

# MULTIGRID CELL SHAPE EVALUATION IN DTM FILTERING OF MOBILE LIDAR POINT CLOUDS

Luis Gézero<sup>1,2</sup>, Carlos Antunes<sup>1,3</sup>

<sup>1</sup> Faculty of Sciences of the Lisbon University, Campo Grande 1749-016 Lisbon Portugal

<sup>2</sup> LandCOBA, Digital Cartography Consulting Lda. - [l.gezero@cobagroup.com](mailto:l.gezero@cobagroup.com)

<sup>3</sup> Dom Luiz Institute, Faculty of Sciences of the Lisbon University, Campo Grande 1749-016 Lisbon, Portugal - [cmantunes@fc.ul.pt](mailto:cmantunes@fc.ul.pt)

Commission IV, WG IV/1

**KEY WORDS:** Point cloud, Mobile LiDAR, Digital terrain models, Filtering, Delaunay triangulation, Barycentric coordinates

## ABSTRACT:

The georeferenced information acquisition for large scales digital terrain models generation is usually a rather time-consuming and costly process. The use of point clouds gathered by mobile LiDAR systems therefore appears as a possible source for the creation of those models, particularly in remote areas where security issues requires a maximum speed field collection information. However, the filtering challenge to separate the points that represents the terrain surface from the remaining, in an automatic and consistent way, is an open challenge that continues to arouse the interest of researchers. Using a multigrid filtering method, this work presents a comparative analysis of different shapes of regular grid cells, including the implementation algorithms for each cell shape. Also, a coordinate grid concept applied to each cell shape is proposed. Finally, a discussion about the obtained results of the method application using point clouds collected in different environments from different ground LiDAR systems are presented.

## 1. INTRODUCTION

The Digital terrain Model (DTM) is probably the single-most universally utilized geographical information source used in disciplines such as geology, geography, geophysics, biology and environment, engineering, land management, etc..

In last decades Mobile LiDAR Systems (MLS), rises as a very efficient technique in the acquisition of precise and dense point clouds. These systems, especially the MLS allows fast collection of large amounts of data along the roads, reducing the fieldwork time, when compared with classic topographic and photogrammetric methods. By lowering the acquisition costs and increasing safety, this technology can then be a solution for gathering fast and precise information that can be used to large scale DTM generation.

However, LiDAR, is a non-selective technique, i.e., the georeferenced point clouds represent the surrounding physical reality at an acquisition moment, indiscriminately, with no classification including: terrain, vegetation, or any other object within the system range. Therefore, to use such type of data for DTM generation, it is necessary to identify and separate the terrain points from the remaining points of the cloud.

This process has been, in the last years, designated in the literature as filtering (Chen et al., 2013, Mongus and Zalik, 2012, Hu et al., 2015). The development of the filtering processes has challenged many researchers since the beginning of the first LiDAR systems.

Chen et al., 2017, presents a good recent review of DTM methods generation, including the filtering algorithms. These methods are classified into six categories: surface-based adjustment; morphology-based filtering, triangulated irregular network (TIN)-based refinement, segmentation and classification, statistical analysis and multi-scale comparison.

Most of the published work in this field of study is applied to point clouds exclusively collected from Aerial LiDAR Systems (ALS) ((Chen et al. 2017, Özcan and Ünsalan, 2017). However, the point clouds collected by these systems have very different characteristics from the collected by MLS, such as, the different collection angle, the higher information density and the complexity of the environment surrounding the system. Based on that, most of the DTM creation algorithms applied to ALS data do not work or are not very efficient for the data collected by MLS. Nevertheless, there are some studies that adapted filters initially designed for ALS or have applied specific algorithms to ground systems collected data (Pfeifer and Mandlbürger, 2008; Fellendorf, 2013; Vallet and Papelard, 2015; Tyagur and Hollaus, 2016; Yang et al., 2016). One of the most well-known methods type, is based on the classification of the lowest point within a cell as ground point. Then, the rest of the points are classified as ground points and non-ground points by analysing the spatial correlation between unclassified points and pre-set ground points. The main assumed rule is: the closer two ground points are, the smaller their elevation difference is. In other hand, if two nearby points have very different elevations, assuming the point with the lowest elevation is on the ground, there is a greater probability that the highest elevation point does not represent or is not part of the terrain.

The present work intends to contribute in this research topic, by presenting a comparative analysis of different regular and irregular cell shapes, starting from a filtering method based in a recursive grid division of the space presented by Gézero and Antunes (2017). The method uses a recursive division of the space, through consecutive iterations where the subsequent cells results from an initial cell division, called along this work, multigrid process. The method links only a cloud point to a

grid cell, however all points in the cloud must be iteratively processed. Therefore, the global method efficiency depends directly of the implementation effectiveness of the condition determination if a specific point cloud is inside the cell, which given the amount of cloud points and grid cells can be very slow. Instead of using directly the point and cell polygon coordinates to check that condition, an alternative method is proposed, based on the grid coordinates row and a column. This proposed algorithm turns easy the elevation comparison between two consecutive cells divisions by changing the cell size value and decrease drastically time consuming of the overall method. Based on this method, the implementation algorithms and a comparative evaluation study of different regular cell shapes is presented. Also, a triangular irregular grid space division is presented, keeping the multigrid concept.

This paper is composed as follows. In section 2, a description of the different regular cell shapes is presented, namely, quadrangular, hexagonal and triangular. An irregular triangular multigrid is also presented, using 3D triangular cells, being the distance to the triangle plane used instead of the elevation point value used in the 2D regular grid cells. The different implementation algorithms are described thought the corresponding pseudo-codes. In section 3, a brief description of the method operation is presented, in the way to help the understanding of the overall method working process flow. The obtained results evaluation in different environments and discussion are described along section 5. Finally, conclusions and future work are presented in the section 6.

## 2. REGULAR AND IRREGULAR GRIDS

Along this work, a grid is considered to be regular when all its cells are equal, not necessarily symmetric, three cell shapes are compared: quadrangular, hexagonal and triangular. Regardless the cell shape, an essential step to implement the method outcomes from the need to identify in which cell grid a point cloud is located. Instead of using the coordinates of the cell polygon, a different implementation is proposed, based in the determination of the grid coordinates row and column (Row, Col). The problem of assigning a cloud point to a cell, based on is planimetric coordinates (X, Y), is then reduced to find the grid coordinates of the cell.

### 2.1 Quadrangular cells

The use of quadrangular cells (particularly squares) is clearly the most intuitive way to divide space on a regular basis. The easy implementation and abstraction of this type of cells makes them the most popular way of representation through a regular grid. The square pixels associated to the raster format of an image are one paradigmatic case of this representation. The formulas to find the grid coordinates of a cell in which a point is inside based on is planimetric coordinates, is described in (1).

$$\begin{cases} \text{Row} = \text{Int}\left(\frac{y}{D_{\text{cel}}}\right) \\ \text{Col} = \text{Int}\left(\frac{x}{D_{\text{cel}}}\right) \end{cases} \quad (1)$$

Where  $D_{\text{cel}}$  – cell size dimension  
 Row – grid row containing the point  
 Col - grid column containing the point  
 X, Y – cloud point planimetric coordinates

### 2.2 Hexagonal cells

A known advantage of using hexagonal cells is related to the determination of their neighborhood. Unlike quadrangular and triangular cells, the distance to all hexagonal neighbors is the same, turning easier, without any ambiguity, the application of this cell shape in methods where the neighborhoods are needed. However, the implementation of hexagonal cells grid is much more complex than quadrangular, since they are not orthogonal to a coordinate system.

In Figure 1, is presented the grid coordinate system defined for the hexagonal cells. The columns are vertical, but the rows are not horizontal, which increases the implementation complexity. Should be noted that the presented version of hexagonal cells is usually called horizontal, that is, the top of each cell is a horizontal edge. There is another version, called vertical, resulting from a 90° rotation of the cells, where the top of each cell is a vertex (also called flat or pointy). The presented algorithm can easily be applied to vertical hexagonal cells, inverting the rows and columns, in that case the rows will be horizontal, but the columns will be not vertical.

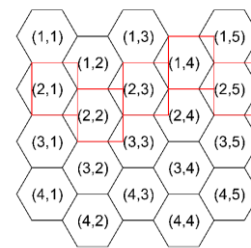


Figure 1. Hexagonal grid cell coordinates and candidate's rows and columns (red rectangles).

Since the hexagonal cells height (h) and width (w) are different, the algorithm first step is to calculate the relationship between h and w. The triangle T, represented in Figure 2, is a right triangle, i.e., the relationship between its height and base is  $1 / \sqrt{3}$ , and the total width of the hexagon will be four times the value of the base of triangle T, then the relationship between h and w can be represented thought (2).

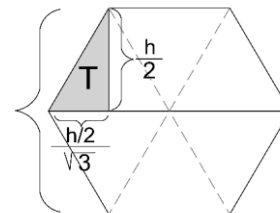


Figure 2. Hexagonal cell weight and height relationship

$$w = 4 \times \frac{h/2}{\sqrt{3}} \quad (2)$$

Where  $w$  - hexagonal cell width  
 $h$  - hexagonal cell Height

The grid coordinates (Row, Col) determination algorithm is then performed in tree steps. In step 1, a rectangular cell candidate is identified (Cc, Rc), whose height is equal to the hexagonal cell height (h) and the width are tree-fourths of the of the hexagonal cell width (red rectangles of Figure 1 and 3). Since the rows are not orthogonal, it is necessary to separate

the cells columns into even and odd. The step 1 formulation is presented in (3). In step 2 its verified if the point is inside the cell corresponding to the rectangle or is in the above or below row, that is, in the areas green and blue inside the rectangle of Figure 3.

Based in the same figure, if the point is in the left fourth of the hexagon width (otherwise the grid coordinates are equal to the candidate coordinates  $C_c$  and  $R_c$ ), it is necessary to check if the point is inside the green or in blue areas. Since booth areas are right triangles, the problem is reduced to check if the point is in the top or bottom half of the left fourth of the hexagon and if is inside those right triangles. The step 2 formulation is presented in (4).

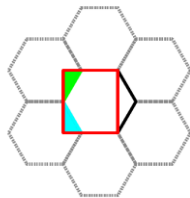


Figure 3. Auxiliary rectangle and above (green) and below (blue) areas

Finally, in step 3, the candidates grid coordinates ( $C_c$ ,  $R_c$ ) are adjusted and the row and col determined, the step conditions are presented in (5).

**Step 1:** (3)

Boolean A =False  
 Boolean B =False

$$C_c = \text{int} \left( \frac{X}{0.75w} \right)$$

$$\text{If } C_c \text{ is even } \begin{cases} R_c = \text{Int} \left( \frac{Y}{h} \right) \\ D_x = X - R_c \end{cases}$$

$$\text{If } C_c \text{ is odd } \begin{cases} R_c = \text{Int} \left( \frac{Y-h}{h} \right) \\ D_x = X - R_c - 1 \end{cases}$$

**Step 2:** (4)

$$D_y = y - C_c$$

$$\text{if } D_x < \frac{w}{4} \begin{cases} \text{if } D_y < \frac{h}{2} \text{ and } \frac{D_x}{D_y} > \sqrt{3} \Rightarrow B \text{ is true} \\ \text{if } D_y \geq \frac{h}{2} \text{ and } -\frac{D_x}{D_y} > \sqrt{3} \Rightarrow A \text{ is true} \end{cases}$$

**Step 3:** (5)

$$\begin{cases} \text{If } A = \text{false and } B = \text{false} \Rightarrow \text{Col} = C_c \text{ and Row} = R_c \\ \text{If } A = \text{true or } B = \text{true} \Rightarrow \text{Col} = C_c - 1 \\ \text{If } A = \text{true and } C_c \text{ is even} \Rightarrow \text{Row} = R_c - 1 \\ \text{If } B = \text{true and } C_c \text{ odd} \Rightarrow \text{Row} = R_c \end{cases}$$

Where  $C_c$  - grid column candidate  
 $R_c$  - grid row candidate  
 $h$  - hexagonal cell height  
 $w$  - hexagonal cell width,  
 $X, Y$  - cloud point planimetric coordinates

**2.3 Triangular cells**

The triangle is the most basic unit of all geometrical patterns, since a rectangle, hexagon or any other shape polygon can be decomposed into a series of triangles.

The proposed regular triangular grid uses vertical isosceles triangles. As in the hexagonal grid, horizontal triangles can be used, by a rotation of 90 ° and an inversion of rows and columns in the presented calculation. The relation of the isosceles triangles height ( $h$ ) and width ( $w$ ) is presented in (6).

$$w = 2 \frac{h}{\sqrt{3}} \quad (6)$$

Where  $w$  - triangular cell width  
 $h$  - triangular cell Height

In the triangular cells case, it's necessary to distinguish between the triangles pointing up and down. To do so a 0.5 value is added to the value of the previous column if the triangle is pointing down. So, all triangles pointing down will have a half unit column index and all triangles pointing up an integer column index.

Considering that the first column corresponds to the number  $K$ , the column sequence will be:  $K, K + 0.5, K + 1, K + 1.5, K + 2, \dots$ . The triangular grid coordinates scheme is presented in Figure 4.

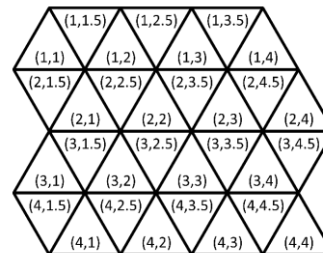


Figure 4. Triangular cells grid coordinates

The algorithm is divided in 2 steps, in the first step presented in (6) a candidate rectangle with the width ( $w$ ) and height ( $h$ ) of the triangle is used to establish the candidate coordinates ( $R_c$ ,  $C_c$ ). In this case the final row value is equal to the candidate row, therefore, only the column value is calculated along the algorithm. In Figure 5, the candidate rectangle is represented in red and the respective formulation is presented in (9). Along the second step the auxiliary rectangle is divided in 3 triangles. The central tested triangle and the remaining both sides right triangles, green and blue, showed in Figure 5.

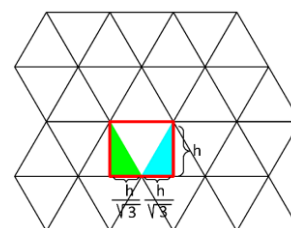


Figure 5. Triangular cells grid coordinates calculation

Dividing the rectangle vertically, a similar situation to the hexagonal cells can be established, by checking if the point lies inside one of the right triangles (blue and green areas of Figure 5). A final adjustment of the column value is then made, adding 0.5 to the cell column index if the triangle is pointing down. The sequential operations and auxiliary values to this calculation are presented in (8).

**Step 1:** (7)

$$\begin{cases} Row = Int\left(\frac{y}{h}\right) \\ Cc = Int\left(\frac{x}{w}\right) \end{cases}$$

**Step 2:** (8)

$$\begin{aligned} Dx &= X - Cc * w \\ Dy &= (Row + 1) * h - y \\ \text{If Row is odd} &\Rightarrow Dy = h - Dy \\ \text{If } Dy > 1 & \\ \begin{cases} \text{If } Dx < \frac{w}{2} \text{ and } \frac{Dx}{Dy} < \frac{1}{\sqrt{3}} \Rightarrow Col = Cc - 0.5 \\ \text{If } Dx \geq \frac{w}{2} \text{ and } \frac{(w - Dx)}{Dy} < \frac{1}{\sqrt{3}} \Rightarrow Col = Cc + 0.5 \end{cases} \end{aligned}$$

Where w - triangular cell width  
 h - triangular cell Height  
 X, Y - cloud point planimetric coordinates

#### 2.4 Irregular grid

Although different approaches, namely quadrangular (Kang, et al., 2015), the most common irregular cell shape to space division and DTM representation, is the triangle, usually called Triangular Irregular Network (TIN). Unlike the regular, the irregular grid requires a pre-established set of initial points along the work area in order to start the recursive triangulation division process. Those initial points can be established using a single initial triangle that contains all the points of the cloud inside (Figure 6), or in order to accelerate the process, a grid of regular cells where the lowest elevation point is identified within each cell, can be used. That set of points are then triangulated and the resulting triangles used to start the recursive triangulation division process.

The triangles division are recursively made, using the point identified within each triangle. Therefore, in each iteration, each triangle is divided in three new triangles.

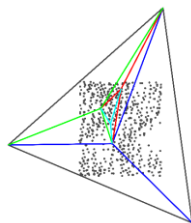


Figure 6. Irregular recursive triangulation

In irregular grids it's not possible to use the coordinate grid concept (Row, Col), but the spatial multigrid relationship between the divided cell and the resulting cells in consecutive iterations is kept.

The concept of point inside a triangle in space can be faced in two different ways, depending if we consider a 2D or 3D approach to the problem. In Figure 7 is represented a triangle T and two points in space, P1 and P2.

Since, the variable used to compare the points inside the cell is the point elevation, if we project the point P1 and P2 along the Z axis, in the X, Y coordinates plane (P1'' and P2''), it may be concluded that P2'' is outside the triangle T' and consequently outside T. In other hand P1'' will be inside T. However, if we consider the projections of P1 and P2 in the triangle plane (P1' and P2'), the opposite can be concluded.

To take advantage of the 3D triangles, especially in terrain slope areas, only the method based in the triangle plane point's projection was considered. The efficient algorithm presented by Heidrich (2005) to compute the triangle barycentric coordinates where used to compute the point projection into the triangle plane and based on that check if the point projection lies inside the triangle.

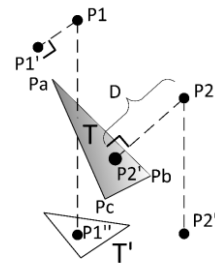


Figure 7 - Point in 3D triangle, and point to plane distance

Based on the triangle T represented in the Figure 7, defined by the points (Pa, Pb, Pc), the algorithm starts by building vectors between two triangle vertices and both a triangle vertex and the point in space P2. The B1, B2 and B3 coefficients, described in (9) are then calculated in way to compute the P2 point projection barycentric coordinates. The verification if the point projection P2' is inside the triangle is reduced to check if that point is in the left side of the triangle edges, i.e., all B1, B2 and B3 coefficients are between 0 and 1.

$$\begin{aligned} \vec{u} &= Pb - Pa \\ \vec{v} &= Pc - Pa \\ \vec{w} &= P2 - Pa \\ \vec{n} &= \text{Cross product}(\vec{u}, \vec{v}) \\ B3 &= \frac{\text{Dot product}(\text{Cross product}(\vec{u}, \vec{w}), \vec{n})}{\text{Dot product}(\vec{n}, \vec{n})} \\ B2 &= \frac{\text{Dot product}(\text{Cross product}(\vec{w}, \vec{v}), \vec{n})}{\text{Dot product}(\vec{n}, \vec{n})} \\ B1 &= 1 - B1 - B2 \end{aligned} \tag{9}$$

$$\text{if } 0 \leq B_2 < 1 \text{ and } 0 \leq B_1 < 1 \text{ and } 0 \leq B_0 < 1 \Rightarrow P2 \text{ in } T$$

Where P - 3d point in space)  
 Pa, Pb, Pc - the tree triangle points  
 B1, B2, B3 - barycentric coordinates of P2 projection in the triangle plane

After checking if the point lies inside the triangle, it is necessary to measure the distance from the point to the triangle

plane. In this case such distance is equal to the distance between the point P2 and its projection in the triangle plane P2''. Then the cartesian coordinates of P2'' can be calculate trough (10).

$$\begin{cases} X'' = B1Xa + B2Xb + B3Xc \\ Y'' = B1Ya + B2Yb + B3Yc \\ Z'' = B1Za + B2Zb + B3Zc \end{cases} \quad (10)$$

Where X'', Y'', Z'' – cartesian coordinates of the point projection in the triangle plane  
 B1, B2, B3 - barycentric coordinates of the point projection in the triangle plane  
 Xi, Yi, Zi - cartesian coordinates of the tree triangle points

Knowing the cartesian coordinates of P2 and P2'', the distance from the point to the triangle plane can be directly calculate by the tridimensional Euclidian distance between these two points.

### 3. METHOD OPERATION AND IMPLEMENTATION

In the overall DTM multigrad method implementation in regular grids, an initial cell size is defined or just the height value in the case of hexagonal and triangular cells. In each iteration, a cell list is created and identified by their grid coordinates (Col, Row) and just a cloud point is associated to each cell. In the first iteration, for each point cloud, the grid coordinates are retrieved, associating to each (Col, Row) pair the minimum elevation point inside each cell. In the follow iterations, a restriction based in a minimum elevation difference regarding the previous cell point is established. The minimum elevation candidate point that fulfils all restrictions, is associate to the new cell. If no point fulfils the restrictions, then the process stops and in the next iteration no point will be considered for the process inside that cell. The method stopping criteria can be defined by a limited number of iterations, a minimum number of points in an iteration or a minimum difference of points between consecutive iterations.

Although the cell size value considered between two consecutive iterations is always half of the previous iteration cell size, the division process is very different regarding the different shapes. Figure 8 shows the different cell shapes division, with half size reduced between two consecutive iterations. Although, both squares and triangles are divided in four new cells, the division is made differently in the cell space. The hexagonal cell division results in seven new cells, furthermore, unlike the others cell shapes the new hexagonal resulting cells do not cover only the previous cell area. This means that the point of some cells can be compared with the neighbor's previous cells and not with the divided cell. This does not represent by itself a problem for the method operation but brings a new variation to the point selection.

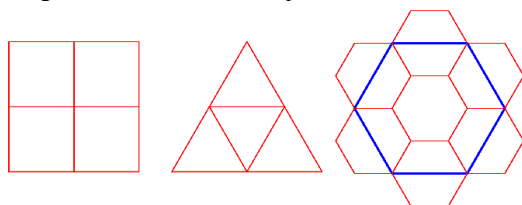


Figure 8. Regular cell division

In the irregular triangular implementation, a relevant aspect of the 3D triangles is the side to which the point is regarding to the triangle plane. The point position relative to the triangle plane, can be obtained by the sign of the dot product between the cross product  $(\vec{u} \times \vec{v})$  and  $\vec{w}$ , defined in (9). Considering two points, if the dot product signs for each point are equal, they lie in some triangle side, on the contrary, they are in opposite sides.

A way to control the side that a point is regarding to a triangle plane is unsure that the triangles points are clockwise ordered. In such case, the dot product will be positive if the point is above a horizontal triangle, and negative if it is bellow that plane. Figure 9 represents a 3D triangle crossing a schematic street curb cross section collected by a point cloud. If the triangle T vertices are clockwise ordered, the defined dot product regarding P1 point is positive and negative for P2 point. In the method implementation, the restrictions are adapted to  $L_{min} < D_i < L_{max}$ , where the candidate point with minimum  $D_i$  is associated to the triangular cell.

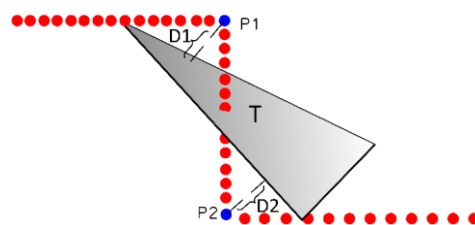


Figure 9. Triangle side illustration

### 4. RESULTS AND DISCUSSION

In the top of the Figure 10, is presented a point cloud collected by a Trimble MX8 MLS, the point cloud has 7800500 points. In the bottom of the same figure the resulting points method application using different cell shapes are presented, squares (green), hexagons (red) and triangles (blue). The presented results are obtained with only two iterations.

At a first glance, the results are very similar. All the points representing elements outside the ground were globally eliminate, i.e., houses, trees, poles, traffic signs, etc.. Also, the resulting point distribution are irregular, less points in flat quasi horizontal areas, and higher density in areas with greater terrain elevation variations. The resulting amount of points, with the initial cell size 1 meter and for each cell type were: 16018 (squares), 17600 (hexagons) and 25310 (triangles).

Despite the same initial cell size used, the difference between the amounts of resulting points are significant, especially between the squares and triangles cells shape. The justification for that, comes from the different areas occupied for each cell shape. Therefore, for the same height value, the space area occupied by the triangles are less and more cells are needed to cover the all area and consequently the resulting DTM contains more points.

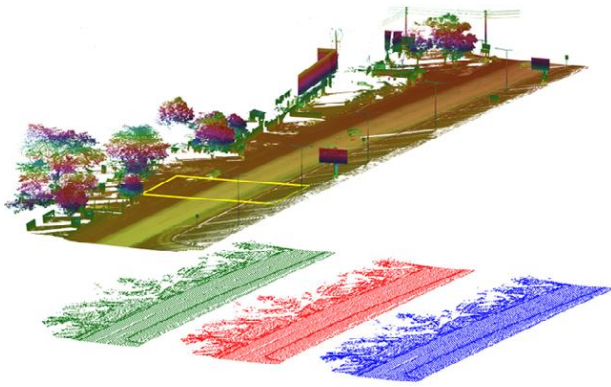


Figure 10. Different cell shape results, squares (green), hexagons (red) and triangles (blue)

To compare the different cell shapes with the same area, a new processing was made using 1 m for the initial values of cell size to the quadrangular cells, 1.075 m for hexagonal cells and 1.316 m for the triangular cells. The resulting amount of points, with the initial cell area of one square meter, for each cell type, were: 16018 (squares), 16223 (hexagons) and 1660 (triangles). In Figure 11 is presented an amplification of the yellow square draw in the Figure 10 point cloud.

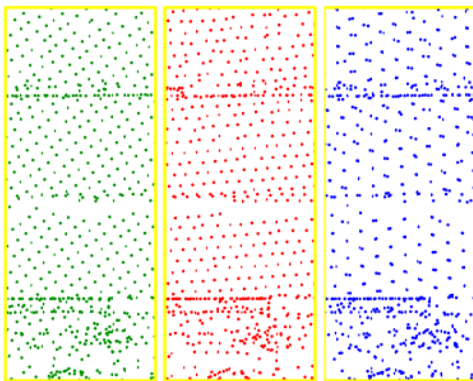


Figure 11. Grid cell shapes resulting points from Figure 10 point cloud yellow zoomed area

In the triangular shape result (blue) of Figure 11, can be observed a "pair of point's" effect, here nearby two by two points appear in flat surfaces. This effect results from the way that the triangles were divided in each iteration, especially due to the middle inverse triangle (Figure 8).

In Figure 12 a gravel road surrounded by dense vegetation is represented by a point cloud collected by a Lynx – Optech MLS. The three presented results are obtained after 4 iterations, with the initial cell area = 4 square meters. The point cloud has 3000500 points, and for the different cell shapes the obtained amount of points were: squares (green) 7587, hexagons (red) 8754 and triangles (blue) 8379.

In this case, the cell shape with highest number of points is the hexagonal grid. That results from a highest space discretization of the hexagons regarding to the other cell shapes (Figure 8). Consequently, more terrain variations between iterations are

detected and more points are included in the final DTM, especially when the number of iterations increases. The resulting points obtained from the hexagons, reveals to be more equidistant, along the road, than the resulting points from the other cell shapes. The reason for that, is the symmetry of the hexagons neighbors, that allows to eliminate the method sensitivity to the terrain inclination.

Although almost all the points representing the vegetation has been eliminated, there are still some points representing vegetation that remains in the resulting DTM points, mostly in the left side of the road. That happens because in that area the points with lower elevation are not in the ground, probably because the emitted LiDAR sensor pulses were reflected in the dense vegetation and did not reach the ground.

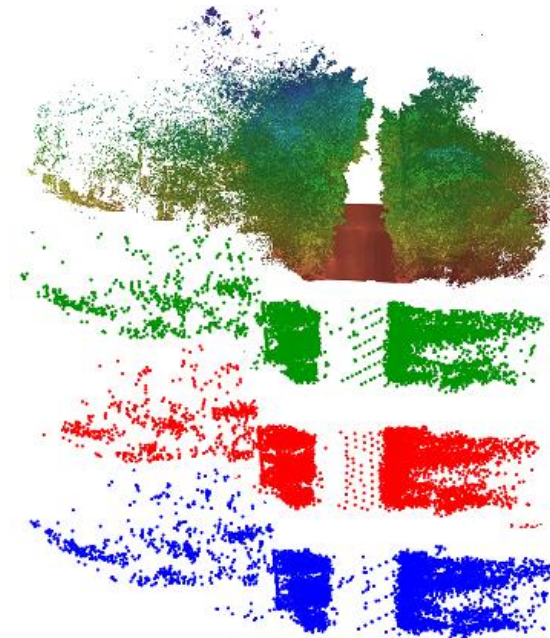


Figure 12. Height vegetation results

A well-known challenge in DTM creation and representation is the terrain modelling along break lines, being the line along the street curb in urban areas a good example. The parked cars along roads are other common issue in DTM in urban environments. The goal of the filtering methods is opposite for these two elements, i.e., keep the curb break-line and remove the cars from the final DTM. In Figure 13 a point cloud collected by a Lynx – Optech MLS, of an urban area are presented. A sidewalk with the respective curb, and a parked truck are collected. In the same figure a triangulation of the resulting points forming the different cell shapes, are shown through a hill shade representation. Once more, for the different cell shapes, the points that represent all elements outside the terrain where almost all eliminated, including the points representing the truck. On the contrary, along the curb, there are a highest point concentration, which allows a better representation of the break-line in the triangulated surface.

After several tests with different size cells and iteration numbers, there are not clear differences between the different cell shapes in the break-line definition. However, significantly

differences are verified in the triangulation method used to represent the resulting points

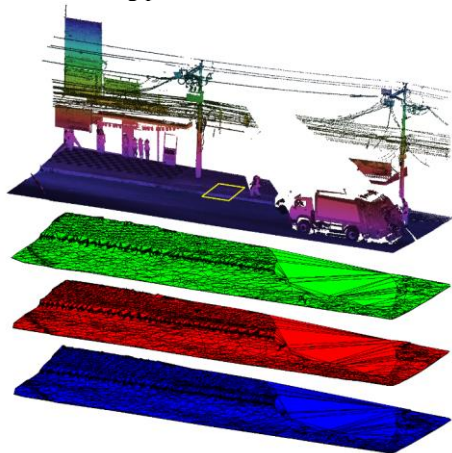


Figure 13. Height vegetation results

The Delaunay triangulation of a point set  $S$  introduced by Boris Nikolaevich Delaunay in 1934, is characterized by the empty circumdisk property: no point in  $S$  lies in the interior of any triangle's circumscribing disk. In two dimensions, it has a striking advantage: among all possible triangulations of set of points, the Delaunay triangulation maximizes the minimum angle. It also optimizes several other geometric criteria related to interpolation accuracy (Cheng et al., 2012). Since only the planimetric coordinates ( $X, Y$ ) are used to define the Delaunay calculations, besides the resulting triangles with 3D coordinates, the Delaunay triangulation are usually known as a 2.5D method. Due its proprieties, this triangulation method is clearly the most used terrain representation through TIN, in the last decades. However, mostly due the resulting point irregular distribution, and the growing cell process from bottom to topo along the terrain, the Delaunay triangulation method reveals some limitations in the curb break-lines representation. Figure 14 shows two different triangulations of an amplification of the yellow square draw in the point cloud of Figure 12. The Delaunay method were used in the triangulation representation. The curb definition is visibly and not well represented, especially the bottom of the break line.

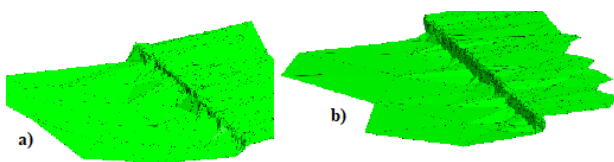


Figure 14. Delaunay triangulation of the DTM resulting points

In Figure 14 b) is presented the exact same points that in Figure 14 a), therefore the triangulation was made using the Poisson reconstruction surface.

The approach of Poisson surface reconstruction is based on the observation that the (inward pointing) normal field of the boundary of a solid can be interpreted as the gradient of the solid's indicator function. Thus, given a set of oriented points sampling the boundary, a watertight mesh can be obtained through 3 steps: 1) Transform the oriented point samples into a

continuous vector field in 3D; 2) Finding a scalar function whose gradients best match the vector field; and, 3) Extracting the appropriate iso-surface (Kazhdan and Hoppe, 2013). In the Poisson surface reconstruction representation, the curb is clearly better defined then in the Delaunay triangulation. The watertight property allows a better top and bottom break-line definition.

In Figure 15 is presented a point cloud sample of a road surrounded by slope areas. The blue points where obtained from the method application using a triangular regular grid after 4 iterations with an initial cell area of 2 square meters. The magenta points result was obtained thought the method application using triangular irregular grid cells. The triangular regular result has 6393 points and the irregular has 2899 points. The significant difference number of points is justified using the point height variable in the regular grid and the triangle plane projection distance in the irregular grid, allowing to decrease the number of points in flat areas.

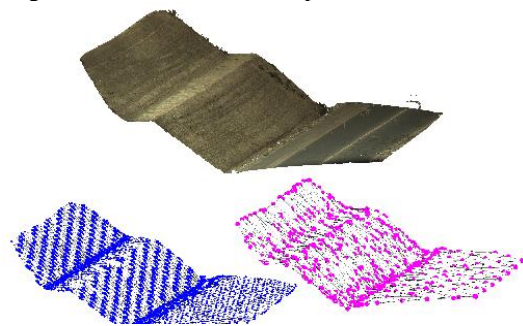


Figure 15. Slope planes comparing results

The drastic execution time difference between the regular cells, where the grid coordinates system was implemented, and irregular cells, is well patent from the analysis of Table 1 values.

| Iterat.       | Regular     |           |             |           |             |           | Irregular   |             |
|---------------|-------------|-----------|-------------|-----------|-------------|-----------|-------------|-------------|
|               | Squares     |           | Hexagons    |           | Triangles   |           | Triangles   |             |
|               | Pnts        | Time (s)  | Pnts        | Time (s)  | Pnts        | Time (s)  | Pnts        | Time (s)    |
| 1             | 141         | 2.6       | 144         | 2.6       | 143         | 2.6       | 143         | 2.6         |
| 2             | 346         | 3.1       | 367         | 4         | 365         | 3.4       | 219         | 178         |
| 3             | 707         | 3.2       | 857         | 4.1       | 878         | 3.4       | 545         | 382         |
| 4             | 778         | 3.3       | 963         | 4.2       | 988         | 3.5       | 1259        | 1073        |
| 5             | 1258        | 3.3       | 1680        | 4.2       | 1651        | 3.9       | 2804        | 3284        |
| <b>Totals</b> | <b>3230</b> | <b>16</b> | <b>4011</b> | <b>19</b> | <b>4025</b> | <b>17</b> | <b>4070</b> | <b>4920</b> |

Table 1. Number of points and execution time statistics

## 5. CONCLUSIONS AND FUTURE WORK

Along this work a multigrid concept is implemented, taking advantage of the grid coordinates (Column and Row), instead of using directly the solid coordinates. This solution makes easy the different cell shapes method implementation, since whole process is similar after the grid coordinates algorithm implementation. A comparative study of different regular grid shapes, namely quadrangular, hexagonal e triangular, is

presented. Also, an irregular triangular grid is implemented, and the results compared with the regular grids. None of the different regular grid shapes comes out as an undoubtedly better solution for the DTM point filtering. Nevertheless, small differences are manifest in the different environments presented along the work. The quadrangular cells are the most intuitive and easiest implementation. The regular triangular cells are the most basic unit geometrical patterns, since all the others can be decomposed into a series of triangles. But these triangular regular grids tend to create pair of points pattern in flat horizontal terrain surfaces. The regular hexagons have the advantage of the neighborhood similarity, what gives this cell shape an insensitivity to the point cloud and terrain direction. Finally, the irregular triangular cells take advantage of 3D triangles, especially along low slope quasi-plane surfaces where less points are needed to represent the terrain. However, the irregular process has a very high time cost when compared with the regular grids, turning this process very inefficient. The implemented algorithms to obtain the grid coordinates turns to be very fast as showed in Table 1. The Delaunay interpolation process reveals to be not suitable to the method resulting points triangulation, especially along break lines. In a future study, a hybrid process of regular and irregular cell shapes can lead to more consistent results, especially in break-lines and vegetation areas. For that, a deeper study to optimize the irregular cell implementation is essential. The method parameters adaptation values along the different iterations can also be a solution for a more efficient point discretization and a better terrain surface representation.

## REFERENCES

- Chen, C.F., Li, Y.Y., Li, W., Dai, H.L., 2013. A multiresolution hierarchical classification algorithm for filtering airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 82, pp. 1–9, doi: 10.1016/j.isprsjprs.2013.05.001.
- Chen, Z., Gao, B., Devereux, B. 2017., State-of-the-Art: DTM Generation Using Airborne LIDAR Data. *Sensors*, Vol. 17: pp. 1-24, doi: 10.3390/s17010150.
- Cheng S., Dey, T. K., Shewchuk, J., 2012. Delaunay Mesh Generation, Chapman and Hall/CRC, pp. 173-187.
- Delaunay, B., 1934. Sur la sphère vide. A la mémoire de Georges Voronoi. *Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles*. Vol. 6, pp. 793–800.
- Fellendorf, M., 2013. Digital terrain models for road design and traffic simulation. In: *Photogrammetric Week '13*, pp. 309-317. Stuttgart, Germany.
- Gézero, L., Antunes, C., 2017. An efficient method to create digital terrain models from point clouds collected by mobile lidar systems. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-1/W1, pp. 289-296. doi: 10.5194/isprs-archives-XLII-1-W1-289-2017.
- Heidrich, W., 2005. Computing the barycentric coordinates of a projected point. *Journal of Graphics Tools*, Vol. 10, pp. 9-12, doi: 10.1080/2151237X.2005.10129200
- Hu, X., Ye, L., Pang, S., Shan, J., 2015. Semi-global filtering of airborne LiDAR data for fast extraction of digital terrain models. *Remote Sensing*, Vol. 7, pp. 10996–11015, doi:10.3390/rs70810996.
- Jianhui, N., Ying, H., Zi, M., 2011. Outlier Detection of Scattered Point Cloud by Classification. *Journal of Computer-Aided Design & Computer Graphics*, Vol. 9, pp. 1526-1532.
- Kang, M., Wang M., Du Q., 2015. A method of DTM construction based on quadrangular irregular networks and related error analysis, *PLOS ONE*, Vol. 10, e0127592. doi: 10.1371/journal.pone.0127592.
- Kazhdan, M., Hoppe, H., 2013. Screened Poisson Surface Reconstruction. *ACM Transactions on Graphics*, 32(3), Article 29. pp. 1-13, doi: 10.1145/2487228.2487237.
- Mongus, D., Zalik, B., 2012. Parameter-free ground filtering of LiDAR data for automatic DTM generation. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 67, pp. 1–12, doi: 10.1016/j.isprsjprs.2011.10.002.
- Özcan, A. H. Ünsalan. C., 2017. LiDAR Data Filtering and DTM Generation Using Empirical Mode Decomposition. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 10, pp. 360-371, doi: 10.1109/JSTARS.2016.2543464.
- Pfeifer, N., Mandlbürger, G., 2008. LiDAR data filtering and DTM generation. In: Shan, J., Toth, C. K. (eds.), *Topographic Laser Scanning and Imaging: Principles and Processing*, pp. 307-334. CSC Press, Taylor and Francis Group, Boca Raton, Florida.
- Shan, Y., Zhang, X., Niu, X., Yang, G., Zhang, J. 2017. Denoising algorithm of airborne LiDAR point cloud based on 3D grid, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 10, pp. 85-92, doi: 10.14257/ijsp.2017.10.3.09.
- Tyagur, N., Hollaus, M., 2016. Digital Terrain Models from Mobile Laser Scanning Data. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLI, Part B3, pp. 387-394, Prague, Czech Republic.
- Vallet, B., Papelard, J., 2015, Road orthophoto/DTM generation from mobile laser scanning. In: *ISPRS Annals of Photogrammetry Remote Sensing and Spatial Information Sciences*, Vol. 3, Part W5, pp. 377-384, La Grande Motte, France.
- Yang B., Huang R., Dong Z., Zang Y., Li J., 2016. Two-step adaptive extraction method for ground points and breaklines from lidar point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 119: pp. 373–389, doi: 10.1016/j.isprsjprs.2016.07.002.
- Zhu, J.-F., Hu, X.-Y., Zhang, Z.-X., Xiong, X.-D., 2015 Hierarchical outlier detection for point cloud data using a density analysis method. *Acta Geodaetica et Cartographica Sinica*, Vol. 44, pp. 282-291, doi: 10.11947/j.AGCS.2015.20130423.