d_{v3i}}{3}|F_i|, \quad (5)$$

$$d_{RMS} = \sqrt{\frac{1}{|S|}\sum_{i=1}^{N_F}\left(\frac{d_{v1i} + d_{v2i} + d_{v3i}}{3}\right)^2|F_i|}, \quad (6)$$

where d_{v1i}, d_{v2i}, d_{v3i} are distances from vertices of the face F_i, $|F_i|$ is the area of F_i and N_F is the number of the faces of the surface S. The notation $|S|$ represents the overall area of the surface S. Mean error and root mean square values are non-negative and equals zero only if surfaces are identical.

3. Implementation

This section covers practical implementation of the suggested method for the measurement of error between similar surfaces. The method was implemented using the C# .Net programming language and tested for performance versus the method presented in 0.

As noted in [1, 4], a full search of the closest element of the surface S' for all vertices $V \subset S$, in general case, leads to the number of iterations equal to $N_F \times N_{F'}$. Here N_F stands for the number of faces $F \subset S$ and $N_{F'}$ is the number of faces $F' \subset S'$. Therefore to reduce the number of iterations, segmentation of the calculation space is applied. The entire calculation space is defined by a box that bounds compared meshes. Segmentation is performed by dividing the bounding box into smaller cubic boxes, called cells [1, 4]. Every cell has its index and references to certain elements of meshes. It is obvious that elements in the same cell are closer to each other than elements in two distant cells. As we seek for the shortest distance, only adjacent cells must be searched.

As it is noted in Section 2.2.1, vertices of the base surface S are used in the calculation of error, so they are assigned to certain cells. In the case of the surface S', not only vertices, but also faces are assigned to cells. Presuming that surfaces are similar, faces are assigned by taking four points: vertices v_1', v_2' and v_3' of a face and its midpoint m' (drawbacks will be noted in Section 4.5) defined as follows:

$$m' = \frac{v_1' + v_2' + v_3'}{3}. \quad (7)$$

The segmentation of the space allows saving the search time, but also decreases the integrity of the calculation space. In fact, the distance between two points in neighboring cells may even be smaller than the distance between points in the same cell. Thus all cells of at least the first rank of adjacency must be searched (except identical surfaces). Apart from both surface mesh information, the algorithm requires a few more parameters, described in the following section.

3.1. Input Parameters

The set of input parameters includes the search radius (recall Section 2), the *number of cells* and the *maximum rank of adjacency*. The search radius (r) defines how far from a given point the search is performed. In case, when space segmentation is applied, the search radius defines the maximum rank of the adjacency of cells to be searched. To simplify the problem, the adjacent cells are taken in the pattern of cube, not sphere. In practice, it is handier to define this parameter in an absolute value. While comparing shapes of differently scaled objects, it is better to handle r relatively to the length of the bounding box diagonal.

The number of cells ($N_{CX} \times N_{CY} \times N_{CZ}$) defines the degree of space segmentation. A higher number of cells reduces the number of search iterations in one cell, but accordingly increases the number of cells to be processed, as r is constant. Another input parameter, the maximum rank of adjacency (R_{CMAX}) denotes how far from a given cell the search must be performed in order to find an appropriate point. The rank $R_C = R^0{}_C = 0$ means the same cell ($N^0{}_C = 1$), $R_C = R^1{}_C = 1$ means all cells adjacent to it, that comprises $N^1{}_C = 26$ cells. Actually, the formula that relates the rank of adjacency $R^i{}_C$ to the number of cells to be processed $N^i{}_C$ is:

$$N_C^i = (2 \cdot R_C^i + 1)^3 - N_C^{i-1}. \quad (8)$$

This gives the maximum number of cells to be processed:

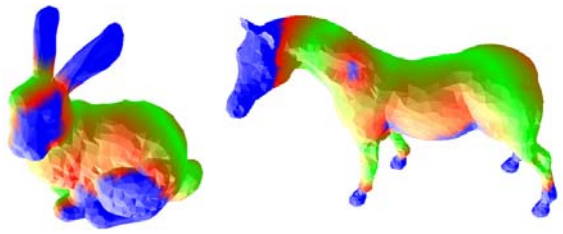
$$N_C = (2 \cdot R_{CMAX} + 1)^3. \quad (9)$$

It is clear that even small ranks, like $R_{CMAX} = 5$, involve a considerable amount of cells to be searched ($N_C = 1331$).

All these parameters are related, but to calibrate the algorithm, they must be altered separately. In this manner, the most effective configuration is found (see Section 4.1). The number of cells and the maximum rank are set automatically, according to the number of faces and the given search radius, relieving a user of the burden. Once the error measurement between surfaces is completed, the algorithm outputs the results.

3.2. Output Parameters

The algorithm returns error values for all vertices of a base surface (Figure 3) as well as marks vertices, where distance was greater than the search radius *r*. Zero distance points are colored in green and points, where the distance equals the search radius, are colored in red. All intermediate colors denote intermediate values. The blue color marks vertices, where an appropriate point on other surface within the bounds of *r* has not been found. Also the algorithm returns several generalized values.



a) b)

Figure 3. Comparison of two dissimilar surfaces: a) a bunny and a horse; b) a horse and a bunny

Apart from the mean error (d_{ME}) and the root mean square (d_{RMS}), defined in Section 2.2.3, there are a few more parameters that describe the error measurement process: *iterations* (*it*), *blindness* (*b*) and the *calculation time* (*t*). Iterations show the number of iterations performed during the search. Blindness evaluates the percentage of vertices for which the distance from a destination surface exceeds the search radius (blue vertices in Figure 3). It is important to note that blue vertices are not involved in the calculation of the mean error and the root mean square. Finally, the *calculation time* involves the duration of assignment to cells, the search of sample points and the calculation of distances. It is not linear dependant on iterations, because iterations do not include cell preparation. Therefore the ratio *t/it* gives information about cell preparation overhead.

In addition, the *precision* (P_d, P_{diag}) parameter is evaluated. As it is noted before, error measurement between similar surfaces requires less sample points and only one-way distance measurement, while error evaluation between dissimilar surfaces demands more sample points and two way distance measurement. The reduction of the number of performed iterations shortens the calculation time, but also reduces precision. To find out how precise the suggested method is, the following expressions are calculated by comparing the acquired mean error value d_{ME} to the values of the forward mean error D_{FME} and the backward mean error D_{BME}, obtained by *Metro* tool. Precision is evaluated with respect to the average mean error value $D_{FME}+D_{BME}/2$, calculated by *Metro*, and to the half of the diagonal length of the bounding box *diag/2*:

$$P_d = 1 - \frac{\left|D_{FME} + D_{BME} - 2d_{ME}\right|}{D_{FME} + D_{BME}}, \qquad (10)$$

$$P_{diag} = 1 - \frac{\left|D_{FME} + D_{BME} - 2d_{ME}\right|}{diag}. \qquad (11)$$

4. Results

In this section the suggested method is referred to as *MESS* (*Measuring Error between Similar Surfaces*). Results obtained by *MESS* were compared to results calculated by *Metro* 0. *Metro* was chosen instead of *Mesh* 0, because its developers offered a more up-to-date version. The newest version of *Metro 4.07* has been downloaded from 0. Also 3D models, as input data, were taken from 0. The performance of algorithms was tested on a Pentium IV 3.20 GHz, 1 GB RAM machine. The following sections cover the parameter calibration and the analysis of the results of error measurement between identical, similar, dissimilar surfaces. Also Section 4.5 describes an exceptional case of simplified surfaces.

4.1. Parameter Calibration

During calibration tests that involve various numbers of cells (Figure 4), calculation time jumps were observed. The jumps occur due to the increment of R_{CMAX}. A larger number of cells means a smaller cell size (l_C – the length of the cell edge), while under the constant search radius, a greater number of cells must be searched. It is also obvious that the increment of the search radius radically extends the calculation time as well as the amplitude of jumps.

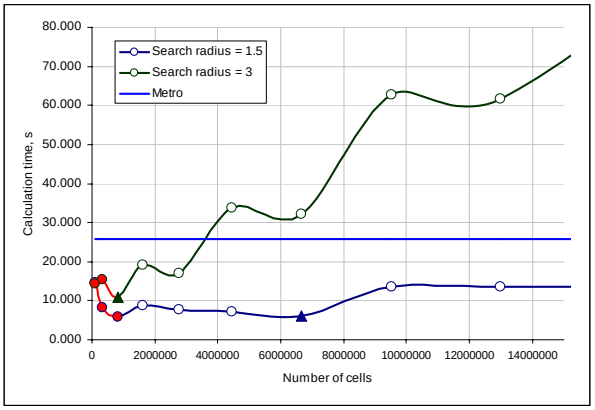


Figure 4. Calculation time versus the number of cells (red circles indicate $b > 0$)

Every jump of the calculation time has its local minimum. Naturally, the first local minimum is the lowest. As described in Section 3.1, it occurs when $R_{CMAX} = 0$, which means a great probability of miscalculation ($b > 0$), since no adjacent cells are searched. Therefore, the second local minimum is taken as the most effective. This happens when $r=2\alpha \cdot l_C$. Experiments show that the value of $\alpha \approx 0.95$ is suitable. Having l_C and r related decreases the performance on relatively low search radii. Also to avoid segmentation overhead, the number of faces per

cell must be regarded. The following formula shows how l_C is evaluated:

$$l_C = \frac{1}{1.9}\max\left(r, \sqrt[3]{\frac{0.02 \cdot diag^3}{N_F + N_{F'}}}\right), \qquad (12)$$

where *diag* is the length of the diagonal of the bounding box, and the constant 0.02 was chosen experimentally. It defines the most optimal experimentally found relation between the number of faces and the number of cells, it can vary for specific surfaces.

4.2. Identical Surfaces

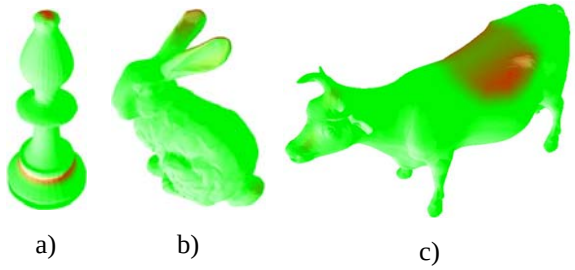
In our first experiment, various surfaces were compared to themselves (see Table 1). Naturally, the factor of similarity (ς) equals 100%. Error measurement between identical surfaces shows that lower resolution surfaces (containing a small amount of faces) were processed 2.6 times faster, while higher resolution surfaces were processed up to 8.3 times faster.

Table 1. Error measurement between identical surfaces: the calculation time and the precision

Surface (face count)	Calc. time, s		P_d, %
	Metro	*MESS*	
BishopH (269 568)	91.600	11.015	100
BunnyH (186 752)	22.953	5.828	100
Cow (5 804)	0.485	0.187	100

4.3. Similar Surfaces

In the tests described in this section, surfaces were compared to a slightly deformed version of themselves. The mesh surface, representing a chess bishop, was deformed at the top and near the base (in Figure 5a) red spots mark-deformed areas). Also 3D model of a bunny had its ears vertices repositioned (Figure 5b), while the mesh of a cow was slightly deformed at the back (Figure 5c). All surfaces retained the discretization step (the number of faces– Table2). The factor of similarity between the compared surfaces equals 99.9% approximately.



a) b) c)

Figure 5. Comparison of similar surfaces: a) a bishop; b) a bunny; c) a cow

In the case of similar surfaces, the suggested method (*MESS*) performed from 2.2 up to 3.2 times faster than *Metro*. Obviously, time saving is significantly lower than in the case of identical surfaces. Also the loss of precision is quite significant in the comparison of cowH and CowH2 meshes (Table 2). On the other hand, absolute values of error between similar surfaces are very small, so precision, relative to a half of the diagonal length of the bounding box, still reaches 99.96%.

Table 2. Error measurement between similar surfaces: the calculation time and the precision

Base (face count)	Destination (face count)	Calc. time, s		P_d (P_{diag}), %
		Metro	*MESS*	
Bishop (7 936)	Bishop2 (7 936)	0.765	0.234	96 (99.99)
BunnyH (186 752)	BunnyH2 (186 752)	25.877	11.296	95 (99.99)
CowH (139 296)	CowH2 (139 296)	22.328	8.370	70 (99.96)

4.4. Dissimilar Surfaces

In the tests described below, surfaces were compared to different surfaces. A visual representation of the error value per vertex is given in Figure 7 and generalized values are shown in Table3. A chess king is compared to a chess pawn (Figure 7a), then surface of a cow is compared to a chess king (Figure 7b) and finally, the mesh of a horse is compared to the mesh of a cow (Figure 7c). The factor of similarity differs in each case, so it is also given in Table 3.

a) b) c)

Figure 6. Comparison of dissimilar surfaces: a) king-pawn; b) cow-king; c) horse-cow

Table 3. Error measurement between dissimilar surfaces: the calculation time, the precision and the factor of similarity

Base (face count)	Destination (face count)	Calc. time, s		P_d (P_{diag}), %	ς, %
		Metro	*MESS*		
King (8 448)	Pawn (4 864)	4.609	10.953	98 (99.88)	93.4
Cow (5 518)	King (8 448)	29.532	139.92	97 (99.36)	78.0
Horse (5 946)	Cow (5 804)	4.766	3.640	77 (98.16)	91.6

The calculation time of *MESS* in two (out of three) cases exceeded the calculation time of *Metro*, since it is closely related to the search radius. Higher *r* values

result in a lower number of cells that pulls further off the optimal ratio of the number of faces to the number of cells. This can be avoided by altering the calibration mechanism. In some cases, the measured mean error is very close to the average value of forward and backward mean distances, measured by *Metro*. Surprisingly, the method maintained decent precision, relative to the half of the diagonal length of the bounding box P_{diag}.

4.5. Simplified Surfaces

Tools, like *Metro* 0 and *Mesh* 0, were developed primarily to compare the original surface to its simplified copy (i.e. a reduced number of vertices and faces). Our method is not suitable for fast calculation of the error value between surfaces that involve a simplified version of a mesh (recall the discretization step in Section 0). The usage of a limited number of sample points, while assigning faces to cells, causes high blindness values (Figure), because the length of the search radius is insufficient to reach those cells. This situation can be avoided by either increasing the search radius, or increasing the number of points, while assigning faces to cells. Both solutions extend the calculation time. The comparison of a simplified version of the surface to the original surfaces does not cause this problem.

Figure 7. Error measurement of a higher resolution surface versus a lower resolution surface

5. Conclusion

The suggested method demonstrated a good performance, while comparing identical and similar surfaces. The algorithm handled the given tasks from 2.2 up to 8.3 times faster than *Metro 4.07*, although in some cases of error measurement it was less precise than *Metro*. The error measurement between dissimilar surfaces, because of a high search radius, in most cases was processed slower, but retained quite a decent precision. The analysis of the results suggests that, under a low search radius, the method proves to be very effective. This is useful in fast detection of flaws or deformed areas, as well as overlapping areas of surfaces. Applying this method can save some time when the identity of a surface must be confirmed, regardless to a different topology. Because of the lack of precision in the error evaluation, the method is not suitable for applications, where high accuracy is required.

References

[1] **N. Aspert, D. Santa-Cruz, T. Ebrahimi.** MESH: measuring errors between surfaces using the Hausdorff distance. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'*02*)*, *August* 2002, *Vol*.1, 705-708.

[2] **J. Bikker.** Raytracing Topics & Techniques – Part 7: Kd-Trees and More Speed. *Flipcode*, *November* 2004.

[3] **P. Bourke.** Determining whether a line segment intersects a 3 vertex facet. *WASP (Western Australian Supercomputer Program), February* 1997. *http://local.wasp.uwa.edu.au/~pbourke/geometry/linefacet/*.

[4] **P. Cignoni, C. Rocchini, R. Scopigno.** Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 1998, *Vol*.17, *No*.2, 167-174.

[5] **D.H. Eberly.** 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics. *Morgan Kaufmann Publishers*, 2001.

[6] **C. Ericson.** Real-time collision detection. *Morgan Kaufmann Publishers*, 2004.

[7] **P. Yiu.** The uses of homogeneous barycentric coordinates in plane Euclidean geometry. *International Journal of Mathematical Education in Science and Technology*, 2000, *Vol*.31, *No*.4, 569-578.

[8] Computer Graphics – Object library. *http://www.cs.technion.ac.il/~cs234325/Software-Sources/Models/index.html*, 2008.

[9] Metro 4.07 tool: *http://sourceforge.net/project/showfiles.php?group_id=98764&package_id=121927&release_id=507635*, 2007.

Received May 2008.