

Towards an Optimal Texture Reconstruction

Zsolt Tóth¹

Department of Computer Graphics and Image Processing
Faculty of Mathematics, Physics and Informatics, Comenius University
Bratislava / Slovakia

Abstract

Visually pleasant texture reconstruction has important role in computer graphics. In this paper we explore the applicability of triangulations for texture reconstruction. Two new algorithms are introduced for generation of data-dependent triangulation. The new deterministic algorithm entitled as image partitioning algorithm (IPA) shifts this reconstruction method closer to real usage. We present a new modification of the optimization technique simulated annealing with generalized look-ahead process (SALA). Also a new way of utilization of color information is presented, to achieve qualitative course of reconstruction of color images. Results show both theoretical and practical superiority over another methods. This work is a part of the APVT project Virtual Bratislava.

Keywords: image reconstruction, data-dependent triangulation, minimum weight triangulation

1 Introduction

Precise reconstruction of images is important for operations on images, where it is relevant to maintain the structure of the image, with minimum amount of artifacts. Such operations are scaling, rotation, warping etc... In real life the perceived information has continuous character. Digitizing this information we get discrete model. Our goal is to recover the continuous model from these discrete samples. It is impossible to design ideal method, but it is necessary to improve the visual quality.

For image reconstruction purposes there exist relatively many approaches. The simplest method is the nearest neighbour interpolation. More successful methods are bilinear interpolation and bicubic interpolation. These methods are using the tensor product of 1D basis functions (linear interpolation and cubic spline). The reconstruction error is significant in direction, which differs from the directions of the applied basis functions. Obviously these methods can be found with the most commercial software tools, because of their high performance. Reconstructing methods, like the nearest neighbour and bilinear interpolation are part of hardware acceleration supporting systems like OpenGL or DirectX. Image processing has also numerous

filtration methods for image reconstruction (Mitchells interpolation, Gauss or Parzen windowed sinc function etc.).

The main research motivation was the technological development and the applications involved. They require improving the quality of the reconstruction methods. For these reasons arose methods, which at first reconstructed visually important areas (mainly edges) with advanced methods, and for the rest are applied the previously described methods. The next step was the application of an edge oriented interpolation for the whole image. Among these methods there belongs the image reconstruction with the help of data dependent triangulation. The root of this idea is only a few years old. The data-dependent triangulation [7] is an intelligent reconstruction method, and it generates the more pleasant results from the described approaches - Figure 1. In this paper we are dealing with new improvements of this idea.

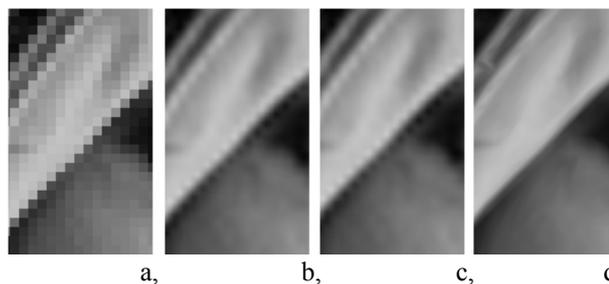


Figure 1: a, nearest neighbour- b, bilinear- c, bicubic- interpolation, d, data-dependent triangulation at 600% magnification

The paper is structured as follows. Section 2 gives a short mathematical background and overview of research that has been dealt with data dependent triangulation methods. A guide for the special setup of data-dependent triangulation for the texture reconstruction can be found in Section 3. A correct use of color information is presented in Section 4. We introduce a method to solve a task to shift this approach closer to real usage – Section 5. On the other hand, a theoretically interesting modification of existing algorithm is introduced in Section 6. Section 7 deals with implementation details. Also we consider the usability of these reconstruction methods in virtual city development. In Section 8 we discuss the results and make some conclusions. Section 9 provides some possible ways of the future work.

¹ zsolt.toth@st.fmph.uniba.sk

2 Background and Related Work

First we introduce some necessary definitions.

Def. 2.1: Given a set of distinct (and not all collinear) vertices $\mathbf{V} = \{(x_i, y_i); i = 1, \dots, n\}$ in the plane. Triangulation of the set \mathbf{V} is called a set of triangles $T(\mathbf{V})$ which satisfy the following properties:

- Each vertex of an arbitrary triangle from $T(\mathbf{V})$ is from set \mathbf{V} and each element from \mathbf{V} is a vertex of at least one triangle in $T(\mathbf{V})$.
- Each edge from triangulation contains exactly two elements from \mathbf{V} .
- Union of all triangles from triangulation $T(\mathbf{V})$ is the convex hull of set \mathbf{V} .
- Intersection of two arbitrary triangles from triangulation $T(\mathbf{V})$ is an empty set or their common vertex or their common edge.

Def. 2.2: The total weight of the triangulation $T(\mathbf{V})$ is a sum of the length (cost) of all its edges:

$$w(T(\mathbf{V})) = \sum_{e \in T(\mathbf{V})} \|c(e)\|,$$

where $c(e)$ denotes the cost function of edge e .

In this work we use l_1 norm (what means the sum of the absolute values), but there exist other possibilities, too [7].

Def. 2.3: The minimum weight triangulation (MWT) of set \mathbf{V} is a triangulation whose weight is minimal between all possible triangulations:

$$w(MWT(\mathbf{V})) = \sum_{e \in MWT(\mathbf{V})} \|c(e)\| = \min\{w(T(\mathbf{V}))\} \quad \forall T(\mathbf{V})$$

Def. 2.4: Data-dependent triangulation (DDT) of set $\mathbf{V} = \{(x_i, y_i); i = 1, \dots, n\}$ in E^2 (Euclidian 2-dimensional space) is a triangulation where the topology of triangulation is dependent on function $f(x, y) = \{z_i = f(x_i, y_i); i = 1, \dots, n\}$.

The DDT allows to construct triangulations in 2D based on properties, which are represented as additional dimension. It also represents a triangulation in 2.5D. It depends on the application which form is more suitable (image reconstruction – 2D, terrain reconstruction – 2.5D). We restrict our considerations to the so-called *edge-based data-dependent triangulations*. There are other possibilities e.g. *vertex-based data-dependent triangulations* [4]. The dependency of the edges from the function $f(x, y)$ can be arbitrary, but we limit our consideration with the following: each interior edge is dependent on the four vertices forming the quadrilateral (created from two adjacent triangles), which contain the edge as a diagonal - Figure 2.

Classical triangulations usually avoid the generation of long tiny triangles. These are suitable for reconstruction of areas where $f(x, y)$ has large derivatives in one direction if compared to other directions. In image reconstruction that areas contain edges in the image, and our primary goal is to reconstruct them correctly - Figure 2.

To solve the MWT problem we need to investigate an exponential number of triangulations, what is extremely time consuming for large data sets. Some properties of MWT have been observed or proved, but the NP-completeness of MWT is still an open problem of the computational geometry. The detailed survey of the

MWT problem was done by Ferko [8]. The number of the existing heuristics, approximations and known subgraphs [9] of this problem for planar triangulations is reduced with the data dependent approach. This is due to special cost assign of the edges.

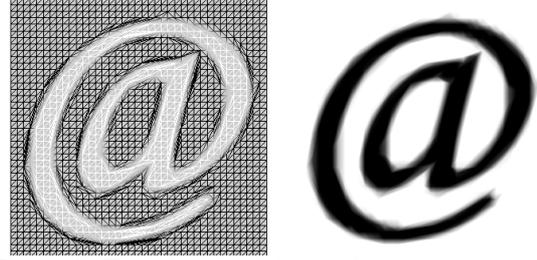


Figure 2: The edge - and the result - representation of the DDT triangulation at 800% magnification

Every interior edge is an intersection of two adjacent triangles, these triangles form a quadrilateral. If the quadrilateral is convex and non degenerate, then the edge can be replaced by another diagonal. This operation is called *edge swap*. The diagonal edge of the quadrilateral with the mentioned properties is called locally optimal if the quadrilateral is optimally triangulated. If it is concave or degenerate then it is also locally optimal. We must note that the local optimality of the edge at DDT depends on the sum of the five edges, depicted on Figure 3. With the edge swapping operation works the most popular data dependent approach, the Lawson optimization [7] (also called locally optimization procedure). It is similar to the Delaunay triangulation for planar triangulations [2], [14]. If the cost function satisfies the well known empty circle property, then they are identical.

The *look-ahead* expansion of the local optimality [24] of edges scans the triangulation structure deeper and triangulates (with MWT property) the area bounded with eight vertices - Figure 3.

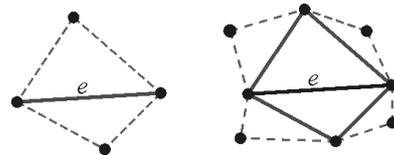


Figure 3: Local optimality and its look-ahead expansion

We generalize the look-ahead principle. Local optimality means in this sense look-ahead of zero degree, the classical look-ahead is marked as a first degree. With enlarging the considered area (the dotted line on Figure 3 means the enlargement at the actual level) we get the look-ahead of second degree etc.

Pixel level data-dependent triangulation was introduced by Su and Willis [19], [20]. This very fast triangulation attain only slightly better results than bilinear or bicubic interpolation. The embedding of the image reconstruction problem into five dimensional space was done by Weimer et. al. [23]. Classical data-dependent methods do better job.

Besides deterministic methods, there are stochastic optimization techniques usable for constructing DDT heuristics of MWT. The simulated annealing approach was described by Schumaker in [18]. Its application for image compressing was done by Kreylos et. al. [15]. The

genetic optimization is another stochastic process, Kolingerova et. al. introduced a data-dependent version [12], recent results can be found in [13].

Mesh refinement and decimation techniques also use DDT approaches. Garland and Heckbert combined the DDT with greedy strategy for digital elevation model reconstructing and simplifying [10].

3 The Main Factors of DDT

This section handles the description of the DDT process factors for image reconstruction purposes.

3.1 The Initial Triangulation of the Domain

The common feature of the majority of existing algorithms is the necessity of initial triangulation. The reason of this is that certainly small number of efficient tools exists for creating DDT approximations and heuristics of MWT. Concretely, the edge swapping and its look-ahead expansion. These operations request an initial triangulation, on which it can operate. In our case the data are organized into the Cartesian grid – the pixel center placement. Our task is to triangulate the domain of the function $f(x, y)$, so generate a classical planar triangulation. The next step is pulling this triangulation into 2.5D, to fit the data vector. The most suitable choice to triangulate the domain of $f(x, y)$ is the regular triangulation – Figure 4, it is both MWT and most equiangular (Delaunay) triangulation for the Cartesian grid data sets.

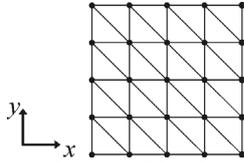


Figure 4: The triangulation of the domain with regularly placed points

Because we know the data organization, the creation of the initial triangulation requires only $O(n)$ time. The existing algorithms mostly have iterative character, so the speed of convergence depends on how far is the initial triangulation from the final stage (what is usually only a local and not global optimum). The regular triangulation guarantees fast convergence speed.

3.2 Data Mining from Color Images

The objective now is to assign costs to edges of the DDT triangulation so, that the MWT approximation of the set results in a suitable reconstruction. In the context of image reconstruction this means low cost for edges, which represents visually relevant parts. The existing approaches works with one data vector, obtained from the color information of the input image. Usually digital images are represented with three components (due to the trichromatic theory of color) e.g.: RGB, YUV. We will restrict to RGB, as other color spaces can be converted into that space.

The simplest progresses deal with linear intensity, what can be easily computed:

$$I = 0.21267R + 0.71516G + 0.07217B, \\ c(e) = c_I(e).$$

In this way the chromatic information is skipped. This causes errors when reconstructing small details. Also there can occur situations, when neighbouring localities have the same intensity but different chroma.

Yu et al [24] presented a model, which uses the chromatic information. The cost for a concrete edge is computed independently for each color component, and then it is combined according to the next formula:

$c(e) = 0.21267c_R(e) + 0.71516c_G(e) + 0.07217c_B(e)$, where $c_N(e)$ means the cost of edge e with respect to data vector N . Authors mentioned that the quality of result is slightly increased, while the changes compared to previous method are minimal.

3.3 Cost Functions

The role of the edges cost functions was mentioned, and now we describe some of these methods. We consider only a small group of these functions, for image reconstructing at work [24]. Generally does not exist any cost function, which creates visually pleasant results for all data sets.

The situation is illustrated on Figure 5, where T_1 and T_2 are triangles at the DDT generated piecewise linear surface. They contain a common edge e . The linear polynomials $P_1(x,y)$ and $P_2(x,y)$ are the planes, which contain the triangles T_1 and T_2 .

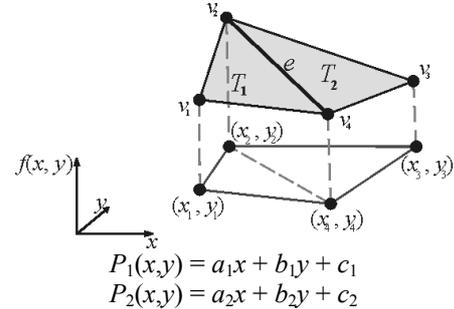


Figure 5: Illustration for the calculation of cost functions

These polynomials are used for the description of several geometrically based cost functions:

ABN (angle between normals) measures the cosine of the angle between the normals of the planes which contains triangles T_1 and T_2 :

$$c^{ABN}(e) = n_1 \cdot n_2,$$

where n_1 and n_2 are normals of triangles T_1 and T_2 .

JND (jump in normal derivatives) is a measure of the change in normals of derivatives P_1 and P_2 across the edge e :

$$c^{JND}(e) = \|n_x(a_1 - a_2) + n_y(b_1 - b_2)\|,$$

where $(n_y, n_x)^T$ is a unit vector in R^2 (xy plane), orthogonal to the direction of e .

DLP (deviations from linear polynomials) measures the errors between linear polynomials P_1 and P_2 evaluated at vertices v_3 and v_1 :

$$c^{DLP}(e) = \|h\|,$$

$$h = \begin{bmatrix} |P_1(x_3, y_3) - f(x_3, y_3)| \\ |P_2(x_1, y_1) - f(x_1, y_1)| \end{bmatrix}.$$

DP (distances from planes) measures the distance between the planes P_1, P_2 and the vertices v_3, v_1 :

$$c^{DP}(e) = \|g\|, \quad g = \begin{bmatrix} \text{dist}(P_1, w_2) \\ \text{dist}(P_2, w_1) \end{bmatrix},$$

$$\text{dist}(P_j, w_k) = \frac{|P_j(x_k, y_k) - f(x_k, y_k)|}{(a_j^2 + b_j^2 + 1)^{1/2}}.$$

SCF (Sederbergs cost function) uses the projection of the T_1 and T_2 triangle normals into plane xy :

$$c^{SCF}(e) = \|\nabla P_1\| \cdot \|\nabla P_2\| - \nabla P_1 \cdot \nabla P_2, \\ \|\nabla P_i\| = (a_i^2 + b_i^2)^{1/2}.$$

The described cost functions except SCF are sensitive for the scaling of the intensity range. For image reconstruction purposes the best results were produced in tests with SCF [24]. This corresponds to our observations - that the SCF reconstructs better the edge areas than other methods. The demonstration of cost functions is shown at Figure 6.

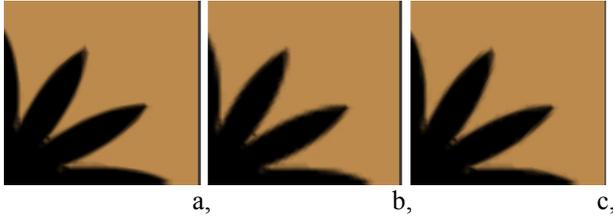


Figure 6: The SCF, the ABN and the DP cost function generated results

Several other approaches exist, which assign the cost to components of triangulation (vertices, edges, triangles, etc.) on the base of various geometrically based and not geometrically based criteria. More information about otherwise constructed cost functions can be found in [1], [4], [7].

4 Correct Usage of the Color

Existing principles of data mining from color information are not completely correct, because these cause some information loss. The source of this is that the color information resampling is done on the base of one triangulation. Computing separate triangulation for each color component seems to be an ideal approach. In [24] authors noticed that this can lead to color bleeding (they used the RGB color model). Our approach is based on the usage of perceptually uniform color spaces, trying to avoid or minimize the color bleeding effect.

Def. : [17] A perceptually uniform (or linear) color model is one in which the perceptual distance between two colors is proportional to the Euclidean distance between their positions in the color space.

Color models can be categorized from many aspects i.e.: device derived, intuitive and perceptually uniform. Our selection is the CIE $L^*u^*v^*$ color model, a device independent, perceptually uniform, and not intuitive color model (can be easily converted into intuitive form).

We note that the perceptual linearity of CIE $L^*u^*v^*$ is not perfect, in the case of large color differences this feature breaks down. Fully perceptually linear color space had not been developed, and the color conversion from RGB and backward conversion from our selection is relatively easy. Also this color space better describes the human visual system sensitivity than RGB.

The suggested method works independently with layers L^*, u^*, v^* , and the bilinear interpolation at resampling also uses this color space - Figure 7.

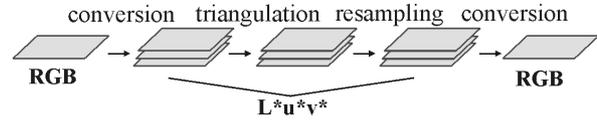


Figure 7: Sequence of correct reconstruction steps

Using CIE $L^*u^*v^*$ for bilinear interpolation, we lose the support of hardware acceleration. On the other hand, the origins of the reconstruction errors are eliminated (for example artifacts at Gourard shading are result of the usage of the RGB model). The outlined method uses the color information without losses, and works in natural way with triangulated data for the human visual system.

5 Image Partitioning

Our new deterministic algorithm tries to combine the advantages of the existing methods. Our main goal was to achieve a process, which allows real usage of data dependent triangulations and does not reduce the quality of the reconstruction like the pixel level data-dependent triangulation.

In the existing triangulations the Euclidean length of edges in 2D projection is only slightly dispersed. These triangulations have rather local character, the occurrence of extremely long edges is exceptional. On this notion is based the conception to divide the picture into blocks - rectangles, each vertex must lie exactly in one block. Inside these blocks there is evaluated the approximation of MWT, or from all the possibilities is selected the MWT. The problem is to interconnect the neighbouring blocks. The initial structure of this interconnection is the regular triangulation - Figure 8.

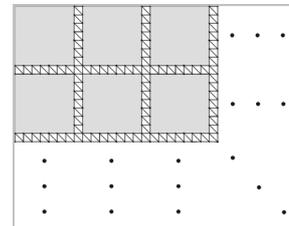


Figure 8: Block segmentation of the image

The quest of the interconnection method is to exude visually pleasant reconstruction and to have a low computational cost. For this intention is used one look-ahead transition (of first degree) across the set V^A - set of active edges. Set V^A consists from edges, which are not part of any block at the initial structure, and it has different orientation as coordinate axes. In other words, we are dealing with the diagonals of the squares, which

are used for the interconnection. Let's consider a concrete edge from the set of active edges – e illustrated in Figure 9. Look-ahead optimization of first degree of the edge e does not change the other elements from V^A , and interconnects the neighbouring blocks.

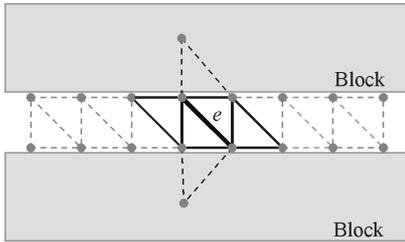


Figure 9: Look-ahead of first degree of edge e

The pseudocode of the algorithm:

Image partitioning algorithm – IPA

Input: digital image
Output: triangulation

1. {
2. divide image into blocks;
3. triangulate each block separately;
4. create the initial triangulation between blocks;
5. build up the list V^A ;
6. **for**(all edges from the set V^A)
7. applicate look-ahead of first degree;
8. }

The computational cost of the IPA mainly depends on the computational cost of the optimization technique of processing individual blocks and on the size selection of the blocks. The computational cost of the initial structure initialization is $O(n)$. Usage of smaller blocks indicates worse quality, big blocks cause performance drop down.

The above described method is suitable for parallel computations. The individual blocks can be computed independently, what seems to be a great advantage. In this case it is necessary to „stitch together” the added block with its neighbouring blocks, for what is used the described method of interconnection. IPA can find a wide range of usage if the image is divided into blocks, which can be triangulated in real time. Individual blocks can be triangulated at various ways, for example we can set their priority with the help of edge detectors.

The image partitioning algorithm is applicable only for data sets, where the projections of the data into xy plane are organized into Cartesian mesh. In other cases the fast interconnection of the blocks is lost. Another practical place of the image partitioning algorithm applicability is the recovery of continuous model from digital elevation models.

6 Simulated Annealing with Look-Ahead

In this section there, is introduced the new modification of the optimization technique called simulated annealing [6], [11]. Our basic idea of the modification of this

stochastic process lies on the usage of the generalized look-ahead process, with the objective to achieve better reconstruction.

Classical simulated annealing considers to proceed as follows: if the edge is locally optimal, then with some probability the edge is swapped. This causes the increase of the triangulation weight, but avoids getting stuck in local minima. The modification is concerning the case when this swap does not happen. Then for the tested edge is it applied the look-ahead searching of optimality. In pseudocode:

Simulated annealing with look-ahead - SALA

Input: initial triangulation
Output: simulated annealing with look-ahead

1. {
2. **for**($k=1$; $k \leq ntemp$; $k++$){
3. $t_A = r^k t_0$;
4. **for**($l=1$; $l \leq nlimit$; $l++$)
5. **while**(number of good swaps $\leq glimit$){
6. select random edge e from triangulation;
7. **if**(exists alternate diagonal of edge e){
8. **if**(strictly convex quadrilateral, which contains e is not optimally triangulated){
9. swap diagonals;
10. }
11. **else**{
12. random choice of number φ
13. $0 \leq \varphi \leq 1$;
14. **if**($\varphi \leq e^{-c(e)/t_A}$)
15. swap diagonals;
15. **else**
16. look-ahead optimization of edge e ;
17. }
18. }
19. }
20. }
21. }

We describe the meaning of parameters:

- $ntemp$ – the number of iterations (in how many steps is the temperature of the annealing process reduced), standard value is around 20.
- $nlimit, glimit$ – number of swaps to be attempted for each temperature, number of good swaps allowed for each temperature. Obviously, these values are choosed as fixed multiplies of the number of edges, often is adjusted by a factor 5 or 10.
- t_0, t_A – initial temperature (0.1), actual temperature.
- r – factor for reducing of the temperatures (the speed of the annealing process), obviously 0.95 or 0.9.
- Under the terminology of *good swap* we means an edge swap (or look-ahead optimization) which reduces the total weight of the triangulation

The above described standard settings work well for simulated annealing. For the modified algorithm need to be these settings changed, because it converges faster,

and with these settings may get stuck in local minima. A new additional parameter is a degree of the look-ahead approach. In our case we employ a look-ahead of first degree. For otherwise distributed data sets the higher degrees can improve reconstruction results. There are no restrictions of the usage from the side of data distribution.

7 Implementation

The software implementation is structured into several parts, as we can see from Figure 10.

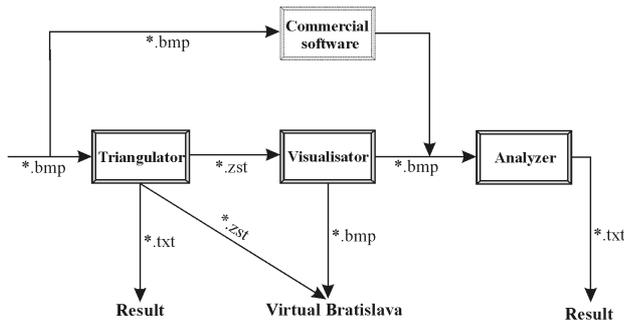


Figure 10: Workflow and the interconnection with project Virtual Bratislava

The three main parts are: *Triangulator*, *Visualisator* and the *Analyzer*. They were developed under Borland C++ Builder 5.0 for Windows platform. These applications provide solid solutions for the problems, still is possible an additional speed up. The purpose of the part, marked as *Commercial software*, is the reconstruction of images with non-DDT methods. The tasks of the main parts are the following:

- *Triangulator* gets bitmap data as input and generates a triangulation as an output.
- The *Visualisator* creates resized images of the triangulated data, which gets from the component *Triangulator*, the output is a bitmap. It is an OpenGL based application, exploiting the possibility of hardware acceleration for huge number triangle rendering, except when using $L*u*v*$ color space.
- *Analyzer* compares the bitmaps from the point of view of perceptual metrics.

In the *Visualisator* we interpolate between the colors at the xy plane and not in the 2.5D. The color calculation at the resampling depends on the ratio of the distances and not on its absolute values. For this reason it is enough to interpolate with Gourard shading in the xy plane. This means faster rendering and smaller amount of transferred data. If the interpolation does not have a linear property this simplification is not possible.

The triangulated data and the resized images will be used in creating panoramas and for reconstructing textures for VRML models in project Virtual Bratislava [21]. Another use is for example magnification of image acquired from mobile devices or printing in large resolutions. There are other possibilities, such as application at videoconferences, due to transfer speed restriction of the internet.

8 Results and Conclusion

Our goal is to provide the most objective comparing of results compared with other works. Their authors use for these purposes: perceptual metrics [20], cross correlation [19] and their own HVS [24].

We have chosen advanced perceptual metrics like the universal image quality index (UIQI) from [22], another work in this field is [5]. The main disadvantage of the perceptual metrics is the urgency of the usage of same size images. At first we must resize the original image (in our case that means minification), after that resize it back into the original size, with some of the introduced reconstruction techniques. The problem is the selection of the method for the first step, because the results will depend on this. With minification the dependency is not as recognizable as with magnification, our selection falls to bilinear interpolation.

We are planning to use in the future feature-based methods [3] for the comparing of the results, to avoid the mentioned problem. The tested images can have arbitrary size, for this approach there is not a problem to compare them with the original image.

At first, the difference is tested between the DDT and the other reconstruction methods - Table 1 - on the included test image - Figure 13.

	UIQI	Cross Correlation	Correlation
Nearest neighbour interpol.	0,10056	0,4720301	94,172
Bilinear interpolation	0,10135	0,4837924	94,418
Bicubic interpolation	0,11064	0,4724295	94,743
Locally optimal triangulation	0,10557	0,4775342	94,664
Look-ahead triangulation	0,10275	0,4776216	94,616

Table 1: Quality measurement of the reconstruction methods

The higher values of the criteries marks better fit to the original image. As we can see the differences are very small and the ordering is changing. The situation was similar with other test images, in other situations reliable comparison tools fails in this problem.

In the other case difference images shows the contrast between the reconstruction methods. In Figure 11 is demonstrated the reconstruction error at the bilinear interpolation case. The error is more distractive for the HVS than the DDT case.

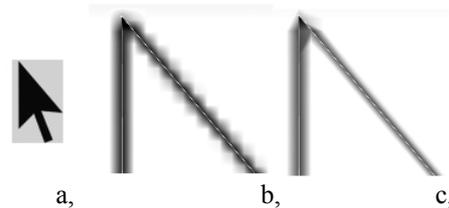


Figure 11: Close up look of difference images, a, the downsampled image, b, bilinear interpolation, c, locally optimal triangulation

The difference at minification is only slightly recognizable. We suggest the usage of fast, non-DDT reconstruction approaches for these cases. On the other

hand, magnification purposes can fully employ the merit of the DDT.

For the test image from Figure 13 we computed the simulated annealing and the SALA. - Figure 12.

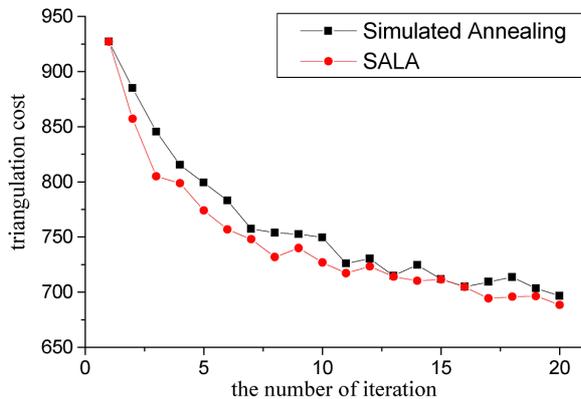


Figure 12: The convergence speed of simulated annealing and SALA

With standard settings described in Section 6 the tested stochastic processes don't work well. Better was the situation, when we lowered $r = 0.8$, or $t_0 = 0.05$. The SALA produces only slightly lower cost triangulation than the simulated annealing. This is due to special data distribution, we can consider the digital images as Markov random fields. The deterministic methods produce more satisfying results for image reconstruction purposes.

The tested deterministic methods: locally optimal triangulation and look-ahead triangulation showed near linear behaviour. For this reason the IPA primarily can find usability in real time applications. In this case is not possible to compute the triangulation for the whole image at once.

From our observations we can make another consequence. The resulting DDT triangulations have rather local character. This implicates that the computationally cheap approximations of MWT give decent results.

The domination of the DDT over the existing reconstruction methods is evident. We believe that with professional software engineering the introduced image partitioning method can achieve real time usability. The wide expansion trend in the near future of the 64-bit processors, PCI-X interfaces, and the rendering power of the existing graphic hardware brings up the inevitable question:

Is it time to design a new reconstruction standard?

9 Future Work

There are several possible ways of the future research, we mention only the most important ones:

- A possible usage of the acquired C^0 piecewise linear surface for generating higher order approximations [16].
- To test the usability of vertex based data dependent methods for texture reconstruction.
- To decimate resulting mesh to obtain image compression. We have an interesting idea to apply for

edge-based DDT built up triangulation vertex-based DDT mesh decimation.

- To test the look-ahead modification of simulated annealing for non-uniformly distributed data sets.
- The precise, speed oriented software development of the introduced image partitioning algorithm.
- To compare the results with feature-based methods.

Acknowledgements

This work has been carried out as part of the project Virtual Bratislava, which is funded from grant APVT 20-0255.

I would like to say thank to my supervisor Andrej Ferko for his continuous help and support during the development and writing of this paper.

References

- [1] Alboul L, Kloostreman G, Traas C, van Damme R. *Best Data-Dependent Triangulations*. Technical Report TR-1487-99, University of Twente, 1999.
- [2] Aurenhammer F. *Voronoi Diagrams- A Survey of a Fundamental Geometric Data Structure*. ACM Computing Surveys, Vol. 23, No. 3., pp. 345-405, 1991.
- [3] Bornik A, Cech P, Ferko A, Perko R. *Beyond Image Quality Comparison*. Eurographics, EG Short, 2003.
- [4] Brown J.L. *Vertex based Data Dependent Triangulations*. Computer Aided Geometric Design 8, pp. 239-251, 1991.
- [5] Čech P. *Perceptual Metrics and Image Compression*. Proceedings of Spring Conference on Computer Graphics, Budmerice, pp. 39-40, 2002. ISSN 1335-5694.
- [6] Černý V. 1985. *A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm*. J. Opt. Th. and Appl. Vol. 45, No.1., pp. 41-52, 1985.
- [7] Dyn N, Levin D, Rippa S. *Data Dependent Triangulations for Piecewise Linear Interpolation*. IMA Journal of Numerical Analysis 10, pp. 137-154, 1990.
- [8] Ferko A. *Approaching the Minimum Weight Triangulation Problem*. Habilitation Lecture Notes, Manuscript, 2004.
- [9] Ferko A, Niepel L, Plachetka T. *Criticism of Hunting Minimum Weight Triangulation Edges*. Proceedings of the 12th Spring Conference on Computer Graphics, Budmerice, pp. 259-264, 1996. ISBN 80-223-1032-8.

- [10] Garland M, Heckbert P.S. *Fast Polygonal Approximation of Terrains and Height Fields*. Technical Report TRCMU-CS-95-181, Carnegie Mellon University, 1995.
- [11] Kirkpatrick S, Gellat jr. C.D, Vecchi M.P. *Optimization by Simulated Annealing*. IBM Technical Report, 1982, also: Science 220 (1983) pp. 671-680, 1983.
- [12] Kolingerová I. *Genetic Approach to Data Dependent Triangulations*. Proceedings of Spring Conference on Computer Graphics, Budmerice, pp. 229-238, 1999.
- [13] Kolingerová I, Ferko A. *Multicriteria-optimized Triangulations*. The Visual Computer 17(6), pp. 380-395, 2001.
- [14] Kolingerová I, Žalik B. *Improvements to Randomized Incremental Delaunay Insertion*. Computers & Graphics 26, pp. 477-490, 2002.
- [15] Kreylos O, Hamman B. *On Simulated Annealing and the Construction of Linear Spline Approximations for Scattered Data*. IEEE Transactions on Visualization and Computer Graphics, Vol. 7, No.1., pp. 17-31, 2001.
- [16] Nielson G.M. *Scattered Data Modeling*. IEEE Computer Graphics & Applications 13(1), pp. 60-70, 1993.
- [17] Rheingans P. *Color Perception and Applications*. SIGGRAPH 97 Course #33, 1997.
- [18] Schumaker L.L. *Computing Optimal Triangulations Using Simulated Annealing*. Computer Aided Geometric Design vol. 10 (3-4), pp. 329-345, 1993.
- [19] Su D, Willis P. *Image Interpolation by Pixel Level Data-Dependent Triangulation*. Submitted to Computer Graphics Forum, 10/2002.
- [20] Su D, Willis P. *Demosaicing of Colour Images Using Pixel Level Data-Dependent Triangulation*. Theory and Practice of Computer Graphics, pp. 16-25, 2003.
- [21] Vnučko I. *Image Based Rendering – Panorámy*. MSc thesis, Bratislava: Comenius University, 2004.
- [22] Wang Z, Bovik A.C. *A Universal Image Quality Index*. IEEE Signal Processing Letters, Vol. 9, No.3., pp. 81-84, 2002.
- [23] Weimer H, Warren J, Troutner J, Wiggins W, Shrout J. *Efficient Co-Triangulation of Large Data Sets*. IEEE Visualization 98, pp. 119-126, 1998.
- [24] Yu X, Morse B.S, Sederberg T.W. *Image Reconstruction Using Data-Dependent Triangulation*. IEEE Computer Graphics and Applications, pp. 62-68, 2001.

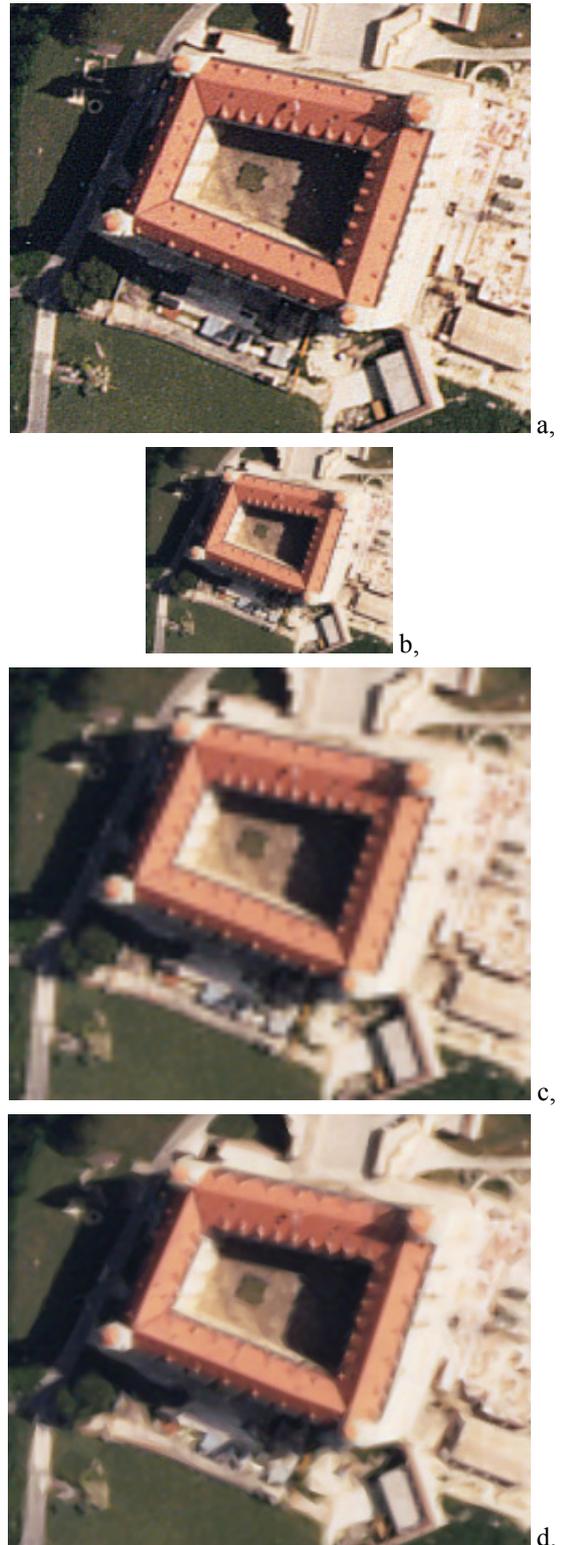


Figure 13: The Bratislava castle test image a, an original orthophotograph, b, downsampled image c, bilinear interpolation, look-ahead triangulation