

LEVELS-OF-DETAIL GENERATION METHOD FOR SKELETAL MESHES

Abstract. This paper is devoted to the development of an algorithm for Levels-Of-Detail generation from skinned meshes. Animated meshes, unlike static ones, cannot be simplified without redistributing or recalculation bone weights. In some cases, objects of rendered scene have redundant details. It happens when their size on a screen, the distance from a virtual camera and other factors are such that there is no sense to display these objects in their full complexity, as it may significantly impact time for rendering one frame. One of the solutions is to create a set of Levels-Of-Detail for each object – a set of meshes and/or texture which represent same object, but with lower level of detail – and change the original object with them, when it is necessary. The simplification of visual models is especially important for visualisation of digital twins of real-world objects, subjects, or processes within the digital twin technology. An analysis of existing algorithms for Levels-Of-Detail generation for animated meshes is presented and discussed. An improved method for Levels-Of-Detail generation is introduced and discussed. The proposed method is based on Houle and Poulin animated mesh simplification. However, there are the following core differences in the proposed method: weights of resulting vertices are interpolated, not just copied; multiple poses are used for simplification input. These new features allow to achieve the animated meshes simplification without significant drawbacks in animation quality and mesh optimization.

Keywords: skeletal animation, graphical modelling, mesh simplification.

Problem Statement. Animation is an important part of visual modeling in many application areas where objects to be visualised have dynamically changing shape and/or position. One of such application areas concerns implementation of a digital twin technology [1]. A digital twin is a complex model of a physical twin which is a real-world object, subject, or process. The digital twin as a complex model includes two essential parts, namely, a visual model and a behavioural model [2]. The digital twin visual model requires animation because the dynamicity is an obvious feature of any digital twin. The process of digital twin visualization is based on the real-time rendering. However, sometimes objects of rendered scene have redundant details. It

happens when their size on a screen, the distance from a virtual camera and other factors are such that there is no sense to display these objects in their full complexity, as it may significantly impact time for rendering one frame. One of the solutions is to create a set of Levels-Of-Detail (LODs) [3-10] for each object – a set of meshes and/or texture which represent same object, but with lower level of detail – and change the original object with them, when it is necessary. The creation of LOD meshes is a resource-intensive process which requires a work of a professional artist. For this reason, the topical task is the creation of methods for mesh simplification. The method proposed in this paper is derived from the method of Haule and Pauline [10], but with improvements such as recalculating weights of resulting vertices and taking into account multiple poses of the original mesh. This method can be employed not only as a part of digital twin technology but for any other applications where skeletal animation is required.

Analysis of Recent Researches and Publications. Most methods for articulated meshes simplification [3-8] work with edge collapse that makes it easier to calculate weights of the resulting mesh. Edge collapse is a method of simplifying a mesh when an edge of the mesh is contracted into one of its points. As a result, two triangles and three edges are removed from the mesh.

Among all methods [3-13] analysed in this research several methods are the most interesting. Thus, a method offered by Huebner [9] is based on collapsing each edge only to one of the extreme points – vertices which make the edge. This makes it trivial to calculate other attributes of resulting mesh and allows all vertices of the mesh to reside of memory, and just use a fraction of them to display polygons. Resulting technique offers a multi-resolution representation of mesh manageable in real-time. Fuhrmann and Schmalstieg [10] propose to decompose a mesh into regions, where vertices of each region correspond to a common set of bones. Edges are collapsed only between vertices of the same region; adjacent regions only affect each other when vertices on the edges of a region are collapsed. Both methods [9, 10] effectively use only a subset of vertices of the original mesh. The advantage of such approach is that all the vertices of the resulting mesh are vertices of the original mesh meaning all their attributes stay valid. This approach may decrease the quality

of resulting mesh, since it forbids optimization of vertex position during the LOD generation. Also, they do not benefit from such software optimizations as face encodings.

The method proposed by Houle and Poulin [11] fixes problems of previously mentioned methods [9, 10]. In order to build simplified mesh, the mesh is firstly simplified as if it was static, using any edge collapse technique with progressive meshes. Progressive meshes are the bases of the LOD meshes display method [12, 13] which uses a special data structure – progressive mesh – to choose and display the best possible LOD mesh for a current mesh. The general idea of the progressive mesh can be described in the following way: the progressive mesh has two operations which can be applied to it – *ecol* (edge collapse) and *vsplit* (vertex split). *Ecol* collapses an edge with the use of predetermined algorithm. *Vsplit* is converse to *ecol*, it returns a structure to the state before previous *ecol* (Fig. 1).

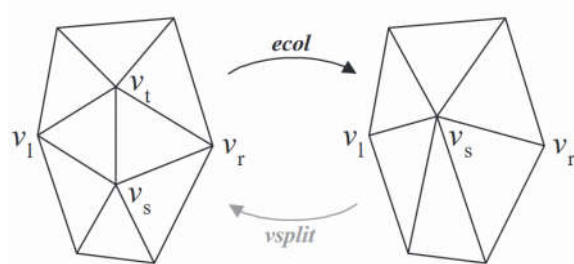


Figure 1 - The illustration of *ecol* and *vsplit* operations [12, 13]

In the method proposed by Houle and Poulin [11] an articulated mesh first is simplified as a static one, in its binding pose, constructing a progressive mesh. All *vsplit* operations from this progressive mesh are copied into the articulated mesh and are interpreted with respect to the mesh skeleton. Since a vertex can move after an edge collapse, its position is backpropagated to other vertices through *vsplit* operations. Weights of the resulting vertices are not changed: the initial values are carried through the simplification process. The result of this method is an articulated mesh which has all attributes preserved and continuous LOD representation. This approach does not address reassigning new weights and only uses one pose of the original mesh to construct an LOD mesh. It may result in visual artefacts during animation of the mesh.

Thus, the analysis of existing methods related to the task of LOD generations allows us to conclude that the development of new methods of animated mesh simplification is a topical task.

Formulation of Research Objective. The objective of this research is to develop a method of animated meshes simplification without significant drawbacks in animation quality and mesh optimization.

Presentation of the Main Research Material. The proposed method is based on Houle and Poulin animated mesh simplification [11]. However, there are the following core differences:

Weights of resulting vertices are interpolated, not just copied;

Multiple poses are used for simplification input.

To start simplification, an articulated mesh in its binding pose and several its copies in different poses, preferably, extreme poses for all bones, and several poses in-between are required. For each of those poses, we need to choose the best edges. It can be achieved by using a memoryless implementation of the quadric error metrics [14-16]. The threshold parameter for error selection t is set to zero, so no non-edge decimation situations occur. This is needed to build a progressive mesh. We choose an edge and a new vertex position with the least error. This edge is collapsed on this pose, and the change is propagated to all other poses. Essentially, this divides the resulting errors between different animations.

For edge contraction, we use the quadric error metric proposed by Garland in [18]. An error at the vertex $v = [v_x v_y v_z 1]^T$ is defined as the quadratic form $\Delta(v) = v^T Q v$. For contraction $v_1 v_2$, the matrix $Q' = Q_1 + Q_2$ is used. The position of v' can be found by minimizing the value of $\Delta v'$ which can be done by solving the equation:

$$v' = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (1)$$

To calculate a bone weight of the resulting vertex we use a weighted sum. Let us suppose that v_1 and v_2 are vertices on the edge, v' is a vertex which is resulted in the collapse of this edge, w_{i1} are weights of v_1 of bone i ,

w_{2i} are weights of v_2 of bone i , w'_i is a weight of resulting vertex v' of bone i . Then, the weights of this vertex can be calculated as follows:

$$w'_i = w_{1i} \|v_1 v'\| / \|v_1 v_2\| + w_{2i} \|v_2 v'\| / \|v_1 v_2\|, \quad (2)$$

for each bone i , where $\|v_1 v_2\|$ is a distance between points v_1 and v_2 .

It is important to do this for each bone, since the sum of the weight for each vertex should always be equal to one. If we wish to remove weights which do not have any significant impact on the vertex, we need to normalize the resulted vector of weights. After this, the process is repeated, until the desired mesh complexity is achieved. The number of iterations for this is trivial to calculate, as each iteration destroys two triangles and three edges.

The resulting method can be described as follows:

1. An animated mesh is set to its binding pose, extreme bone positions, and multiple poses in-between.

2. For each vertex matrix Q is calculated.

3. An optimal point v' for each edge is calculated.

4. An edge with the minimal error on all poses is chosen.

5. This edge is collapsed on a mesh with lowest error; this change is applied on all other poses.

6. A bone weigh of the resulting vertex is calculated as a weighted sum of the parent vertices where the weight is a ratio of the distance of its parent vertex to the distance between them.

7. The cost of changed pairs is recalculated.

8. The process is repeated until a mesh of the desired complexity is achieved.

In the articulated meshes obtained as a result of this method, an overall shape and detail are preserved during an animation and small details, like individual fingers, start to lose their shape only on very low triangle counts. It can be noticed in Fig. 2.



Figure 2 - The result of applying the proposed method to an animated mesh; the triangle counts from left to right: 5546, 4700, 2800, 1500, 550

To evaluate the difference between both the original and the simplified mesh, the error metrics described by Garland [18] and modified further was used:

$$E_i = \frac{1}{(|X_i| + |X_n|)B} \left(\sum_{v \in X_n} d^2(v, M_i) + \sum_{v \in X_i} d^2(v, M_n) \right), \quad (3)$$

where X_n and X_i are sets of points sampled on M_n (the original mesh) and M_i (the simplified mesh); v is a point of the set; $d(v, M)$ is a distance from the point v to the closest point on the mesh M ; B is a length of the mesh bounding box diagonal.

The method can be implemented as it is shown on Algorithm 1.

Algorithm 1. Simplification of articulated mesh

<p>Input: The articulated mesh M_n with a set of vertices \bar{v}, a set of weights \bar{w}, a set of triangles \bar{t}; The desired number of triangles on the simplified mesh c which is smaller than the size of \bar{t}.</p>
<p>Output: An articulated mesh M_n with the desired triangle count.</p>
<p>While size of $\bar{t} > c$, do: For each $v_i \in \bar{v}$: $\Delta i = v^T Q v$ For each v_i, v_j, on same edge: $Q' = Q_i + Q_j$</p>

$$v'_{ij} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

For $\min(\Delta')$ from contraction v_i, v_m :
 Remove v_i, v_m from \bar{v}
 Add v' to \bar{v}
 $w'_i = w_{ii} \|v_i v'\| / \|v_i v_m\| + w_{mi} \|v_m v'\| / \|v_i v_m\|$

The bounding box was added to normalize error results to a size of the mesh. The experiments were carried out by using a computer with Intel® Core™ i7-4810MQ processor clocked at 2.8 GHz. The proposed method has been applied to articulated meshes. As a result of the experiments, it was found that the time elapsed for simplification of this mesh from 5546 triangles to 310 triangles is 300 ms on single thread. The resulting error is shown in Fig. 3. The obtained results confirm the quality of the method.

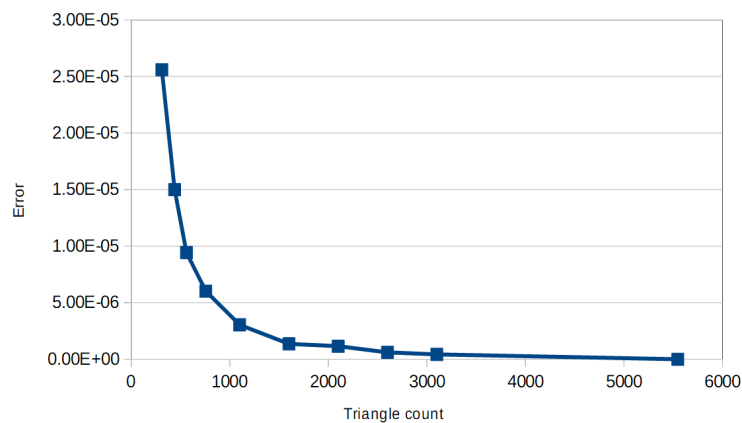


Figure 3 - Error of mesh simplified by using the proposed method

Thus, the developed method can be used for applications where the simplicity of 3D models is important for ensuring real-time rendering. In digital twin technology it can be used for preliminarily view of a physical twin model to be resulted in a digital twin visual model. In this case, the procedure of a digital twin visual model creation includes the following main stages:

1. Appearance data measuring/estimation;
2. Template model individualisation;
3. LODs generation;

4. Template model previewing;
5. Physical and digital twins matching.

The use of LODs generation in the procedure of the digital twin visual model creation enables decreasing the workload on computational resources. It can be especially important in cases when a physical twin has a complex shape and requires simplification while its previewing that needs to be carried out without visualisation artefacts appearance.

Conclusions. The proposed method of skeletal meshes simplification can be applied to any animated mesh and enables mitigating the problems which other existing methods suffer from. It is achieved by recalculating weights of resulting meshes and splitting the error between different animations. The method presented in this paper can be used in many application areas including digital twin technology applications, especially in cases when computation resources are limited.

Since the method relies on skinning, the further research should be focused on studying both interactions with skinning methods, including dual quaternion skinning, and optimal choice of input poses number.

ЛИТЕРАТУРА / ЛІТЕРАТУРА

1. Madni, A. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems* / Azad M. Madni, Carla C. Madni, Scott D. Lucero. – 2019. – No 7, Issue 7. – pp. 1–13. – doi: 10.3390/systems7010007
2. Dychka I. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing / I. Dychka, Ye. Sulema, Iu. Bukhtiiarov // *Visnyk of Kherson National Technical University*. – 2019. – No 3 (70). – p. 112–119. – doi: 10.35546/kntu2078-4481.2019.3.12
3. Cignoni, P. A comparison of mesh simplification algorithms / P. Cignoni, C. Montani, R. Scopigno // *Computers & Graphics*. – 1998. – Vol. 22(1). – pp. 37–54. – doi: 10.1016/S0097-8493(97)00082-4
4. Cheng X. Detail Resolution: A New Model to Describe Level of Detail Information of Vector Line Data / Xiaoqiang Cheng, Huayi Wu, Tinghua Ai, Min Yang // *Spatial Data Handling in Big Data Era*. – Springer. – 2017. – pp. 167–177. – doi: 10.1007/978-981-10-4424-3_12

5. Erikson, C. Polygonal simplification: An overview / C. Erikson // UNC Chapel Hill Computer Science Technical Report TR96-016. – 1996.
6. Akenine-Möller T. Texture Level of Detail Strategies for Real-Time Ray Tracing / Tomas Akenine-Möller, Jim Nilsson, Magnus Andersson, Colin Barré-Brisebois, Robert Toth, Tero Karras // Ray Tracing Gems. – Springer. – 2019. – pp. 321–345. – doi: 10.1007/978-1-4842-4427-2_20
7. Garland, M. Multiresolution modeling: Survey & future opportunities / M. Garland // State of the art report. – 1999. – pp. 111–131.
8. Zienkiewicz Ja. Monocular, Real-Time Surface Reconstruction Using Dynamic Level of Detail / Jacek Zienkiewicz, Akis Tsiotsios, Andrew Davison, Stefan Leutenegger // Proceedings of the 4th International Conference on 3D Vision. – 2016. – pp. 37–46. – doi: 10.1109/3DV.2016.82
9. Huebner R. Application of continuous level-of-detail for 3D games / Robert Huebner // Proceedings of SIGGRAPH 2000. – 2000. – No 39.
10. Fuhrmann A. Coarse view-dependent levels of detail for hierarchical and deformable models / Anton Fuhrmann, Dieter Schmalstieg // Technical Report TR-186-2-99-20. – 1999.
11. Houle J. Simplification and real-time smooth transitions of articulated meshes / J. Houle, P. Poulin // Proceedings of 27th Graphics Interface Conference. – 2001. – pp. 55–60. – doi: 10.20380/GI2001.07
12. Hoppe H. Progressive meshes / H. Hoppe // Proceedings of the 23rd Conference on Computer graphics and Interactive Techniques. – 1996. – pp. 99–108. – doi: 10.1145/237170.237216
13. Hoppe H. Efficient implementation of progressive meshes / H. Hoppe // Computers & Graphics. – 1998. – Vol. 22(1). – pp. 27–36. – doi: 10.1016/S0097-8493(97)00081-2
14. Biljecki F. The effect of acquisition error and level of detail on the accuracy of spatial analyses / Filip Biljecki, Gerard B.M. Heuvelink, Hugo Ledoux, Jantien Stoter // Cartography and Geographic Information Science. – 2018. – Vol. 45, Issue 2. – pp. 156–176. – doi: 10.1080/15230406.2017.1279986

15. Hoppe H. New quadric metric for simplifying meshes with appearance attributes / H. Hoppe // Proceedings of IEEE Conference Visualization'99. – 1999. – pp. 59–510. – doi: 10.1109/VISUAL.1999.809869
16. Garland M. Surface simplification using quadric error metrics / M. Garland, P. S. Heckbert // Proceedings of the 24th Conference on Computer graphics and Interactive Techniques. – 1997. – pp. 209–216. – doi: 10.1145/258734.258849

REFERENCES

1. Azad M. Madni, Carla C. Madni, Scott D. Lucero. Leveraging Digital Twin Technology in Model-Based Systems Engineering. Systems, Vol. 7, No. 1, p. 7, 2019. doi: 10.3390/systems7010007
2. Dychka I., Sulema Ye., Bukhtiarov Iu. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing. Visnyk of Kherson National Technical University, 2019, No 3 (70), pp. 112–119. doi: 10.35546/kntu2078-4481.2019.3.12
3. Cignoni, P., Montani, C., Scopigno, R. A comparison of mesh simplification algorithms. Computers & Graphics, 1998, 22(1), 37–54. doi: 10.1016/S0097-8493(97)00082-4
4. Xiaoqiang Cheng, Huayi Wu, Tinghua Ai, Min Yang. Detail Resolution: A New Model to Describe Level of Detail Information of Vector Line Data. Spatial Data Handling in Big Data Era, Springer, 2017, pp. 167–177. doi: 10.1007/978-981-10-4424-3_12
5. Erikson, C. Polygonal simplification: An overview. UNC Chapel Hill Computer Science Technical Report TR96-016, 1996
6. Tomas Akenine-Möller, Jim Nilsson, Magnus Andersson, Colin Barré-Brisebois, Robert Toth, Tero Karras. Texture Level of Detail Strategies for Real-Time Ray Tracing. Ray Tracing Gems, Springer, 2019, pp. 321–345. doi: 10.1007/978-1-4842-4427-2_20
7. Garland, M. Multiresolution modeling: Survey & future opportunities. State of the art report, 1999, pp. 111–131.
8. Jacek Zienkiewicz, Akis Tsiotsios, Andrew Davison, Stefan Leutenegger. Monocular, Real-Time Surface Reconstruction Using Dynamic Level of Detail.

Proceedings of the 4th International Conference on 3D Vision, 2016, pp. 37–46. doi: 10.1109/3DV.2016.82

9. Robert Huebner. Application of continuous level-of-detail for 3D games. Proceedings of the Conference SIGGRAPH 2000, No 39, 2000.

10. Anton Fuhrmann, Dieter Schmalstieg. Coarse view-dependent levels of detail for hierarchical and deformable models. Technical Report TR-186–2–99–20, 1999.

11. Houle, J., Poulin, P. Simplification and real-time smooth transitions of articulated meshes. Proceedings of 27th Graphics Interface Conference, 2001, pp. 55–60. doi: 10.20380/GI2001.07

12. Hoppe, H. Progressive meshes. Proceedings of the 23rd Conference on Computer graphics and Interactive Techniques, 1996, pp. 99–108. doi: 10.1145/237170.237216

13. Hoppe, H. Efficient implementation of progressive meshes. Computers & Graphics, 1998, Vol. 22(1), pp. 27–36. doi: 10.1016/S0097-8493(97)00081-2

14. Filip Biljecki, Gerard B.M. Heuvelink, Hugo Ledoux, Jantien Stoter. The effect of acquisition error and level of detail on the accuracy of spatial analyses. Cartography and Geographic Information Science, 2018, Vol. 45, Issue 2, pp. 156–176. doi: 10.1080/15230406.2017.1279986

15. Hoppe, H. New quadric metric for simplifying meshes with appearance attributes. Proceedings of IEEE Conference Visualization'99, 1999, pp. 59–510. doi: 10.1109/VISUAL.1999.809869

16. Garland, M., Heckbert, P. S. Surface simplification using quadric error metrics. Proceedings of the 24th Conference on Computer graphics and Interactive Techniques, 1997, pp. 209–216. doi: 10.1145/258734.25884

Received 02.12.2019.

Accepted 04.12.2019.

Метод генерації рівнів деталізації для скелетних моделей

У цій статті вирішується проблема спрощення анімованих моделей зі скелетом та створення рівнів деталізації для них. Спрощення візуальних моделей особливо важливе, коли обчислювальні ресурси є обмеженими. У цій статті наводяться та аналізуються методи створення рівнів деталізації для анімованих моделей. Представлено і обговорено покращений метод генерації рівнів деталізації. Він дозволяє спрощення анімованих мешів без істотних недоліків в якості анімації.

Levels-Of-Detail Generation Method for Skeletal Meshes

This paper is devoted to the development of an algorithm for Levels-Of-Detail generation from skinned meshes. The simplification of visual models is especially important when computing resources are limited. An analysis of existing algorithms for Levels-Of-Detail

generation for animated meshes is presented and discussed. An improved method for Levels-Of-Detail generation is introduced and discussed. It allows to achieve the animated meshes simplification without significant drawbacks in animation quality.

Сулема Євгенія Станіславівна – к.т.н., доцент кафедри програмного забезпечення комп’ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Лось Ігор Анатолійович – магістрант кафедри програмного забезпечення комп’ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Сулема Евгения Станиславовна – к.т.н., доцент кафедры программного обеспечения компьютерных систем Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского».

Лось Игорь Анатольевич – магистрант кафедры программного обеспечения компьютерных систем Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского».

Sulema Yevgeniya – PhD, associate professor at Computer Systems Software Department of National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

Los Ihor – Master Program student at Computer Systems Software Department of National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.