

A Perception Correlated Comparison Method for Dynamic Meshes

Libor Váša and Václav Skala, *Member, IEEE*

Abstract—There are multiple areas of computer graphics where triangular meshes are being altered in order to reduce their size or complexity, while attempting to preserve the original shape of the mesh as closely as possible. Recently, this area of research has been extended to cover even a dynamic case, i.e., surface animations which are compressed and simplified. However, to date very little effort has been made to develop methods for evaluating the results, namely the amount of distortion introduced by the processing. Even the most sophisticated compression methods use distortion evaluation by some kind of mean squared error while the actual relevance of such measure has not been verified so far. In this paper, we point out some serious drawbacks of the existing error measures. We present results of the subjective testing that we have performed, and we derive a new measure called Spatiotemporal edge difference (STED) which is shown to provide much better correlation with subjective opinions on mesh distortion.

Index Terms—Animation, distortion, evaluation, error, measure, discrepancy, dynamic mesh.

1 INTRODUCTION

FOR about 20 years, we have seen attempts to reduce the memory requirements of triangular meshes based on sophisticated encoding or simplification of surface meshes. These efforts do not lose their value with increasing availability of high performance hardware, because it is always useful to be able to store and transmit more detailed models. On the other hand, it had been the increase of processing power of current hardware that recently enabled the extension of the original static mesh compression problem to the more computationally demanding dynamic case, i.e., the compression of surface animations.

Most of the techniques for both static and dynamic mesh compression presented so far are lossy, i.e., the decompressed data is slightly different from the original data. The goal of lossy compression is to achieve the best possible rate/distortion ratio, i.e., we want to obtain a mesh which can be stored in a low number of bits, and at the same time, we want to preserve some property of the mesh (low distortion). Quite surprisingly, there has been only very little effort invested into the evaluation techniques. Most of the papers simply use the mean squared error (MSE), or some of its derivatives. This might seem sufficient, because the MSE behaves intuitively in cases such as adding Gaussian error with growing deviation (the MSE grows as well) or adding sinusoidal error with growing amplitude (the MSE again grows).

However, as soon as we start adding different kinds of distortion, the MSE fails completely. It is unable to distinguish between smooth error (neighboring vertices

shifted by similar amount and in similar direction) and random error (neighboring vertices shifted by a random amount in random directions), even though intuitively there is a big perceptual difference between these two. Such situation is actually very common in practical compression and processing tasks, where usually multiple parameters control the quality of the result (roughness of quantization, number of basis vectors used for PCA based schemes, etc.), however, each parameter may introduce an error of different character. These parameters are usually set by some kind of rate/distortion optimization, and it is therefore, **essential** to have an error measure which accurately measures the property of the processed which we want to preserve. MSE works well only as long as the property being preserved is “vertices located near their original location,” however, we will show that “visual similarity with the original” is a different property and it should be therefore evaluated by a different measure.

In the next section, we will show the mesh comparison methods used so far, identifying the same problems for all of them. Motivated by this insufficiency, we had performed a series of subjective experiments which showed that the correlation of the MSE (and other measures) with subjective distortion perception is only very limited. Based on the results, we have designed a new measure, the STED error, which will be described and configured to fit the subjective testing results. Finally, we will show that the STED measure had predicted some counterintuitive phenomena which were confirmed by our final subjective experiment.

2 USED NOTATION

Throughout the rest of the paper, we will be using following symbols:

- F stands for the total number of frames of the animation.
- V stands for the number of vertices in each frame.
- T stands for the number of triangles in each frame.

• The authors are with the Centre of Computer Graphics and Visualization, Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic. E-mail: {loasa, skala}@kiv.zcu.cz.

Manuscript received 12 Dec. 2008; revised 22 June 2009; accepted 23 Sept. 2009; published online 12 Feb. 2010.

Recommended for acceptance by C. Gotsman.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCG-2008-12-0203. Digital Object Identifier no. 10.1109/TVCG.2010.38.

- v_i^f stands for a three component vector of XYZ coordinates of the i th vertex in the f th frame.
- xv_i^f stands for the X coordinate of the i th vertex in the f th frame.
- Overline denotes values or functions of a distorted version of the dynamic mesh, i.e., \bar{v}_i^f stands for a three component vector of XYZ coordinates of the i th vertex in the f th frame of a distorted dynamic mesh.

3 STATE OF THE ART

Although dynamic mesh compression has been thoroughly investigated only recently, there are already countless methods starting probably from Lengyel [1], through Dynapack by Ibarria and Rossignac [2] up to recent advances such as the Scalable encoder by Stefanoski et al. [3], the Coddyc scheme by Váša and Skala [4], [5], [6], and the new MPEG standard, the FAMC algorithm [7]. There is no point in going into detail about these methods. All we need to know is that they produce a decompressed mesh which is a modified version of the original. We will focus on the methods used to evaluate the amount of distortion. The most intuitive, and also widely used method is the MSE. It can be applied to the cases when the connectivity has not been changed, i.e., when we know that each original vertex v_i corresponds to a decompressed vertex \bar{v}_i . The error is a simple sum of squares of the deviations over the total number V of vertices:

$$MSE = \frac{1}{V} \sum_{i=1}^V (\|v_i - \bar{v}_i\|)^2. \quad (1)$$

Some papers dealing with dynamic mesh compression, [8], [9], [10], simply show the MSE for each of the F frames in a graph, others choose to sum or average the MSE values. Karni and Gotsman [11] have proposed to perform a different measurement which has been later adopted by multiple papers, [12], [13], [3], and which is known as the KG error. They suggest to reform the dynamic mesh into a matrix A of size $3V \times F$ where each row represents the temporal development of one coordinate of one vertex, and therefore, each column represents one frame of the animation. Similarly, the decompressed animation is represented by matrix A' , and the error is expressed as:

$$KG_{error} = 100 * \frac{\|A - A'\|}{\|A - E(A)\|}, \quad (2)$$

where $\|\cdot\|$ denotes Frobenius norm, $E(A)$ is a matrix of size $3V \times F$, which is computed from the matrix A by replacing the values in each column with the mean value of the given coordinate in the given column. The denominator term ensures that the value of KG error is unchanged when the data set is uniformly scaled.

Some authors prefer to use the sophisticated Hausdorff distance measure. This is basically a “largest smallest distance,” i.e., the algorithm searches for a point which is most distant from the other surface. Since this point may be located anywhere on the surface (not only at the mesh vertices), the error is usually approximated by sampling, using some advanced speedup techniques like those presented in the implementations Metro [14] and Mesh [15]. Again, the measure does not take into account the

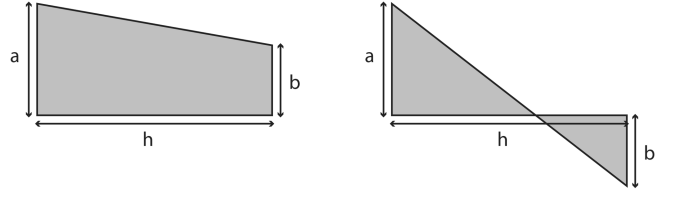


Fig. 1. Straight and twisted trapezoids used in the evaluation of the D_A error.

actual error distribution, and it must be extended to cover the case of dynamic meshes, by using averaging or so-called Root Mean Squared Error (RMSE; used by Mamou [16]), which is a root of squared Hausdorff distances for each frame (this definition of RMSE follows from the Metro tool, and it is often misinterpreted as square root of MSE, which is also sometimes used as an error measure).

Another often used method for dynamic mesh comparison is the D_A error, also known as the ribbon error, which has been introduced in [17] and used by [18], [19]. This measure focuses on the virtual ribbons created by the error vector over time. It works with each coordinate separately, and expresses the error as a sum of areas of the straight or twisted ribbons.

For the non-twisted case, the area of a ribbon can be computed as an area of a trapezoid (see Fig. 1):

$$D(a, b, h) = \frac{1}{2} (a + b) * h. \quad (3)$$

A slightly more complex version is required in the case of a twisted ribbon:

$$D(a, b, h) = \frac{a^2 + b^2}{2(a + b)} * h. \quad (4)$$

The algorithm computes the areas of ribbons formed by the original and distorted positions of each coordinate of each vertex between two subsequent frames, and subsequently averages the ribbon areas over all the vertices and all the frames. For the X coordinate values, we can write:

$$D_X = \sum_{f=1}^{F-1} \sum_{i=1}^V D(|xv_i^f - \bar{xv}_i^f|, |xv_i^{f+1} - \bar{xv}_i^{f+1}|, 1/25). \quad (5)$$

The algorithm is scale-invariant by incorporating a constant W equal to the largest span of coordinate values over the length of the animation:

$$\begin{aligned} M_x &= \max_{i=1..V} \left(\max_{f=1..F} (xv_i^f) \right), m_x = \min_{i=1..V} \left(\min_{f=1..F} (xv_i^f) \right), \\ M_y &= \max_{i=1..V} \left(\max_{f=1..F} (yv_i^f) \right), m_y = \min_{i=1..V} \left(\min_{f=1..F} (yv_i^f) \right), \\ M_z &= \max_{i=1..V} \left(\max_{f=1..F} (zv_i^f) \right), m_z = \min_{i=1..V} \left(\min_{f=1..F} (zv_i^f) \right), \\ W &= \max(M_x - m_x, M_y - m_y, M_z - m_z). \end{aligned} \quad (6)$$

With the constant W , we can express the D_A error as:

$$D_A = \frac{D_X + D_Y + D_Z}{3WF}. \quad (7)$$

This measure does reflect the temporal behavior of the mesh. However, it is intuitively inverse to what one would

expect. If the error oscillates, then the measure yields smaller values than in the case of a constant direction error. Moreover, since the coordinates are processed separately, we get different error values for a rotated version of the model. In other words, if we simply rotated both original and decompressed models, then we would get different error measure values, which is certainly an unexpected behaviour.

A slightly different point of view is taken in the work of Mathur et al. [20]. In their paper, they attempt to obtain a measure of importance of vertices in an animation, which can be used later during compression to locally steer the quantization. Their assumption is that vertices of higher change in curvature are more important, and thus should be quantized more accurately. Their other criterion is based on the knowledge of animation skeleton, which is generally not guaranteed. However, in order to evaluate their results, they use a variation of the PSNR error metric.

To sum up, we see the following serious drawbacks of the presented error measure methods:

1. Some of the methods are not rotation invariant, i.e., if we rotate both original and decompressed meshes, we get different results.
2. None of the methods has been tested to correlate with perceptual quality.
3. None of the methods captures the character of the introduced error, i.e., difference between random and smooth distortion.
4. None of the methods even attempts to capture spatial artifacts, with exception of ribbon measure, which actually favors(!) oscillating vertices.

4 PERFORMED SUBJECTIVE TESTING

We have performed a series of subjective tests in order to approve/disapprove the relation of various measures to subjective evaluation of distortion amount. In order to do so, we have prepared multiple distorted versions of some of the available data sets, using random (1-6), smooth (7-9) and real-world (10-13) distortions:

1. Gaussian noise added to vertex positions (noises with various deviations have been used). Different values were generated for each vertex and each frame.
2. Gaussian noise added to vertex positions. One random distortion vector was generated for each vertex, used in all the frames.
3. Gaussian noise added to vertex positions. One random distortion vector was generated for each frame, used for all the vertices in the frame (i.e., each frame has been shifted by a random amount).
4. Gaussian noise added to vertex positions, random values generated for each vertex and each frame, noise deviation selected for each vertex according to the length of the edges incident with the given vertex (vertices with shorter edges had smaller noise deviation).
5. Gaussian noise added to vertex positions, random values generated for each vertex and each frame, noise deviation selected for each vertex according to the speed of the movement of the given vertex (slow moving vertices had smaller noise deviation).

TABLE 1
Used Distortions

dataset	chicken	dance	cloth	mocap
A	11	2	2	10
B	1	3	10	12
C	11	1	3	13
D	2	10	9	6
E	11	2	1	13
F	5	11	10	12
G	1	11	5	13
H	8	4	9	6
I	7	4	8	11

6. Uniformly distributed random value added to vertex positions. Different values generated for each vertex and each frame.
7. Value $Asin(xv_i^f\omega)$ added to each coordinate of each vertex, i.e., a smooth distortion of amplitude A and frequency ω .
8. Value $Asin(f\omega)$ added to each coordinate of each vertex, i.e., a smooth temporal shifting of the whole mesh.
9. Value $Asin(xv_i^f\omega)$ added to each coordinate of each vertex, i.e., a smooth distortion of amplitude A and frequency ω . The amplitude A has been selected for each vertex according to the speed of the movement of the given vertex (slow moving vertices had smaller distortion amplitude).
10. Result of the compression scheme Coddyc using a coarse quantization.
11. Result of the compression scheme Coddyc using a low number of basis trajectories.
12. Result of the Dynapack [2] compression scheme.
13. Result of the compression scheme by Alexa and Müller [21].

This way, we have achieved distortions of varying nature. We have prepared a set of nine distorted versions of the data sets, namely, chicken (artificial sequence, 3,030 vertices, 400 frames), dance (artificial humanoid sequence, 7,061 vertices, 201 frames), falling cloth (created by a garment simulation in Autodesk 3ds Max, 9,987 vertices, 200 frames) and mocap dance (created by applying mocap data onto a figure in eFrontier Poser 7.0, 14,409 vertices, 263 frames), giving together 40 data sets (including the originals). The details of the distortion types used are listed in Table 1.

In each test, a set of versions of a data set has been shown to a group of subjective evaluators. The evaluation took place in a computer laboratory with a projection screen and 10 computers. One computer at the front of the lab and its associated projection screen was playing the original version of the animation, while each computer played back one of the nine distorted versions. Such approach is equivalent to the multiple-stimulus testing technique with known gold standard which had been previously successfully applied on subjective testing of audio recordings [22].

The displays we have used were identical 19" LCD screens, including the one playing the original sequence. The projection screen has only been used for quick reference, the users could view the original sequence on an identical display at any time.

TABLE 2
Correlations

chicken (43 subjects)											
	A	B	C	D	E	F	G	H	I	Pearson	Spearman
MOS	7.77	9.84	1.63	7.70	2.16	6.30	6.64	0.95	1.79		
Std. deviation	1.73	0.43	1.63	1.88	2.09	2.19	2.01	1.17	1.65		
95% confidence interval	0.52	0.13	0.49	0.56	0.63	0.65	0.60	0.35	0.49		
KG error	2.19	1.86	1.86	1.85	2.95	1.79	0.93	3.95	1.83	-0.53	-0.23
Ribbon mean $\times 10^4$	3.38	2.56	2.19	3.13	3.54	1.67	1.28	7.56	3.45	-0.49	-0.37
Ribbon peak $\times 10^4$	0.45	0.21	0.91	0.16	1.11	0.50	0.10	0.12	0.06	-0.33	0.02
Hausdorff $\times 10^2$	1.04	0.94	1.35	0.96	2.25	0.67	0.48	0.88	0.61	-0.32	0.02
RMS $\times 10^3$	1.70	1.57	1.89	1.56	3.11	0.91	0.85	4.15	2.13	-0.69	-0.63

dance (43 subjects)											
	A	B	C	D	E	F	G	H	I	Pearson	Spearman
MOS	8.93	1.95	3.58	6.74	9.56	1.88	2.02	7.95	6.35		
Std. deviation	1.05	1.73	1.80	1.57	0.96	1.90	1.66	1.68	1.92		
95% confidence interval	0.31	0.52	0.54	0.47	0.29	0.57	0.50	0.50	0.57		
KG error	0.48	6.53	0.23	0.48	0.61	0.49	2.73	0.46	0.28	-0.54	-0.24
Ribbon mean $\times 10^3$	0.53	5.86	0.20	0.43	0.64	0.44	2.54	0.46	0.28	-0.53	-0.02
Ribbon peak $\times 10^2$	0.28	3.12	0.16	0.33	0.61	0.67	5.23	0.38	0.20	-0.60	-0.40
Hausdorff $\times 10^2$	0.28	1.56	0.13	0.28	0.40	0.42	2.90	0.38	0.22	-0.56	-0.36
RMS $\times 10^3$	0.51	8.56	0.24	0.51	0.71	0.71	4.12	0.57	0.35	-0.57	-0.30

falling cloth (37 subjects)											
	A	B	C	D	E	F	G	H	I	Pearson	Spearman
MOS	9.24	7.86	2.22	1.32	2.73	8.78	6.00	2.46	2.32		
Std. deviation	1.28	1.32	1.98	1.23	1.72	1.47	1.47	2.00	1.84		
95% confidence interval	0.41	0.42	0.63	0.39	0.55	0.47	0.47	0.63	0.59		
KG error	0.34	0.26	0.21	0.23	2.02	0.21	0.21	0.65	0.22	-0.27	0.14
Ribbon mean $\times 10^4$	1.89	1.34	1.06	0.88	8.19	1.02	0.61	2.51	1.49	-0.24	0.13
Ribbon peak $\times 10^2$	0.16	0.29	0.06	0.16	3.31	0.09	0.41	1.17	0.02	-0.29	0.20
Hausdorff $\times 10^2$	0.23	0.21	0.04	0.15	2.03	0.11	0.20	0.74	0.04	-0.26	0.36
RMS $\times 10^3$	0.38	0.29	0.26	0.19	2.20	0.20	0.13	0.75	0.27	-0.28	0.20

mocap dance (49 subjects)											
	A	B	C	D	E	F	G	H	I	Pearson	Spearman
MOS	9.64	5.36	3.42	8.60	2.32	8.82	1.74	5.62	1.10		
Std. deviation	0.83	1.70	2.65	1.44	1.96	1.28	1.60	1.34	1.40		
95% confidence interval	0.23	0.47	0.74	0.40	0.54	0.35	0.44	0.37	0.39		
KG error	0.48	0.23	2.05	0.48	1.03	0.46	0.51	0.24	0.55	-0.34	-0.50
Ribbon mean $\times 10^3$	0.55	0.25	2.52	0.55	1.24	0.51	0.60	0.28	0.59	-0.33	-0.49
Ribbon peak $\times 10^2$	0.14	0.07	2.80	0.14	1.57	0.14	0.77	0.07	1.16	-0.62	-0.53
Hausdorff $\times 10^2$	0.20	0.10	1.40	0.20	0.83	0.20	0.39	0.10	0.52	-0.53	-0.53
RMS $\times 10^3$	0.52	0.25	3.05	0.52	1.52	0.46	0.76	0.27	0.81	-0.42	-0.50

The evaluators were briefly introduced into the problem of dynamic mesh compression, and then they have been asked to perform an evaluation in accordance with the following instructions:

- Have a look at all the animations, ideally in their full length. Focus on possible artifacts and differences with respect to the original.
- Find an animation with the worst degradation and assign a mark 10 to it.
- Have a look again at all the animations, and assign marks 0-10 to them according to how acceptable the distortion is. Try to keep the marks proportional, i.e., double the mark value means twice less acceptable distortion.
- Give a mark 0 only to such data set where you cannot find any difference with respect to the original.
- Do not consult the marks with others, or show the nature or location of the problematic parts to each other. Work on your own.

These rules were followed by the evaluators, and thus there was no influencing between the subjects. We have had more than 170 voluntary evaluators—students of the MSc degree in computer science. No evaluator has been engaged more than twice.

The test results are shown in Table 2; we are only giving the mean marks (will be denoted Mean Opinion Score (MOS)), the standard deviation of the marks and the 95 percent confidence intervals. The actual data sets used as versions A-I are published on the Internet (<http://graphics.zcu.cz/compression>) along with details of the introduced distortion. However, such details are irrelevant to the following considerations, because we are trying to reach a measure that does not rely on the knowledge of the error character.

5 ERROR MEASURES EVALUATION

For every distorted animation, we can evaluate any of the error metrics presented so far. Subsequently, we can observe the match of the computed error values, and the

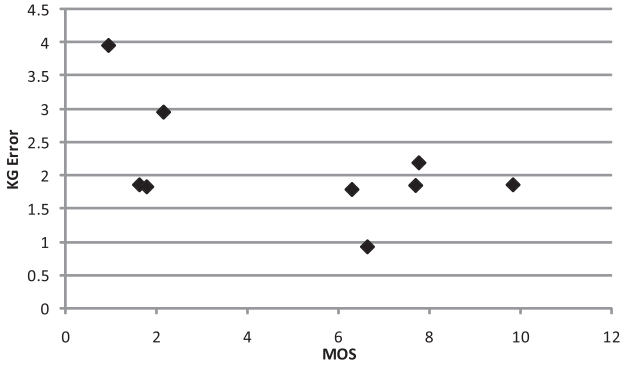


Fig. 2. Example of the relation of the subjective testing results with the KG error measure.

results of subjective testing. Note that we do not want to achieve matching values, because we do not know anything about the magnitude of the error. What we do want to achieve is the correlation between the computed values and the subjective testing results.

As a measure of correlation, we use the Pearson correlation coefficient which is defined for two variables X and Y as:

$$\rho_{X,Y} = \frac{E[(X - E(X))(Y - E(Y))]}{\sigma_X \sigma_Y}, \quad (8)$$

where $E(X)$ is the mean value of the variable X . The coefficient takes values from the interval $\langle -1; 1 \rangle$. If $\rho(X, Y) = 1$, then there is a perfect increasing linear dependence between the variables. The coefficient value of -1 shows a perfect decreasing linear dependence. The zero value shows that there is no linear relation between the values of X and Y .

The correlation coefficient can be estimated from a limited sample of the values by the following equation:

$$\rho_{X,Y} = \frac{\sum_x \sum_y (x - \hat{x})(y - \hat{y})}{\sqrt{\sum_x \sum_y (x - \hat{x})^2 (y - \hat{y})^2}}, \quad (9)$$

where \hat{x} denotes average value. This way, we have evaluated the correlation of existing metrics with the subjective testing. The results are summarized in Table 2. Fig. 2 shows an example of the correlation of the KG error with the results of the subjective testing.

All the existing error measures provide a small negative linear correlation (see Table 2, column "Pearson") with the results of a subjective testing which involves multiple kinds of distortion. In other words, there is no linear correlation present at all. One possible explanation is that the measures do correlate with subjective testing results, but the correlation is not linear. We have, therefore, also evaluated the Spearman [23] correlation coefficient for all the measures. The Spearman coefficient is in fact similar to the Pearson coefficient, only instead of evaluating the actual values, it works with ranks. The result can be interpreted as how well does the measure preserve the order of the results. Should there be, for example, a logarithmic relation between the perceived and the measured error, then the Pearson coefficient may be small, however, the Spearman coefficient

should be large. However, Table 2 shows that not even the Spearman coefficient shows any significant correlation, and it is also negative in a number of cases. Also by looking at the example of the relation shown in Fig. 2, one can see that scaling the measured values will probably not yield any better results.

The results we are presenting were obtained mainly from male evaluators (only four females participated in the presented experiments), however, we have later repeated the experiment with the mocap dance sequence with 30 female subjects. The results of females have correlated with the results of males with a Pearson correlation coefficient of 0.984, and thus, we state that it is quite likely that the results are gender independent.

6 SPATIOTEMPORAL EDGE DIFFERENCE

Motivated by the formal insufficiency of the existing error measures and by the nonexistent correlation between subjective testing results, we propose a new method for measuring discrepancy between two dynamic meshes. Our proposed measure is the spatiotemporal edge difference (STED). The main ideas of the measure are the following:

- We measure the error of some local property independent of its absolute position. We have experimented with the area of triangles, but we have finally chosen edge lengths, because it is not possible that the endpoints of the edge are moved relative to each other, and the property remains unchanged (this can easily happen in the case of triangle area).
- Densely sampled areas of the mesh are likely to contain fine geometric details, and thus are more sensitive to distortion. This fact can be exploited by using relative edge length difference rather than its absolute length.
- Temporal artifacts and distortions can be included in the same framework by considering virtual edges connecting subsequent positions of a vertex.
- We focus on the local changes of the error rather than on the absolute value of it. In order to do so, we express a standard deviation of the edge difference around each vertex, and sum these values to obtain the overall error.

6.1 Spatial Error

Formally, the error can be derived as follows: first, we denote length of an edge connecting vertices v_i and v_j (i.e., of the edge e_{ij}) in frame f as follows:

$$el(e_{ij}, f) = \|v_i^f - v_j^f\|. \quad (10)$$

Note that the el function applies separately on original and distorted meshes. Now, we can define relative edge difference as a property of a spatial edge connecting vertices v_i and v_j (i.e., of the edge e_{ij}) in frame f as follows:

$$ed(e_{ij}, f) = \left\| \frac{el(e_{ij}, f) - \bar{el}(e_{ij}, f)}{el(e_{ij}, f)} \right\|. \quad (11)$$

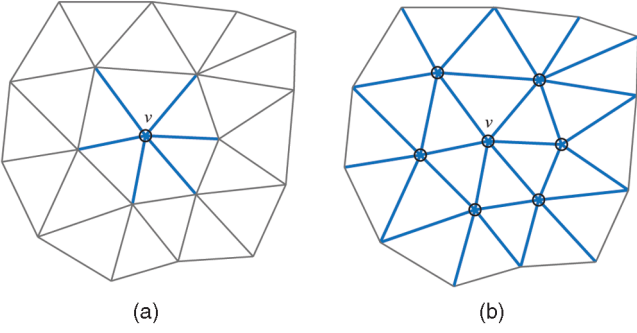


Fig. 3. Neighborhoods of the vertex v . The black circles denote the elements of $NV(v)$, the thick edges denote elements of $NE(v, d)$. The case (a) shows the situation when allowed topological distance d is 0, the case (b) shows the situation when the allowed topological distance d is 1.

Subsequently, for each vertex, we assign the local standard deviation across the edges of given topological distance from the vertex. The user specifies a topological distance d . Subsequently, a set of vertices of topological distance lower or equal to d is found for i th vertex and denoted $NV(i, d)$. Finally, a set $NE(i, d)$ of edges incident with any of the vertices of each $NV(i, d)$ is found (see Fig. 3).

Now, we will compute the average relative edge difference around a vertex. Due to the fact that the surroundings of the vertex may contain edges of very varying length, we compute a weighted average. The weight of an edge is determined by its original length:

$$avged(i, f, d) = \frac{\sum_{e \in NE(i, d)} ed(e, f) el(e, f)}{\sum_{e \in NE(i, d)} el(e, f)}. \quad (12)$$

Now, we can express the local deviation around a vertex v in frame f . Note that we are again using weighting by edge length:

$$dev(i, f, d) = \sigma(ed(NE(i, d), f)) = \sqrt{\frac{\sum_{e \in NE(i, d)} ((ed(e, f) - avged(i, f, d))^2 el(e, f))}{\sum_{e \in NE(i, d)} el(e, f)}}. \quad (13)$$

Finally, we have to average the value over all the vertices and all the frames. Note that the value of (11) has a character of a ratio, i.e., it is scale-independent, and therefore, also the values of (13) and (14) are scale-independent.

$$STED_s(d) = \frac{1}{VF} \sum_{i=1}^V \sum_{f=1}^F dev(i, f, d). \quad (14)$$

6.2 Temporal Error

The reasoning behind the derivation of temporal error is coherent with the derivation of the spatial error, only this time, we will consider virtual edges connecting vertices in subsequent frames. The idea of computing the relative edge length which has been used in the spatial error case to increase sensitivity in high-precision areas will be used again to increase sensitivity in areas of a very slow motion.

This feature is based on the behavior of the falling cloth sequence where some compression methods introduce errors to the static parts of the scene, i.e., the static torus

starts to move slightly. This is very noticeable, and therefore, such artifact should be detected by an error measure.

However, we cannot use the exact equivalent of (11), because in some animations, such as the dance sequence, it is possible that some vertices become almost static for short periods of time (usually the legs of the dancer). However, the time period over which the movement becomes static is too short for human observers to start detecting disturbing artifacts.

In order to evaluate the steadiness of the movement of i th vertex in a frame f , we compute the average speed of the vertex within a temporal window of size w around the frame f . First, we define temporal edge length tel as follows:

$$ld = \max_{i=1..V, j=1..V} (\|v_i^1 - v_j^1\|), \quad (15)$$

$$\begin{aligned} dx(i, f) &= \frac{xv_i^{f+1} - xv_i^f}{ld}, \\ dy(i, f) &= \frac{yv_i^{f+1} - yv_i^f}{ld}, \\ dz(i, f) &= \frac{zv_i^{f+1} - zv_i^f}{ld}, \end{aligned} \quad (16)$$

$$tel(i, f, dt) = \sqrt{dx(i, f)^2 + dy(i, f)^2 + dz(i, f)^2 + dt^2}.$$

Note that this value is not defined in the last frame. The dt (distance in time) term is used to determine the temporal distance between subsequent frames of the animation. The main purpose of dt is to avoid zero length temporal edges (static vertices), which would later cause infinity or near to infinity results.

We can also interpret the dt constant as an actual temporal distance $t_{i+1} - t_i$ between subsequent frames expressed in temporal units (i.e., $1/25$ s in our case) multiplied by an unknown relating constant α , which describes the relation between spatial and temporal units from the point of view of distortion perception, i.e., $dt = \alpha \cdot (t_{i+1} - t_i)$. However, for the simplicity, we will later only try to determine the value of the dt constant.

Also note that the character of (16), i.e., the presence of square root and a constant dt , introduces a need for an early spatial normalization. However, we want to achieve rotation independence, and thus, we cannot use the main diagonal of the first frame bounding box. Instead, we use the distance ld of the two most distant points in the first frame of the original sequence. In this way, we achieve a more robust relative distance which is rotation-invariant. The computation of ld in (15) is not necessarily quadratic and can be speeded up to almost linear (for details, see [24]).

The average spatiotemporal speed s of the i th vertex in the temporal vicinity of width w around the f th frame is defined as:

$$s(i, f, w, dt) = \frac{\sum_{lf=\max(1, f-w)}^{\min(F, f+w)} tel(i, lf, dt)}{\min(F, f+w) - \max(1, f-w)}. \quad (17)$$

Note that due to using the constant dt in (16), this value never becomes zero. Now, we can define the relative temporal edge difference as follows:

$$ted(i, f, w, dt) = \frac{\|tel(i, f, dt) - \overline{tel}(i, f, dt)\|}{s(i, f, w, dt)}. \quad (18)$$

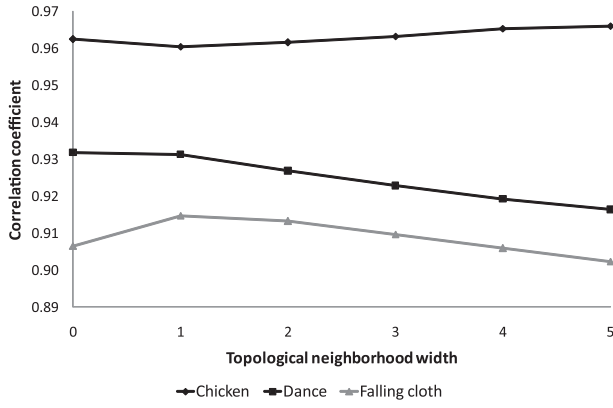


Fig. 4. Dependence of the correlation coefficient on the topological distance d used in (13).

Finally, the overall temporal error is defined as an average over all the vertices and all the frames:

$$STED_t(w, dt) = \frac{1}{V(F-1)} \sum_{i=1}^V \sum_{f=1}^{F-1} ted(i, f, w, dt). \quad (19)$$

6.3 Overall Error and its Parameters

We define the overall error as a hypotenuse of weighted spatial and temporal error:

$$STED(d, w, dt, c) = \sqrt{STED_s(d)^2 + c^2 \cdot STED_t(w, dt)^2}. \quad (20)$$

There are four constants that we have used so far in the definition without discussion of their value. These are:

- topological distance d used to compute the vertex neighbourhood $NV(i, d)$,
- temporal distance value between subsequent frames dt used in (16),
- temporal window size w used in (17),
- the relating constant c used in (20).

The actual values of these constants have been determined using the results of the subjective testing, achieving the best possible correlation of (20) with the subjective testing results.

7 STED PARAMETERS

In this section, we will estimate the values of the parameters for the STED measure. The main objective is to achieve as high correlation with the MOS as possible.

First, we will only consider the spatial part of the STED value, i.e., the result of (14). The equation requires one

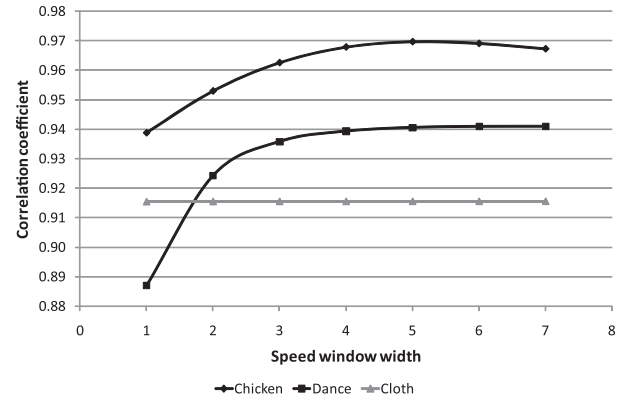


Fig. 5. Dependence of the correlation coefficient on the vertex speed window w used in (17).

parameter, the width of the topological neighborhood d over which the deviation (13) is computed. For the experiment, we have only considered the spatial error, achieving the best results with topological distance 1 for the neighboring vertices, i.e., the edge neighborhood of the shape equivalent to Fig. 3b. Fig. 4 shows the measured dependence of the correlation coefficient with the topological neighborhood width, Table 3 gives the measured values for neighborhood of distance 1.

Now, we will set the parameter values for the temporal error. We have to set three parameters: the window size w for the computation of the vertex speed in (17), the temporal frame distance parameter dt used in (16), and the weight c relating the spatial and the temporal error in (20).

Through a series of experiments, we have found values for these constants so that the overall error expressed by (20) correlates with the subjective testing results as much as possible. We are presenting graphs that describe the behavior of the correlations around the values we are using. Figs. 5, 6 and 7 show the development of the correlation coefficient around the selected values. The figures demonstrate that the computation is not critically dependent on the exact values of these constants.

The best results have been obtained for the speed window of width $w = 5$ and the temporal distance coefficient $dt = 0.0003$. The weighting coefficient c is easiest to optimize, and the highest correlation has been obtained for the value $c = 9.144 \times 10^{-5}$. Using these constants, we can evaluate the error values shown in Table 4 along with the correlations with the MOS.

Note that we have only used the first three data sets to determine the parameters. The mocap dance data set has been excluded from the optimization and serves as a check that the algorithm works in cases for which it has not been

TABLE 3
Result of the Spatial Term of the STED Error

dataset	A	B	C	D	E	F	G	H	I	pearson	spearman
chicken ($\times 10^3$)	4.48	5.06	0.64	5.02	0.91	3.00	2.52	0.00	1.24	0.960	0.950
dance ($\times 10^3$)	1.06	0.00	0.50	1.06	1.16	0.13	0.41	0.92	0.57	0.931	0.946
falling cloth ($\times 10^3$)	0.58	0.25	0.00	0.03	0.17	0.38	0.22	0.08	0.00	0.915	0.959
mocap dance ($\times 10^3$)	1.62	0.68	0.17	1.62	0.11	1.32	0.07	0.78	0.14	0.976	0.916

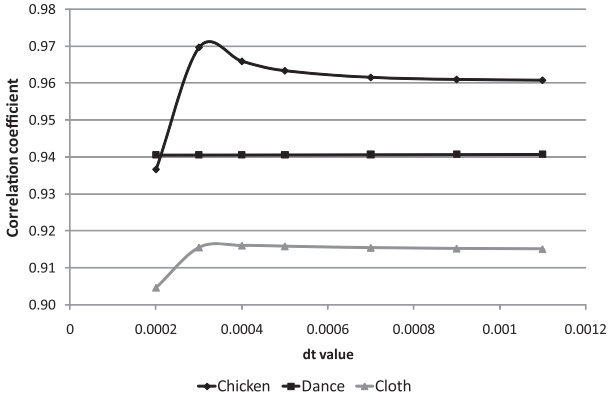


Fig. 6. Dependence of the correlation coefficient on the temporal frame distance dt used in (16).

optimized. The achieved correlation of 0.976 asserts that the algorithm yields high correlation even for data not used for optimization. Note that there are distortion types only used for the mocap dance data set, which shows that the algorithm also works for some distortion types not used for optimization.

8 PRACTICAL APPLICATION

In order to evaluate the practical importance of the proposed measure, we have performed a series of tests using a different data set, the human jump sequence (scanned dynamic mesh, 15,830 vertices, 222 frames, for details see [25], [26]). We have experimented with the Coddyc compression scheme [4], using different configurations of the compressor. Generally, there are two parameters that influence the performance of the compressor: the number of basis trajectories, and the quantization constant. Both parameters can be used to steer the rate/distortions in a similar way. By increasing the parameter value the error should decrease, but the data rate should increase. We have performed tests with the number of basis vectors from 5 to 40, and quantization constants from 7 to 13. The results using the KG error measure is shown in Fig. 8, the STED results are shown in Fig. 9.

The first observation is that the used measures provide significantly different results. The basic development of the curves remains unchanged, i.e., finer quantization leads to lower error in both cases. However, the similarity fails when different numbers of basis trajectories are used. In the KG error measure curves, the error decreases with the number of basis trajectories, which seems to be intuitive. In the STED measures, this relation is more complex: for fine quantization the relation holds, but for coarse quantization the relation is reversed, i.e., more basis vectors introduce

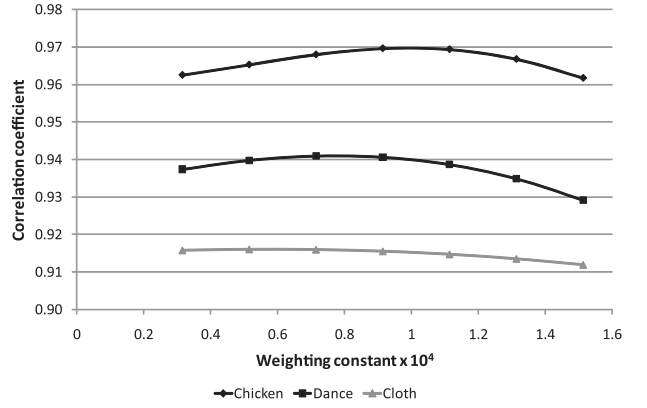


Fig. 7. Dependence of the correlation coefficient on the weighting constant c used in (20).

more error. This can be explained by the fact that the STED mainly focuses on the local error which exhibits an additive behavior, and therefore, increases when more basis vectors with error are added together. In other words, adding a low-importance basis vector will bring the vertices closer to their original positions, however, it will increase the local deviation of the error vectors (neighboring vertices will be distorted in a different way).

The second observation is that the selection of compression parameters (number of basis vectors and quantization constant) is strongly dependent on the measure used. The KG measure prefers higher number of basis vectors, while the quantization may remain coarse. On the other hand, the STED measure usually drops significantly with finer quantization, while the effect of adding basis vectors is less significant.

We have prepared a series of distorted versions of the human jump sequence, for which the STED and the KG measures are inconsistent in decision about which one is more acceptable. We have performed a blind subjective test to determine whether the actual observations match the results of the STED or the KG error.

Table 5 summarizes the findings from 21 subjective evaluators. The column headers give the compression parameters used in the format $\langle \text{number of Basis vectors} \rangle - \langle \text{Quantization constant} \rangle$. We have intentionally selected such parameters so that we could test hypotheses following from Figs. 8 and 9. The findings are:

1. According to Fig. 9, the settings 10-8 should provide result with smaller error than settings 40-8. Table 5 confirms this with values 9.62 for 40-8 and 7.00 for 10-8. This result is very unexpected and surprising, because it shows that adding more basis vectors leads to an increase of the perceived error. This

TABLE 4
Result of the STED Error Measure

dataset	A	B	C	D	E	F	G	H	I	pearson	spearman
chicken ($\times 10^3$)	4.48	5.57	0.64	5.02	0.91	3.73	2.58	0.13	1.24	0.970	0.950
dance ($\times 10^3$)	1.06	0.24	0.50	1.06	1.16	0.13	0.41	0.92	0.57	0.941	0.962
falling cloth ($\times 10^3$)	0.58	0.25	0.06	0.03	0.17	0.39	0.22	0.08	0.00	0.915	0.950
mocap dance ($\times 10^3$)	1.62	0.68	0.17	1.62	0.11	1.32	0.07	0.78	0.14	0.976	0.916

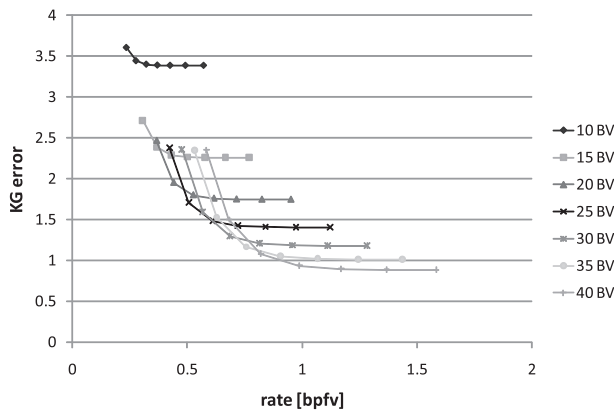


Fig. 8. Rate/distortion curves for the jump sequence using the Coddyc algorithm, using the KG error measure. Each curve represents a constant number of basis vectors, and shows the dependency on the quantization constant, changing from 7 bits/main diagonal (leftmost, coarsest quantization) to 13 bits/main diagonal (rightmost, finest quantization).

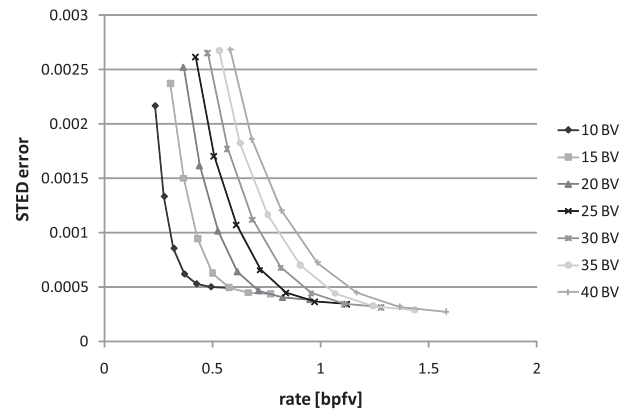


Fig. 9. Rate/distortion curves for the jump sequence using the Coddyc algorithm, using the STED error measure. Each curve represents a constant number of basis vectors, and shows the dependency on the quantization constant, changing from 7 bits/main diagonal (leftmost, coarsest quantization) to 13 bits/main diagonal (rightmost, finest quantization).

phenomenon is not detected by any existing error measure.

2. At data rate approximately 0.75 bpfv, according to Fig. 8, the optimal configuration of the coder is 35-9 while according to Fig. 9, it should be 15-13. The test showed that result of configuration 15-13 achieved the MOS 1.95, while the configuration 35-9 performed significantly worse, reaching 6.57.
3. Similarly, at bit rate approximately 0.5 bpfv, according to Fig. 8, the optimal configuration of the coder is 25-8 while according to Fig. 9, it should be 10-12. The test showed that result of configuration 10-12 achieved MOS 1.23 while the configuration 25-8 performed significantly worse, reaching 9.23.
4. According to Fig. 8, no significant improvement is achieved when improving quantization from configuration 20-9 to 20-13, which contradicts Fig. 9, where such change leads to drop of error to about one half. The subjective results confirm the second conclusion by MOS 6.00 for configuration 20-9 versus 1.52 for configuration 20-13.
5. Similarly, according to Fig. 9, no significant improvement is achieved when adding base trajectories from configuration 15-13 to 30-13, which contradicts Fig. 8, which shows that such change leads to drop of error to about one half. The subjective results confirm the first conclusion by MOS 1.95 for configuration 15-13 versus 1.62 for configuration 30-13.

All the tested assumptions following from the STED measures have been confirmed by the experiment. Some of the examples can be seen in Fig. 2. There is, however, one unsolved contradiction following from the test. The

configuration 15-13 has reached the MOS 1.95, which is worse than the result of configuration 10-12, which has scored 1.23. This relation should be the other way round, because 15-13 has both finer quantization and more basis vectors. We believe that this is caused by the fact that even at such low rates the visual error is so small that the observers could not distinguish between the two versions, and the difference is caused by random factors. This explanation is supported by the standard deviation values which are of similar magnitude as the MOS values themselves.

The tests 2-5 show that used error measure has an important impact on dynamic mesh algorithm configuration. When configuring an algorithm, one has to decide on **what is the property of the data we wish to preserve**, and design the property measure accordingly. From this point of view, scaling the measure values has no effect on the algorithm configuration, i.e., a configuration which is optimal in KG error optimization will be also optimal in the MSE error optimization, because the KG error is no more than a scaled version of the MSE. STED error, on the other hand, is **fundamentally different** and has an important impact on the choice of the optimal configuration.

9 CONCLUSIONS

We have proposed a novel method for measuring the discrepancy between two dynamic meshes. Our contributions are:

- We have shown that there is only a **weak relation** between the MSE and the perceived similarity of dynamic meshes, and as such is of limited use for

TABLE 5
Subjective Experiment Results for the Human Jump Sequence

configuration	20-13	25-8	40-8	20-9	10-8	30-13	15-13	35-9	10-12
MOS	1.52	9.24	9.62	6.00	7.00	1.62	1.95	6.57	1.24
Std. deviation	1.17	0.89	0.59	1.58	1.73	1.53	1.53	1.80	1.30

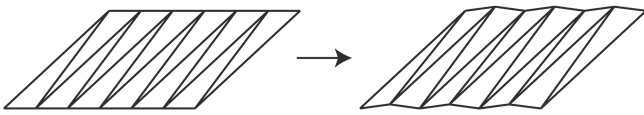


Fig. 10. The “newspaper” artifact, not detected by the STED measure.

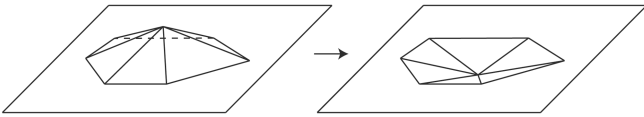


Fig. 11. The “hill/valley” artifact, not detected by the STED measure.

real tasks, such as configuration of a compression algorithm by a rate/distortion optimization.

- We have determined the form of equations which yield results with **high correlation** with subjective testing.
- We have determined constants to be used in the equations. These constants may not be exact, however, the result does not critically depend on their value.

Using the proposed STED measure, we are able to robustly determine an error introduced by a compression. We can detect certain kinds of spatial and temporal artifacts, namely the difference between regular and random vertex shifts. The measure we are proposing yields results significantly different from the ones provided by existing measures, which has a substantial influence on dynamic mesh compression algorithm configuration, or even dynamic mesh compression algorithm design.

Quite surprisingly, our experiments show that the influence of temporal error is relatively small. The contribution of the temporal error term in (20) is about 12 percent for the chicken sequence, and less than 2 percent for the dance and cloth sequences. Nevertheless, the inclusion of the temporal term improves the correlation and may be important in cases where very little spatial error is introduced into the animation.

The method has some drawbacks as well. In some cases, such as rigid translation, rotation, and scale of the animation, it is possible that the measure does not report any difference. We consider such eventuality highly unlikely and easy to detect and compensate. It is possible that in some cases of extremely smooth deformation, such as global taper or shear of the animation, the measured error might be smaller than expected.

We have also identified a class of artifacts which will not be detected by the measure at all. One can easily imagine that the deformation depicted by Fig. 10 is well visible, while all the edge lengths remain unchanged. This is not only a problem of meshes with borders and with one edge around which the mesh is rotated. Situation depicted by Fig. 11 is also not detected by the measure. Such case may occur in practise, however, a larger deformation that leaves edge lengths unchanged is again very unlikely.

Despite the drawbacks, we have received measures that correlate with the subjective testing with Pearson correlation coefficient constantly higher than 0.9 during the testing. In the case of the mocap dance sequence, we have reached to a value of 0.976. In our experiments, such high correlations are not achieved by any existing error measure

technique. Moreover, the STED measure is fast to evaluate, especially compared to the Hausdorff distance based measures such as the one used by the Metro tool.

10 FUTURE WORK

In the future, we intend to perform a study of existing compression algorithms and their performance with respect to the proposed STED measure. We have shown that moving from the KG error to the STED error requires reconfiguration of the Coddycoder in order to obtain the best possible result. Similarly, other compression algorithms have multiple configuration parameters (quantization constant, level of simplification, number of clusters, depth of used space subdivision structure, etc.) which need to be optimized to obtain an optimal result with respect to a given error measure. It is therefore not easy to perform a fair comparison, and thus, we have not attempted to do such study in this paper.

Currently, the measure only works for compression evaluation, i.e., for cases when original and distorted sequences share the connectivity. There is a possibility to extend the measure also to the case of mesh sequences of unequal connectivity. This could be done by fine resampling of both meshes, which will convert them into a shared topology, using some strain minimization criterion. There are some techniques used for this purpose, such as the Face2Face algorithm [27], [28], the wavelet decomposition [29], or remeshing [30]. Extending the measure in this way is a part of the intended future work.

We also intend to use the spatial part of the STED measure to test the possibility of measuring the difference between static meshes in order to compare results of static mesh compression algorithms.

Detailed information about our research in the field of dynamic mesh compression can be found on our website <http://graphics.zcu.cz/compression>.

ACKNOWLEDGEMENTS

The authors would like to thank all the volunteers who have helped them with the subjective testing. Also the critical comments from their colleagues as well as the anonymous reviewers have been very valuable for them. This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics). The chicken character was created by Andrew Glassner, Tom McClure, Scott Benza, and Mark Van Langeveld. This short sequence of connectivity and vertex position data is distributed solely for the purpose of comparison of geometry compression techniques.

REFERENCES

- [1] J.E. Lengyel, “Compression of Time-Dependent Geometry,” *Proc. Symp. Interactive 3D Graphics (SI3D '99)*, pp. 89-95, 1999.
- [2] L. Ibarria and J. Rossignac, “Dynapack: Space-Time Compression of the 3D Animations of Triangle Meshes with Fixed Connectivity,” *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '03)*, pp. 126-135, 2003.
- [3] N. Stefanoski, X. Liu, P. Klie, and J. Ostermann, “Scalable Linear Predictive Coding of Time-Consistent 3D Mesh,” *Proc. True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON)*, <http://www.3dtk-con.org>, May 2007.

- [4] L. Váša and V. Skala, "Coddyc: Connectivity Driven Dynamic Mesh Compression," *Proc. True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON)*, May 2007.
- [5] L. Váša and V. Skala, "Combined Compression and Simplification of Dynamic 3D Meshes," *Computer Animation and Virtual Worlds*, vol. 20, no. 4, pp. 447-456, 2008.
- [6] L. Váša and V. Skala, "COBRA: Compression of Basis of PCA Represented Animations," *Computer Graphics Forum*, vol. 28, no. 6, pp. 1529-1540, 2009.
- [7] *MPEG4 Part 16 AMD2: Frame-Based Animated Mesh Compression*, ISO/IEC JTC1/SC29/WG11, 2007.
- [8] R. Amjoun, "Efficient Compression of 3D Dynamic Mesh Sequences," *J. WSCG*, vol. 15, pp. 99-106, Feb. 2007.
- [9] H.M. Briceno, P.V. Sander, L. McMillan, S. Gortler, and H. Hoppe, "Geometry Videos: A New Representation for 3D Animations," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '03)*, pp. 136-146, 2003.
- [10] J. Zhang and C.B. Owen, "Octree-Based Animated Geometry Compression," *Proc. Conf. Data Compression (DCC '04)*, p. 508, 2004.
- [11] Z. Karni and C. Gotsman, "Compression of Soft-Body Animation Sequences," *Computers and Graphics*, vol. 28, no. 1, pp. 25-34, <http://citeseer.ist.psu.edu/karni04compression.html>, 2004.
- [12] K. Mamou, T. Zaharia, and F. Preteux, "A Skinning Approach for Dynamic 3D Mesh Compression: Research Articles," *Computer Animation and Virtual Worlds*, vol. 17, nos. 3-4, pp. 337-346, 2006.
- [13] F. Payan and M. Antonini, "Wavelet-Based Compression of 3D Mesh Sequences," *Proc. IEEE Int'l Conf. Machine Intelligence (ACIDCA ICMI '05)*, Nov. 2005.
- [14] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167-174, <http://citeseer.comp.nus.edu.sg/cignoni98metro.html>, 1998.
- [15] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: Measuring Errors Between Surfaces Using the Hausdorff Distance," *Proc. IEEE Int'l Conf. Multimedia and Expo*, vol. 1, pp. 705-708, http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1035879, 2002.
- [16] K. Mamou, T. Zaharia, and F. Preteux, "Multi-Chart Geometry Video: A Compact Representation For 3D Animations," *Proc. Int'l Symp. 3D Data Processing Visualization and Transmission*, pp. 711-718, 2006.
- [17] E.S. Jang, J.D.K. Kim, S.Y. Jung, M. Han, and S.O. Woo, "Interpolator Data Compression for MPEG-4 Animation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 7, pp. 989-1008, July 2004.
- [18] K. Müller, A. Smolic, M. Kautzner, and T. Wiegand, "Rate-Distortion-Optimized Predictive Compression of Dynamic 3D Mesh Sequences," *Signal Processing: Image Comm.*, vol. 21, no. 9, pp. 812-828, Oct. 2006.
- [19] M. Sattler, R. Sarlette, and R. Klein, "Simple and Efficient Compression of Animation Sequences," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation (SCA '05)*, pp. 209-217, 2005.
- [20] P. Mathur, C. Upadhyay, P. Chaudhuri, and P. Kalra, "A Measure for Mesh Compression of Time-Variant Geometry," *Computer Animation and Virtual Worlds*, vol. 15, nos. 3-4, pp. 289-296, 2004.
- [21] M. Alexa and W. Müller, "Representing Animations by Principal Components," *Computer Graphics Forum*, vol. 19, no. 3, pp. 411-418, <http://citeseer.ist.psu.edu/alexa00representing.html>, 2000.
- [22] A. Mason, "Mushra (Multi Stimulus Test with Hidden Reference and Anchor)," ITU-R BS.1534, BBC R&D White Paper WHP 038, 2002.
- [23] C. Spearman, "The Proof and Measurement of Association Between Two Things," *Am. J. Psychology*, vol. 15, pp. 72-101, 1904.
- [24] V. Skala and V. Ondračka, "Fast Maximum Distance of Points in E2 & E3," technical report, Univ. of West Bohemia, 2008.
- [25] P. Sand, L. McMillan, and J. Popović, "Continuous Capture of Skin Deformation," *Proc. ACM SIGGRAPH '03 Papers*, pp. 578-586, 2003.
- [26] N. Anuar and I. Guskov, "Extracting Animated Meshes with Adaptive Motion Estimation," *Proc. Conf. Vision, Modeling, and Visualization*, pp. 63-71, 2004.
- [27] M.M. Bronstein, A.M. Bronstein, A.M. Bruckstein, and R. Kimmel, "Matching Two-Dimensional Articulated Shapes Using Generalized Multidimensional Scaling," *Proc. Conf. Articulated Motion and Deformable Objects (AMDO)*, pp. 48-57, 2006.
- [28] M.M. Bronstein, A.M. Bronstein, and R. Kimmel, "Face2face: An Isometric Model for Facial Animation," *Proc. Conf. Articulated Motion and Deformable Objects (AMDO)*, pp. 38-47, 2006.
- [29] J.-H. Yang, C.-S. Kim, and S.-U. Lee, "Semi-Regular Representation and Progressive Compression of 3D Dynamic Mesh Sequences," *IEEE Trans. Image Processing*, vol. 15, no. 9, pp. 2531-2544, Sept. 2006.
- [30] P. Klie, E. Okon, N. Stefanoski, and J. Ostermann, "A Framework for Scene-Flow Driven Creation of Time Consistent Dynamic 3D Objects Using Mesh Parametrizations," *Proc. True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON)*, <http://www.tnt.uni-hannover.de/staff/klie/papers/Klie3DTV-CON07Framework.pdf>, May 2007.



compression algorithm design.

Libor Váša received the PhD degree at the University of West Bohemia in 2008 with thesis "Methods for size reduction of dynamic meshes." Since then, he is working at the Centre of Computer Graphics and Data Visualization in the field of dynamic mesh compression. Since 2007, he is working as a full time researcher for the Center for Computer Graphics (LC-CPG). Currently, he is mainly focused on methodologies for distortion evaluation and their impact on


Graphics, Visualization and Computer Vision (<http://wscg.zcu.cz>). He is a member of the IEEE.

Václav Skala is a full professor of computer science at the University of West Bohemia in Plzen and the head of the Centre of Computer Graphics and Visualization (<http://graphics.zcu.cz>). He is a member of the Eurographics (since 1985). He is a member of the editorial boards of *Computers and Graphics* (Elsevier) and *The Visual Computer* (Springer Verlag). He is a founder of the WSCG International Conferences in Central Europe on Computer

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.