ON AUTOMATIC TEXTURE GROWING FROM VORONOI COMPUTED BUILDING FAÇADE SURFACES USING COLOR SIMILARITY DETECTION

Ersilia ONIGA – teaching assistant, „Gheorghe Asachi” Technical University of Iasi, Faculty of Hydrotechnical Engineering, Geodesy and Environmental Engineering, Department of Terrestrial Measurements and Cadastre, ersilia.oniga@ tuiasi.ro
Marius SUTU – associate teaching assistant, „Gheorghe Asachi” Technical University of Iasi, Faculty of Automatic Control and Computer Engineering, Department of Automatic Control and Applied Informatics, msutu@tuiasi.ro

***Abstract**: Terrestrial laser scanners are becoming increasingly a source and a good alternative for 3D data acquisition. The scanning process produces an impressive number of spatial points, each of them being characterized as position by the X, Y and Z co-ordinates, by the value of the laser reflectance and their real color, expressed as RGB (Red, Green, Blue) values. This paper proposes a solution for the automatic texture growing for the 3D building facades. The starting point is to create the textures for the surfaces corresponding to the building facades. By computing the corresponding Voronoi diagrams, the RGB values of every point that has been taken using terrestrial laser scanning technology are mapped on the surfaces defining the buildings facades. The next step is the texture growing in each Voronoi diagram by merging cells based on colour similarity detection. The proposed algorithm identifies a match between adjacent cells by a weighted euclidian distance on their RGB color codes, and merges them with an average of the respective colour codes. The textures are processed in MATLAB and rendered with AutoCad. By using this algorithm we aim to obtain in an automated manner, photorealistic textures for a 3D model.*

Keywords: Point Cloud, Façade, Digital Image, Voronoi Diagram, Texture, RGB Color Space, Color Similarity

1. Introduction

One of the main challenges of the 3D urban modelling is how to generate 3D models of buildings and obtain a photorealistic appearance. Capturing, processing and aligning surface textures for the purpose of 3D modelling (for example the textures of the facades) still represent costly and time-consuming tasks. The most frequently used method is to project photographs, based on finding correspondences between images or between an image and a 3D surface or a point cloud.

The automatic surface texture generation represents a key element in the creation and practical use of the 3D models. The algorithms presented in the specialized publications use images in order to generate the textures, the multiple images being preferred because of the numerous projection centers. They can be high-resolution aerial photographs taken in the oblique position, or close range digital images taken with digital (metric) photocameras.

Commercial software like Photo Modeler and Photo Modeler Scanner by Eos Systems Canada, Shape Capture and Shape Scan by ShapeQuest Inc., 3DImage by Dr. Eng. Ahmed Abdelhafiz, Photo3D by SoftCube Co. Japan or SketchUp by Google are computing 3D textured models only using single or multiple digital images with accurate photorealistic effect but lacking precision towards representing dimensional elements such as distances, areas or volumes.

The precision shortcommings can be solved by using 3D - point clouds taken with a terrestrial laser scanner. A high accuracy and a high scanning speed in acquiring the points are obtained, while keeping the errors in the range of milimeters - for the ScanStation 2 terrestrial laser scanner, also used in this paper, the error for 50 m being 6 mm (Thomas P. Kersten et. al., 2008). Some commercial 3D systems such as: Cyclone by Leica Geosystems, SceneVision-3D by 3rdTech Inc., Bentley CloudWorx by Bentley Systems Inc., Faro Scene by Faro Technologies, ISiTE Studio by I-SiTE, RiSCAN PRO and Phidias by Riegl, RealWorks Survey by Trimble Navigation, LFM Modeller by Z+F, Geomagic Studio by Geomagic etc., are processing point clouds while also rendering the captured RGB values of each LIDAR point with the camera integrated in the system.

In the above mentioned point cloud processing, several methods found in literature are being used. The most encountered approaches are to create either polygon meshes (Hu C., 2008), or NURBS surfaces (Non Uniform Rational Basis Spline) or using CAD objects and apply textures from digital images. When using CAD objects on building modelling, the resulting model is parametric and editable, being based on dimensional elements. An approach in this direction is presented in the following chapters, starting with a brief introduction to point cloud acquisition**,** several aspects on point cloud modelling techniques with some details on using Voronoi diagrams for texture generation. An algorithm for texture growing from the generated Voronoi cells is presented, together with the obtained results for a considered test scenario.

2. Point cloud acquisition using terrestrial laser scanner

The laser scanner communicates with an attached computing device (typically a laptop) in order to store the acquired points information as well as controlling the scanning patterns. The field-of-view is determined and the proper resolution is set (the distance between two adjacent measured points for a specified distance to the scanner), prior to starting the scanning process. The operation is almost fully automated providing a hight point density, while the points scanned are directly visualized in three dimensions on the screen providing an overview of the area that has already been scanned (Quintero M., 2008). Each of this points is characterized as position by the X, Y and Z co-ordinates, by the value of the laser reflectance (I) and his real color, expressed as RGB (Red, Green, Blue) values, since most laser scanners have a digital camera integrated in their system.

The scanner camera resolution (in the considered case of Leica ScanStation2 - 5 mega pixel), leads to a quality of the textured point color that is not always optimal, and, moreover, it is not always the desired texture. Another inconvenience is that the process of taking the images doesn't always take place in ideal conditions, especially for outdoor applications. In this case, the time difference between the recorded images will result in varying light conditions and changing shadows, thus the images will have different radiometric properties. Such problems may disturb the appearance of the resulting textured model (Alshawabkeh et. al, 2005). Therefore it is more useful to acquire geometry using laser scanning technology in an close to optimal light context.

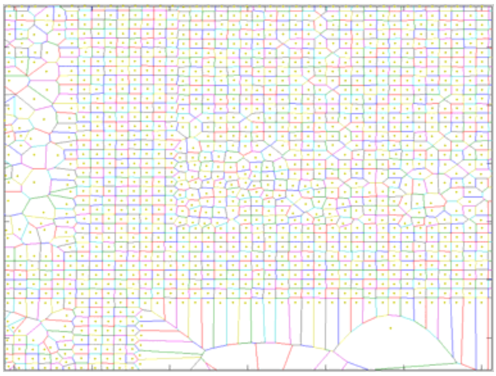
3. Point cloud based modelling

The point cloud modelling starts with the segmentation procedure in which the scanned points are approximately grouped dependent on the surface they belong to. After that, the surfaces are constituted from a group of adjacent points which fit well into their respective surface (patch/plane, cylinder, sphere, etc.). In this way, distinctive elements of the facades are delimited (Region Grow command for Leica Cyclone), but the color information of each point is not further used for texturing the surfaces, an aleatory color from the software pallete being generally used. As a result, the 3D model is textured having little to no correspondence with the reality in the object space.

In their assertion from 2009, Pelagottia and his collaborators (Pelagottia et. al, 2009) propose a fully automatic approach for multispectral texture mapping. The method relies on the extraction from the model geometry of a depth map, in form of an image, whose pixels maintain an exact correspondence with vertices of the 3D model; the subsequent step is registration, between the depth map and the image to be mapped, with a very robust registration algorithm, based on Maximization of Mutual Information. While the resulted color fidelity on the 3D model data (points or meshes) is highly accurate, the solution is not based on dimensional elements, thus not parametric and editable. Also, the generated texture is based on aditional digital images and not relying on the laser scanner acquired RGB colors for each point.

Solving these texturing issues requires some supplementary steps like using another software to create the final textured model (for example Autodesk 3D Studio Max) in which the textures are created using fragments from the digital photos of the building facades. The downside of this solution is that it does not reflect entirely the terrain reality and the results prove unsatisfying.

In (Oniga E., 2012) an algorithm is proposed for the semiautomatic texture generation using as color information the RGB values of every LIDAR point. The point information is taken by terrestrial laser scanning technology and the 3D surfaces defining the buildings facades are generated with the Leica Cyclone software. The point cloud modelling is performed using lines and surfaces with the existing Cyclone functions. The surfaces are exported as a *.dxf file and subsequently imported into AutoCAD platform, leading to the creation of the 3D model for the building based on CAD objects. The corresponding textures are generated by calculating the Voronoi diagram while using Fortune's algorithm (Fortune S., 1987) as presented in Fig. 1.



(a)

(b)

Fig. 1. (a) Sample Voronoi cells for a surface, (b) The textured surface

When applying the algorithm for a various set of planes from the 3D model, the corresponding Voronoi diagram sizes are dirrectly influenced by the laser scanner spatial resolution. When processing the 3D points, these are projected on the corresponding planes, and even if they are filtered by specific surface contours and relative distance to the projection planes, most of the analyzed points pass and a Voronoi cell is being created for every point. As presented in Table 1, the representation complexity is high, the resulting textures being composed in some cases even of hundreds of thousands polygons – one for each Voronoi cell.

Table 1. Representation complexity for Voronoi computed textures

Surface	No. of. initial points	Threshold (cm)	No. of polygons
Secondary façade detail	4661	1.5	4430
Window frame element	9484	3	8540
Front façade plane 1	92733	2	90653
Front façade plane 2	149027	1.5	140896
Door element	9612	2	7429

The high geometric complexity given by the amount of polygons in the resulting textures is a bottleneck when mapping them to the actual 3D model, so a texture growing solution is presented in the next chapter.

4. The proposed algorithm - texture growing for each surface

As detailed next, texture growing is a process consisting in merging adjacent polygons based on color code similarity. The RGB triplet represents the three-dimensional coordinate of a point in the RGB color space, which is represented as a cube having the black color in its origin and the white color in the opposite corner.

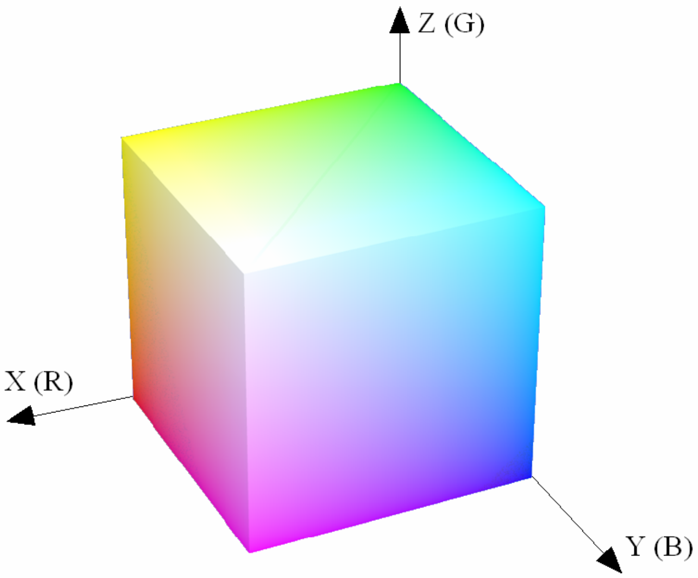


Fig. 2. The RGB color space

The algorithm considers a threshold when deciding that there is a similarity between two RGB color codes. The similarity is calculated using the weighted Euclidian distance, or perceptual Euclidean distance (PED), between two color representations in the RGB color space as presented in:

$$PED(c_i, c_j) = \sqrt{w_R(R_i - R_j)^2 + w_G(G_i - G_j)^2 + w_B(B_i - B_j)^2} \tag{1}$$

where: w_R, w_G and w_B are the weights for the RGB color channels with w_R= 0.26, w_G =0.70, w_B=0.04 and w_R+ w_G+w_B=1. PED, with these weight-coefficients finds its roots in human vision and correlates significantly higher than all other distance measures (Gijsenij A., et. al., 2008).

The Voronoi diagram computational representation consists of two arrays, one with all the vertices and one with the polygonal representation for every cell – an ordered set of vertices. The algorithm identifies the adjacent cells by the vertex they converge into, and if the PED between the considered cells RGB colors falls in a desired error margin, the cell growing process can take place. From a topological point of view, the narrowed set of cells is a level one neighbourhood around the chosen vertex.

The merge algorithm computes a new cell by preserving all the boundary edges and removing the common vertex as well as the vertices corresponding to the adjacent edges between any of the merged cells (Fig. 3). The resulting cell color is set as the rounded average of the red, green and blue components of the merged cells.

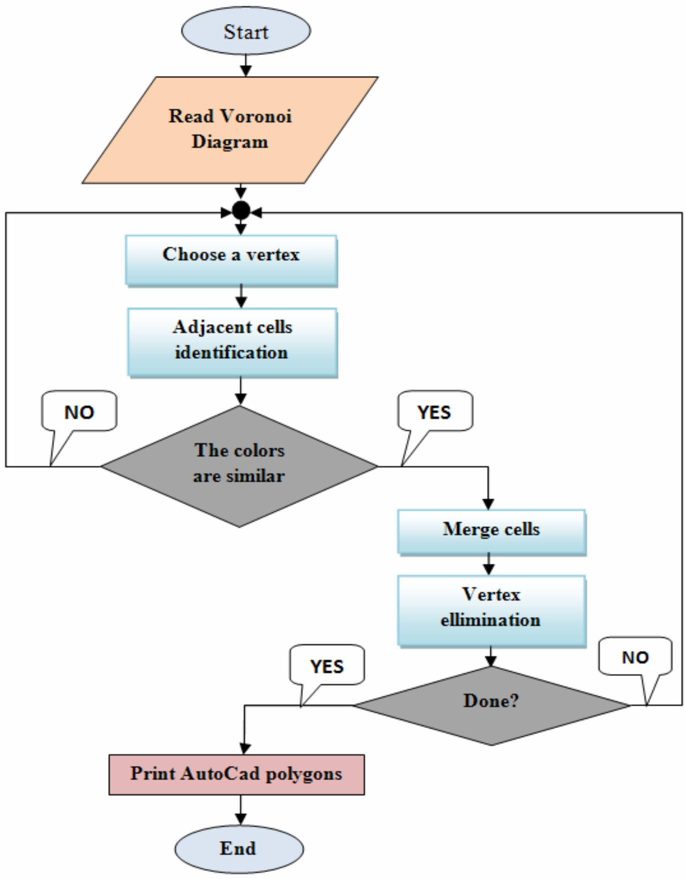


Fig. 3. The proposed algorithm logical schema

The newly created cell is merged into the first cell in the identified set, while the contents of the other cells in set become void, in order to preserve the consistency of the arrays defining the Voronoi diagram. The merge process repeats as long as there are vertices that can lead to identifying color similar sets of cells, with the newly created cells taking part in the next iterations of the merge process (Fig. 4).

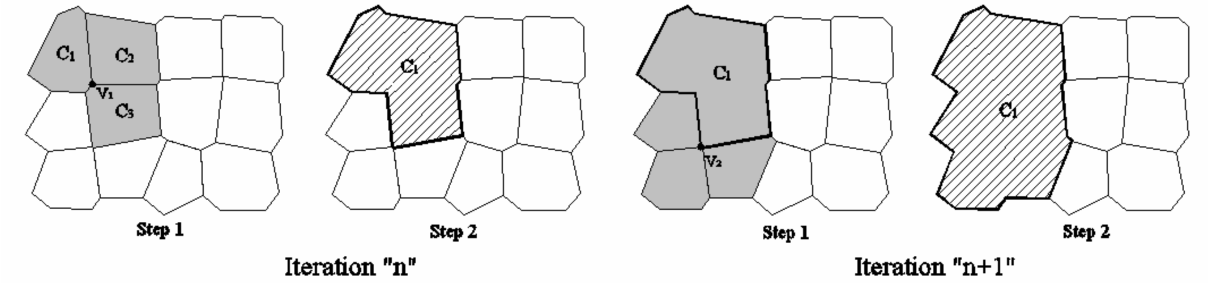
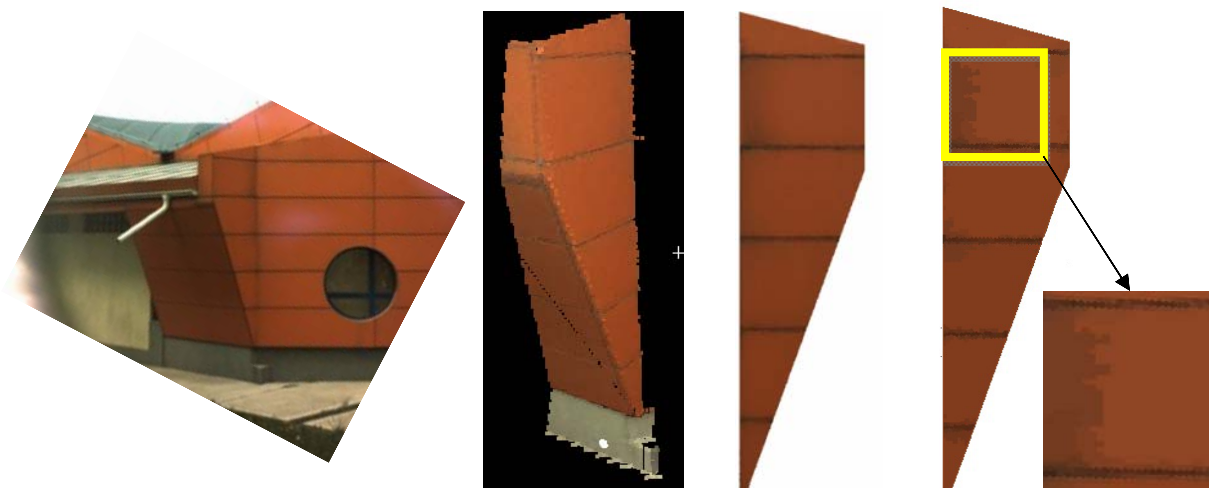


Fig. 4. Texture growing sequence

At iteration "n", starting from a set of three cells C_1, C_2 and C_3 identified by the a common vertex V_1, based on their colors being similar by the above presented criteria, the merge process creates a new cell and places its content in the C_1 location. At next iteration, the newly created cell is found to be color similar with a new set of cells around vertex V_2 and the merge process takes place again, reshaping cell C_1 to include the new cells.

5. Results and discussion

The considered test scenario was to apply the algorithm on the component patches of the 3D model of the dean's office building from the "Faculty of Hydrotechnics, Geodesy and Environmental Engineering", Iasi city. The images were acquired from each station point with the terrestrial laser scanner, using a 5.0 megapixel camera already integrated in the system, each point of the point cloud being textured using the well-known collinearity equations. The error margin used for the tests was 2%, corresponding to a PED less or equal to 5.1, on a scale from 0 to 255, with 0 meaning identical colors and 255 the PED between black and white. A test case is presented in Fig. 5.



(a) (b) (c) (d)

Fig. 5. Façade plane (a) The image taken with the camera integrated in the scanner system (b) Textured point cloud (c) Textured surface in AutoCad (d) Surface after texture growing

The obtained texture after applying the texture growing algorithm preserves the geometric and color properties of the original texture with a good fidelity, while reducing the computational representation to aproximately 1/10 of the original amount of cells. While applying the algorithm, a negative side effect can be observed because the texture irregularities get accentuated in the process, the respective area of the texture being marked with a yellow rectangle.

6. Conclusions and future work

By using the Voronoi diagrams for the texture generation of the 3D CAD model, from 3D LIDAR points with the RGB value, this paper attempts to solve the texturing problem without having to drap digital images on the 3D model. The main issue was to reduce the computation complexity of the resulted textures by reducing the amount of needed polygons to represent the Voronoi cells. This process is supposed to not affect the texture fidelity and make possible the texture mapping on a 3D model without stressing the used software (AutoCad).

The implemented texture growing algorithm didn't solve all the existing problems in texturing 3D models, being sensitive to brightness variation, occlusion areas (not visible surface parts of the real object), or color irregularities due weather effects, but has the following advantages:

- The texture representation is entirely based on CAD objects, thus parametric and editable while being based on dimensional elements;
- The colors used to generate the textures maintain a good correspondence to the RGB values of the LIDAR points on which the Voronoi diagram is generated, thus mostly achieving the needed photorealistic effect;
- The representation complexity for the resulted textures is greatly reduced with little cost on the quality of the representation.

This paper merely presents an incipient effort in reducing the computation complexity for the textures, as the tested error margin of 2% for color similarity leads to accentuating the irregularities in the actual building appearance. Further tests have to be done in order to establish an optimum error margin that will ensure the photorealistic effect while still being effective regarding the resulting number of polygons.

Future work must lean towards a study on adjusting the color computation algorithms for the merged cells, as the current solution to average the RGB channels is also propagating the texture irregularities. Since the colors for the scanned LIDAR points are sensitive to the environment lighting – intensity and position of the light sources, the photorealist effect for the 3D model is tied to one viewpoint. Solving this issue can imply filtering the light variation effect during the texture growing process.

7. References

1. *Alshawabkeh, Y. & Haala, N. – Automatic Multi-Image Photo Texturing of Complex 3D Scenes*. CIPA IAPRS Vol. 34- 5/C34, pp. 68-73, *2005*.
2. *Fortune, S. – A sweepline algorithm for Voronoi diagrams. Algorithmica, 2(2):153-174, 1987.*
3. *Gijsenij, A., Gevers, T., Lucassen, M. P. - A Perceptual Comparison of Distance Measures for Color Constancy Algorithms, in Computer Vision - ECCV 2008, Part I, D.*

Forsyth, P. Torr, and A. Zisserman (Eds.), LNCS 5302, Springer-Verlag , pp. 208–221, 2008.

4. *Hu, C., Wang, Y., & Yu, W. – Mapping digital image texture onto 3D model from LIDAR data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII, Part B5, Beijing, pp. 611-613, 2008.*
5. *Kersten, T., Mechelke, K., Lindstaedt, M., Sternberg, H. – Geometric accuracy investigations of the latest terrestrial laser scanning systems. Integrating Generations, FIG Working Week 2008, Stockholm, Sweden 14-19 June 2008.*
6. *Oniga, E. –A new approach for the semi-automatic texture generation of the buildings facades, from terrestrial laser scanner data. The international Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Melbourne, Australia - to be published.*
7. *Pelagottia, A., Del Mastiob, A., Uccheddub, F. & Remondino, F. – Automated multispectral texture mapping of 3D models. 17th European Signal Processing Conference (EUSIPCO 2009) Glasgow, Scotland, pp. 1215-1219, 2009.*
8. *Quintero, M., S., Genechten, B., Bruyne, B., Poelman, R., Hankar, M., Barnes, S., Caner, H., Budei, L. Heine, E., Reiner, H, García, J. L., Biosca Taronger, J. M. – Theory and practice on Terrestrial Laser Scanning. Training material based on practical applications, prepared by the learning tools for advanced three-dimensional surveying in risk awareness project (3DRiskMapping), Version 4 June 2008.*
9. *www.leica-geosystems.com.*