# Towards an Optimal Texture Reconstruction

Zsolt Tóth*

**Abstract.** *Visually pleasant texture reconstruction plays an important role in computer graphics. In this paper we propose special triangulations for texture reconstruction. We introduce two new algorithms for data-dependent triangulation. The new deterministic algorithm named 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. It supports achieving qualitative way of reconstruction of color images. Results obtained demonstrate both theoretical and practical superiority over another methods. This work is a part of the Virtual Bratislava research.*

**Keywords:** Image Reconstruction, Data-dependent Triangulation, Minimum Weight Triangulation

## 1. Introduction

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

There are many approaches for image reconstruction. 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. The image reconstruction with the help of data dependent triangulation belongs among these methods. The root of this idea is only a few years old. The data-dependent triangulation [5] is an intelligent reconstruction method, which generates the most

*email: toth@sccg.sk

*Faculty of Mathematics, Physics and Informatics, Comenius University, SK-84248 Bratislava, Slovakia

pleasant results from the described approaches (see Figure 1). In this paper we are dealing with new improvements of this idea.

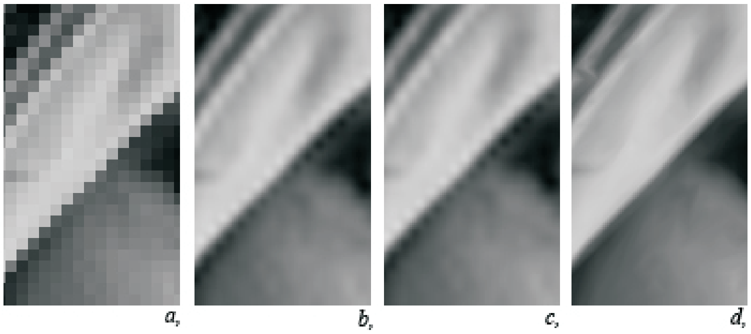

**Figure 1. a) nearest neighbour- b) bilinear- c) bicubic- interpolation, d) data-dependent triangulation at 600% magnification**

The paper is structured as follows. Section 2 gives a short mathematical background and an overview of research that has 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 in Section 5. 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.

## 2. Background and Related Work

First we introduce some necessary definitions.
**Def. 2.1:** A set of distinct (and not all collinear) points $\mathbf{V} = \{(x_i, y_i); i = 1, \ldots, n\}$ is given in $E^2$ (Euclidean 2-dimensional space). Triangulation of the set **V** is called a set of triangles $T(\mathbf{V})$ which satisfy the following properties:

(i) Each vertex of an arbitrary triangle from $T(\mathbf{V})$ is from the set **V** and each element from **V** is a vertex of at least one triangle in $T(\mathbf{V})$.
(ii) Each edge from the triangulation contains exactly two elements from **V**.
(iii) The union of all triangles from $T(\mathbf{V})$ is the convex hull of the set **V**.
(iv) An intersection of two arbitrary triangles from $T(\mathbf{V})$ 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 lengths (costs) of all its edges:

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

where $c(e)$ denotes the edge length of edge $e$.
In this work we use the $l_1$ norm, but there exist other possibilities, too [5].
**Def. 2.3:** The minimum weight triangulation (MWT) of a set **V** is a triangulation whose weight is minimal among all possible triangulations:

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

**Def. 2.4:** Data-dependent triangulation (DDT) of a set $\mathbf{V} = \{(x_i, y_i); i = 1, \ldots, n\}$ in $E^2$ is a triangulation whose topology is dependent on a function $f(x, y) = z_i = f(x_i, y_i); i = 1, \ldots, n$.
The DDT allows to construct triangulations in 2D based on additional information. 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 to 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 5.

Triangulations usually avoid the generation of long tiny triangles. However, these triangles are suitable for reconstruction of areas, where $f(x, y)$ has large derivatives in one direction if compared to other directions. In image reconstruction such areas contain edges in the image, and our primary goal is to reconstruct them as correctly as possible - 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. The detailed survey of the MWT problem was done by Ferko [6]. The number of the existing heuristics, approximations and known subgraphs [7] of this problem for planar triangulations is reduced with the data dependent approach. This is due to special edge cost.

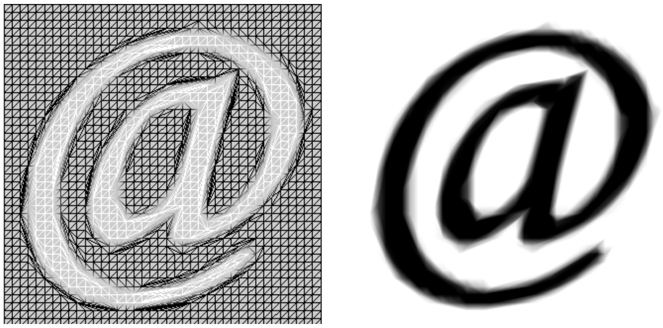

**Figure 2. Local optimality and its look-ahead expansion**

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 (MWT) 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. The most popular data dependent approach, the Lawson optimization [5] (also called locally optimization procedure) works with the edge swapping operation. It is similar to the Delaunay triangulation for planar case [2, 12]. 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.

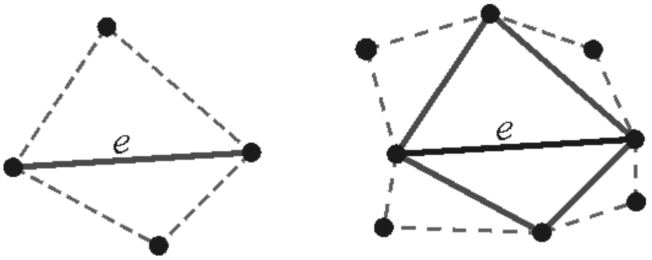

**Figure 3. Local optimality and its look-ahead expansion**

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 dashed 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 [17, 18]. 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]. However, data-dependent methods give better results.

Besides deterministic methods, there are stochastic optimization techniques used for constructing DDT heuristics of MWT. The simulated annealing approach was described by Schumaker in [16]. Its application for image compression was done by Kreylos *et al.* [13]. The genetic optimization is another stochastic process, Kolingerova introduced a data-dependent version [10], recent results can be found in [11].

Mesh refinement and decimation techniques also use DDT approaches. E.g. Garland and Heckbert combined the DDT with greedy strategy for digital elevation model reconstructing and simplifying [8].

## 3. The Main Steps of DDT

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

### 3.1. Initial Triangulation of the Domain

First, we need an initial triangulation. The reason is that nearly no efficient tools exist for creating DDT approximations and heuristics of MWT. Edge swapping and its look-ahead expansion require an initial triangulation. 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 create a planar triangulation. The most suitable choice to triangulate the domain of $f(x, y)$ is the regular triangulation - Figure 4, it is both MWT and the most equiangular (Delaunay) triangulation for the Cartesian grid data sets.

Because we know the data organization, the creation of the initial triangulation requires only $O(n)$ time. The existing algorithms to improve the initial $T(\mathbf{V})$ mainly have iterative character. The speed of convergence depends on how far is the initial triangulation from the final stage (what is usually only a local optimum). The regular triangulation guarantees fast convergence speed, as we have observed.

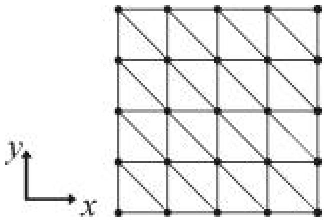

**Figure 4. The triangulation of the domain with regularly placed points**

### 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 setting low cost for edges, which represent visually relevant parts. The existing approaches work with one data vector obtained from the color information of the input image. Digital images are usually represented with three components, e.g. RGB, YUV. We will use standard RGB.

The simplest approach uses linear intensity, what can be easily computed:

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

where $c_I(e)$ means the cost of edge $e$ with respect to intensity values. In this way the chromatic information is omitted. This causes errors when reconstructing small details. Also there can occur situations, when neighbouring pixels 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 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 $N$. The authors report that the quality of the result was slightly increased.

### 3.3. Cost Functions

Now we describe the role of the edges cost. We consider only a small group of these functions, for image reconstructing [24]. Generally, there 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 $T_2$.

$$P_1(x, y) = a_1x + b_1y + c_1$$
$$P_2(x, y) = a_2x + b_2y + c_2$$

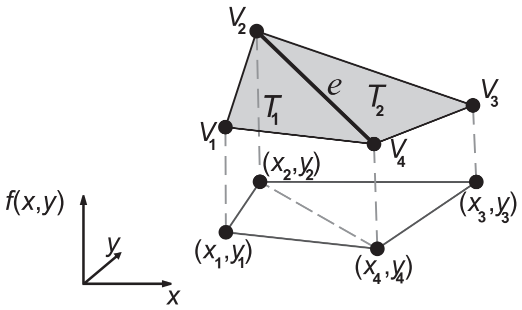

**Figure 5. Illustration for the calculation of cost functions**

The above 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 $(n_y, n_x)^{\boldsymbol{T}}$ is a unit vector in $E^2$ ($xy$ 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 $v_1$:

$$c^{DLP}(e) = \|h\| \; ,$$
$$h = [|P_1(x_3, y_3) - f(x_3, y_3)|, |P_2(x_1, y_1) - f(x_1, y_1)|] \; .$$

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

$$c^{DP}(e) = \|g\|, g = [dist(P_1, v_3), dist(P_2, v_1)],$$
$$dist(P_j, v_k) = \frac{|P_j(x_k, yk) - f(x_k, y_k))|}{(a_j^2 + b_j^2 + 1)^{1/2}} \; .$$

**SCF** (Sederbergs cost function) uses the projection of the $T_1$ and $T_2$ triangle normals into the 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 of SCF are sensitive to the scaling of the intensity range. For image reconstruction purposes the best results were produced in tests with SCF [24]. This corresponds with our observations - that the SCF control reconstructs better the edge areas than other methods. The demonstration of performance of the cost functions is shown at Figure 6.

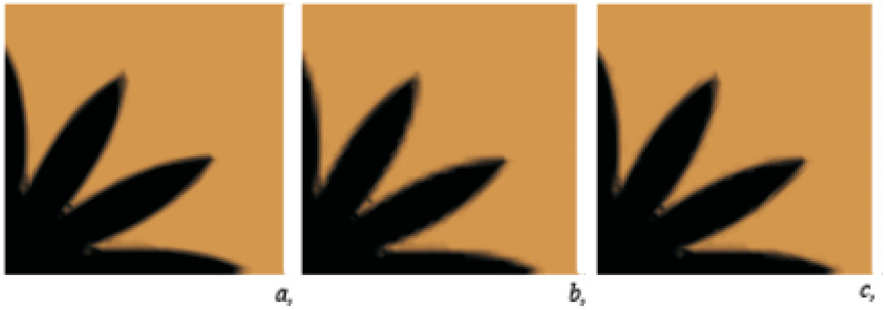

**Figure 6. The SCF, the ABN and the DP cost function generated results**

Several other approaches exist, which assign the cost to components of triangulation (vertices, edges, triangles, etc.) on the base of various geometrically based and non-geometrically based criteria. More information about other cost functions can be found in [1, 4, 5].

## 4. Correct Usage of Color

Data mining from color information is not completely correct in the existing approaches, 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 the separate triangulation for each color component seems to be an ideal approach. In [24] authors noticed that this can lead to color bleeding with the RGB model. Our approach is based on the usage of perceptually uniform color spaces, trying to avoid or minimize the color bleeding effect.
**Def.:** [15] 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 (it 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 between RGB and L*u*v* is relatively easy. Also this color space describes the human visual system sensitivity better than RGB.

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

By using CIE L*u*v* for bilinear interpolation, we loose the support of hardware acceleration. On the other hand, the origins of the reconstruction errors are eliminated (for example the artifacts at Gouraud shading are result of using the RGB model). The outlined method uses the color information without losses, and works in a natural way, which is better for the human vision.

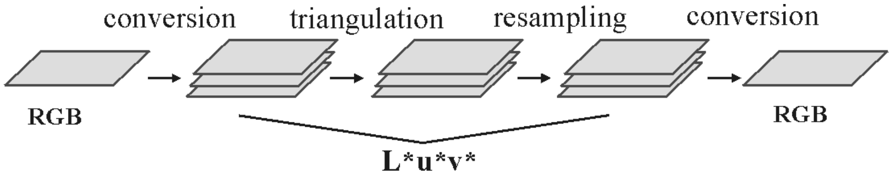

**Figure 7. Sequence of correct reconstruction steps**

## 5. Image Partitioning

Our new deterministic algorithm named IPA tries to combine the advantages of the existing methods. Our main goal was to find an approach, which allows a 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. The conception to divide the picture into blocks - rectangles is based on this observation. Each vertex must lie exactly in one block. The approximation of the MWT is evaluated inside these blocks, or the MWT is selected from all the possibilities. The problem is to interconnect the neighbouring blocks. The initial structure of this interconnection is the regular triangulation - Figure 8.

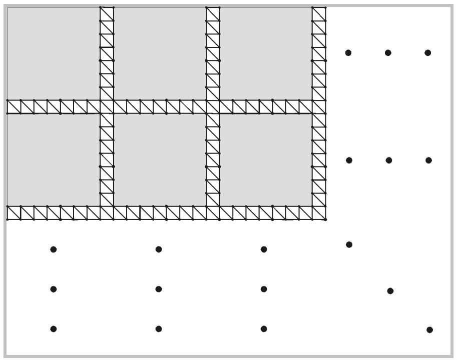

**Figure 8. Block segmentation of the image**

The task of the interconnection method is to aid visually pleasant reconstruction and to have a low computational cost. One look-ahead transition (of first degree) is used for this intention across the set $\mathbf{V}^{\mathbf{A}}$ - set of active edges. Set $\mathbf{V}^{\mathbf{A}}$ consists of edges, which are not part of any block at the initial structure, and they have different orientation as the 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 $e$ from the set of active edges - illustrated in Figure 9. The look-ahead optimization of the first degree of the edge $e$ does not change the other elements from $\mathbf{V}^{\mathbf{A}}$, and interconnects the neighbouring blocks. Figure 10 shows the pseudocode of the algorithm.

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

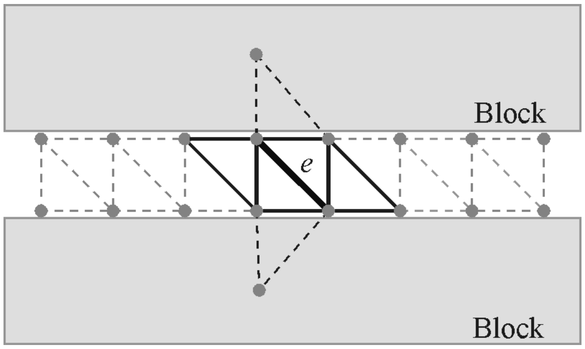

**Figure 9. Look-ahead of first degree of edge $e$**

**Image partitioning algorithm - IPA**
**Input:** digital image
**Output:** triangulation

```
{
  divide image into blocks;
  triangulate each block;
  create the initial triangulation between blocks;
  build up the list $V^A$;
  for(all edges from the set $V^A$)
    applicate look-ahead of first degree;
}
```

**Figure 10. The pseudocode of the IPA**

IPA 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 the described method of interconnection is used. 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 in various ways, for example we can set their priority with the help of edge detectors.

The IPA 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 a continuous model from digital elevation models.

## 6. Simulated Annealing with Look-Ahead

In this section, there the new modification of the optimization technique called simulated annealing [9, 20] is introduced. Its basic idea employs the generalized look-ahead process, with the goal to achieve better reconstruction.

Classical simulated annealing proceeds as follows: if the edge is locally optimal, then with some

**Simulated annealing with look-ahead - SALA**
**Input:** initial triangulation
**Output:** simulated annealing with look-ahead

```
{
 for($k = 1; k \leq ntemp; k++$){
   $t_A = r^k t_0$;
   for($l = 1; l \leq nlimit; l++$)
     while(number of good swaps $\leq$ glimit){
       select random edge $e$ from triangulation;
       if(exists alternate diagonal of edge $e$){
         if(strictly convex quadrilateral, which contains $e$ is not optimally triangulated){
           swap diagonals;
         }
         else{
           random choice of number $\phi$, $0 \leq \phi \leq 1$;
           if($\phi \leq e^{-c(e)/t_A}$)
             swap diagonals;
           else
             look-ahead optimalization of edge $e$;
         }
       }
     }
 }
}
```

**Figure 11. The pseudocode of the SALA**

probability the edge is swapped. This causes the increase of the triangulation weight, but avoids getting stuck in a local minima. Our modification processes the case when this swap does not happen. Then for the tested edge the look-ahead searching of optimality is applied. Figure 11 shows the pseudocode of the algorithm.

We describe the meaning of the 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 chosen as fixed multiplies of the number of edges, often adjusted by a factor 5 or 10.
- $t_0, t_A$ - initial temperature (0.1), actual temperature.
- $r$ - factor for reducing the temperatures (the speed of the annealing process), typically 0.95 or 0.9.

We mean by *good swap* an edge swap (or look-ahead optimization) which reduces the total weight of the triangulation.

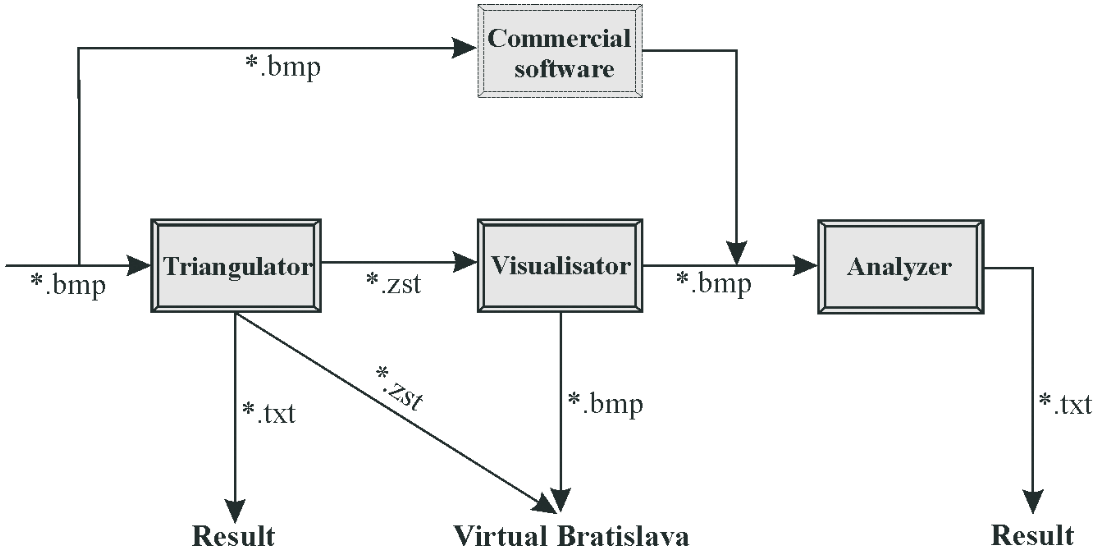

**Figure 12. Workflow and the interconnection with project Virtual Bratislava**

The above described standard settings work well for simulated annealing. For the modified algorithm there is a need for changes, because it converges faster, and with standard settings it 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 the first degree. For various distributed data sets the higher degrees can improve reconstruction results.

## 7. Implementation

The software implementation is structured into several parts, as we can see from Figure 12.

The three main parts are: *Triangulator*, *Visualisator* and *Analyzer*. They were developed under Borland C++ Builder 5.0 for Windows platform. These applications provide solid solutions for the problems. However, it is still 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 the bitmap data as input and generates a triangulation as an output.
- *Visualisator* creates resized images of the triangulated data, (from the component *Triangulator*), the output is a bitmap. It is an OpenGL based application, exploiting the possibility of hardware acceleration for huge triangle count rendering, except when using L*u*v* color space.
- *Analyzer* compares the bitmaps using perceptual metrics.

In the *Visualisator* we interpolate between the colors in 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 their 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 the project Virtual Bratislava [21]. Another use is for example magnification of an 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 our results with other works. Their authors use for these purposes: perceptual metrics [17], cross correlation [18] and their own HVS [24].

We have chosen an advanced perceptual metrics like the universal image quality index (UIQI) from [22], another work in this field is [19]. The main disadvantage of the perceptual metrics is the urgency of the usage of equal 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 fails on bilinear interpolation.

In the future we are planning to use feature-based methods [3] for the comparison of the results, to avoid the above 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 15.

| | UIQI | Cross Correlation | Correlation |
|---|---|---|---|
| Nearest neighbour interpolation | 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 criteria means better fit to the original image. As we can see the differences are very small and the ordering is changing. The situation was similar on other test images.

In the other case, difference images show the contrast between the reconstruction methods. In Figure 13 the reconstruction error of the bilinear interpolation case is demonstrated. The error is more distractive for the HVS than the DDT case.

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 15 we computed the simulated annealing and the SALA (see Figure 14).

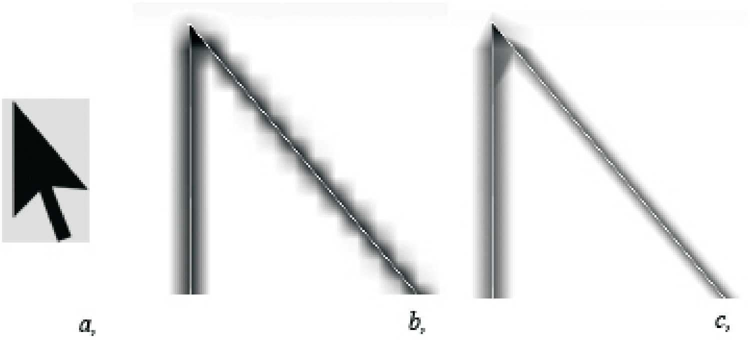

**Figure 13. Close up look of difference images, a, the downsampled image, b, bilinear interpolation, c, locally optimal triangulation**

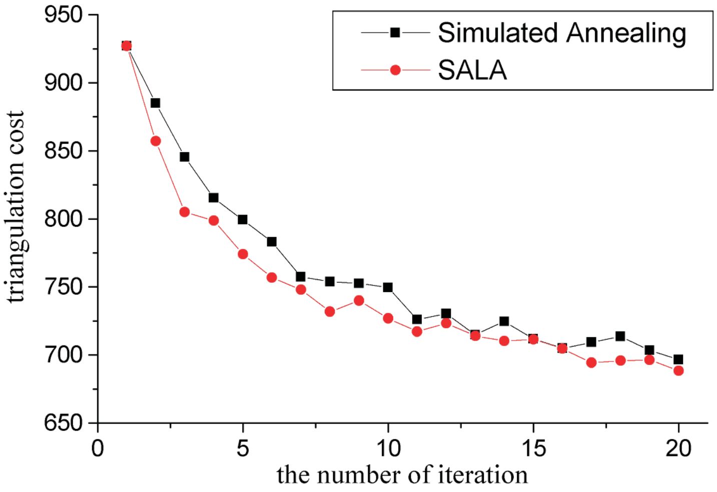

**Figure 14. The convergence speed of simulated annealing and SALA**

With standard settings described in Section 6 the tested stochastic processes don‘t work well. Better situation was, when we decreased $r = 0.8$, 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 nearly linear behaviour. For this reason the IPA can find usability primarily in real time applications. In this case it 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 implies 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 of the 64-bit processors, PCI-X interfaces, and the rendering power of the existing graphic hardware in the near future 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 [14].
- To test the usability of vertex based data dependent methods for texture reconstruction.
- To decimate the resulting mesh to obtain image compression. We have an idea to apply vertex-based DDT mesh decimation for triangulation built up by edge-based DDT.
- 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.

## 10. Acknowledgements

This work has been carried out as part of the Virtual Bratislava research, which is partly funded from the grant VEGA 1/3083/06.

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] L. Alboul, G. Kloostreman, C. Traas, and R. V. Damme. Best data-dependent triangulations. *J. Comput. Appl. Math.*, 119(1-2):1–12, 2000.

[2] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[3] A. Bornik, P. Čech, A. Ferko, and R. Perko. Beyond image quality comparison. In *Eurographics, EG Short 2003*, pages 271–276, 2003.

[4] J. Brown. Vertex based data dependent triangulations. *Computer Aided Geometric Design*, 8(3):239–251, 1991.

[5] N. Dyn, D. Levin, and S. Rippa. Data dependent triangulations for piecewise linear interpolation. *IMA Journal of Numerical Analysis*, (10):137–154, 1990.

[6] A. Ferko. Solving the minimum weight triangulation problem. Habilitation Lecture Notes, Bratislava: Comenius University (manuscript), 2004.

[7] A. Ferko, L. Niepel, and T. Plachetka. Criticism of hunting minimum weight triangulation edges. In *Proceedings of the 12th Spring Conference on Computer Graphics*, pages 259–264, 1996.

[8] M. Garland and P. S. Heckbert. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181, 1995.

[9] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598:671–680, 1983.

[10] I. Kolingerová. Genetic approach to data dependent triangulations. In *Proceedings of Spring Conference on Computer Graphics*, pages 229–238, 1999.

[11] I. Kolingerová and A. Ferko. Multicriteria-optimized triangulations. *The Visual Computer*, 17(6):380–395, 2001.

[12] I. Kolingerová and B. Zalik. Improvements to randomized incremental delaunay insertion. *Computers & Graphics*, (26):477–490, 2001.

[13] O. Kreylos and B. Hamann. On simulated annealing and the construction of linear spline approximations for scattered data. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):17–31, 2001.

[14] G. M. Nielson. Scattered data modeling. *IEEE Comput. Graph. Appl.*, 13(1):60–70, 1993.

[15] P. Rheingans. Color perception and applications. In *Siggraph'97. Course # 33*, 1997.

[16] L. L. Schumaker. Computing optimal triangulations using simulated annealing. In *Selected papers of the international symposium on Free-form curves and free-form surfaces*, pages 329–345. Elsevier Science Publishers B. V., 1993.

[17] D. Su and P. Willis. Demosaicing of colour images using pixel level data-dependent triangulation. In *TPCG '03: Proceedings of the Theory and Practice of Computer Graphics 2003*, page 16, Washington, DC, USA, 2003. IEEE Computer Society.

[18] D. Su and P. Willis. Image interpolation by pixel-level data-dependent triangulation. *Comput. Graph. Forum*, 23(2):189–202, 2004.

[19] P. Čech. Perceptual metrics and image compression. In *Proceedings of Spring Conference on Computer Graphics*, pages 39–40, 2002.

[20] V. Černy. A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–52, 1985.

[21] I. Vnučko. Image based rendering - panorámy. Master's thesis, Comenius University, Bratislava, 2004.

[22] A. Wang and A.C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[23] H. Weimer, J. Warren, J. Troutner, W. Wiggins, and J. Shrout. Efficient co-triangulation of large data sets. In *IEEE Visualization 98*, pages 119–126, 1998.

[24] X. Yu, B. S. Morse, and T. W. Sederberg. Image reconstruction using data-dependent triangulation. *Computer Graphics and Applications*, 21(3):62–68, 2001.

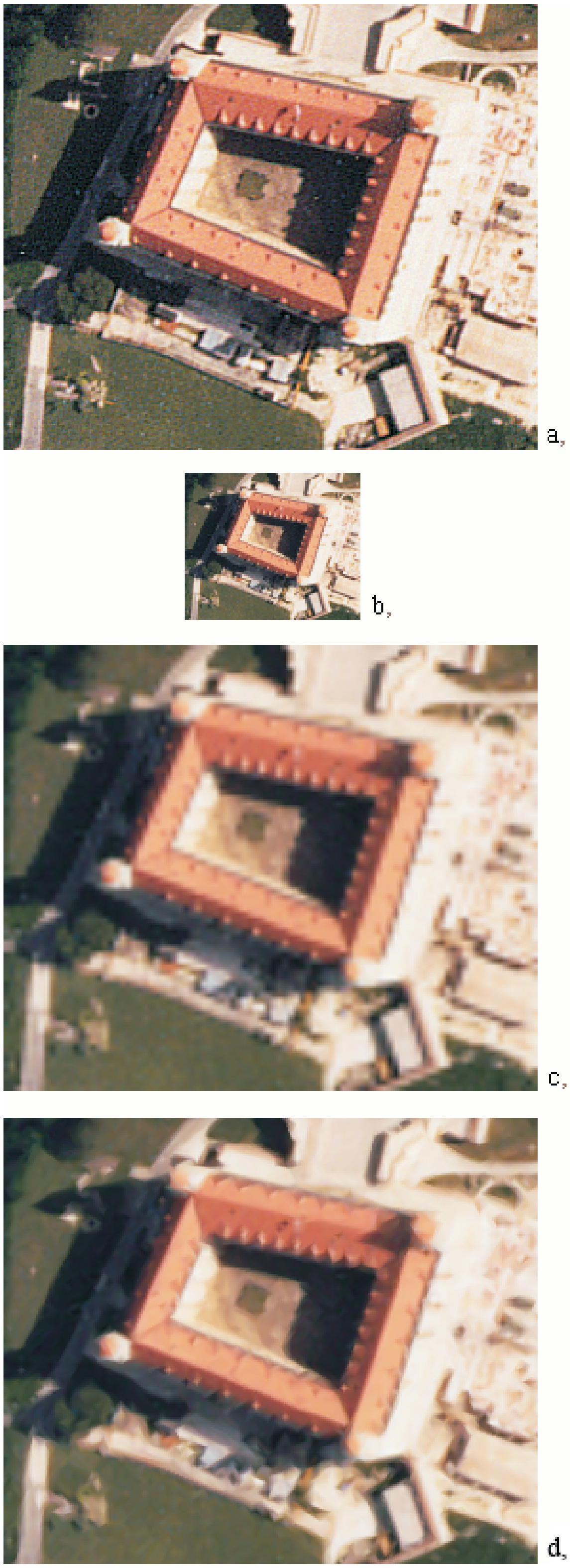

**Figure 15. The Bratislava castle test image a) an original orthophotograph, b) downsampled image, c) bilinear interpolation, d) look-ahead triangulation. Data courtesy EUROSENSE, a.s.**