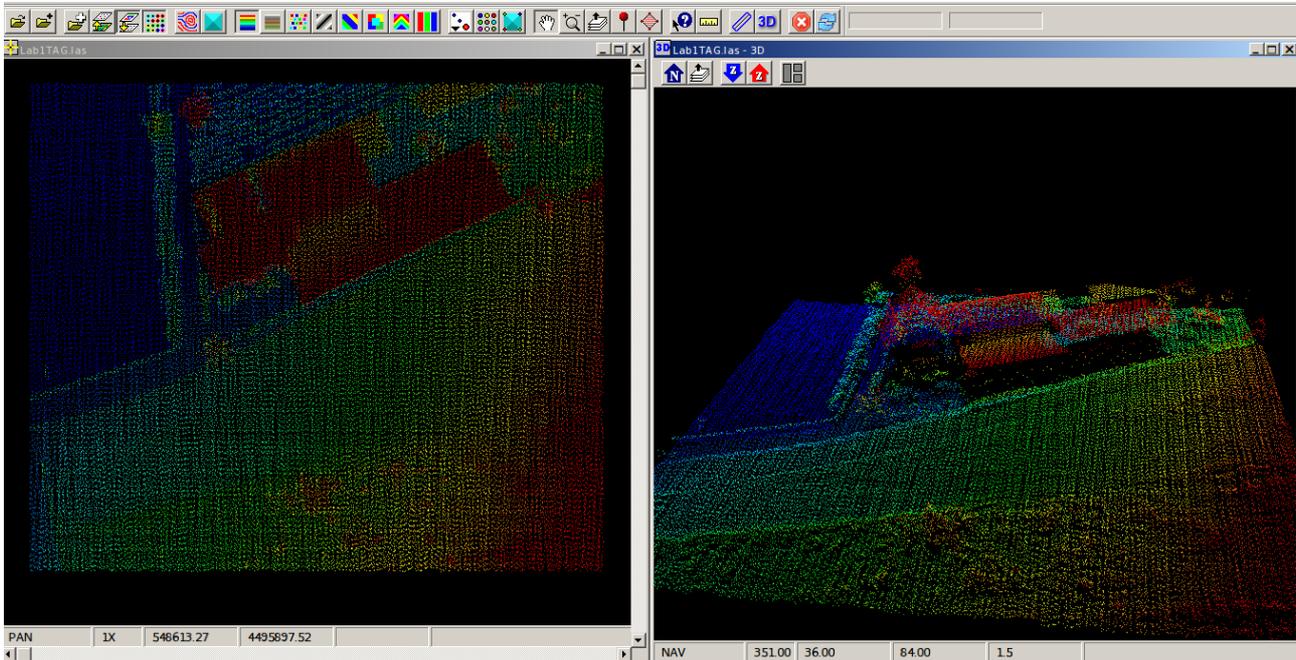


## 1. Objectivos:

Implementação em Python do filtro morfológico de (Zhang et al. , 2003) para nuvens de pontos LIDAR em formato LAS



## 2. Dados<sup>1</sup>:

Os dados para efectuar este estudo consistem em:

1. Um ficheiro LAS contendo uma nuvem de pontos LiDAR. As especificações destes ficheiros podem ser encontradas em: <http://www.asprs.org/Committee-General/LASer-LAS-File-Format-Exchange-Activities.html>

## 3. Métodos

### Introdução

A filtragem da nuvem de pontos em pontos terreno e não-terreno é a componente nuclear de qualquer software de processamento de dados lidar. Para uma dada nuvem de pontos 3D, só é possível gerar um Modelo Digital de Terreno se esta tiver sido previamente filtrada. Depois do MDT ter sido gerado, podemos então realizar a extração de entidades geográficas (por exemplo, árvores e edifícios) e extrair alguns dos seus atributos geométricos (por exemplo, a altura).

A ideia principal dos filtros morfológicos progressivos consiste em aproximar iterativamente a superfície do terreno utilizando operações morfológicas sucessivas tais como a abertura. Comparados com outros filtros, os filtros morfológicos são conceptualmente simples e facilmente implementáveis em qualquer pacote de software de processamento de imagens que contenham as operações básicas da morfologia matemática.

<sup>1</sup> O ficheiro LAS pode ser obtido em: <http://www.mat.uc.pt/~gil/downloads/Lab1TAG.las>

## Morfologia Matemática

A Morfologia Matemática surgiu na década de sessenta e tem as suas bases na teoria dos conjuntos. No processamento digital de imagens é utilizada para extrair entidades geométricas das imagens (incluindo topologia) através duma transformação feita por um elemento estruturante.

**Definição de erosão.** Segundo (Gonzalez et al. , 2003), a erosão em níveis de cinza de  $f$  por um elemento estruturante  $b$  é definida como:

$$(f \ominus b)(x, y) = \min \{f(x + x', y + y') - b(x', y') | (x', y') \in D_b\} \quad (1)$$

onde  $D_b$  é o domínio de  $b$  e  $f(x, y)$  é assumido tomar os valores  $+\infty$  fora do domínio de  $f$ . Conceptualmente podemos visualizar esta operação como uma operação de convolução onde o elemento estruturante é rodado e depois é transladado sobre a imagem: em cada nova posição os valores do elemento estruturante rodado são adicionados aos valores dos pixeis da imagem e é calculado o valor mínimo dentro desta vizinhança.

Na prática a erosão é feita habitualmente utilizando um elemento estruturante plano, no qual o valor (cota) de  $b$  para todas as coordenadas para o qual este é definido, ou seja:

$$b(x', y') = 0 \quad \forall (x', y') \in D_b$$

Neste caso a operação  $\min$  é especificada completamente pelo padrão dos 0s e 1s da matrix binária  $D_b$ , e a equação de erosão para níveis de cinzento simplifica-se para:

$$(f \ominus b)(x, y) = \min \{f(x + x', y + y') | (x', y') \in D_b\}$$

Portanto podemos dizer que a erosão em níveis de cinzento é um operador local de mínimo, onde o mínimo é calculado numa vizinhança de pixeis definida pela forma do elemento estruturante.

**Definição de dilatação.** A dilatação de  $f$  por um elemento estruturante  $b$  é definida por

$$(f \oplus b)(x, y) = \max \{f(x - x', y - y') + b(x', y') | (x', y') \in D_b\} \quad (2)$$

Tal como na erosão, a dilatação é habitualmente efectuada utilizando um elemento estruturante plano, o que simplifica muito a definição:

$$(f \oplus b)(x, y) = \max \{f(x - x', y - y') | (x', y') \in D_b\}$$

Consequentemente a dilatação em níveis de cinzento (gray-scale) é um operador de máximo local, no qual o máximo é calculado numa determinada vizinhança de pixeis definida pela forma do elemento estruturante  $D_b$ .

**Definição de abertura.** A abertura morfológica duma imagem  $f$  por um elemento estruturante é definida como a erosão de  $f$  por  $b$  seguida da dilatação da imagem resultante:

$$f \circ b = (f \ominus b) \oplus b \quad (3)$$

**Definição de fecho.** O fecho morfológico duma imagem  $f$  por  $b$  é definida como a dilatação de  $f$  por  $b$  seguida da erosão da imagem resultante

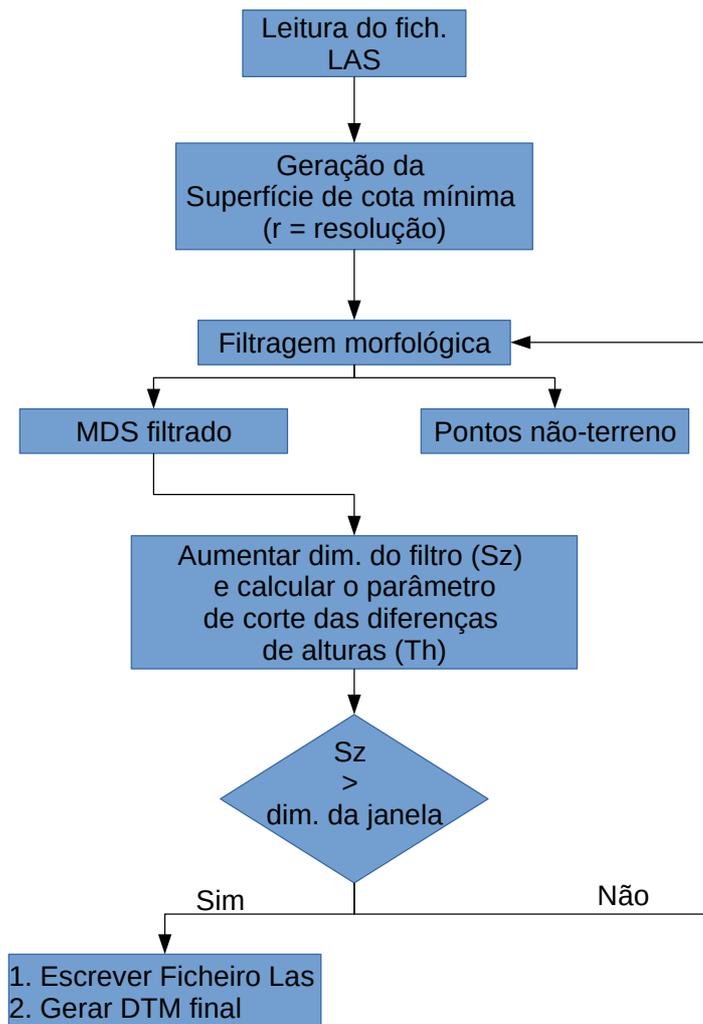
$$f \bullet b = (f \oplus b) \ominus b \quad (4)$$

Note-se, por último, que os nossos dados iniciais não são imagens mas sim nuvens de pontos 3D. Neste contexto definimos uma nuvem de pontos  $N$  como um conjunto de pontos  $p$  do espaço 3D possuindo um atributo  $a$

$$V = \{p | p(x \in \mathbb{R}^3, a \in \mathbb{R})\} \quad (5)$$

## Algoritmo de Zhang

Quando existem pulsos LidAR suficientes a atingirem o solo, a abertura morfológica executada com uma janela (ou elemento estruturante) pequena pode remover efectivamente os objectos localizados sobre a superfície do terreno natural e gerar uma superfície que aproxima o terreno natural. Quando não existem pulsos suficientes a atingirem o solo, por exemplo nas áreas cobertas por objectos com dimensões importantes, a abertura morfológica não remove estes objectos se o tamanho do elemento estruturante não contiver espacialmente estes objetos. No entanto, o problema de utilizar aberturas morfológicas com tamanhos grandes do elemento estruturante é que estes produzem superfícies com alguns artefactos nomeadamente terraços planos. Consequentemente um dos grandes desafios destes métodos consiste em manter as entidades do terreno inalteráveis quando se utilizam tamanhos grandes para o elemento estruturante.



**Figura 1: Fluxograma do filtro morfológico progressivo de Zhang**

A ideia do filtro de Zhang consiste em aumentar gradualmente a janela do elemento estruturante. Em (Zhang et al. , 2003) é mostrado o princípio de funcionamento do filtro em 1D. A implementação deste filtro em 2D pode ser feita em três fases:

- 1ª Fase: A partir da nuvem de pontos 3D  $(x, y, z)$  em formato LAS constrói-se a superfície de cota mínima selecionando a cota mínima dos pontos que estão dentro da célula. Se não houver pontos dentro da célula atribui-se a cota da célula vizinha.
- 2ª Fase: É aplicada uma abertura morfológica à superfície mínima com o tamanho mínimo do elemento estruturante. Seguidamente entra-se num processo iterativo onde os parâmetros do filtro são definidos a partir do resultado da abertura morfológica executada na iteração anterior.
- 3ª Fase: O tamanho ( $Sz$ ) do elemento estruturante da abertura morfológica é aumentado e é calculado um novo parâmetro de corte para as alturas dos pontos ( $Th$ ). Repetem-se as fases 2 e 3 até se atingir o tamanho máximo do elemento estruturante.
- 4ª Fase: A última fase consiste em gerar o DTM a partir dos pontos classificados como terreno e exportar o ficheiro LAS contendo os pontos classificados.

Por motivos de simplicidade apresentamos um possível algoritmo em 1D para um filtro morfológico progressivo

---

**Algorithm 1:** Algoritmo de Zhang

---

**Entrada:**

- Nuvem de pontos LiDAR
- Tamanho  $c$  da célula
- Parâmetro  $b$
- Declive médio do terreno  $s$
- Limite inicial de altura  $dh_0$
- Diferença máxima de altura  $dh_{\max}$

**Saída:**

- Dois conjuntos representando os pontos classificados como pertencentes ao terreno e não pertencentes ao terreno

**início**

```
 $m = \text{floor}[(\max(y) - \min(y))/c] + 1$           /* número de linhas */  
  
 $n = \text{floor}[(\max(x) - \min(x))/c] + 1$           /* número de colunas */  
  
Calcular a matriz  $A[m, n]$  a partir da nuvem de pontos LiDAR  
 $\text{flag}[m, n] \leftarrow 0$                           /* inicialização da matriz */  
  
para  $k = 1$  até  $M$  faça  $w_k = 2b^k + 1$   
 $dh_T = dh_0$   
para cada janela  $w_k$  faça  
  para  $i = 1$  até  $m$  faça  
     $P_i = A[i;]$   
     $Z \leftarrow P_i$   
     $Z_f = \text{erosão}(Z, w_k)$   
     $Z_f = \text{dilatação}(Z_f, w_k)$   
     $P_i \leftarrow Z_f$   
     $A[i;] = P_i$   
    para  $j = 1$  até  $n$  faça  
      se  $Z[j] - Z_f[j] > dh_T$  então  $\text{flag}[i, j] = w_k$   
    fim  
  fim  
  se  $dh_T > dh_{\max}$  então  
     $dh_T = dh_{\max}$   
  senão  
     $dh_T = s(w_k - w_{k-1})c + dh_0$   
  fim  
fim  
para  $i = 1$  até  $m$  faça  
  para  $j = 1$  até  $n$  faça  
    se  $B[i, j](x) > 0$  e  $B[i, j](y) > 0$  então  
      se  $\text{flag}[i, j] = 0$  então  
         $B[i, j]$  é um ponto do terreno  
      senão  
         $B[i, j]$  não é um ponto do terreno  
    fim  
  fim  
fim  
fim
```

---

## Leitura e escrita de ficheiros LAS

Para lermos e escrevermos os ficheiros contendo as nuvens de pontos em formato LAS podemos utilizar a biblioteca LIBLAS (<http://www.liblas.org/>) para uma implementação em C/C++. No caso de optarmos por uma implementação em python existem várias opções:

- A API python do LIBLAS
- A Laspy (<http://laspy.readthedocs.org/en/latest/index.html>)
- ...

No caso de pretendermos utilizar o ARCGIS existe já uma função disponível que permite ler estes ficheiros e convertê-los numa geometria do tipo multipoint numa geodatabase ou numa shapefile. Além disso podemos também utilizar a toolbox Lastools (<http://www.cs.unc.edu/~isenburg/lastools/>)

## 4. Tarefas a realizar

---

1. Implementação dum função python que classifique os pontos existentes no ficheiro LAS

## 5. Elementos a entregar

---

1. Relatório do trabalho efectuado utilizando a template do IJRS, com um máximo de 10 páginas. Utilize a seguinte estrutura para o seu relatório:
  - Introdução
  - Metodologia
  - Resultados e Discussão
  - Conclusões
2. Ficheiros python
3. Ficheiro las classificado

## Bibliografia

---

### References

Gonzalez, R. C.; Woods, R. E. & Eddins, S. L., 2003. *Digital Image Processing Using MATLAB*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Zhang, K.; Chen, S.-C.; Whitman, D.; Shyu, M.-L.; Yan, J. & Zhang, C. (2003). *A progressive morphological filter for removing nonground measurements from airborne LIDAR data*, *Geoscience and Remote Sensing*, IEEE Transactions on 41 : 872-882.

### Anexo: API

---

Os APIs que irá necessitar para a realização deste trabalho são:

- python-liblas ou funções python de leitura e escrita de ficheiros LAS
- python-mahotas para implementação das operações morfológicas de dilatação e erosão
- scipy e numpy para implementação das funções de interpolação definidas sobre matrizes multidimensionais e para a implementação das funções de processamento de imagens

Além disso para visualizar os ficheiros LAS pode utilizar o Frugoviewer disponível em

<http://www.fugroviewer.com/>