

Pontifícia Universidade Católica
do Rio de Janeiro



Dennis Ritchie

Modelo de tese e dissertação PUC-Rio

Tese de Doutorado

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática, do Departamento de Informática da PUC-Rio .

Orientador: Prof. Marcelo Gattass

Rio de Janeiro
Março de 2018

Pontifícia Universidade Católica
do Rio de Janeiro



Dennis Ritchie

Modelo de tese e dissertação PUC-Rio

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática da PUC-Rio . Aprovada pela Comissão Examinadora abaixo:

Prof. Marcelo Gattass

Orientador

Departamento de Informática – PUC-Rio

Prof. Alberto Barbosa Raposo

PUC-Rio

Prof. Waldemar Celes Filho

PUC-Rio

Rio de Janeiro, 8 de Março de 2018

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Dennis Ritchie

Graduado em ciência da computação pela Universidade de Harvard.

Ficha Catalográfica

Ritchie, Dennis

Modelo de tese e dissertação PUC-Rio / Dennis Ritchie; orientador: Marcelo Gattass. – 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. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Aos meus pais, irmãos e família
pelo apoio e encorajamento.

Agradecimentos

Ao meu orientador Professor Marcelo Gattass pelo estímulo e parceria para a realização deste trabalho.

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus amigos Júlio, Sara, Carmen e Antônio Miguel por todo apoio, paciência e compreensão.

Aos meus pais, pela educação, atenção e carinho de todas as horas.

À minha professora Maria da Silva, pelas importantes contribuições e palavras de apoio.

Aos meus colegas da PUC-Rio.

Aos professores que participaram da Comissão examinadora.

A todos os professores e funcionários do Departamento pelos ensinamentos e pela ajuda.

A todos os amigos e familiares que de uma forma ou de outra me estimularam ou me ajudaram.

Para alunos contemplados com qualquer tipo de Bolsa da CAPES, cuja defesa ocorreu a partir de 04/09/2018 deixar o seguinte trecho: O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

Ritchie, Dennis; Gattass, Marcelo. **Modelo de tese e dissertação PUC-Rio**. 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.

Abstract

Ritchie, Dennis; Gattass, Marcelo (Advisor). **Thesis and dissertation template PUC-Rio**. 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.

Sumário

1	Introdução	15
2	Trabalhos relacionados	16
3	Método proposto	19
4	Resultados	21
4.1	Resultados comparativos	21
5	Conclusão e trabalhos futuros	22
6	Referências bibliográficas	23

Lista de figuras

Figura 1.1	Meshes generated from medical data. Data obtained from the AIM@SHAPE Shape Repository (AIM@SHAPE. . . ,)	15
Figura 2.1	A set of three subfigures: (a) describes the first subfigure; (b) describes the second subfigure; (c) describes the third subfigure.	16
	(a) Bamboo-pile Vertically Inserted Position	16
	(b) Bamboo-pile Normal Inserted Position	16
	(c) bamboo-pile Inserted 45° angle	16
Figura 2.2	A set of six subfigures in two pages.	17
	(a) Bamboo-pile Vertically Inserted Position	17
	(b) Bamboo-pile Normal Inserted Position	17
	(c) bamboo-pile Inserted 45° angle	17
	(d) Bamboo-pile Vertically Inserted Position	18
	(e) Bamboo-pile Normal Inserted Position	18
	(f) bamboo-pile Inserted 45° angle	18

Lista de tabelas

Tabela 4.1 Results for devil mesh

21

Lista de algoritmos

Algoritmo 1 Escolha das amostras iniciais

20

Lista de Códigos

Código 1	Mean Filter	19
Código 2	Mean Filter	22

Lista de Abreviaturas

ADI – Análise Digital de Imagens

BIF – *Banded Iron Formation*

My beautiful epigraph

Wassily Kandinsky, *Regards sur le passé.*

1

Introdução

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.

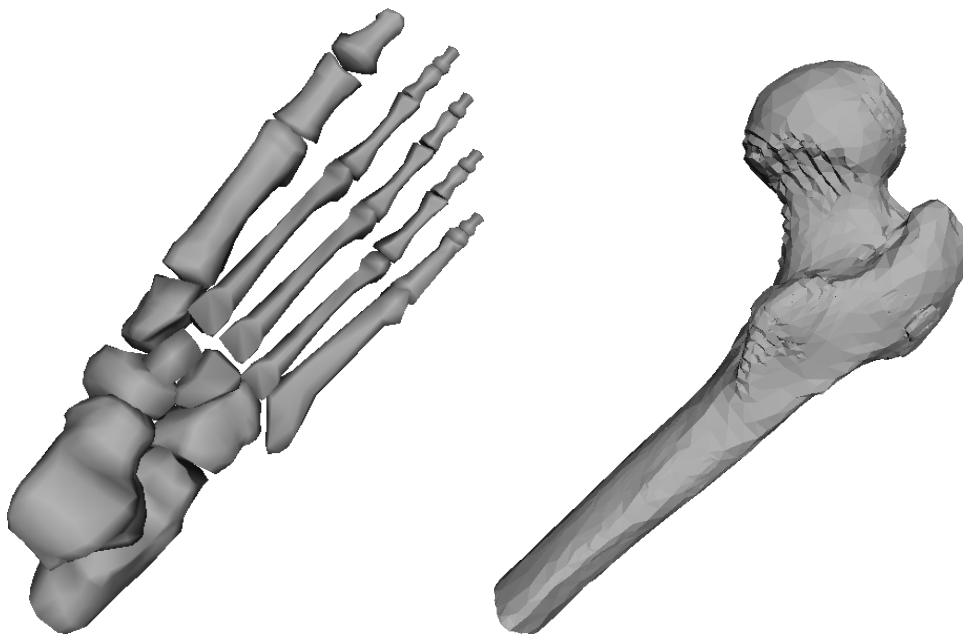
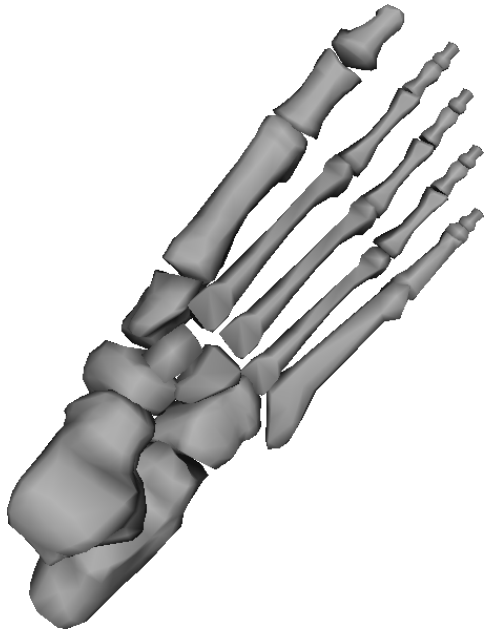


Figura 1.1: Meshes generated from medical data. Data obtained from the AIM@SHAPE Shape Repository (AIM@SHAPE. . . ,)

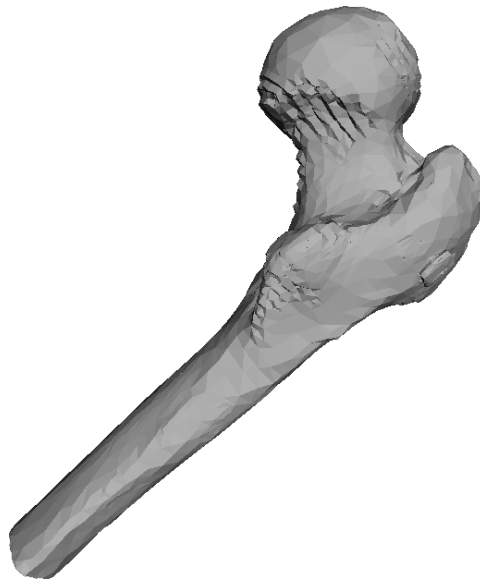
2

Trabalhos relacionados

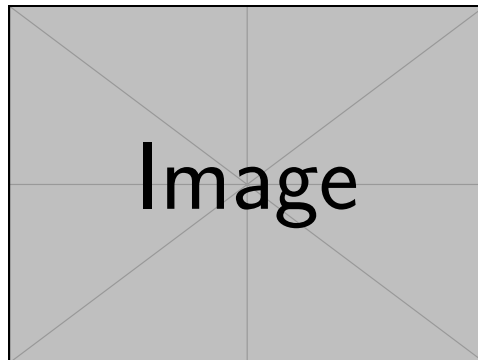
Early smoothing methods tried to minimize... In the figure 2.2d we see...



(a) Bamboo-pile Vertically Inserted Position

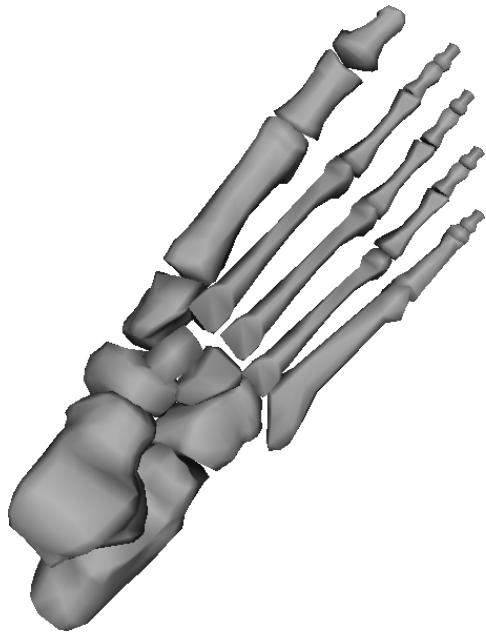


(b) Bamboo-pile Normal Inserted Position

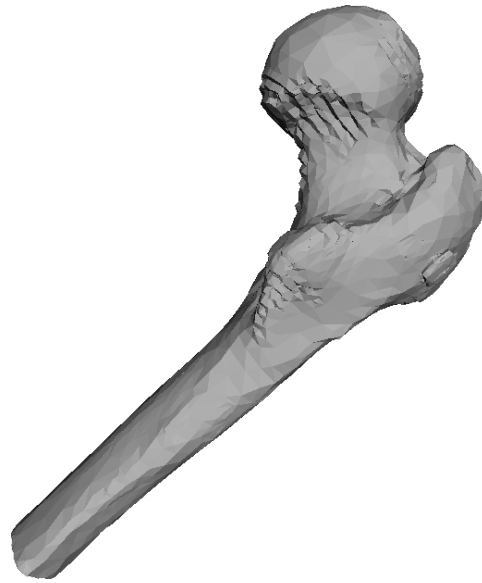


(c) bamboo-pile Inserted 45° angle

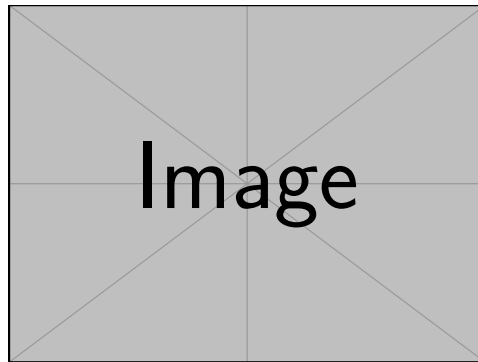
Figura 2.1: A set of three subfigures: (a) describes the first subfigure; (b) describes the second subfigure; (c) describes the third subfigure.



(a) Bamboo-pile Vertically Inserted Position

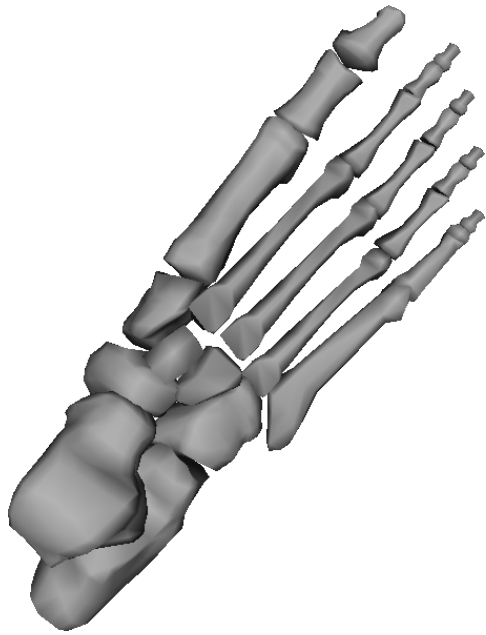


(b) Bamboo-pile Normal Inserted Position

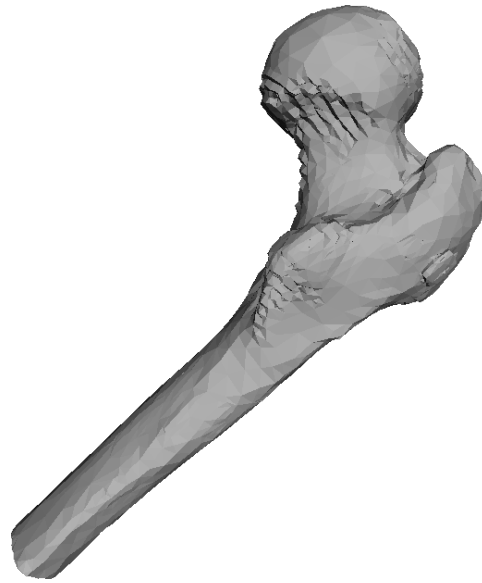


(c) bamboo-pile Inserted 45° angle

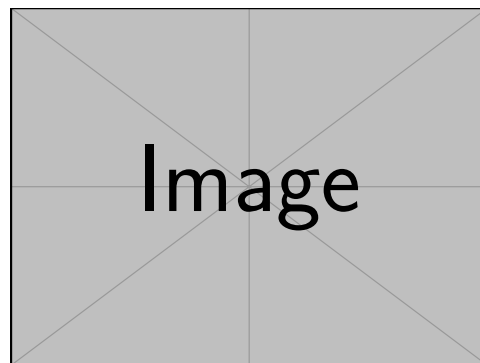
Figura 2.2: A set of six subfigures in two pages.



(d) Bamboo-pile Vertically Inserted Position



(e) Bamboo-pile Normal Inserted Position



(f) bamboo-pile Inserted 45° angle

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

3 Método proposto

Equation example 1:

$$\begin{aligned} \min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} \|x' - x_i\| u_i da \\ \text{s.t. } u \in [0, 1] \wedge \int_{x_i \in X} u da = a_0, \end{aligned} \quad (3-1)$$

Equation exmample 2:

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

Equation example 3:

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

Código 1: Mean Filter

```

1 # -----#
2 # Create filter function
3 # l is the width of window
4 # -----#
5 meanfilter <- function( l, imagem ) {
6   if( l%%2 == 0 )
7     print("Please, type an odd number!")
8   imagem.result <- imagem
9   lp1d2 <- (l-1)/2
10  L <- dim(imagem)[1]
11  C <- dim(imagem)[2]
12  for( j in as.integer(lp1d2+1) : as.integer(C-lp1d2)) {
13    for( i in as.integer(lp1d2+1) : as.integer(L-lp1d2)) {
14      imagem.result[i,j] <- mean(imagem[as.integer(i-lp1d2):as.
15                                integer(i+lp1d2), as.integer(j-lp1d2):as.integer(j+lp1d2)
16                                ])
15    }
16  }

```

```

17  print("Image filtered with success!")
18  return(imagem.result)
19  }
20  #
-----#
21  # End of Script.
22  #
-----#

```

Algoritmo 1: Escolha das amostras iniciais

Entrada: Malha e quantidade de pontos a ser amostrado

Saída: Pontos amostrados na malha

- 1 Crie um vetor de números randômicos entre $[0, 1]$ com a quantidade de pontos a ser amostrada e ordene-o
 - 2 Calcule a área total dos triângulos da malha
 - 3 **para** $i = 0$ **até** numeroDePontos **faça**
 - 4 Navegue entre as faces acumulando a sua $\frac{\text{area}}{\text{areaTotal}}$ até achar a face com valor acumulado $\geq \text{numerosRandomicos}[i]$
 - 5 Pegue um ponto randômico dentro da face utilizando o método de Turk e adicione no vetor do resultado
-

4

Resultados

Table example. Table 4.1 shows results.

Tabela 4.1: Results for devil mesh

	Mean Vertex Dis- tance	L2 Ver- tex Ba- sed	Mean Quadric	MSAE	L2 Nor- mal Ba- sed	Tangential	Mean Discrete Curva- ture	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; DU- RAND; DES- BRUN, 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; SCHA- EFER, 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; REITE- BUCH; 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

Resultados comparativos

5 Conclusão e trabalhos futuros

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

Código 2: Mean Filter

```
1 #
   -----#

2 # Create filter function
3 # l is the width of window
4 #
   -----#

5 meanfilter <- function( l, imagem ) {
6   if( l%%2 == 0 )
7     print("Please, type an odd number!")
8   imagem.result <- imagem
9   lp1d2 <- (l-1)/2
10  L <- dim(imagem)[1]
11  C <- dim(imagem)[2]
12  for( j in as.integer(lp1d2+1) : as.integer(C-lp1d2)) {
13    for( i in as.integer(lp1d2+1) : as.integer(L-lp1d2)) {
14      imagem.result[i,j] <- mean(imagem[as.integer(i-lp1d2):as.
15        integer(i+lp1d2), as.integer(j-lp1d2):as.integer(j+lp1d2)
16      ])
17    }
18  }
19 }
20 #
   -----#

21 # End of Script.
22 #
   -----#
```

6

Referências bibliográficas

AIM@SHAPE Shape Repository. <<http://visionair.ge.imati.cnr.it>>. Accessed: 2017-05-01. Citado 2 vezes nas páginas 9 e 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>>. Citado na página 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. Citado na página 21.

HE, L.; SCHAEFER, S. Mesh denoising via l_0 minimization. **ACM Transactions on Graphics (TOG)**, ACM, v. 32, n. 4, p. 64, 2013. Citado na página 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. Citado na página 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. Citado na página 21.

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