Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Vieira Cortes

Type Reference

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Informática, do Departamento de Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Informática.

Advisor: Prof. Roberto Ierusalimschy

Rio de Janeiro
March 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Felipe Vieira Cortes

Type Reference

Thesis presented to the Programa de Pós–graduação em Informática da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Informática. Approved by the Examination Committee:

Prof. Roberto Ierusalimschy
Advisor
Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo
PUC-Rio

Prof. Waldemar Celes Filho
PUC-Rio

Rio de Janeiro, March 8th, 2018

Felipe Vieira Cortes

Graduated in computer science by the Harvard University.

Bibliographic data

Cortes,Felipe V.

Type Reference / Felipe Vieira Cortes; advisor: Roberto Ierusalimschy. – 2018.

23 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2018.

Inclui bibliografia

1. Informática – Teses. 2. Procesamento Geométrico. 3. Remoção de ruído de malha. 4. Vizinhança adaptativa. I. Ierusalimschy, Roberto. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

To my parents, for their support
and encouragement.

Acknowledgments

To my adviser Professor Marcelo Gattass for the stimulus and partnership to carry out this work.

To CNPq and PUC-Rio, for the aids granted, without which this work does not could have been accomplished.

For students contemplated with any CAPES scholarship, whose defense occurred as of 04 September 2018 leave the following passage: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Cortes,Felipe V.; Ierusalimschy, Roberto (Advisor). **Type Reference**. Rio de Janeiro, 2018. 23p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The acquisition of triangular meshes typically introduces undesired noise...

Keywords

Geometry Processing; Mesh Denoising; Adaptive Patches.

Resumo

Cortes,Felipe V.; Ierusalimschy, Roberto. **Referência de tipos**. Rio de Janeiro, 2018. 23p. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A aquisição de malhas triangulares normalmente introduz ruídos indesejados...

Palavras-chave

Procesamento Geométrico; Remoção de ruído de malha; Vizinhança adaptativa.

Table of contents

List of figures

List of tables

List of algorithms

List of codes

List of Abreviations

ADI – Análise Digital de Imagens

BIF – *Banded Iron Formation*

My beautifull epigraph

Wassily Kandinsky, *Regards sur le passé.*

1
Introduction

Nowadays 3D surface models are used in several fields and industries such as medicine, engineering, entertainment, geo-exploration, architecture, cultural heritage and so on. These models can be acquired from a variety of sources like 3D scanning, 3D imaging, multi-view stereo reconstruction, CAD modeling, etc. The data generated by these techniques should be processed to be available for production or any task where it can be used (visualization, simulation, animation, interaction, etc.). This processing step is called digital geometry processing which is a field of computer science that uses mathematical models and algorithms (BOTSCH et al., 2010). Figure 1.1 shows some examples of noisy meshes.

This document is structured as follows. In Chapter 2 we present some previous work relevant to our problem. In Chapter 3 we explain our proposal. In Chapter 4 we show our results. Finally, in Chapter 5 we present our conclusion and future work.

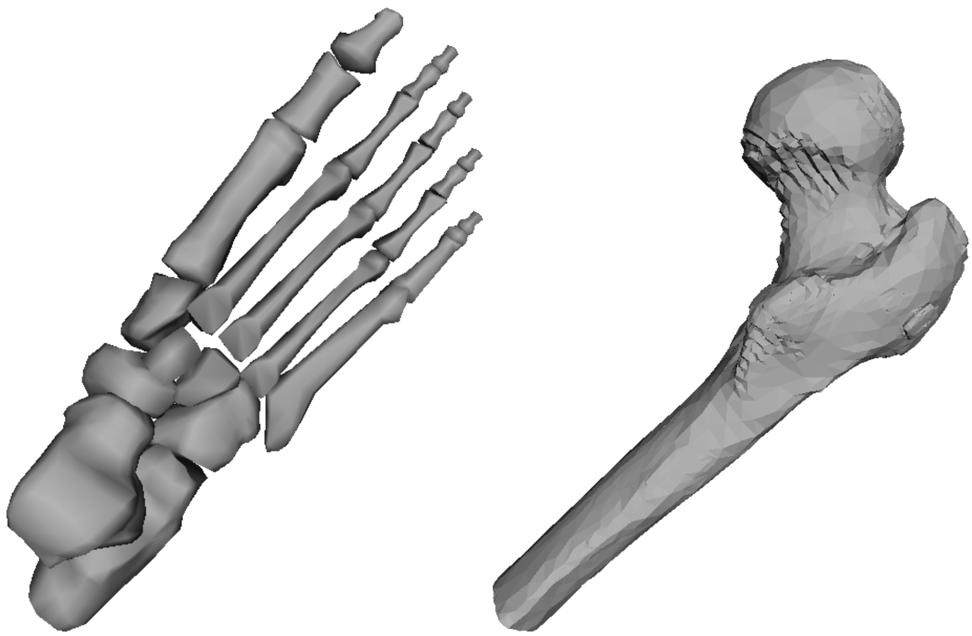
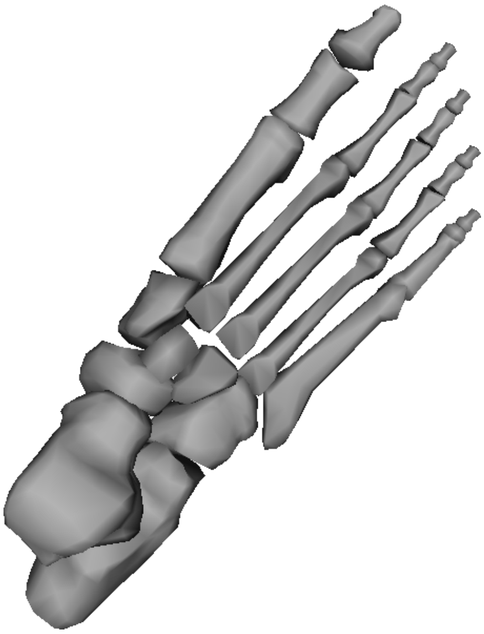


Figure 1.1: Meshes generated from medical data. Data obtained from the AIM@SHAPE Shape Repository (AIM@SHAPE. . . ,)

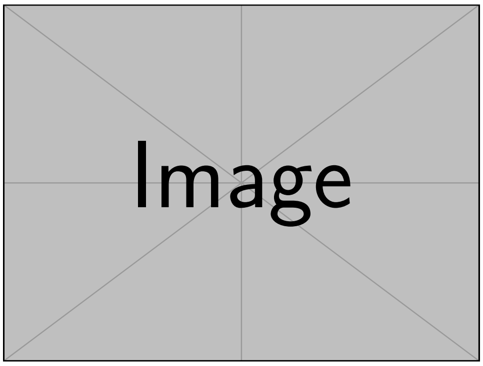
2
Previous Work

Early smoothing methods tried to minimize... In the figure 2.2d we see...



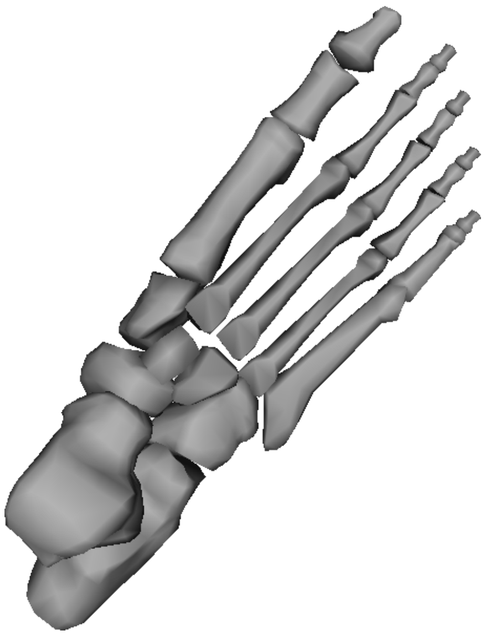
(a) Bamboo-pile Vertically Inserted Position

(b) Bamboo-pile Normal Inserted Position

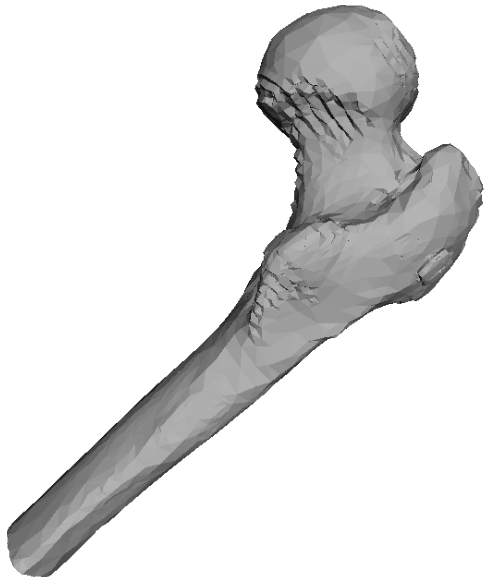


(c) bamboo-pile Inserted 45° angle

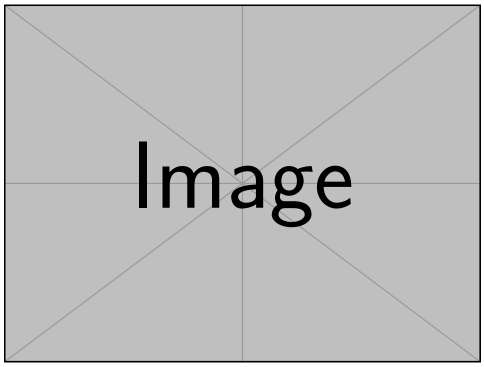
Figure 2.1: A set of three subfigures: (a) describes the first subfigure; (b) describes the second subfigure; (c) describes the third subfigure.



(a) Bamboo-pile Vertically Inserted Position

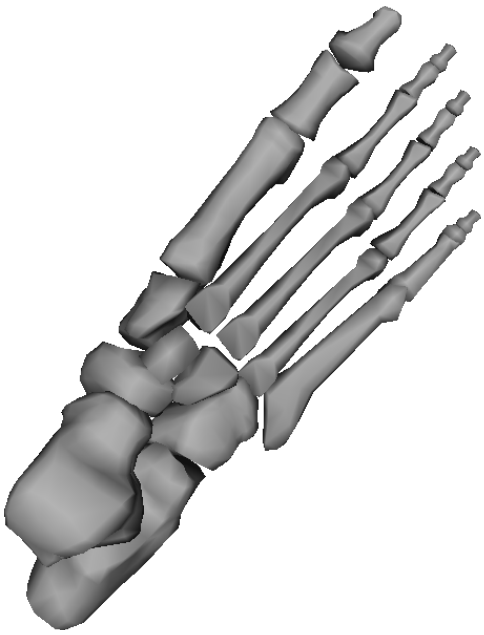


(b) Bamboo-pile Normal Inserted Position

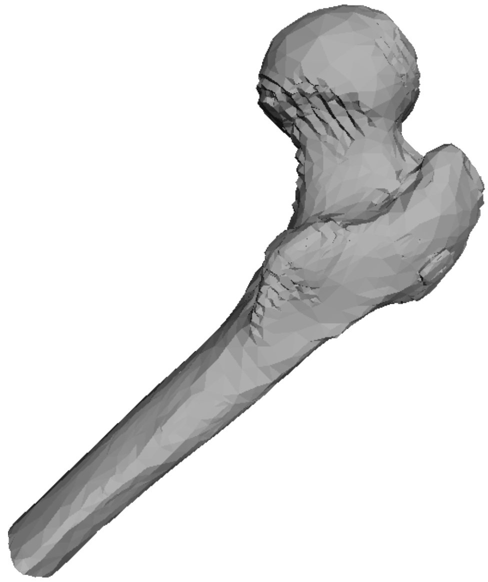


(c) bamboo-pile Inserted 45° angle

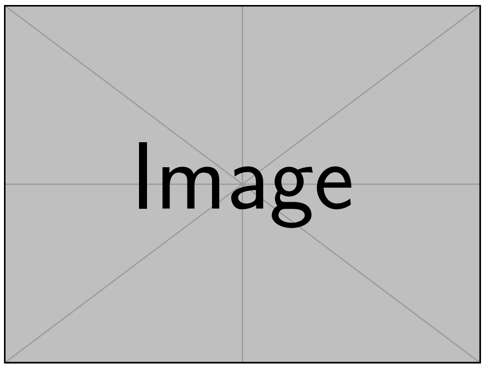
Figure 2.2: A set of six subfigures in two pages.



(d) Bamboo-pile Vertically Inserted Position



(e) Bamboo-pile Normal Inserted Position



(f) bamboo-pile Inserted 45° angle

Figure 2.2: A set of six subfigures in two pages.(Continuation)

3
Proposal

Equation example 1:

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \int_{x_i \in X} ||x' - x_i|| u_i da \\ s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0, \tag{3-1}$$

Equation exmaple 2:

$$\min_{\mathbf{u}} \alpha \mathbf{u}^T \mathbf{A}^T \mathbf{Q} \mathbf{A} \mathbf{u} + \beta \mathbf{d}^T a' \mathbf{A} \mathbf{u} + \gamma \mathbf{u}^T \mathbf{G}^T \mathbf{G} \mathbf{u} + \delta \mathbf{f}^T a' \mathbf{A} \mathbf{u} \\ s.t. \quad \mathbf{0} \leq \mathbf{u} \leq \mathbf{1} \wedge \mathbf{a}^T \mathbf{u} = a_0. \tag{3-2}$$

Equation example 3:

$$\mathbf{G} = (g_{ij}) = \begin{cases} \sum_{f_k \in N_f(f_i)} l_{ik} & i = j \\ -l_{ij} & e_{ij} \in E \\ 0 & \text{otherwise} \end{cases} \tag{3-3}$$

Code 1: Mean Filter

```
# ---------------------------------------------------------------------#
# Create filter function
# l is the width of window
# ---------------------------------------------------------------------#
meanfilter <- function( l, imagem ) {
  if( l%%2 == 0 )
    print("Please, type an odd number!")
  imagem.result <- imagem
  lp1d2 <- (l-1)/2
  L <- dim(imagem)[1]
  C <- dim(imagem)[2]
  for( j in as.integer(lp1d2+1) : as.integer(C-lp1d2)) {
    for( i in as.integer(lp1d2+1) : as.integer(L-lp1d2)) {
      imagem.result[i,j] <- mean(imagem[as.integer(i-lp1d2):as.integer(i+lp1d2), as.integer(j-lp1d2):as.integer(j+lp1d2)])
    }
  }
```

```
  print("Image filtred with success!")
  return(imagem.result)
}
# --------------------------------------------------------------------#
# End of Script.
# --------------------------------------------------------------------#
```

Algorithm 1: Escolha das amostras inicias

Input: Malha e quantidade de pontos a ser amostrado

Output: Pontos amostrados na malha

Crie um vetor de números randômicos entre $[0, 1]$ *com a quantidade de pontos a ser amostrada e ordene-o*

Calcule a área total dos triângulos da malha

for $i = 0$ **to** *numeroDePontos* **do**

  Navegue entre as faces acumulando a sua $\frac{area}{areaTotal}$ *até achar a face com valor acumulado* $\geqslant$ *numerosRandomicos[i]*

  Pegue um ponto randômico dentro da face utilizando o método de Turk e adicione no vetor do resultado

4 Results

Table example. Table 4.1 shows results.

Table 4.1: Results for devil mesh

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
(FLEISHMAN; DRORI; COHEN-OR, 2003)	0.061277	0.110973	0.236219	19.697900	0.055170	0.047678	0.090284	0.051443	0.045645
(JONES; DURAND; DESBRUN, 2003)	0.001293	0.002800	0.002289	21.237300	0.021589	0.013026	0.087991	0.000364	0.002621
(SUN et al., 2007)	0.001439	0.002880	0.003540	14.043200	0.012654	0.008911	0.055849	0.007806	0.000582
(ZHENG et al., 2011)	0.000713	0.001537	0.001824	12.171400	0.009654	0.005781	0.054567	0.005617	0.000425
(HE; SCHAEFER, 2013)	0.002531	0.004560	0.007108	13.830100	0.017459	0.010314	0.114528	0.001686	0.001786
(ZHANG et al., 2015)	0.001623	0.003079	0.005048	10.454200	0.015233	0.008054	0.094668	0.002629	0.001326
(YADAV; REITEBUCH; POLTHIER, 2016)	0.000737	0.001548	0.001493	16.880800	0.014129	0.006974	0.079952	0.000209	0.002375
Ours	0.000987	0.001902	0.002686	11.574200	0.010632	0.006796	0.075106	0.003970	0.000722

4.1 Comparison

5
Conclusion and future work

We proposed an algorithm for triangular mesh denoising with detail preservation...

Code 2: Mean Filter

```
# --------------------------------------------------------------------#
# Create filter function
# l is the width of window
# --------------------------------------------------------------------#
meanfilter <- function( l, imagem ) {
  if( l%%2 == 0 )
    print("Please, type an odd number!")
  imagem.result <- imagem
  lp1d2 <- (l-1)/2
  L <- dim(imagem)[1]
  C <- dim(imagem)[2]
  for( j in as.integer(lp1d2+1) : as.integer(C-lp1d2)) {
    for( i in as.integer(lp1d2+1) : as.integer(L-lp1d2)) {
      imagem.result[i,j] <- mean(imagem[as.integer(i-lp1d2):as.integer(i+lp1d2), as.integer(j-lp1d2):as.integer(j+lp1d2)])
    }
  }
  print("Image filtred with success!")
  return(imagem.result)
}
# --------------------------------------------------------------------#
# End of Script.
# --------------------------------------------------------------------#
```

6 Bibliography

AIM@SHAPE Shape Repository. <http://visionair.ge.imati.cnr.it>. Accessed: 2017-05-01. Cited 2 times in pages 9 and 15.

BOTSCH, M. et al. **Polygon Mesh Processing**. Taylor & Francis, 2010. (Ak Peters Series). ISBN 9781568814261. Disponível em: <https://books.google.com.br/books?id=8zX-2VRqBAkC>. Cited in page 15.

FLEISHMAN, S.; DRORI, I.; COHEN-OR, D. Bilateral mesh denoising. In: ACM. **ACM transactions on graphics (TOG)**. [S.l.], 2003. v. 22, n. 3, p. 950–953. Cited in page 21.

HE, L.; SCHAEFER, S. Mesh denoising via l 0 minimization. **ACM Transactions on Graphics (TOG)**, ACM, v. 32, n. 4, p. 64, 2013. Cited in page 21.

JONES, T. R.; DURAND, F.; DESBRUN, M. Non-iterative, feature-preserving mesh smoothing. In: ACM. **ACM Transactions on Graphics (TOG)**. [S.l.], 2003. v. 22, n. 3, p. 943–949. Cited in page 21.

SUN, X. et al. Fast and effective feature-preserving mesh denoising. **IEEE transactions on visualization and computer graphics**, IEEE, v. 13, n. 5, p. 925–938, 2007. Cited in page 21.

YADAV, S.; REITEBUCH, U.; POLTHIER, K. Mesh denoising based on normal voting tensor and binary optimization. **arXiv preprint arXiv:1607.07427**, 2016. Cited in page 21.

ZHANG, W. et al. Guided mesh normal filtering. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. [S.l.], 2015. v. 34, n. 7, p. 23–34. Cited in page 21.

ZHENG, Y. et al. Bilateral normal filtering for mesh denoising. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 17, n. 10, p. 1521–1530, 2011. Cited in page 21.