



## Moment-based metrics for mesh simplification

Hui Tang, Huazhong Shu, Jean-Louis Dillenseger, Xudong D. Bao, Limin M. Luo

### ► To cite this version:

Hui Tang, Huazhong Shu, Jean-Louis Dillenseger, Xudong D. Bao, Limin M. Luo. Moment-based metrics for mesh simplification. *Computers and Graphics*, 2007, 31 (5), pp.710-718. 10.1016/j.cag.2007.05.001 . inserm-00182475

**HAL Id: inserm-00182475**

**<https://inserm.hal.science/inserm-00182475>**

Submitted on 26 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Moment-based metrics for mesh simplification

H. Tang<sup>1, 2, 3, 4</sup>, H. Z. Shu<sup>1, 4</sup>, J. L. Dillenseger<sup>2, 3, 4</sup>, X. D. Bao<sup>1, 4</sup>, L. M. Luo<sup>1, 4</sup>

<sup>1</sup>Laboratory of Image Science and Technology, Department of Computer Science and

Engineering, Southeast University, 210096, Nanjing, China

<sup>2</sup>INSERM, U642, Rennes, F-35000, France

<sup>3</sup>Université de Rennes 1, LTSI, Rennes, F-35000, France

<sup>4</sup>Centre de Recherche en Information Biomédicale Sino-Français (CRIBs)

Information about the corresponding author:

Hui Tang

Laboratory of Image Science and Technology

School of Computer Science and Engineering

Southeast University, 210096, Nanjing, China

Tel: 00-86-25-83 79 42 49

Fax: 00-86-25-83 79 26 98

E-mail: corinna@seu.edu.cn

E-mail address of the other authors:

Hua Zhong Shu: shu.list@seu.edu.cn

Jean-Louis Dillenseger: jean-louis.dillenseger@univ-rennes1.fr

Xu Dong Bao: bao.list@seu.edu.cn

Li Min Luo: luo.list@seu.edu.cn

**Abstract:** Mesh simplification is an important research topic in scientific visualization and virtual reality. The simplification metric is a key issue of a simplification algorithm. In this study, two new simplification metrics based on surface moments and volume moments are proposed, which take the difference between the moments defined by the original mesh and those of the simplified mesh as the objective function. These metrics were used in an edge collapse scheme in order to prove their usability in the mesh simplification procedure. For a given maximum order and the number of triangles required, the optimal mesh with a minimum moment difference from the original mesh can be determined. The procedures are applied to some models and better results are obtained in comparison with some known algorithms.

**Keywords:** Mesh simplification; Decimation metric; Surface moments; Volume moments.

## 1. Introduction

The most commonly used three-dimensional (3D) boundary representation in computer graphics and vision is the polygonal representation. In this case an object is represented by a mesh of polygons. Triangle meshes, owing to their geometrical properties and mathematical simplicity, have become a common standard in computer graphics. But on many occasions, to achieve the requirement of high fidelity, high resolution model is generally comprised of millions of triangle primitives. The total number of triangles used to represent an object often largely exceeds the capacity of the graphics hardware real-time rendering, which represents one of the virtual simulation bottlenecks. One natural way to solve this problem consists of simplifying the mesh models by eliminating elements of polygons (vertices, edges, faces) for topologically-simple surfaces or by reducing the geometric and topological complexity for topologically-rich models [1], replacing the original object with fewer triangles while trying to keep its main characteristics.

The mesh simplification methods can be classified into four groups: sampling, adaptive subdivision, decimation and vertex merging. Among these methods, the decimation techniques which principally consist in iteratively removing vertices, edges or triangles from the mesh are very fast and relatively simple to program. According to the deleted elements such as vertices, edges or triangles, the simplification method is called vertex removal, edge collapse or triangle removal respectively. Most of the decimation approaches are based on the following iterative framework: 1) assigning a cost to each valid operation which represents the amount of change introduced in the model, 2) applying the operation with minimum cost, 3) recalculating the costs of the operation belonging to the modified mesh portion. This iteration continues until the desired resolution is reached.

Mesh simplification based on decimation methods introduces a metric between the original mesh and the approximated one. In the design of a mesh simplification algorithm, an important issue is the selection of an element to be deleted. Choosing optimal elements guarantees minimization of the metric between the original mesh and the final approximation. Oliver et al. [2] compared several metrics for mesh simplification and indicated the importance of the metric to the quality of approximations. Park et al. [3] proposed an area-based metric and demonstrated its efficiency. Instead of combining the error volume with additional constraints into quadric matrices [4], Oliver et al. [2] used the volume itself as a simplification metric and the experimental result demonstrated that it is suitable for applications that require fast and memoryless simplification while generating approximations with considerable quality.

Most of the proposed metrics are based on local properties which guaranty preservation of local features. The area-based metric and the volume based metric are based on the object's global features, but their preserved characteristics are single and lack of enough information. Moments and moment invariants contain much information about the object and they are widely used in object representation and recognition [5]. Since the moments of lower order (up to two) can be used to describe the shape of boundary segment, they have been successfully used to detect the image

boundary [6, 7]. Shu et al. [8] proposed moment-based methods for polygonal approximation of digitized curves. Inspired by this research work, a surface moments based metric and a volume moments based metric are developed in this paper with the purpose of generating low error approximations and being simple to implement. An implementation based on edge collapse is used in the experiments, with the aim of comparing the different metrics independently of other aspects related to the simplification method. A uniform framework is applied for the simplification of all models, and only the investigated metrics are different.

The rest of this paper is organized as follows. Section 2 reviews briefly some relevant previous works on simplification metrics. The proposed moment based metrics are presented in detail in Section 3. These metrics are evaluated and the results are discussed in section 4. Finally, the conclusions are given in Section 5.

## 2. Related previous works

As mentioned before, the decimation based simplification process often takes an iterative method. The metric according to which we decide the simplification sequence is crucial during the iterative process. Some existing metrics are presented in this section.

Simplification metrics can be divided into two classes: local or global features based metrics. Within the class of local feature based metrics methods, one of the first vertex removal based simplification methods was proposed by Schroeder et al. [9]. The implemented simplification metric is the distance of the vertex to the average plane formed by its adjacent triangles. This metric is very simple to compute, but generates low-quality approximated models. Kim et al. [10] proposed a discrete curvature metric for simplification which has been demonstrated as not satisfactory because it always generates low-quality simplifications [2]. Vollmer et al. [11] used the standard deviation of a vertex set as a simplification metric. It has the effect of smoothing a triangle mesh during its simplification. Graland et al. [12] proposed a quadric error metrics (QEM) based algorithm. This algorithm made use of the quadric error metric to choose the edge to be simplified and the new vertex after contraction. It estimates the error introduced by a pair collapse operation as the distance from a vertex to a quadratic surface, represented as a symmetric matrix. The algorithm provides high-quality results because the quadric matrices are accumulated during the simplification process. Lindstrom et al. [4] added volume preservation and boundary perseveration constrains to the quadratic objective functions when selecting the position of the new vertex. Hoppe [13] introduced an energy function to describe the complexity and fidelity of mesh and tracked simplification quality minimizing it. Hoppe's energy function requires many vertex distance evaluations so that it reduces the computational speed. Klein et al. [14] evaluated the Hausdorff distance between the original and simplified models, allowing precise error control. Hussain et al. [15] proposed a metric which is the summation of geometric change combined with vertex visual importance.

The above presented metrics are all designed according to the local feature changes. Wu et al. [16] modified QEM to postpone the simplification of global

features by adding the assigned weight to the contraction cost. The global feature preservation is used as constraints within the simplification process and the metric itself is still according to the local features.

The method based on global feature change is another efficient way to determine the approximated meshes. Among this kind of methods, Park et al. [3] proposed a global feature based metric, which compares the area difference of the original mesh  $S$  and its approximation  $S'$ . The cost function is defined as:

$$AD = |AREA(S) - AREA(S')| \quad (1)$$

A volume-based metric as the geometric deviation measure was proposed by Alliez et al. [17]. It minimizes the volume between the simplified mesh and the original mesh. The cost function is as follows:

$$VD = |VOLUME(S) - VOLUME(S')| \quad (2)$$

Inspired by these two metrics, we propose two moment based simplification metrics with the purpose of improving the approximated results.

### 3. Proposed moments based metrics

In this section, the two moments based metrics are presented. They are based on surface moments and volume moments respectively. The fast computation methods proposed by Tuzikov et al. [18] are applied to reduce the computational burden.

#### 3.1. Surface moments based metric

The surface moment of order  $k_1+k_2+k_3$  of a 3D compact body  $P$  is defined as:

$$m_{k_1k_2k_3}S(P) = \int_{S(P)} x^{k_1} y^{k_2} z^{k_3} dS. \quad (3)$$

where the integral is taken on the surface of  $P$ .

Notice that  $m_{000}S(P)$  is the area of the model's surface which is used by Park et al. [3] as a simplification metric (Eq. 1) in their method. Since the higher order moments contain more information about the object's surface, we will use the moment difference to measure the similarity between the original model surface  $S$  and the simplified model surface  $S'$ . Let  $m_{k_1k_2k_3}S(P)$  and  $m_{k_1k_2k_3}S'(P)$  be the surface moments defined by  $S$  and  $S'$ , respectively. Then we define the following simplification cost function:

$$SMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q (m_{p-q,q-r,r}S(P) - m_{p-q,q-r,r}S'(P)) \right| \quad (4)$$

where  $M$  is the maximum order of moments that we will use.

Comparing Eq. (4) with Eq. (1), it can be seen that Eq. (1) corresponds to a special case of Eq. (4) (with  $M = 0$ ). Therefore, we can expect to obtain better results using Eq. (4) with  $M > 0$  contrasting with Eq. (1). However, the algorithm directly based on Eq. (4) could be time consuming because the moment calculation by a straightforward method is very expensive. This problem can be solved using a simple

and fast surface moment computing algorithm which was proposed by Tuzikov [18]. A brief description of this algorithm is given below.

Assume that the object is represented by a mesh of  $N$  triangles, each triangle defined by its vertices  $\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic}$ ,  $i = 1, 2, \dots, N$ . For each triangle we form a tetrahedron denoted by  $T_i$  defined by the three vertices and the coordinate origin. Let  $S_0(T_i)$  be the tetrahedron facet that is opposite to the coordinates origin, then Eq. (3) can be rewritten as:

$$m_{k_1 k_2 k_3} S(P) = \int_{S(P)} x^{k_1} y^{k_2} z^{k_3} dS = \sum_i m_{k_1 k_2 k_3} S_0(T_i) \quad (5)$$

Eq. (5) shows that we need only to calculate the surface moment  $m_{k_1 k_2 k_3} S_0(T_i)$  of order  $k_1 + k_2 + k_3$ . To do this, let  $T_i = T_i(\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic})$  be the tetrahedron formed by the coordinate origin and the three vertices  $\mathbf{v}_{ia} = (a_1, a_2, a_3)$ ,  $\mathbf{v}_{ib} = (b_1, b_2, b_3)$ ,  $\mathbf{v}_{ic} = (c_1, c_2, c_3)$  and the vertices  $\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic}$  are arranged in counter-clockwise order with respect to the outward normal of the surface  $S_0(T_i(\mathbf{v}_{ia}, \mathbf{v}_{ib}, \mathbf{v}_{ic}))$ .

Let us introduce some notations. Denote by  $A = (A_{ij})$  the following matrix

$$A = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix}.$$

Given integers  $k_1, k_2, k_3$  denote by  $\zeta$  a set of such  $3 \times 3$  matrices  $(k_{ij})$  with integer values  $k_{ij}$ ,  $0 \leq k_{ij} \leq k_i$  such that  $\sum_{j=1}^3 k_{ij} = k_i$ ,  $i = 1, 2, 3$ . The following formula is derived for computing the surface moment  $m_{k_1 k_2 k_3} S_0(T)$  of order  $k = k_1 + k_2 + k_3$ :

$$m_{k_1 k_2 k_3} S_0(T) = \frac{2Ar(S_0)k_1!k_2!k_3!}{(k+2)!} \times \sum_{(k_{ij}) \in \zeta} \frac{\prod_{j=1}^3 \left( \left( \sum_{i=1}^3 k_{ij} \right)! \right)}{\prod_{i,j=1}^3 (k_{ij}!)} \prod_{i,j=1}^3 A_{ij}^{k_{ij}} \quad (6)$$

where  $Ar(S_0)$  denotes the area of the face  $S_0(T)$ .

The fast surface moment computing algorithm allows us to develop a mesh simplification method based on higher order moments.

Using Eq. (5), Eq. (4) becomes:

$$SMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q \left( \sum_i m_{p-q,q-r,r} S_0(T_i) - \sum_i m_{p-q,q-r,r} S'_0(T_i) \right) \right| \quad (7)$$

Eq. (7) can be used to determine a sequence of approximation meshes with different number of triangles. Let  $F_N$  be the original mesh with  $N$  triangles. For a specified number  $n$ , the objective may be to find an approximated mesh  $F_n$  so that the moment differences of  $F_n$  from the initial mesh  $F_N$  is minimal. The candidate  $F_n$  is uncountable because the three-dimensional model is complicated. In order to simplify the optimization procedure, we take the iterative decimation methods (vertex removal, face removal or edge collapse) to determine a sequence of approximated meshes of

the initial model. During the iterative decimation procedure, only the moments related to the modified triangles are changed for each step. So Eq. (7) can be simplified to a more efficient form. Here we take edge collapse procedure for illustration.

As illustrated in figure 1, each operation will remove two triangles adjacent to the deleted edge  $e_{ij}$ . That is, we start with the initial mesh  $F_N$  to get the first simplified mesh  $F_{N-2}$ , then the simplified mesh  $F_{N-4}$  from  $F_{N-2}$ , and so on until finding  $F_n$ . The greedy strategy we use in the approximation process is that  $F_{N-2}$  is derived from  $F_N$  such that the surface moment difference between  $F_{N-2}$  and  $F_N$  is minimal (global minimum). To obtain the approximated mesh  $F_{N-2}$  from  $F_N$ , the kernel operation of the approximation process is to collapse an edge  $e_{ij}$  to a point  $k$ . Since all the other triangles have no change except the triangles adjacent to the two points  $i$  and  $j$ , which are the two vertices of the deleted edge. We can deduce from Eq. (7) that the moment difference of order up to  $M$  between  $F_{N-2}$  and  $F_N$  is as follows:

$$ESMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q \left( \sum_{v \in (T_i + T_j - T_{e_{ij}})} m_{p-q, q-r, r} S_0(T_v) - \sum_{v \in T_k} m_{p-q, q-r, r} S_0(T_v) \right) \right| \quad (8)$$

where  $T_v$  denotes the triangles adjacent to a vertex  $v$  and  $T_{e_{ij}}$  denotes the two triangles adjacent to the edge  $e_{ij}$ . Eq. (8) shows that the computation of adjacent triangle moments is taken instead of the whole object triangle moments.

This principle can be extended to vertex or face removal methods by only taking their respective triangle changes caused by each operation into account.

### 3.2. Volume moments based metric

The volume moment of order  $k=k_1+k_2+k_3$  of a 3D compact body  $P$  is defined as:

$$m_{k_1 k_2 k_3} V(P) = \int_P x^{k_1} y^{k_2} z^{k_3} dx dy dz. \quad (9)$$

where the integral is taken on the volume of  $P$ .

Notice that  $m_{000}V(P)$  is the volume of the model  $P$  which is used by Alliez et al. [17] as a simplification metric (Eq. 2) in their method. The simplification cost function by using the volume moments is similar to that of the surface moments. It is defined by

$$VMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q (m_{p-q, q-r, r} V(P) - m_{p-q, q-r, r} V'(P)) \right| \quad (10)$$

where  $m_{k_1 k_2 k_3} V(P)$  and  $m_{k_1 k_2 k_3} V'(P)$  denote the  $(k_1+k_2+k_3)$ th-order of volume moments defined on the volume  $P$  and simplified mesh  $P'$ , respectively. Comparing Eq. (10) with Eq. (2), it can be seen that Eq. (2) corresponds to a special case of Eq. (10) (with  $M = 0$ ).

Tuzikov et al. [18] also proposed a fast algorithm for computing the volume moments. The computation method of volume moments is similar to that of surface moments. Similar to Eqs. (5), (6), the corresponding formulas for volume moments based simplification are as follows:



$$m_{k_1 k_2 k_3} V(P) = \int_P x^{k_1} y^{k_2} z^{k_3} dx dy dz = \sum_i m_{k_1 k_2 k_3} V(T_i) \quad (11)$$

$$m_{k_1 k_2 k_3} V(T) = \frac{|A| k_1! k_2! k_3!}{(k+3)!} \times \sum_{(k_{ij}) \in \zeta} \frac{\prod_{j=1}^3 \left( \left( \sum_{i=1}^3 k_{ij} \right)! \right)}{\prod_{i,j=1}^3 (k_{ij}!)} \prod_{i,j=1}^3 A_{ij}^{k_{ij}} \quad (12)$$

where  $|A|$  is the determinant of  $A$ .

Using Eq. (11), Eq. (10) becomes:

$$VMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q \left( \sum_i m_{p-q,q-r,r} V(T_i) - \sum_i m_{p-q,q-r,r} V'(T_i) \right) \right| \quad (13)$$

For iterative mesh simplification methods, only some local modification presents to each iteration. Similar to that of surface moments based metric, we take edge collapse operation to illustrate the simplified form of Eq. (13) for a series of decimation methods. We can deduce from Eq. (13) that the moment difference of order up to  $M$  between  $F_N$  and  $F_{N-2}$  is as follows:

$$EVMD = \left| \sum_{p=0}^M \sum_{q=0}^p \sum_{r=0}^q \left( \sum_{v \in (T_i + T_j - T_{\epsilon_{ij}})} m_{p-q,q-r,r} V(T_v) - \sum_{v \in T_k} m_{p-q,q-r,r} V(T_v) \right) \right| \quad (14)$$

where  $EVMD$  denotes the global volume moment difference of an edge collapse operation, which is called the cost function. As mentioned previously, this metric can be easily extended to vertex or face removal operation.

## 4. Experiments

The experiments were performed on a PC Pentium 4 2.66GHZ CPU with 512MB RAM, running on Windows XP operating system. Visual C++ and OpenGL were taken as development tools. The results for the cow model (2904 vertices and 5804 triangles) and the North America model (2025 vertices and 3872 triangles) are presented. The original models are shown in Fig. 2.

### 4.1. Mesh simplification process

In order to evaluate the proposed metrics, we follow the same framework as described in [2]. In this paper, the authors suggest to employ the edge collapse operation (Fig. 1) for the simplification procedure because it preserves topology and it is easy to implement. In our opinion, it is a good choice because the approximation results mainly rely on the metrics, without the involvement of any other factors (like retriangulation method choice).

Based on the edge collapse method, our greedy strategy for determining the approximated meshes can be summarized as follows. In each step of the approximation process, i.e., when we pass  $F_N$  and  $F_{N-2}$ , the edge among all the edges of  $F_N$  is removed if it gives the minimum moment difference value. The corresponding algorithm is as follows:

Input: The original mesh  $F_N$  containing  $N$  triangles, the maximum order of moments

$M$  and the number of triangles  $n$  required.

Step 1: Compute every triangle's moments using Eq. (6) or Eq. (12) and then compute moment difference for each edge collapse operation using Eq. (8) or Eq. (14).

The moment differences are stored as a contraction cost for every edge.

Step 2: Sort the costs in an increasing order.

Step 3: Select the top edge in the queue, and check if it can be contracted. If not, remove it from the queue and return to the beginning of step 3. If yes, contract this edge and recalculate the information of the affected triangles and edges.

Step 4: Update the position of the affected edges in the cost queue.

Step 5: Repeat step 3 and step 4 until the required triangle number of the approximation is reached.

Each time when one edge is collapsed, two triangles and one edge will be deleted. In order to perform the operation, we must choose a position for  $k$ . A simple scheme is implemented, which is to choose  $v_i$ ,  $v_j$  or  $(v_i + v_j)/2$  depending on the lowest value of simplification cost that produced by the new position. In order to preserve the vision characteristic, if the largest normal direction change of one edge's adjacent triangles is greater than a certain threshold (here we choose  $\pi/4$ ) after edge contraction, we keep the edge.

#### 4.2. Metrics evaluation method

The two moments based metrics were evaluated qualitatively and estimated by measuring the global simplification error assessment. Since the root mean squared (*RMS*) error measures the global average error between the model and its approximation, it is commonly used as the efficiency measurement in the mesh simplification algorithms. We used the error detection tool Metro [19] to calculate the *RMS* error between one model and its approximation. The *RMS* error from  $S'$  to  $S$  is defined as:

$$d_{rms}(S, S') = \sqrt{\frac{1}{AREA(S)} \int_S d(p, S')^2 ds}.$$

Besides the error detection method, the moments themselves can also be used as evaluation tools. The moment difference between the approximation and the original model is an efficient measure tool to evaluate the moments preservation ability of the simplification methods. We have done some experiments to demonstrate this evaluation method.

#### 4.3. Experimental results

Fig 3 and 4 present the qualitative results of a 70% model simplification rate (from 5804 triangles to 1704 triangles) using respectively surface and volume moments based metrics for different maximum moment order values  $M$ . In all the cases we can see that they still contain the major topology characteristic of the initial model. From Fig. 3, we can see that the choice of  $M > 0$  gets qualitatively better results than that of  $M = 0$ . When  $M = 0$ , the surface moments based metric becomes the area-based metric of Eq. 1. Fig 4 shows that the increase of the moment order for

volume based method slightly improves the approximated results, but it performs better than the surface moments based simplification.

Fig. 5 and Fig. 6 show the simplification errors of the two moments based metrics for different values of  $M$  in terms of the simplification rate. The results presented in Fig 5(a) confirm quantitatively the visual effect of Fig. 3. For the two moments based metrics,  $M = 0$  represents area-based metric and volume based metric respectively. When the required facet rate in the approximation is low, it seems that  $M > 0$  can get a lower simplification error than  $M = 0$ . That is to say, the moments based metrics can get a better result in the simplification error sense.

In order to compare the two moments based metrics, a graph of their simplification errors is drawn in Fig. 7. For each metric, we choose the lowest simplification error for  $M$  varying from 0 to 3. It seems that the volume moments based metric provides lower simplification error.

The aim of the proposed metrics is to preserve the moment features of the original models. So in order to demonstrate the preservation ability of the two metrics, we compute the moment difference between the original model and the approximation, using our metrics and the quadric error metric (QEM) respectively. The results are illustrated in Fig. 8 and Fig. 9, for SM based metric and VM based metric respectively. Here we choose  $M = 2$  for illustration. The results show that our methods have a better ability to preserve the moment features. These results also demonstrate that our global moment based metrics can be used to evaluate the several mesh simplification methods.

## 5. Conclusions

We have proposed two new metrics for mesh simplification, with the aim of preserving the original model's global features in the approximation. They are designed based on the surface moments and volume moments by extending the two global metrics, which are area based and volume based metrics, respectively. The experimental results demonstrated that the extended metrics can get better result since they contain more information about the model.

The experiments showed that the speed of convergence of our algorithms is fast enough with the help of fast computation of moments. To reduce the computational time further, our algorithms can be implemented in a parallel processing machine with minor modifications.

## References

- [1] Andujar C, Brunet P, Ayala D, Topology-reducing surface simplification using a discrete solid representation. *ACM Trans. On Graphics* 2002; 21(2): 88-105.
- [2] Oliver M, Hélio P, A comparative evaluation of metrics for fast mesh simplification. *Computer Graphics Forum* 2006; 25 (2): 197-210.
- [3] Park I, Shirani S, Capson DW, Mesh simplification using an area-based distortion measure, *Journal of Mathematical Modelling and Algorithms* 2006; 5: 309-329.
- [4] Lindstrom P, Turk G, Fast and memory efficient polygonal simplification, in *IEEE Visualization 98 Conference Proceedings*, Oct. 1998: 279-286.
- [5] Prokop RJ, Reeves AP, A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models Image Process* 1992; 54 (5): 438-460.
- [6] Ghosal S, Mehrotra R, Orthogonal moment operators for subpixel edge detection, *Pattern Recognition* 1993; 26 (2): 295-306.
- [7] Luo LM, Hamitouche C, Dillenseger JL, Coatrieux JL, A moment based three dimensional edge operator, *IEEE Trans. Biomed. Eng.* 1993; 40 (7): 693-703.
- [8] Shu HZ, Luo LM, Zhou JD, Bao XD, Moment-based methods for polygonal approximation of digitized curves. *Pattern Recognition* 2002; 35: 421-434.
- [9] Schroeder WJ, Zarge JA, Lorensen WE, Decimation of triangle meshes. *Computer Graphics* 1992; 26(2): 65-60.
- [10] Kim SJ, Jeong WK, Kim CH, LOD generation with discrete curvature error metric. In *proceedings of 2nd Korea Israel Bi-National Conference on Geometrical Modeling and Computer Graphics in the WWW Era* 1999: 97-104.
- [11] Vollmer J, Mencl R, Muller H, Improved Laplacian smoothing of noisy surface meshes, *Computer Graphics Forum* 1999; 18(3): 131-138.
- [12] Garland M, Heckbert PS, Surface simplification using quadric error metrics. *Computer Graphics* 1997; 31: 209-216.
- [13] Hoppe H, Progressive meshes, *Computer Graphics* 1996; 30: 99-108.
- [14] Klein R, Liebich G, Strasser W, Mesh reduction with error control, 7<sup>th</sup> *IEEE Visualization* 1996 (VIS'96): 311-318.
- [15] Hussain M, Okada Y, Niijima K, Efficient and feature-preserving triangular mesh decimation. *Jurnal of WSCG* 2004; 12(1-3): 167-174.
- [16] Wu Y, He Y, Cai H, QEM-based mesh simplification with global geometry features preserved, *Computer graphics and interactive techniques in Australasia and South East Asia*, 2004: 50-57.
- [17] Alliez P, Laurent N, Sanson H, Schmitt F, Mesh approximation using a volume-based metric, in *Pacific Graphics 99 Conference Proceedings*, 1999 : 292-301.
- [18] Tuzikov AV, Sheynin SA, Vasiliev PV, Computation of volume and surface body moments. *Pattern Recognition* 2003; 36: 2521-2529.
- [19] Cignoni P, Rocchini C, Scopigno R, Metro: measuring error on simplified surfaces. *Computer Graphics forum* 1998; 17(2): 167-174.

Figure 1: Process of edge collapse from  $e_{ij}$  to vertex  $k$

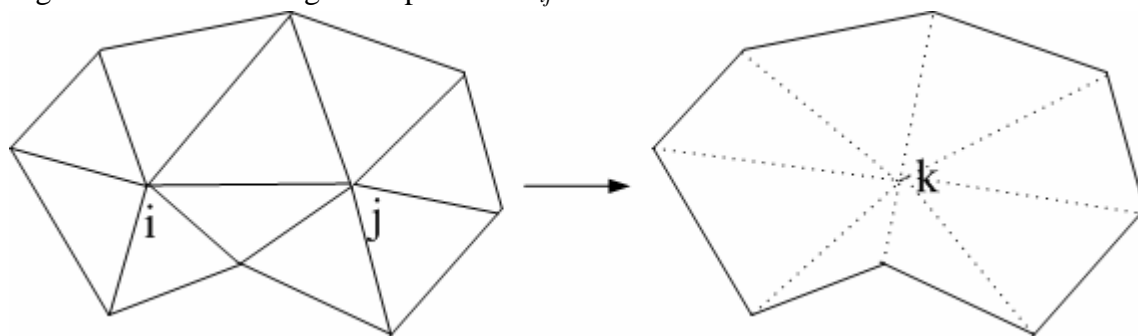
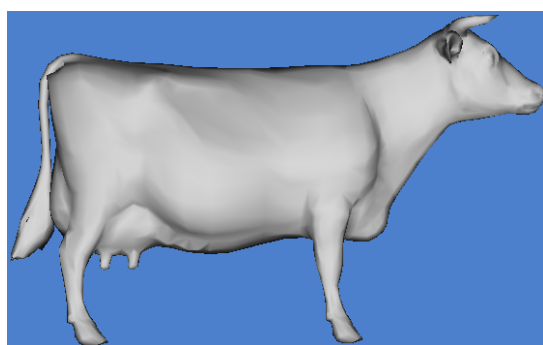
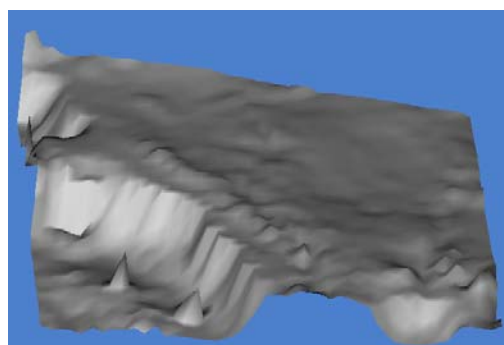


Figure 2: Models for experiment



(a) Cow model



(b) North America model

Figure 3: 70% simplification (possessing 30% of the original faces) for cow model using surface moments based metric up to different maximum order  $M$

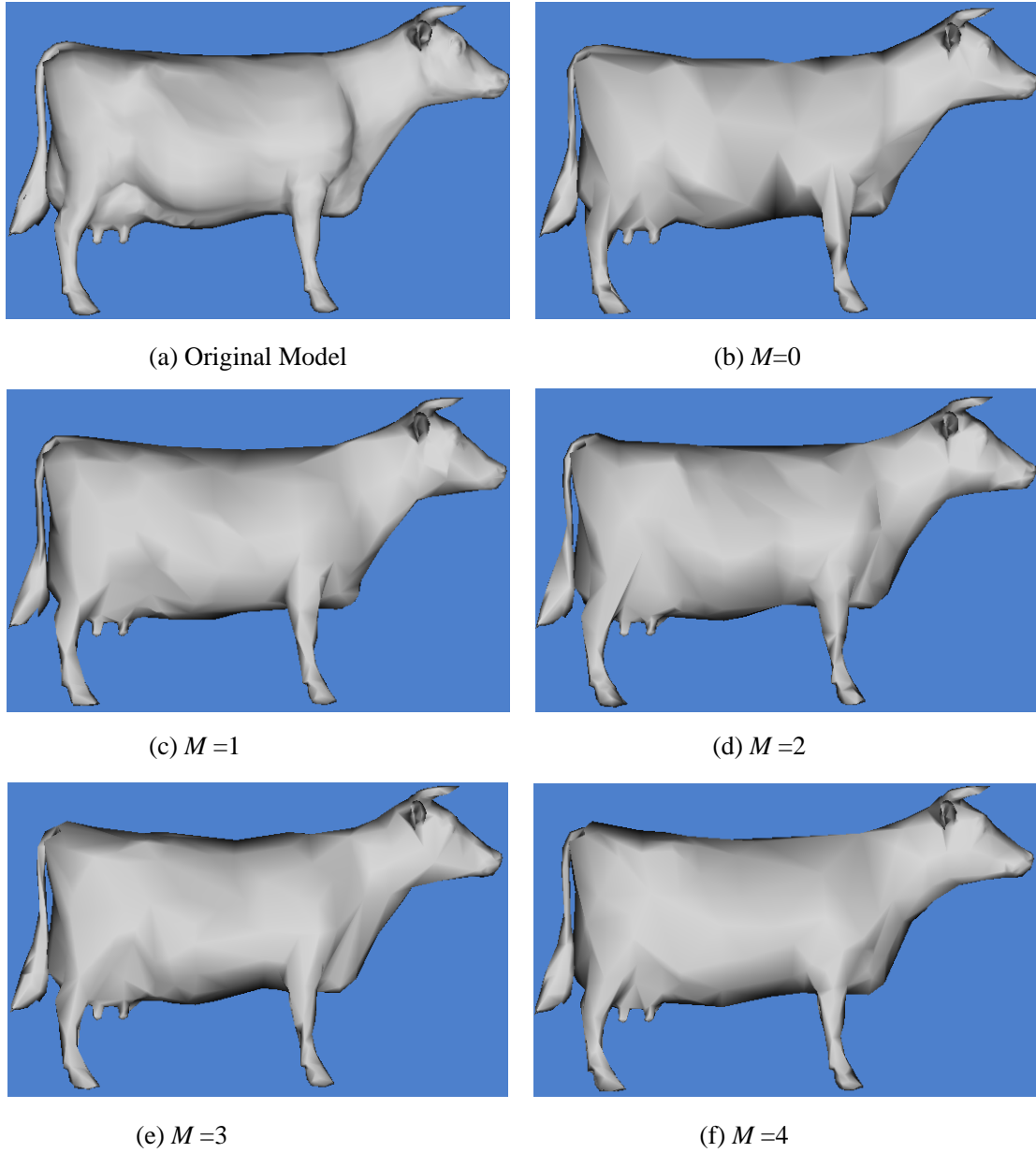


Figure 4: 70% simplification (possessing 30% of the original faces) for cow model using volume moments based metric up to different maximum order  $M$

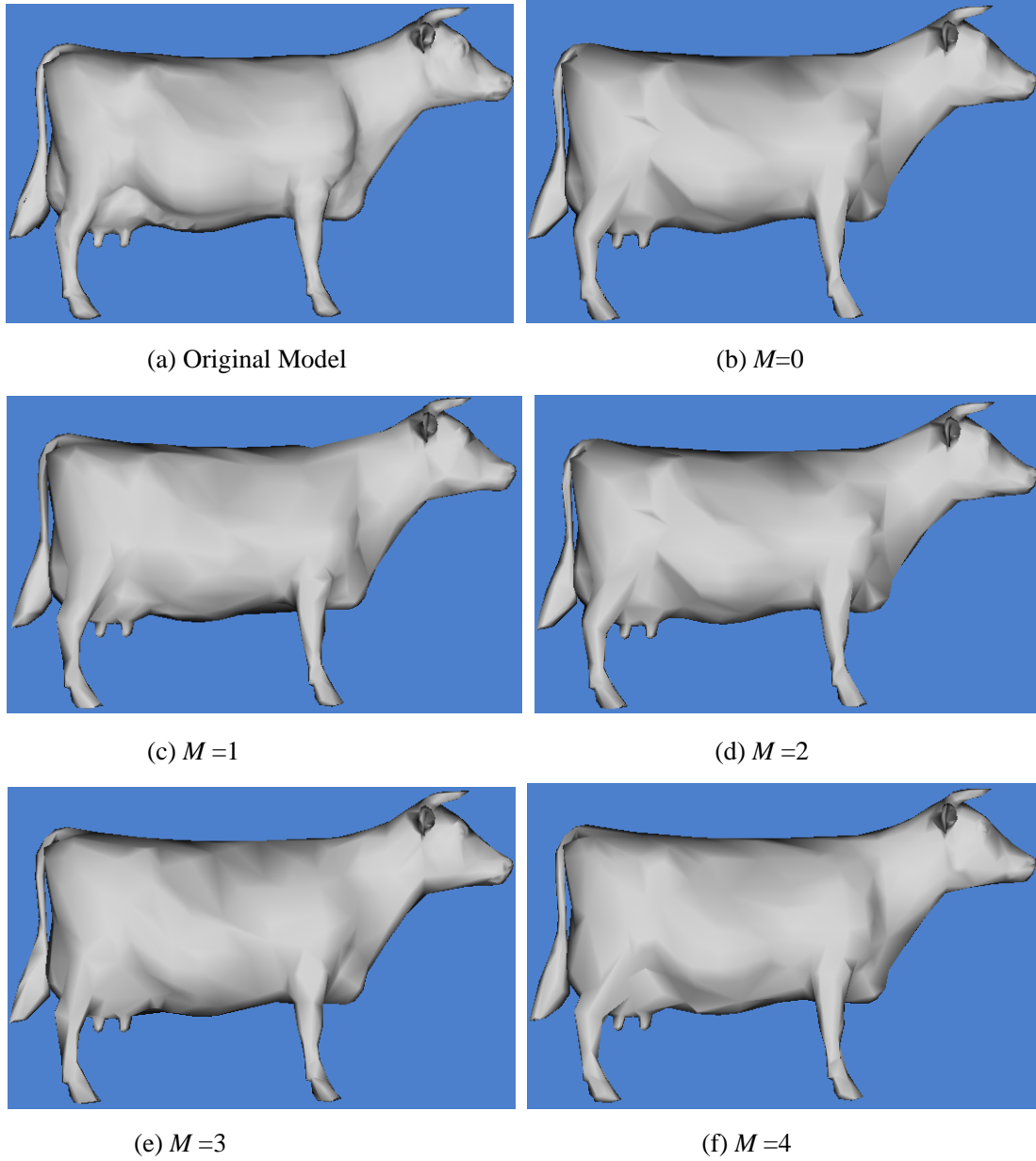
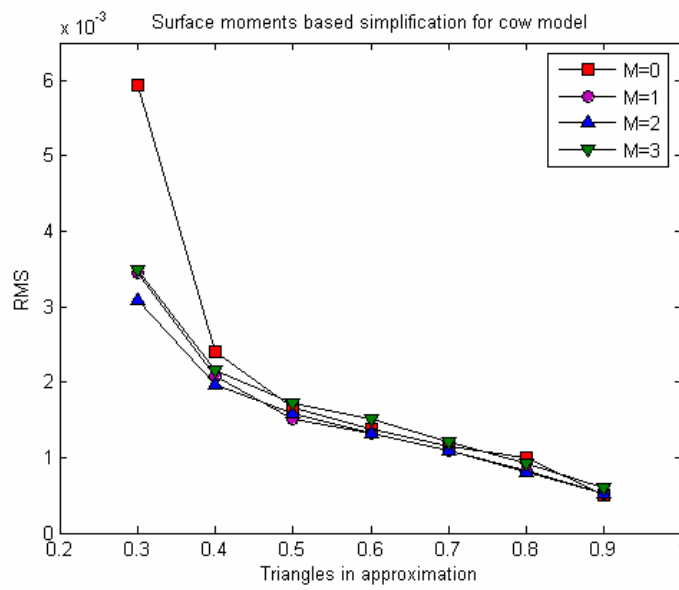
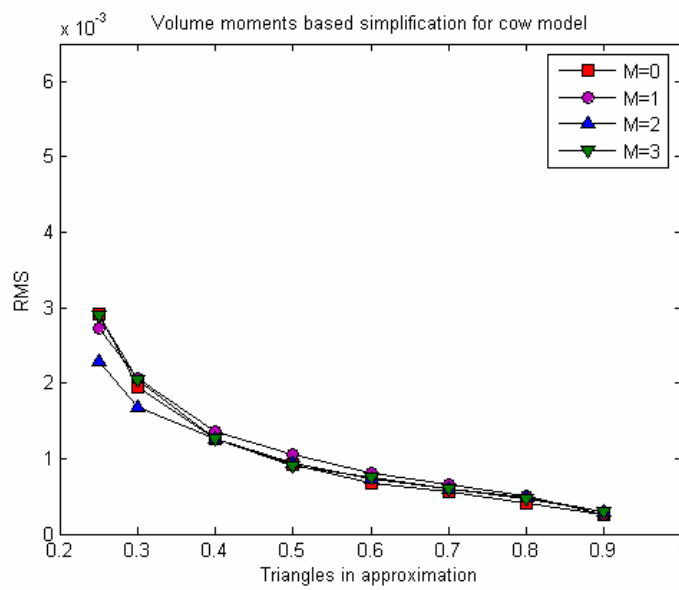


Figure 5: Influence of the moment order on the *RMS* error between the cow model and its approximation for different simplification rate: (a) Surface moments (*SM*) based method. (b) Volume moments (*VM*) based method.



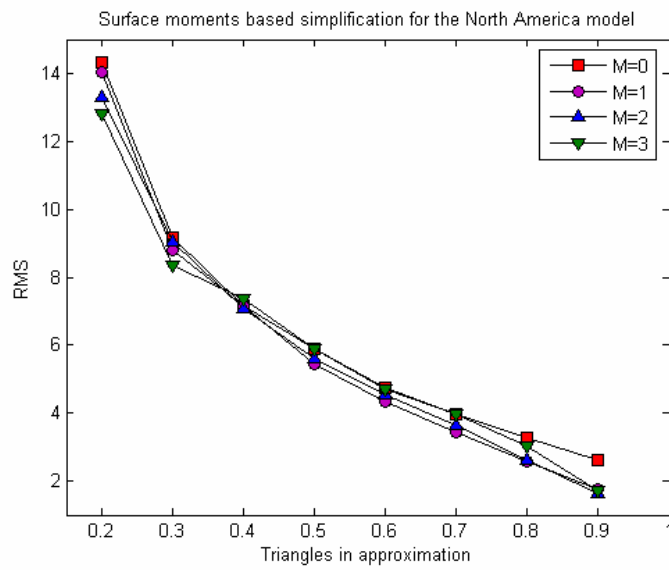
(a) Surface moments



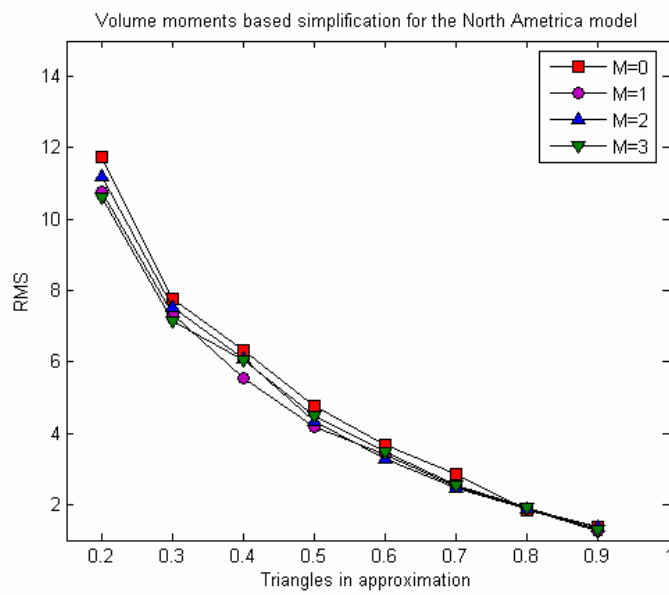
(b) Volume moments



Figure 6: Influence of the moment order on the *RMS* error between the North America model and its approximation for different simplification rate: (a) Surface moments (*SM*) based method. (b) Volume moments (*VM*) based method.

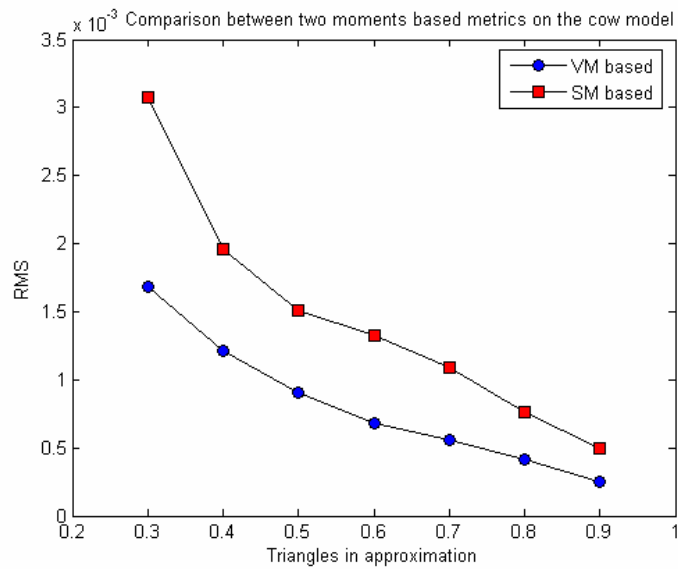


(a) Surface moments

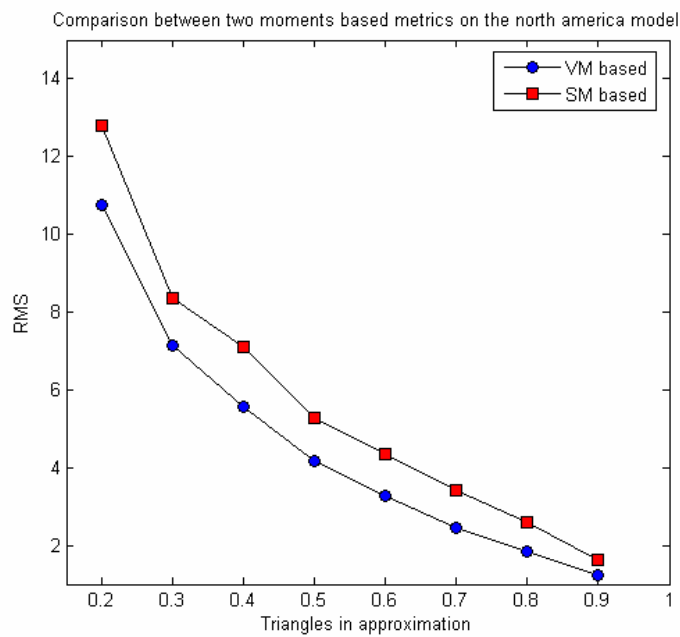


(b) Volume moments

Figure 7: Comparison of *VM* based method and *SM* based method: (a) the cow model.  
(b) the North America model



(a) the cow model



(b) the North America model

Figure 8: Comparison of SM differences between SM based method and QEM

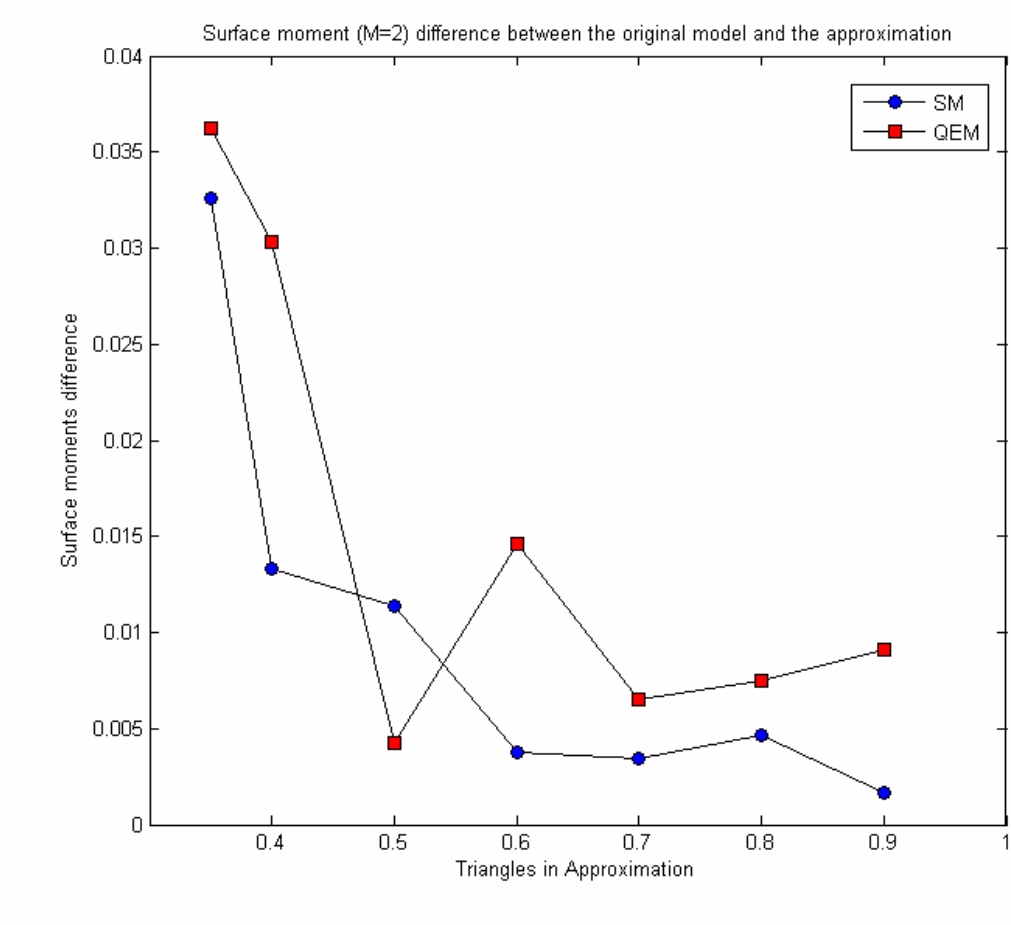


Figure 9: Comparison of VM differences between VM based method and QEM

