

A NEW APPROACH FOR THE SEMI-AUTOMATIC TEXTURE GENERATION OF THE BUILDINGS FACADES, FROM TERRESTRIAL LASER SCANNER DATA

E. Oniga^a

^aDepartment of Terrestrial Measurements and Cadastre, Technical University „Gheorghe Asachi” of Iasi, D. Mangeron Street, Iasi, Romania - ersylya83@yahoo.com

KEYWORDS: Point Cloud, Building, Digital Image, Texture, Laser scanning, Photo-realism.

ABSTRACT:

The result of the terrestrial laser scanning is 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. The color code for each LIDAR point is taken from the georeferenced digital images, taken with a high resolution panoramic camera incorporated in the scanner system. In this article I propose a new algorithm for the semiautomatic texture generation, using the color information, the RGB values of every point that has been taken by terrestrial laser scanning technology and the 3D surfaces defining the buildings facades, generated with the Leica Cyclone software. The first step is when the operator defines the limiting value, i.e. the minimum distance between a point and the closest surface. The second step consists in calculating the distances, or the perpendiculars drawn from each point to the closest surface. In the third step we associate the points whose 3D coordinates are known, to every surface, depending on the limiting value. The fourth step consists in computing the Voronoi diagram for the points that belong to a surface. The final step brings automatic association between the RGB value of the color code and the corresponding polygon of the Voronoi diagram. The advantage of using this algorithm is that we can obtain, in a semi-automatic manner, a photorealistic 3D model of the building.

1. INTRODUCTION

1.1 General Assertions

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. Data acquisition, as well as textures generation, imply manual steps. Hence, 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 photocaleras.

The textured 3D models are being used in a variety of applications such as: navigation, visualization, computer animation (virtual studio production, trick film production, advertising, driving/flight simulators, etc.), archaeology, architecture, dentistry, education, fashion and textiles, reverse engineering and also in preservation and documentation of cultural heritage.

After obtaining a 3D model, it is necessary to map textures over it in order to obtain a photorealistic result. Textures are available as a collection of images, which often are the same images used for 3D reconstruction. For this reason, some commercial 3D systems such as: Cyclone, produced by Leica Geosystems, SceneVision-

3D, produced by 3rdTech Inc., Bentley CloudWorx, produced by Bentley Systems Inc., Faro Scene, produced by Faro Technologies, ISiTE Studio, produced by I-SiTE, RiSCAN PRO and Phidias, produced by Riegl, RealWorks Survey, produced by Trimble Navigation, LFM Modeller, produced by Z+F, already provide model-registered color texture by capturing the RGB values of each LIDAR point with a camera integrated in the system. Some other software, creates 3D textured models by only using multiple images taken with a digital camera.

The software that create textured 3D models of the buildings based only on photographs are presented in Table 1.

1.2 Overview to Related Work

Several methods for texturing a 3D model have been proposed in literature. The most frequently used method is to project photographs, by calculating the camera's calibrated parameters (the exterior and interior orientation elements), based on finding correspondences between images or between an image and a 3D surface (point cloud).

In this sense, Chunmei Hu (Chunmei Hu et. al., 2008) develops an experimental system to realize the data fusion between laser scanning data and digital image, in order to map the corresponding image onto point cloud model and TIN model, using the DLT (Direct Linear Transformation) of iterative least squares to calculate the precise values of parameters, and creates the textured point cloud model and the textured TIN model. At the same time, it projects this image to the basic level of range image to make out an orthophoto map. The texturing of 3D point clouds

(namely point-based techniques) allows a faster visualization, but for detailed and complex 3D models it is not an appropriate method (Pelagottia et. al, 2009).

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 such image and the chosen texture, with a very robust registration algorithm, based on Maximization of Mutual Information.

System	Developer/ Researcher	Input data	Description
Photo Modeler	Eos Systems Canada	One or more images	Measurement and precise 3D modeling using one or multiple images
Photo Modeler Scanner	Eos Systems Canada, 2010	Multiple images	Obtaining scans of facades, esp. those with complex shapes and natural textures
Shape Capture	ShapeQuest Inc.	Multiple images	Accurate measurement, detailed 3D modeling
Shape Scan	ShapeQuest Inc.	Multiple images	Dense stereo matching to obtain point cloud data from images
3DImage	Dr. Eng. Ahmed Abdelhafiz	Multiple images	Automatic coloring the laser scanner point cloud and texturing the mesh models
Photo3D	SoftCube Co., Japan	One or more images	Creates 3D models from photographs
SketchUp	Google	Multiple images	Creates 3D models of buildings and not only

Table 1. Some of the existent commercial software for 3D textured models creation using photographs

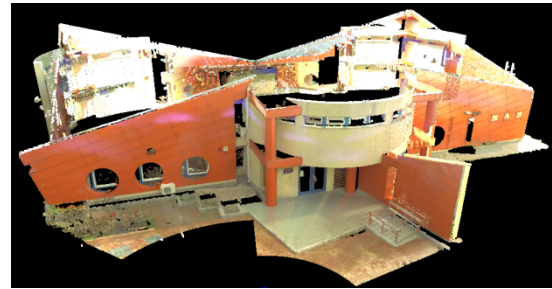
The point cloud modelling using the Leica Cyclone software starts with the segmentation procedure in which the scanned points are approximately grouped dependent on the surface they belong to. After that, the surfaces constituted from a group of adjacent points which fit well into the surface and define distinctive elements of the facades are delimited using the Region Grow command. But the color information of each point is not further used for texturing the surfaces. An aleatory color from the software palette is used resulting in a 3D model with a texture that has no correspondence with the reality. This issue requires some supplementary steps like using another soft to create the final textured model (for example Autodesk 3ds Max), but this solution does not reflect entirely the

terrain reality and the results prove unsatisfying. In this article we propose a new algorithm for the semiautomatic texture generation, using the color information, the RGB values of every LIDAR point that has been taken by terrestrial laser scanning technology and the 3D surfaces defining the buildings facades, generated with the Leica Cyclone software.

For this case study the dean's office building from the “Faculty of Hydrotechnics, Geodesy and Environmental Engineering”- „Gheorghe Asachi” University of Iasi, Romania, has been chosen.



(a)



(b)

Figure 1. The study zone (a) Digital image (b) Textured point cloud, resulted from scanning process with ScanStation2 laser scanner

2. THE 3D MODEL GENERATION

2.1 Terrestrial Laser Scanner Surveying

For this case study, the dean's office building from the “Faculty of Hydrotechnics, Geodesy and Environmental Engineering” was scanned for three hours, with ScanStation2 laser scanner, achieving five scans (ScanWorlds) taken from five different scanning stations, each of them lasting approximately 45 minutes. The scanning stations were made as a tachymetric traverse, having the base (SW1-7000), with known orientation. For bringing all the 5 stations in a common system the direct georeferencing process was necessary. After the calculation of the tachymetric traverse, the following errors were obtained: $e_x=0.009m$, respectively $e_y=0.023m$. The data base obtained after the georeferencing process (Iasi.imp) was imported in the Cyclone software for data processing.

2.2 Data from ScanStation2 Processing

An important operation in post-processing is the filtering of the resulted data, so in this way, the measured points, additionally

taken by the instrument, were manually eliminated, being easily identified by analyzing the scanned area.

Modelling the point cloud was performed by using lines and surfaces with the existing functions in the Cyclone v. 6.0 software. The surfaces created in Cyclone were exported as a *.dxf file which was imported into AutoCAD 2009 platform and after that the 3D model of the building was created (Figure 2). In order to apply the textures, the 3D model, obtained in AutoCAD 2009, needs to be imported into another software (for example Autodesk 3ds Max), in which the textures are created using fragments from the digital photos of the building facades.

To eliminate this inconvenience we propose a new algorithm for the texture generation which doesn't require more images or any other software.

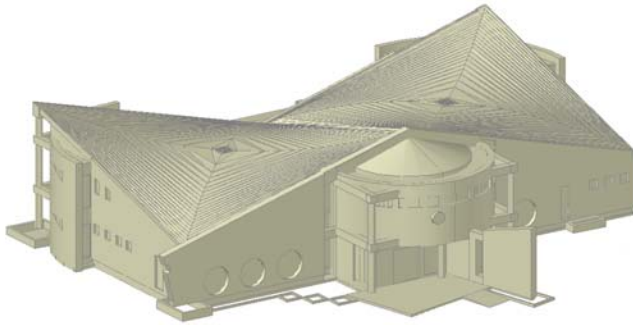


Figure 2. The 3D model, without texture, of the dean's office building, Faculty of Hydrotechnics, Geodesy and Environmental Engineering, Technical University „Gheorghe Asachi” of Iasi

3. THE PROPOSED ALGORITHM

3.1 Design of the Algorithm

The research task is to create the texture 3D model of the building. The implemented approach can be subdivided into five steps, as follows:

- 1). Defining the minimum distance between a point and the closest surface.
- 2). Calculating the distances drawn from each point to the closest surface and the X, Y, Z coordinates of the feet of the perpendiculars.
- 3). Eliminating the points that do not belong to the plan, depending on the limiting value and those who are outside the polygon boundary.
- 4). Calculating the Voronoi diagram for the points that belong to a surface.
- 5). The final step brings automatic association between the RGB value of the color code and the corresponding polygon of the Voronoi diagram.

The input data for the suggested algorithm are:

- a .txt file containing the following information: the point number, the X, Y and Z coordinates corresponding to the terrain system of coordinates, the reflectance value and the RGB color code.
- a .dxf file, exported by the Cyclone software, containing the plans created by this software.

The Cyclone software interpolates the selected points to a surface which can be: a patch, a cylinder, a sphere or a smooth surface, but for the presented algorithm we will take in consideration only plan surfaces of varying shape.

The logical diagram of this algorithm is presented in Figure 3.

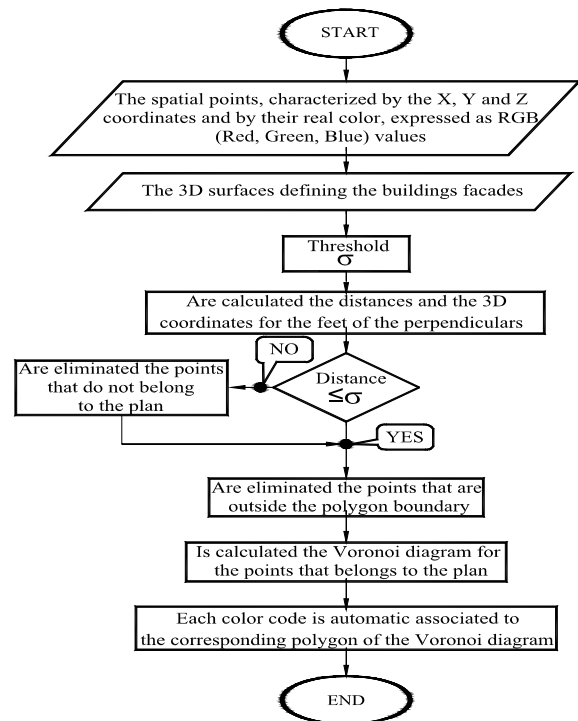


Figure 3. The proposed algorithm logical diagram

3.3 Detailed Implementation of the Algorithm

3.3.1 Defining the Threshold: When calculating the Voronoi diagram it is important all the points to be contained in the plan surface that is about to be textured. The points that are interpolated to that plan are not situated on the plan and for this reason they have to be projected on it. To eliminate the points situated too far from the plan, the operator establishes a threshold limit which represents the maximum distance admitted.

3.3.2 Calculating the Distances: This step consists in calculating the distances drawn from each point to the closest surface and the X, Y, Z coordinates for the feet of the perpendiculars. The general equation of the plan is as follows:

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \quad (1)$$

where: A, B, C, D= the plan parameters

The distance from the point $P_i(x_i, y_i, z_i)$ to the π plan defined by relation (1) is:

$$d(P_i, \pi) = \frac{|Ax_i + By_i + Cz_i + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (2)$$

To calculate the coordinates for the foot of the perpendicular leading from the point P_i to the plan, the following relation was used:

$$\begin{cases} x_0 = x_i + \lambda A \\ y_0 = y_i + \lambda B \\ z_0 = z_i + \lambda C \end{cases} \quad \lambda = -\frac{Ax_i + By_i + Cz_i + D}{A^2 + B^2 + C^2} \quad (3)$$

where: x_0, y_0, z_0 = the coordinates of the foot of the perpendicular;
 x_i, y_i, z_i = the coordinates of the starting point of the perpendicular;
 A, B, C, D= the plan parameters;
 λ = constant.

3.3.3 Eliminating the Points that do not Belong to the Plan:

To obtain the correct texture of the plan the points that do not belong to it must be eliminated. This points are the ones situated at a higher distance than the threshold limit and the ones situated outside the polygon area. To eliminate the points outside the plan area we calculate the determinant $((x_i, y_i, 1), (x_{i+1}, y_{i+1}, 1), (x_p, y_p, 1))$ for $i=1$ to N values (when $i=N$, $i+1=1$ is considered). the point $P(x_p, y_p)$ is situated inside the polygon if and only if all the determinants have the same sign. Otherwise, the point is situated outside the polygon which defines the plan.

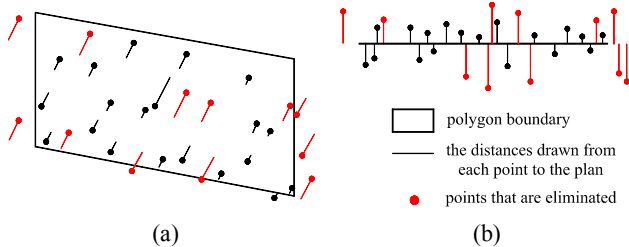


Figure 4. Calculating the distances drawn from each point to the plan (a) Perspective view (b) Top view

3.3.4 Calculating the Voronoi Diagram: In order to calculate the Voronoi diagram for the points that belong to a surface, Fortune's algorithm was used.

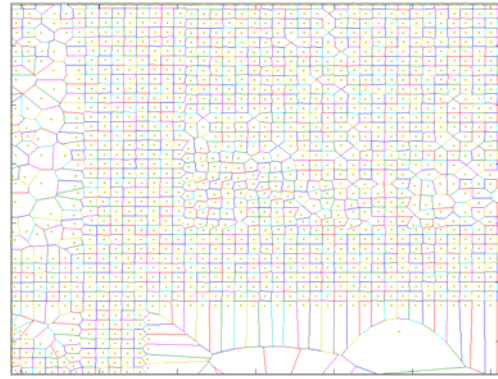


Figure 5. Voronoi diagram polygons calculated for the points that belong to the window's plan (bottom-right corner)

3.3.5 Automatic Coloring of the Voronoi Diagram Polygons:

The final step brings automatic association between the RGB value of the color code and the corresponding polygon of the Voronoi diagram.

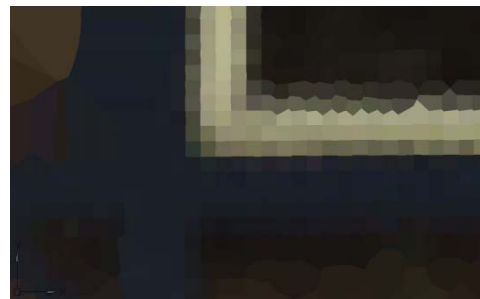


Figure 6. Voronoi diagram polygons calculated for the points that belong to the window's plan, with the corresponding RGB value

2. VORONOI DIAGRAM

In mathematics, a Voronoi diagram is a special kind of decomposition of a given space, e.g., a metric space, determined by distances to a specified family of objects (subsets) in the space. These objects are usually called the sites or the generators (but other names such as "seeds" are in use) and to each such an object one associates a corresponding Voronoi cell, namely the set of all points in the given space whose distance to the given object is not greater than their distance to the other objects. Voronoi diagrams can be found in a large number of fields in science and technology, even in art, and they have found numerous practical and theoretical applications (Aurenhammer, 1991).

In the simplest and most familiar case (the Voronoi diagram of a random set of points in the plane), we are given a finite set of points $P := \{p_1, \dots, p_n\}$ in the Euclidean plane. In this case each site p_k is simply a point, and its corresponding Voronoi cell (also called Voronoi region or Dirichlet cell) R_k consisting of all points whose distance to p_k is not greater than their distance to any other site. Each such cell is obtained from the intersection of

half-spaces, and hence it is a convex polygon. The segments of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest sites. The Voronoi vertices (nodes) are the points equidistant to three (or more) sites.

The formal definition of the Voronoi diagram is: Let X be a space (a nonempty set) endowed with a distance function d . Let K be a set of indices and let $(P_k)_{k \in K}$ be a ordered collection of nonempty subsets (the sites) in the space X . The Voronoi cell, or Voronoi region, R_k , associated with the site P_k is the set of all points in X whose distance to P_k is not greater than their distance to the other sites P_j , where j is any index different from k . In other words, if $d(x,A)=\inf\{d(x,a): a \text{ in } A\}$ denotes the distance between the point x and the subset A , then

$$R_k = \{x \text{ in } X: d(x, P_k) \leq d(x, P_j) \text{ for all } j \neq k\} \quad (4)$$

To implement Voronoi diagram the Matlab programming language and Fortune algorithm with time related complexity $O(n \log n)$ have been used.

4. RESULTS AND DISCUSSION

The described algorithm was tested 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 point station of the terrestrial laser scanner, using a camera already integrated in the system, that has 5 mega pixel resolution, and each point of the point cloud was textured using the well-known collinearity equations:

$$\begin{aligned} x' &= \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \\ y' &= \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} \end{aligned} \quad (5)$$

where: x', y' = image coordinates
 X, Y, Z = the object point coordinates in ground coordinate system
 X_0, Y_0, Z_0 = coordinates of projection center
 $r_{11}, r_{12}, r_{21}, r_{22}, \dots$ = the coefficients of the rotation matrix as function of k, φ, ω .

This resolution, of 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 ideal conditions for taking the images are not always met, especially for outdoor applications. In this case, the large time differences will result in varying light conditions and changing shadows, thus the recorded 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 at optimal time for texturing.

For the first surface, a plan that is part of the secondary façade, a threshold of 1.5 cm has been chosen and a number of 4430

polygons have been created. For the second surface, a window from the front façade of the building, a threshold of 3 cm has been chosen, due to the fact that many points were reflected by the surface, and a number of 8540 polygons have been created. For the third surface, a plan that is part of the front façade, a 2 cm. threshold has been chosen and a number of 90653 polygons have been created and for the last surface a number of 140896 polygons have been obtained using a threshold of 1.5 cm.

The experimental results are shown in Figures 7, 8, 10 and 11.

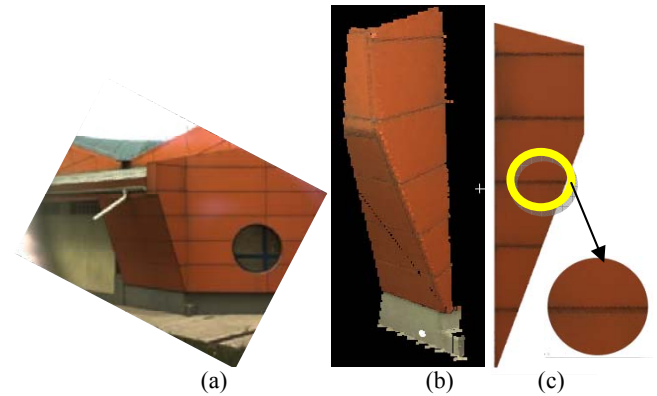


Figure 7. A plan that is part of the secondary façade of the building (a) The image taken with the camera integrated in the scanner system (b) Textured point cloud (c) The surface textured in AutoCad by using the presented algorithm

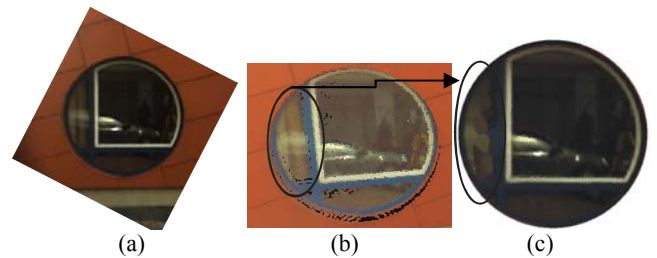


Figure 8. Window from the front façade of the building (a) The image taken with the camera integrated in the scanner system (b) Textured point cloud (c) The surface of the window textured in AutoCad by using the presented algorithm

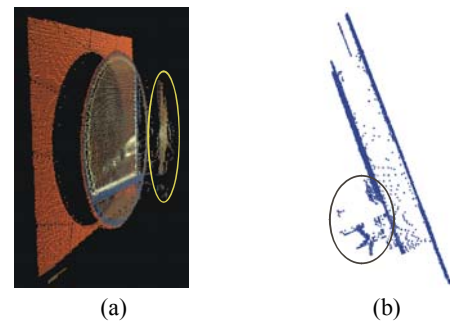


Figure 9. Point cloud representing the surface of the window (a) Perspective view (b) Top view

We can observe in Figure 8 (c) that on the left side of the window's surface the Voronoi cells are quite big comparing to the rest, this due to the fact that the points from that area are situated at a longer distance from the plan than the threshold established by the operator; in this case the threshold was 2 cm. These points can be very well noticed in Figure 9.

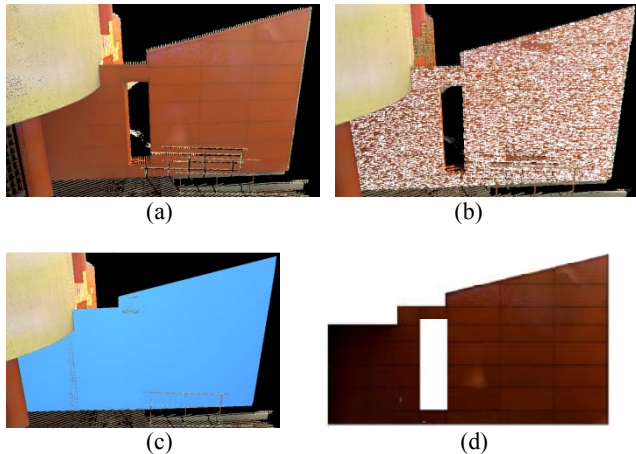


Figure 10. A plan that is part of the front façade of the building
(a) Textured point cloud (b) Creating the patch in Cyclone software (c) Textured patch in Cyclone software using an aleatory color from the software palette (d) The surface textured in AutoCad by using the presented algorithm



Figure 11. A plan that is part of the front façade of the building
(a) The digital image (b) The surface textured by using the presented algorithm

5. CONCLUSIONS

A new algorithm used for the texture mapping, from 3D LIDAR points with the RGB value onto 3D CAD models, has been presented. The algorithm in discussion didn't solve the existing problems in texturing 3D models, which are: brightness correction, eliminating occlusion areas (not visible surface parts of the real object), but has the following advantages:

- 3D models creation and texturing are made semi-automatically by using only two software, Leica Cyclone and AutoCAD Map 3D to which we have implemented the presented algorithm, without taking more images.
- this algorithm combines CAD with 3D textured scan data so you can create parametric, editable solid models of virtually anything you scan.

If this algorithm were implemented in the Cyclone software, the user would be capable to create textured 3D models using the textured point clouds obtained by laser scanner technology directly into this software, thus reducing the time required to process the scan data.

At the same time, the system should also be improved so that it can be used for other types of surfaces like cylinders, spheres, NURBS, this thing being possible by using the 3D Voronoi diagram. The future work in developing this theme will be directed to obtain "true color" images of the 3D model, by eliminating the shadows and the occlusion areas and by adopting an optimum illumination source for each façade of the 3D model or by choosing the optimum position of the illuminating source, regardless the conditions in which the scanner system is taking the images

REFERENCES

References from Journals:

- Alshwabkeh, Y. & Haala N., 2005. Automatic Multi-Image Photo Texturing of Complex 3D Scenes. *CIPA IAPRS Vol. 34-5/C34*, pp. 68-73.
- Aurenhammer, F., 1991. Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, 23(3):345-405, 1991.
- Hu, C., Wang, Y., & Yu, W., 2008. 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
- Pelagottia, A., Del Mastiob, A., Ucheddub, F. & Remondino, F., 2009. Automated multispectral texture mapping of 3D models. *17th European Signal Processing Conference (EUSIPCO 2009)* Glasgow, Scotland, pp. 1215-1219.
- Zeng, J. & Li, T., 2011. An Algorithm of 3D Texture Synthesis based on Point Cloud. *Journal of Computational Information Systems* 7:5, pp. 1592-1599

References from Books:

- McGlone, J., 2004. *Manual of Photogrammetry*. Fifth Edition, American Society for Photogrammetry and Remote Sensing.
- Kasser, M., & Egls, Y., 2002. *Digital Photogrammetry*. Taylor & Francis, London.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to prof. Florea Zavoianu for his unconditional support and contribution to the presented work. I am thankful to Dumitru Onose, Adrian Savu and Aurel Negri (UTCB, Romania), for their help in scanning the building, and also to Marius Sutu (UTI, Romania) for his support in programming the algorithm.