EFFICIENT CONTENT ADAPTIVE MESH REPRESENTATION OF AN IMAGE USING BINARY SPACE PARTITIONS AND SINGULAR VALUE DECOMPOSITION

Michel Sarkis, Oliver Lorscheider and Klaus Diepold

Technische Universität München, Institute for Data Processing, Arcisstr. 21, 80290, Munich, Germany

Emails: michel@tum.de, oliver_lorscheider@web.de, kldi@tum.de

ABSTRACT

Content adaptive mesh generation is an important research area with many applications in image processing and computer vision. The main issue is to represent an image with the pixels that preserve most of the amount of its information. The obtained pixels are then used to generate a mesh that approximates the original image. This work presents a novel iterative method that simultaneously reduces the number of the pixels and generates the mesh approximation of an image. The main idea is to incorporate binary space partitions along with singular value decomposition to cluster the pixels into planes and thus the nodes of the mesh are nothing but the pixels that define each plane. Compared to previous techniques, the proposed method leads to a 30% reduction in the size of the approximating mesh. In addition, the method minimizes the artifacts obtained from the reconstruction of the original image from the approximating mesh.

***Index Terms*—** Image sampling, image coding, clustering methods, mesh generation, singular value decomposition

1. INTRODUCTION

Content adaptive mesh representation of an image is an important research field with many applications in image processing [1–5] and in computer vision [6]. The aim of these techniques is to approximate an image with a polygonal mesh. The approximation mesh is usually constructed in two steps. First, the redundant pixels are removed from the image and only the ones with most of the information, *the significant pixels*, are retained. Then, a mesh is generated from the remaining pixels using a triangulation scheme.

The advantage of such approximation is that it allows the image to have a more compact representation. The approximating or representing mesh usually possesses a lot less number of pixels (referred to as the mesh nodes) than the original image. In addition, the removed pixels from the original image are reconstructed by interpolating the significant pixels using the mesh [4, 5].

The major issue in this research is in the methodology that should be used to find the lowest possible number of significant pixels in an image while preserving its content. As a consequence, the aim becomes equivalent to finding the non-uniform samples of the image [1, 4, 7, 8]. This makes the methods developed for progressive image coding and transmission, as in [7, 9], relevant since non-uniform sampling is used there to derive the coding schemes. These can then be easily followed by a triangulation method to generate the adaptive mesh as done in [4].

The main disadvantage in most of the developed techniques, is the necessity to construct the approximating mesh in two steps. The non-uniform samples have to be first extracted then the mesh is formed upon them. This makes the mesh relatively dependent on the samples found while it should be the other way around. This is justified since reconstructing the missing samples using the mesh highly depends on the ability of each triangle of the mesh to represent and recover the missing samples.

To overcome this issue, a novel iterative method is derived that generates a triangular mesh while reducing the number of points at the same time. The key issue is to combine the binary space partitions (BSP) along with the singular value decomposition (SVD) to model the pixels as intensity planes in order to both cluster the pixels and generate the mesh. The idea behind this work is that each triangle in the mesh can be defined as a plane while a plane can also be uniquely defined by three points. Thus, by defining a plane, a triangle of the mesh is obtained where the nodes of the triangles are the non-uniform samples.

Section 2 summarizes the state of the art algorithms. Section 3 establishes the proposed content adaptive mesh generation method. Section 4 assesses the results and compare the performance of the proposed algorithm with other techniques. Finally, conclusions are drawn in Section 5.

This research is sponsored by the German Research Foundation (DFG) as a part of the SFB 453 project, High-Fidelity Telepresence and Teleaction.

2. RELATED WORK

2.1. Yang's algorithm

In [4], or Yang's algorithm, a Delaunay triangulation based content adaptive mesh representation algorithm was derived in which the nodes of the mesh are the non-uniform adaptive samples of the image. A pixel is chosen to be a non-uniform sample if the second directional derivative is significant. Let $G(x, y)$ denote the largest magnitude of the second derivative of each image pixel $I(x, y)$:

$$\mathbf{G}(\mathbf{x}, \mathbf{y}) =_{\theta \in [\mathbf{0}, \mathbf{2}\pi]}^{\max} \left| \mathbf{I}_{\theta}^{''}(\mathbf{x}, \mathbf{y}) \right| \tag{1}$$

where $\theta \in [0, 2\pi]$ is the direction of the second derivation of the image function. All pixels where $G(x, y)$ is above a predefined threshold are put into a feature map. Then a modified version of the classical Floyd-Steinberg algorithm is used to diffuse the error of a pixel to its four neighbors. As experiments showed, Yang's algorithm worked better if the serpentine raster order was used instead of the standard raster order of the diffusion algorithm. The reason for the improvement is that the diffused errors of the pixels are propagated in a more balanced manner with its neighbors.

2.2. Ramponi's algorithm

In [7], or Ramponi's algorithm, the main issue was to derive non-uniform sampling technique that can be used for progressive image coding. Given an image, the initial estimate of the reduced set of pixels in the image are determined by computing the skewness of the pixels. The skewness s of a pixel evaluated on a $m \times n$ mask K is defined as:

$$s\left(x,y\right) = \frac{1}{m \times n} \sum_{k=1}^{m} \sum_{l=1}^{n} \left(I\left(k,l\right) - \mu\left(k,l\right)\right)^{3}, \tag{2}$$

where μ is the mean of the intensity values of the pixels in K. To make this measure insensitive to the dynamic range of the image, it is better to normalize (2) as:

$$\bar{s}\left(x,y\right) = \frac{\left|s\left(x,y\right)\right|}{\max\left(\left|s\left(x,y\right)\right|\right)}, \tag{3}$$

where $\bar{s}$ is the normalized skewness and the denominator is the maximum skewness obtained in the image. Then, a threshold is defined and all the pixels whose normalized skewness $\bar{s}\left(x,y\right)$ is higher than the threshold are saved in a feature image F.

To further reduce the number of pixels and obtain a more compact representation of the image, the feature image is decimated by defining a forbidden circular area of radius F around each pixel in it. Consequently, if a pixel belongs to the area of another pixel, it is deleted from F.

To generate the adaptive mesh, Ramponi's algorithm was followed in this work by a Delaunay triangulation step to generate a primary mesh. Then, new pixels were recursively added to the mesh at the locations where the error, see Section 3.1, of the reconstructed pixels is higher than a predefined threshold as was proposed in [6].

3. THE PROPOSED CONTENT ADAPTIVE MESH REPRESENTATION METHOD

Given an input image, the main objective of this work is to construct an adaptive mesh that approximates the original image with the lowest number of non-uniform samples. The proposed method is based on the Binary Space Partitioning (BSP) principle [10]. BSP clusters a data set recursively. In each step, one cluster is divided into two sub-clusters if some predefined criteria are not met. This is the main motivation behind using BSP to represent the image with a content adaptive mesh.

The criteria defined in this work to subdivide a triangle, are the maximum number of points that the triangle can hold and the reconstruction error of the pixel intensities from the triangle. The flowchart of the BSP based algorithm is illustrated in Fig. 1 and the rest of this section is dedicated to explain it. For simplicity, the proposed technique is derived for gray-scale images. Its extension to color images is easily obtained.

3.1. Error measurement

To give a qualitative statement about the mesh approximation, it is necessary to have a quantity that measures the error between the luminance values $I(x,y)$ of the original image and those of the approximated image $\hat{I}(x,y)$. A well known image quality measure is the peak signal to noise ratio (PSNR) [11]. The PSNR for a greyscale $W \times H$ image where the intensity values vary between 0 and 255 is:

$$\mathrm{PSNR} = 10 \log\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{4}$$

where MSE is the mean squared error defined by:

$$\mathrm{MSE} = \frac{1}{W \cdot H} \sum_{k=0}^{W-1} \sum_{l=0}^{H-1} \left(I\left(x,y\right) - \hat{I}\left(x,y\right)\right)^{2}. \tag{5}$$

A typical value for the PSNR to obtain a good image reconstruction is between 30 and 40 dB. One might also think of using the MSE as a measure instead of the PSNR operation as in [1, 4] since both are equivalent.

3.2. Modeling the Pixels with Intensity Planes

The points of an image describe a 3D space represented by the coordinates of the pixel in the image and the corresponding intensity. When constructing the mesh, a triangle T of the mesh should represent the pixels that it covers as accurately as possible. Each triangle is formed by three points which are parts of the nodes or the vertices of the mesh. In a similar manner, a plane can also be defined by the three vertices of the triangle.

Let $V_i(x_i, y_i, I_i)$ with $i = 1, 2, 3$ be the three vertices of T. The plane P describing these vertices is defined using the normal equation as:

$$\overrightarrow{n} \cdot P(x,y,I) + d = 0, \tag{6}$$

where $P(x,y,I)$ is a point lying on the plane, d is a real constant and $\overrightarrow{n}$ is the normal vector of the plane $\overrightarrow{n} = \left[\begin{smallmatrix} a \\ b \\ c \end{smallmatrix}\right]$. The normal vector can be computed with the three vertices $V_i(x_i, y_i, I_i)$ as the cross product of any two edges of the triangle:

$$\overrightarrow{n} = \left[V_2 - V_1\right] \times \left[V_3 - V_1\right]. \tag{7}$$

From (7), the equation of the plane can be then directly obtained:

$$\hat{I} = -\frac{ax + by + d}{c}, \tag{8}$$

where $\hat{I}$ is the modeled intensity of all the pixels lying inside the triangle T. Using this equation, it is possible now to reconstruct the intensity of each pixel that lies inside T to compute the PSNR. Consequently, one can easily check now how does each triangle represent the pixels that lies within by simple computation of the PSNR described in (4). If the PSNR of the reconstructed intensities of T is lower than a predefined threshold ϵ, e.g. 30 dB, then T should be further decomposed into two smaller triangles. This step is repeated until ϵ is satisfied across the image.

What remains is to determine the pixels that lies inside T. On the one hand, the points inside the triangle have in common that they lie on the same side of each of the triangle's edges. On the other hand, the center $w(x,y)$ of T is always lying inside the triangle. Thus, every point in T should be lying on the common side of all triangle's edges as well. In order to verify this property, we need to recall the 2D normal equation of a triangle's edge:

$$\overrightarrow{n_{edge}} \cdot V(x,y) + d = 0. \tag{9}$$

The vector $\overrightarrow{n_{edge}}$ is the normal to the edge defined as:

$$\overrightarrow{n_{edge}} = \begin{bmatrix} y_i - y_j \\ x_j - x_i \end{bmatrix}. \tag{10}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the vertices V_i and V_j of the triangle's T edge in the image [12].

A point $P(x,y)$ is defined to be lying in the triangle if the following equation is fulfilled to the three edges of the triangle simultaneously:

$$sign(\overrightarrow{n_{edge}} \cdot P(x,y)) = sign(\overrightarrow{n_{edge}} \cdot w(x,y)), \tag{11}$$

where $sign$ is the sign operator. This equation suggests that a point P lies inside T if the sign of dot product of P with the normal edge $\overrightarrow{n_{edge}}$ is the same as that of the center of the triangle w for each edge of T [12].

3.3. Determining the Optimal Partition Line

As previously said, if the PSNR does not satisfy ϵ or the number of points is larger than the predefined maximum triangle's size, a triangle T in the mesh has to be divided into two new ones. In order to let BSP decide how a triangle should be divided, the Singular Value Decomposition (SVD) is employed [13].

To accomplish this issue, the points $P(x, y, I)$ of the triangle have to be arranged into a measurement matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} x_1 & x_2 & x_3 \dots & x_n \\ y_1 & y_2 & y_3 \dots & y_n \\ I_1 & I_2 & I_3 \dots & I_n \end{bmatrix}. \tag{12}$$

$\mathbf{K}$ has then to be centered, i.e. the mean is subtracted, as in any SVD based clustering algorithm to obtain the centered measurement matrix $\tilde{\mathbf{K}}$. Then, the SVD of $\tilde{\mathbf{K}}$ is computed as:

$$\tilde{\mathbf{K}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \tag{13}$$

The columns of $\mathbf{U}$ show the directions of the largest variances of the measurement matrix $\tilde{\mathbf{K}}$. The first column of $\mathbf{U}$, i.e. $\mathbf{u} = (x_u, y_u)$, shows the direction of largest variance in $\tilde{\mathbf{K}}$. It can also be shown that $\mathbf{u}$ is perpendicular to the sought partition line [13]. Therefore, the vector of the partition line is nothing but:

$$\vec{l} = \begin{bmatrix} y_u \\ -x_u \end{bmatrix}. \tag{14}$$

Using $\vec{l}$, it is possible to compute the new vertex, or the non-uniform sample, as the intersection of the partition line and the triangle's edge. The line equation of a triangle's edge is given by:

$$V_1 + \alpha \cdot (V_2 - V_1), \tag{15}$$

and that of the partition line starts in the opposite vertex of the triangle's edge:

$$V_3 + \beta \cdot \vec{l}. \tag{16}$$

By combining (15) and (16), we obtain the following equation:

$$\begin{bmatrix} (V_2 - V_1) & \vec{l} \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = V_3 - V_1, \tag{17}$$

were α and β has to be solved. Then, the coordinates of the new vertex or non-uniform sample is determined by solving (15). In the case if α is equal to 0 or 1, Equation 15 becomes ill conditioned and the new vertex cannot be computed anymore. To overcome this deficiency, the subdivision of the triangle is conducted by placing the new vertex in the middle of the triangle's longest edge. Note that the constructed mesh in the proposed algorithm does not have necessarily to fulfill the Delaunay condition, see [14] for more information about Delaunay triangulation.

4. RESULTS AND COMPARISON

Yang's algorithm has a better performance in term of quality and compression when compared to other state of the art content adaptive meshing techniques [4]. Thus, it will be used with the comparisons done with the proposed method. In addition, the modified version of Ramponi's algorithm explained in Section 2.2 will also be analyzed since no comparisons have been made with it. All computations are performed by an Intel Pentium M Core Solo Processor (2,0 Ghz, 2MB L2 Cache, 1024 MB RAM) using Matlab.

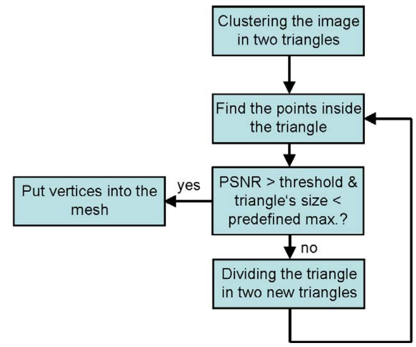

Fig. 1. Flow chart of the proposed method.

The tests consist of measuring both the performance of the algorithms and the quality of the results obtained. The performance will be tested by measuring both the mesh size, i.e. the number of the triangles, and compression ratio given by:

$$compression\ ratio = \frac{original\ image\ size}{number\ of\ feature\ points}, \tag{18}$$

while the quality will be judged by visual comparison of the reconstruction of the images from the mesh. In the proposed algorithm, the mesh is reconstructed by computing the intensities of the pixels using (8). In order to construct the intensity values in both Yang's and Ramponi's algorithm, the weighted sum of the intensities of the vertices will be used. The weights assigned to each of the vertices are the barycentric coordinates of the pixel in the corresponding triangle [14]. This choice is made since it is equivalent to the intensity reconstruction of the proposed method in (8).

Fig.2 shows the results of the compression ratio and number of triangles obtained with the different algorithms when applied to the Lena image. The results are obtained while varying the PSNR between 25 and 40 dB. As can be seen, the proposed method achieves a better compression ratio than both of the other techniques especially at low values of the PSNR. As the PSNR increases, i.e 40 dB, the improvement of the proposed method to the other becomes smaller. However, by looking at the number of the triangles of the mesh, it can be seen that the amelioration is tremendous. The proposed technique results in 30% less number of triangles than the other techniques even at 40 dB PSNR which corresponds to a high quality of the reconstruction.

Moreover, the proposed algorithm has a stable transition while varying the PSNR. It has no abrupt jumps as in Ramponi's algorithm since it considers the quality of the reconstruction of each pixels when constructing the mesh while Ramponi's method seeks for the non-uniform samples by simple filtering operations. Such stability is also noticed in Yang's technique since it propagates the error using a diffusion algorithm in order to choose the non-uniform samples.

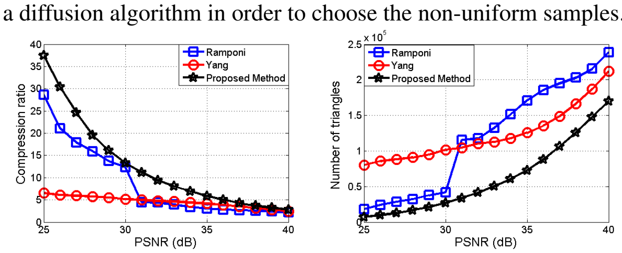

Fig. 2. Results of the algorithms on the Lena image. Left: Compression ratio. Right: Number of the triangles in the constructed mesh.

Figures 3 and 4 show the original image of the Lena along with its reconstruction using all of the methods and by setting the PSNR at 30 dB. By comparing the reconstructed images, it can be directly seen that the proposed method results in less artifacts than the two other algorithms. This can be noticed at the wooden bar in the left and in the mirror for example. In addition, the proposed method preserves the edges. By looking at Ramponi's method, this cannot be noticed. Examples are the hat in the mirror and the wooden bar in the background. With Yang's method, some artifacts can also be noticed. Although the edges are sharply represented, but if one is found in an area with low intensity variation, the reconstructed edge become blurred and noisy. For convenience, the content adaptive mesh representations of the images resulting from the proposed method and Ramponi's algorithm are shown in Fig. 5.

Fig. 3. Left: The Lena image. Right: The reconstructed lena image using the proposed method with 30 dB PSNR.

Fig. 4. Left: The reconstructed lena image using Ramponi's method with 30 dB PSNR. Right: The reconstructed lena image using Yang's method with 30 dB PSNR.

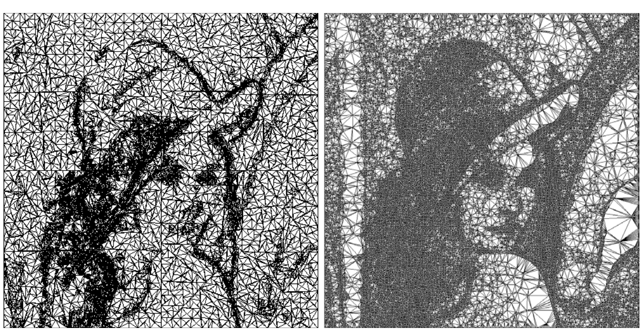

Fig. 5. The content adaptive mesh representation of the Lena image at 30 dB PSNR. Left: Outcome of the proposed algorithm. Right: Outcome of Ramponi's algorithm.

5. CONCLUSION

This work presents a new content adaptive meshing technique based on the combination of BSP and SVD to represent the images. The main idea is to model the intensity variation of the pixels inside a triangle by a planar equation since both are specified using three points, which are defined as the non-uniform samples of the image. This makes the algorithm locate the non-uniform samples of the image by determining the triangles of the mesh that best describe the points lying within. Results show that the proposed technique has a better compression ratio than other state of the art techniques and the obtained mesh size is about 30% less even for high PSNR values. In addition, visual results show that the main technique reduces the artifacts in reconstructing the images from the mesh due to adaptivity gained from the application of both BSP and SVD.

6. REFERENCES

[1] M. A. García, B. X. Vintimilla, and A. D. Sappa, "Efficient approximation of gray-scale images through bounded error triangular meshes," in *International Conference on Image Processing (ICIP)*, October 1999, pp. 168–172.

[2] M.A. García and B.X. Vintimilla, "Acceleration of filtering and enhancement operations through geometric processing of gray-level images," in *International conference on Image Processing (ICIP)*, September 2000, pp. 97–100.

[3] Y. Yang, J. G. Brankov, and M. N. Wernick, "Content-adaptive mesh modeling for fully-3D tomographic image reconstruction," in *IEEE International Conference on Image Processing (ICIP)*, September 2002, pp. 621–624.

[4] Y. Yang, M. N. Wernick, and J. G. Brankov, "A fast approach for accurate content-adaptive mesh generation," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 866–880, August 2003.

[5] L. Demaret, N. Dyn, and A. Iske, "Image compression by linear splines over adaptive triangulations," *Signal Processing Journal*, vol. 86, no. 7, pp. 1604–1616, July 2006.

[6] M. Sarkis and K. Diepold, "A fast solution to the approximation of 3D scattered point data from stereo images using triangular meshes," in *IEEE RAS Conference on Humanoid Robots*, Nov. 2007.

[7] G. Ramponi and S. Carrato, "An adaptive sampling algorithm and its application to image coding," *Image and Vision Computing*, vol. 19, no. 7, pp. 451–460, May 2001.

[8] F. Marvasti, Ed., *Nonuniform Sampling*, chapter 6, Springer, first edition, 2001.

[9] K.-L Hung and C.-C. Chang, "New irregular sampling coding method for transmitting images progressively," in *IEE Proceedings on Vision, Image and Signal Processing*, February 2003, pp. 44–50.

[10] H. Radha, M. Vetterli, and R. Leonardi, "Image Compression Using Binary Space Partitioning Trees," *IEEE Transactions on Image Processing*, vol. 5, no. 12, pp. 1610–1624, 1996.

[11] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, SPIE Press, first edition, 1991.

[12] J. A. Vince, *Vector Analysis for Computer Graphics*, Springer, first edition, 2007.

[13] S.M. Savaresi, D. Boley, S. Bittanti, and G. Gazzaniga, "Cluster selection in divisive clustering algorithms," in *The Second SIAM International Conference on Data Mining*, April 2002, pp. 299–314.

[14] F. Nielsen, *Visual Computing: Geometry, Graphics, and Vision*, Charles River Media, first edition, 2005.