JOINT MESH AND TEXTURE COMPRESSION USING MARGINAL ANALYSIS

S.Horbelt, L.Balmelli, M.Vetterli

Swiss Federal Institute of Technology, 1015 Lausanne EPFL, Switzerland

ABSTRACT

In many computer graphic applications, a texture is mapped onto a meshed surface. Under complexity constraints, both texture and mesh need to be approximated. In this paper, we present a framework for the joint simplification of texture and mesh with respect to an error measure in screen space. In order to achieve optimal operating points, we choose two efficient simplification algorithms: Optimal multiresolution mesh simplification based on a quadtree structure and multiresolution view-dependent texture compression based on wavelets. Together, these two algorithms are used for joint simplification using an efficient heuristic based on marginal analysis. As an application example, we study the mapping of aerial orthophotographs onto a terrain model of the swiss alps, and show that the resolution of the terrain and the aerial photograph is efficiently traded-off.

keywords: Compression Algorithms, Texture Mapping, Triangle Decimation, Wavelets, Marginal Analysis

1. PROBLEM FORMULATION

In this paper we address a simple yet basic question of computer graphics: given a surface specified by a mesh and a texture to be mapped on that mesh, what is the resolution required for the mesh and the texture to achieve the best combined display quality? This question is relevant every time we are resource-constrained, and we need to simplify meshes and/or textures mapped on the meshes. The resource constraint might be computational (complexity of rendering meshes and mapping textures), storage space or bandwidth (bitrate needed to represent, transmit or update a complex model and its associated texture information) or both. In such cases, it is critical to choose the correct resolution or level of detail for both the mesh and the texture mapped on it. If more resource become available (e.g. in progressive downloads over the Internet), it is important to know if the mesh or the texture needs to be refined first.

To make our discussion more concrete, consider a specific case where the interplay of texture and mesh resolution is particularly intuitive, namely very large terrain models with aerial photographs to be mapped on the terrain models. Consider two extreme cases. First, consider mapping a very fine texture (a very detailed aerial photograph) onto a very coarse terrain model. This gives an unsatisfactory result: While many details are available, the surface model is too simplistic compared to the texture resolution. At the other extreme, take a very detailed terrain model, but with an overly coarse texture (e.g. one value only per surface element). The mesh is now very complex, but the texture is trivial, thus not reaching a good trade-off. The above two extreme cases hint at a better solution, where the correct trade-off between mesh and texture resolution is found. Let us now formalize the problem. Given is a surface model at full resolution M_0, and a family of simplified models $\{M_i\}_{i=1...N}$. We associate a cost function $C_M(M_j)$ to each model which reflects its complexity (number of bits needed to represent the model, or computational cost for rendering the model). Likewise, given a full resolution texture T_0, and a family of simplified textures $\{T_i\}_{i=1...M}$, we associate a cost function $C_T(T_j)$ to each texture, which measures its complexity (again, bitrate or computational cost of rendering). Finally, for a given pair (M_i, T_j), we define a distortion measure $D(M_i, T_j)$ which evaluates the error in screen space between the image generated by the full resolution version (M_0, T_0) and the approximated version (M_i, T_j). Calling the image on the screen $I(M_i, T_j)$ to reflect its dependence on the underlying model and texture, a possible distortion is the l_2-norm, or

$$D(M_i, T_j) = \|I(M_0, T_0) - I(M_i, T_j)\|_2 \qquad (1)$$

Note that $I(M_i, T_j)$ depends on the rendering process (e.g. lighting), but we assume the same rendering process for the various approximate images.

The statement of the problem is now the following: Given a total budget for the combined complexity of the mesh and texture:

$$C = C_M + C_T \qquad (2)$$

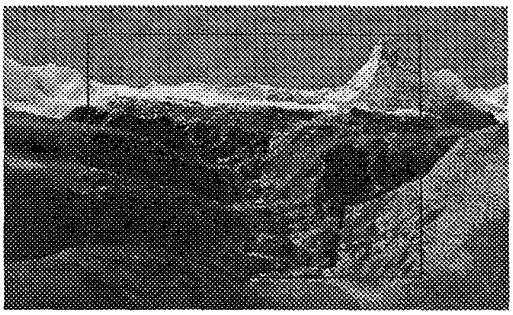


Figure 1: View-dependent compressed texture of Figure 3 mapped on simplified mesh seen from the viewer position. The field-of-view is depicted by the red box.

0-7803-6725-1/01/$10.00 ©2001 IEEE

find the pair (M_i, T_j) that minimizes $D(M_i, T_j), i \in [0 \ldots N]$, $j \in [0 \ldots M]$:

$$D_{min} = \min_{i,j}(D(M_i, T_j)) \quad (3)$$

under the constraint that $C_M(M_i) + C_T(T_j) \leq C$.

Such a formulation is very reminiscent of compression systems, and indeed, when the cost is the required bitrate for model and texture, the above gives a best compression scheme in an operational rate-distortion sense.

The search for the minimum error in eq. (3) can be done by exhaustive search, but this is clearly impractical. Also, generating a set of models $\{M_i\}$ and a set of textures $\{T_i\}$ leading to various costs C_M and C_T can be very complex unless some structure is imposed. Thus it is critical to come up with computationally efficient methods for searching a wide variety of possible approximations. In the following, we focus our attention on the case where the cost is the bitrate required for representing the model and texture. This is a well defined cost, and it is very relevant for applications where storage and/or transmission is involved. Also, we restrict our attention to successive refinement or progressive models and textures. While this is a restricted class, it is a very important one for case where interactivity and communication is involved (e.g. progressive downloads).

2. SOLUTION METHOD

We review two multiresolution approaches to mesh and texture simplification that we used in our joint simplification algorithm. Note that our joint simplification algorithm works with other simplification methods as well.

2.1. Multiresolution meshes

Many multiresolution meshing strategies exist, and they are reviewed for example in [1]. In our case, we use a multiresolution mesh based on an underlying quadtree structure [2]. This allows very efficient pruning methods that minimize distortion in either surface space or screen space for a given bit budget. The algorithm takes $O(N \log N)$ to build a progressive description of a model of N vertices. The efficiency is such that very complex models can be simplified or refined in quasi-real-time (See Figure 2).

2.2. Multiresolution textures

Inspired from work in image compression using wavelets, we develop a multiresolution texture mapping based on wavelet methods. That is, the texture is represented by its wavelet coefficients. These are pruned depending on view point and on the desired resolution. This is similar to the work in [3] about view-dependent compressed images, which predicts the content of projected textures in the frequency domain of the original texture. Given a full resolution view-dependent texture T_0 viewed at a distance d_0, a view-dependent multiresolution pyramid of simplified texture $\{T_i\}_{i=1\ldots M}$ is defined by exponentially increasing the distance $d_i = d_0 \cdot 2^{i/k}$. A scalable bitstream for the pyramid $\{T_i\}_{i=0\ldots M}$ is achieved by coding first the coefficients of the lowest resolution T_M, then adding the additional coefficient from T_{M-1} until the full texture T_0 is reached (See Figure 3).

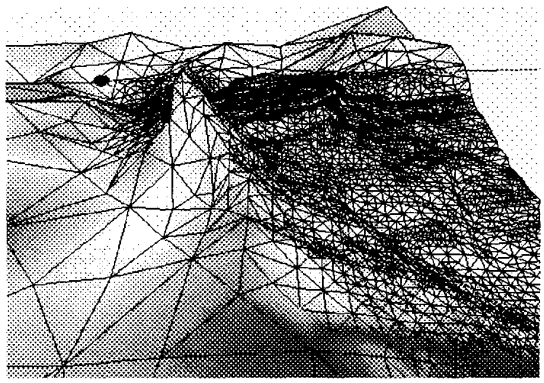


Figure 2: View dependent simplified mesh of ~8K triangles (original model has 131K triangles). The simplification algorithm minimizes the displacement of vertices projected in the screen. The viewer position and the view frustum are respectively depicted by the blue ball and the red lines.

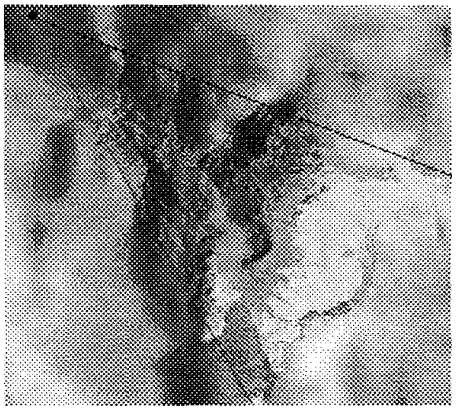


Figure 3: Detail of view-dependent compressed texture. The viewer position and the field-of-view are respectively depicted by the blue circle and the red lines.

2.3. Efficient joint mesh and texture simplification

As outlined in the problem formulation, a brute force method consists in minimizing the screen distortion over all possible pairs of mesh and texture resolutions satisfying the bitrate constraint. Instead, we propose a method based on marginal analysis, a technique used in optimizing compression algorithms [4]. Using this method, a necessary condition for a pair (M_i, T_j) to be a candidate achieving a good trade-off between mesh and texture resolution is that removing a certain bit budget from either mesh or texture will lead to a similar increase in distortion. The intuition behind the result is simple: If it were not so, some of the bits could be moved from mesh to texture (or vice versa) in order to reduce the distortion. While this result is exact only when assuming independence of the mesh and texture (which is an approximation), the heuristic based on it is quite competitive. Note that it can be used either as a greedy refinement or a greedy pruning technique.

3. JOINT MESH-TEXTURE SIMPLIFICATION

Both meshes and texture are usually treated separately in the current literature. The originality of our work is to take a joint approach to simplify a mesh and its texture: joint texture and mesh coding. The analogy in communications theory is joint source and channel coding. In the previous sections, we outlined methods to reduce/increase the resolution of textures mapped onto surfaces, as well as reduce/increase the resolution of surfaces defined by meshes. In both cases, the algorithms depend on the representation chosen for the texture and mesh, respectively. For textures, cosine bases and wavelet bases were used, and for meshes, an efficient quadtree structure was involved. In all cases, a smooth tradeoff between the complexity of the representation (as measured in bitrate) and quality of the representation (as measured by an l_2 error in world or screen space).

Clearly, when mapping textures onto surfaces, there is a non-trivial interaction between the respective resolution of texture and mesh. For example, visually, a high resolution texture will look pleasing (giving the "illusion" of high resolution) but an underlying coarse surface will lead to a large objective error (since the high resolution texture is not placed at the correct location).

The complex dependence between these effects would require an exhaustive search of the optimal trade-off. Instead, we propose to use an efficient heuristics that has proven to be efficient in other contexts where exhaustive search is too expensive [4]. The method, called marginal analysis, was presented intuitively in the introduction. It searches greedily for the best tradeoff while reducing or increasing the resolution of both mesh and texture. In the pruning mode, a full resolution model with full resolution texture mapped onto it is successively coarsened. At each step, given a target complexity reduction of B bits, the best of reducing the mesh or the texture description by B bits is chosen (or possibly, reducing both by $B/2$). The degradation is simply computed by comparing the signal-to-noise ratio of the pruned versions with respect to the original, either in world space or in screen space (the latter being preferred). While this is a greedy approach, it performs quite close to an optimal exhaustive search in our experiments. Optimality would only be reached in general when the two entities are independent, in which case this becomes standard Lagrangian optimization [5]. In the tree growing mode, the converse operation is performed in order to find the better of increasing the resolution of texture or mesh. Note that tree growing is doubly greedy, and would not even perform optimally in the case of independence, and thus the pruning method is preferred.

The **Marginal analysis algorithm** in tree pruning mode takes as input a given multiresolution pyramid of meshes M_i and texture T_j. The algorithm works as follows:

◁ **Output**: A SET OF PROGRESSIVE PAIRS (M_l, T_m), WITH $l \in \{0, \ldots, N\}$ AND $m \in \{0, \ldots, M\}$

1 **for** ($i = N, j = M$; $i \geq 0$ **and** $i \geq 0$;)

2 **if** ($D(M_{i-1}, T_j) \leq D(M_i, T_{j-1})$)

3 { STORE THE PAIR (M_{i-1}, T_j); $i = i - 1$; }

4 **else** { STORE THE PAIR (M_i, T_{j-1}); $j = j - 1$; }

4. EXPERIMENTAL RESULTS

In this section, we present experimental results on large terrain data and aerial photographs. We compare various rate configurations and the performance in both complexity and distortion are analysed. The terrain model consists of a regular mesh of 256 by 256 vertices and the texture has 1024 by 1024 pixels resolution. To evaluate the quality of the rendered image, we measure the peak signal-to-noise ratio (PSNR) in the screen. The PSNR is a well established objective image error measure, albeit it does not reflect the subjective image quality perceived by a human viewer [6].

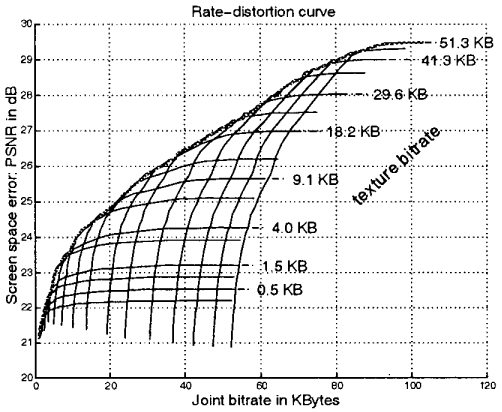


Figure 4: **Experience 1 (exhaustive search)**: The bold red envelope shows the optimal rate-distortion curve for joint mesh and texture simplification found by exhaustive search. Each thin line represents pairs of all used mesh resolutions with the same fixed texture resolution.

We first conduct an exhaustive search experiment for a set of pairs (M_i, T_i) of meshes and textures optimized for a specific viewpoint. The viewer location, corresponding to the one in Figure 1, is such that only 10% of the terrain model is visible. To generate the approximation M_i and T_j, we increase exponentially the rate of the mesh and the texture from 0.1% to 10% of the available vertices and wavelet coefficients. In total, 32 textures and 60 meshes are generated giving a total of 1920 pairs. Figure 4 shows the result of the experiment: For each pair $(M_i, T_j)_{i=0\ldots59, j=0\ldots31}$, the PSNR with respect to the full resolution pair (M_0, T_0) is evaluated in the screen space. Each blue line in the graph corresponds to a fixed texture rate. The rate of the mesh increases on the x-axis, while the y-axis shows the resulting PSNR. Note that only some of the texture rates are indicated in the figure. The optimal path is the outer hull of all curves and is depicted by the red line. Following this path gives the highest PSNR for a given bitrate.

The second experience consists in applying the marginal analysis and comparing the obtained path to the optimal path found by exhaustive search in experience 1. Figure 5 shows, for all combinations of the 60 meshes and 32 textures, their PSNR and model complexity. The mesh complexity increases along the x axis, whereas the texture complexity increases along the y axis.

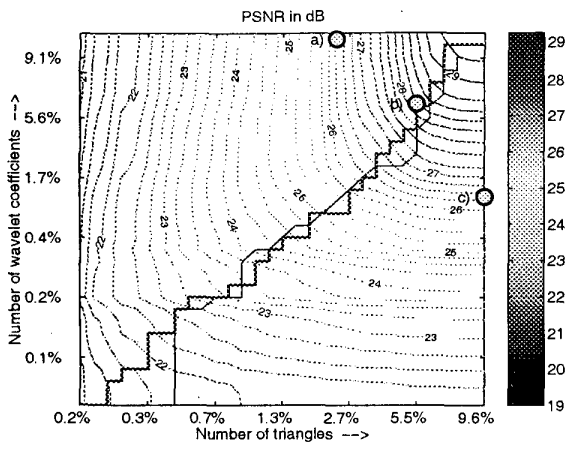
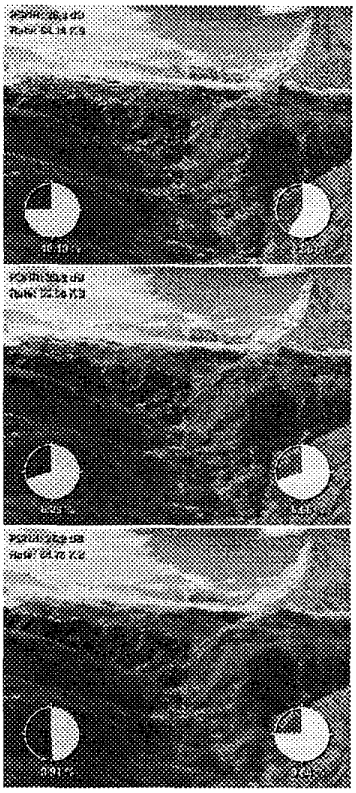


Figure 5: **Experience 2 (marginal analysis): ISO-lines of the PSNR** of all combination between mesh rate and texture (each color line is a line of constant PSNR, as indicated by the right side bar). The thick red line shows the optimal rate-distortion path found by exhaustive search. The thin blue line is the path found by marginal analysis, which gives a good approximation of the optimal path.

The path found by the marginal analysis has a length of 84 frames. To find it, twice as much frames had to be rendered and their screen space error evaluated. Figure 5 shows that the path is close to the optimal path found by exhaustive search. Exhaustive search required to evaluate about 2000 mesh-texture combination, whereas marginal analysis needed only 168 evaluations. We can conclude that marginal analysis is an efficient tool to select the pairs (M_i, T_i) that are close to optimal. In Figure 6 we compare three pairs that have the same joint bitrate to illustrate the possible trade-offs.

5. CONCLUSION

The framework we set up to jointly simplify mesh and textures can be extended to other problems as well. For example, in all progressive transmission cases where refinements have to be sent, it is possible to use the marginal analysis idea to decide which information has to be refined first. Clearly, our marginal analysis algorithm is an efficient way to decide which refinements to send first. In conclusion, we have presented an efficient framework to solve the problem of jointly refining textures and meshes. The ingredients that were used include efficient multiresolution meshes on quadtrees and multiresolution texture mapping based on wavelets. Together with an algorithm based on marginal analysis to search for the best refinement (tree growing) or simplification (tree pruning) of the combined texture and mesh (with respect to an objective screen space error measure, the l_2 norm), we obtained a new combined algorithm. In experimental results, the proposed algorithm performs very close to the optimal exhaustive method, and this at a fraction of the computational cost.



a) Higher texture resolution: texture: **10%**, mesh: 2.4%, PSNR = 26.3dB, joint rate = 64KB.

b) Optimal solution: texture: 6.2%, mesh: 5.6%, PSNR = **28.2dB**, joint rate = 65KB.

c) Higher mesh resolution: texture: 0.9%, mesh: **9.6%**, PSNR = 26.2dB, joint rate = 65KB.

Figure 6: Trade-off between texture and mesh resolution. In a) parts of the image are displaced slightly and in c) the image is blurred. The best compromise b) is in between both. The complexity is given in percent of the full model.

6. REFERENCES

[1] P.S. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," *Carnegie Mellon University Technical Report*, May 1997.

[2] L. Balmelli, S. Ayer, and M. Vetterli, "Efficient algorithms for embedded rendering of terrain models," *Proc. of IEEE Int. Conf. Image Proc. (ICIP)*, vol. 2, pp. 914–918, October 1998.

[3] F. Jordan, S. Horbelt, and T. Ebrahimi, "Photorealistic texture mapped on 3d surface," *Proc. of Int. Conf. Image Proc. (ICIP)*, vol. 2, pp. 390–393, 1997.

[4] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. on Image Proc.*, vol. 6, pp. 677–693, May 1997.

[5] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, April 1993.

[6] H. Rushmeier, B. Rogowitz, and C. Piatko, "Perceptual issues in substituting texture for geometry," *SPIE Proc., Human Vision and Electronic Imag. V, B. Rogowitz and T. Pappas, Eds.*, vol. 3959, pp. 372–383, 2000.