

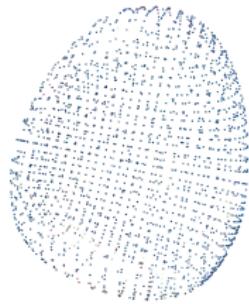
# Trabalho de conclusão de curso: Compressão do sinal de cor de uma nuvem de pontos com codificador aritmético

Miguel de Carvalho Pachá

Departamento de Engenharia Elétrica, Universidade de Brasília

27 de novembro de 2019

# Introdução



# Introdução



# Sumário

- 1 Motivação
- 2 Teoria
- 3 Metodologia
- 4 Resultados
- 5 Conclusão

# Table of Contents

1 Motivação

2 Teoria

3 Metodologia

4 Resultados

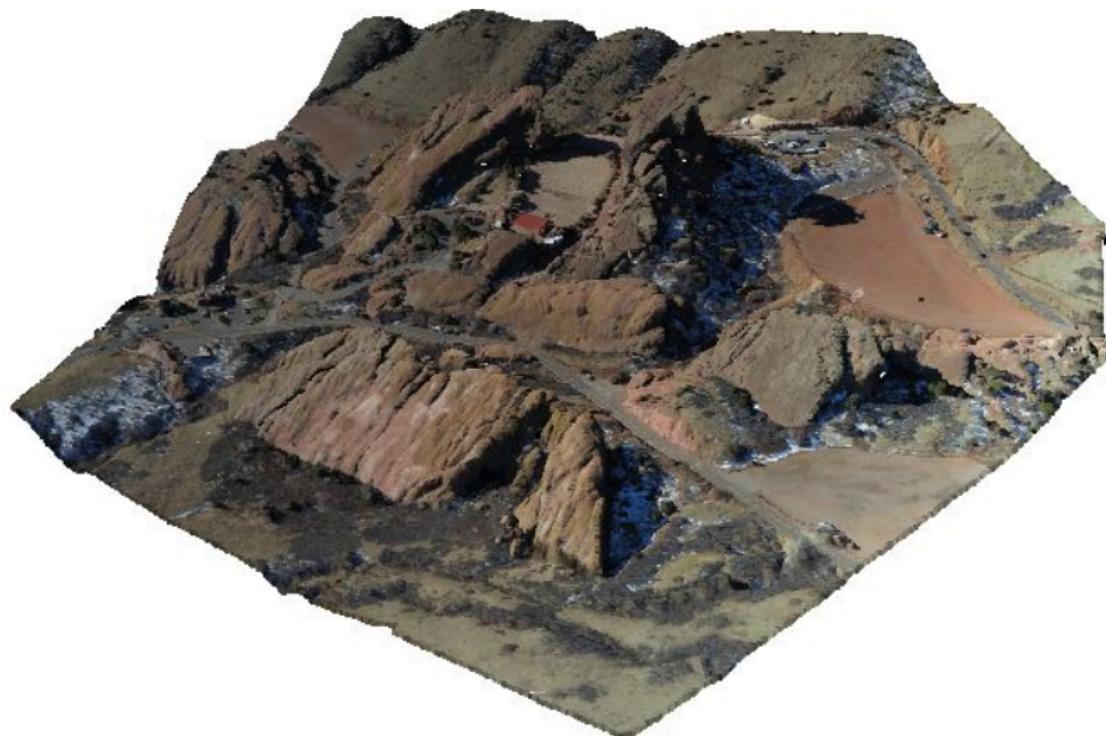
5 Conclusão

# Motivação: Aplicações



Realidade virtual

# Motivação: Aplicações



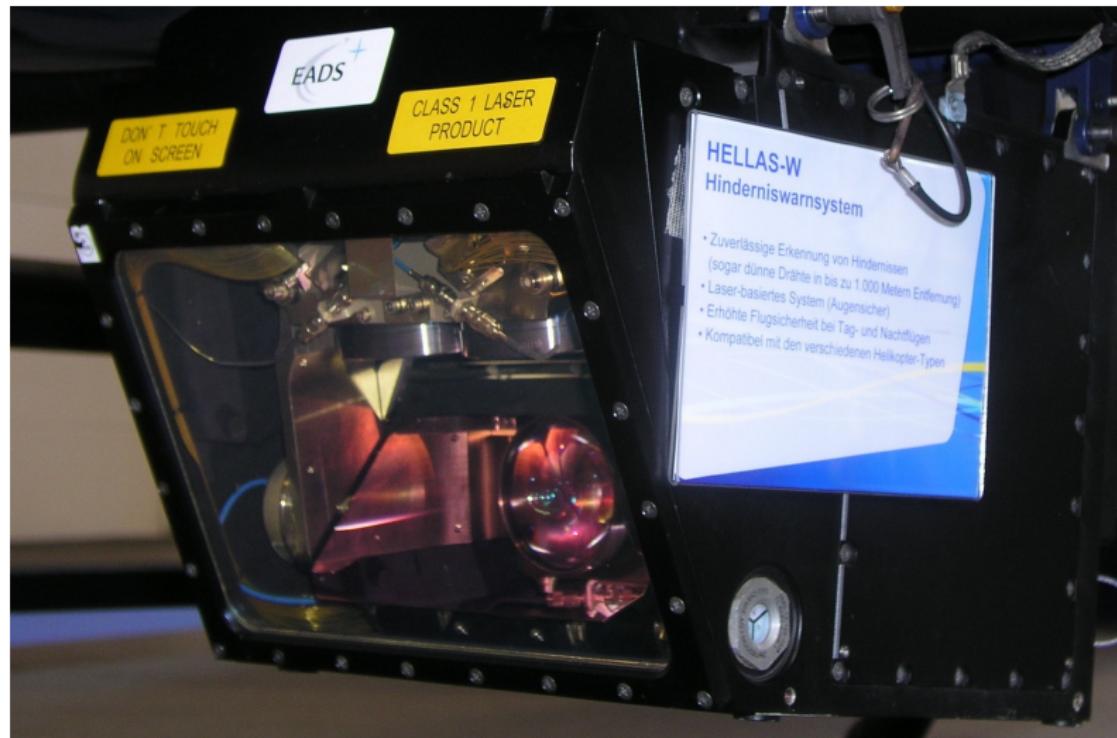
Sistemas de referenciamento geográfico

# Motivação: Aplicações



Carros autônomos

# Obtenção de imagens



Um LiDAR embarcado

# Obtenção de imagens



Câmera RGBD

# Table of Contents

1 Motivação

2 Teoria

3 Metodologia

4 Resultados

5 Conclusão

## Definição

Uma nuvem de pontos é um conjunto de duplas  $(X_i, A_i)$ , com

- $X_i = (x_i, y_i, z_i)$  (vetor de posição)
- $A_i = (A_{1,i}, A_{2,i}, \dots, A_{D,i})$  (vetor de atributos)

## Definição

Uma nuvem de pontos é um conjunto de duplas  $(X_i, A_i)$ , com

- $X_i = (x_i, y_i, z_i)$  (vetor de posição)
- $A_i = (A_{1,i}, A_{2,i}, \dots, A_{D,i})$  (vetor de atributos)
- *Voxel*: Ponto com coordenadas quantizadas

## Exemplos de atributo

- LiDAR: refletância.
- Câmeras RGBD: sinal de cor

Medindo a informação de uma nuvem de pontos

Taxa: bits gastos para cada *voxel* ocupado (bpov).

Medindo a informação de uma nuvem de pontos

Taxa: bits gastos para cada *voxel* ocupado (bpov).

## Exemplos

- Coordenadas com 9 bits  $\times$  3 dimensões: taxa da geometria = 27 bpov

Medindo a informação de uma nuvem de pontos

Taxa: bits gastos para cada *voxel* ocupado (bpov).

## Exemplos

- Coordenadas com 9 bits  $\times$  3 dimensões: taxa da geometria = 27 bpov
- Cor de 8 bits  $\times$  3 canais: taxa do sinal de cor = 24 bpov

# Table of Contents

1 Motivação

2 Teoria

3 Metodologia

4 Resultados

5 Conclusão

# Metodologia

- Divisão em filamentos
- Codificação dos filamentos

# Metodologia

- Divisão em filamentos
  - ▶ Corte em camadas
  - ▶ Geração de um caminho
  - ▶ Corte em filamentos: limiar de distância de corte
- Codificação dos filamentos

# Metodologia

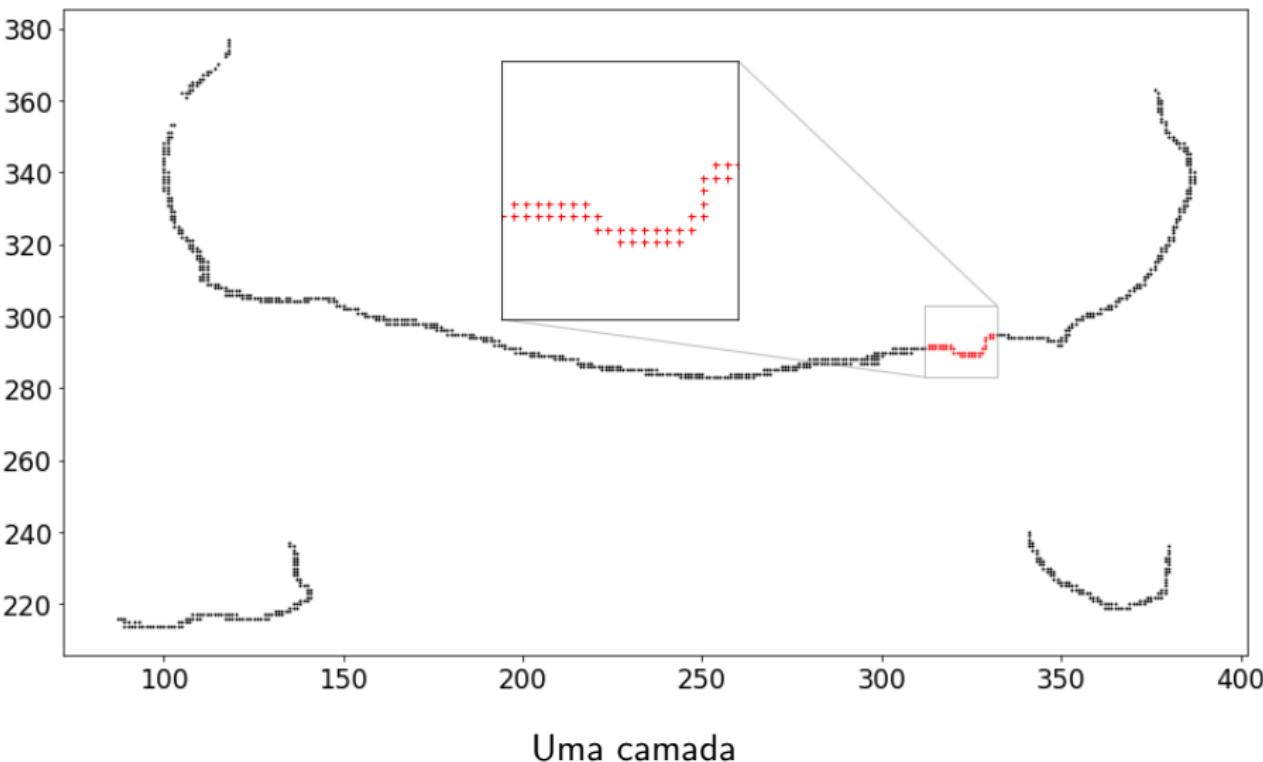
- Divisão em filamentos
  - ▶ Corte em camadas
  - ▶ Geração de um caminho
  - ▶ Corte em filamentos: limiar de distância de corte
- Codificação dos filamentos
  - ▶ Escolha do codec
  - ▶ Codificação diferencial
  - ▶ Codificação aritmética

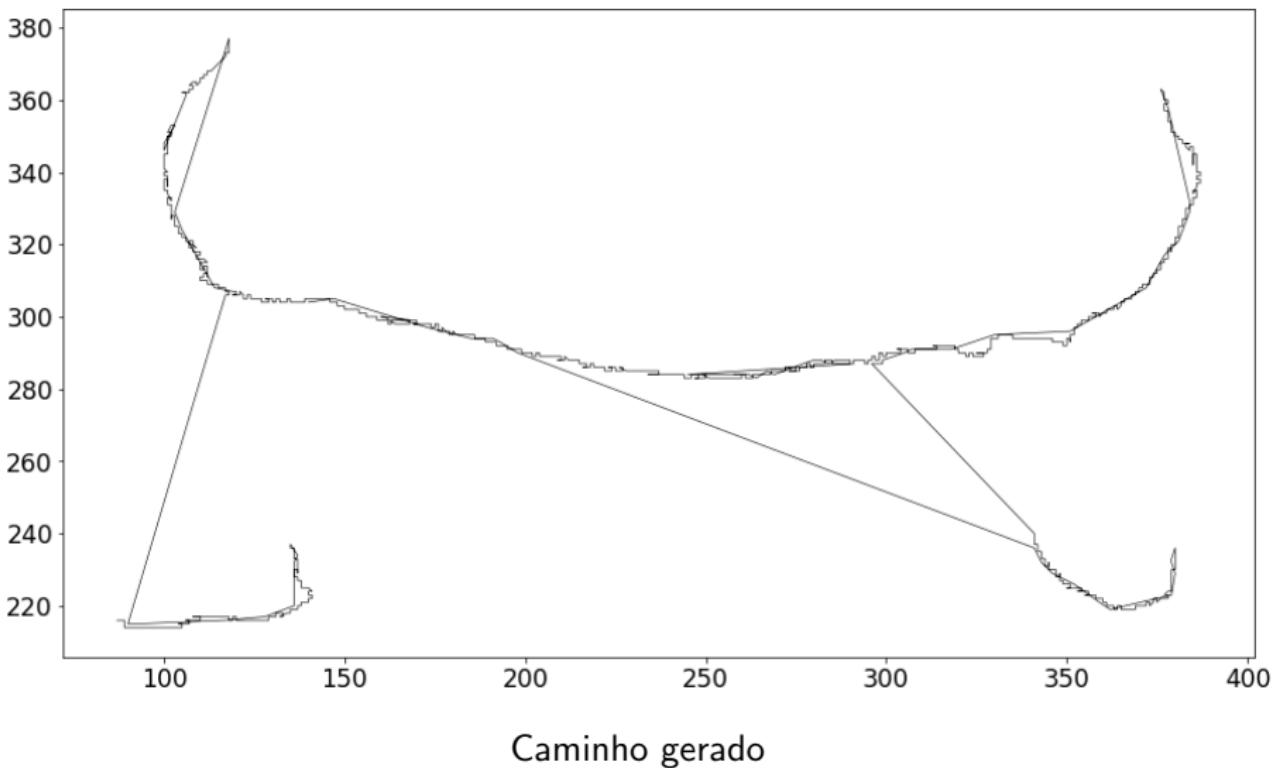
# Metodologia

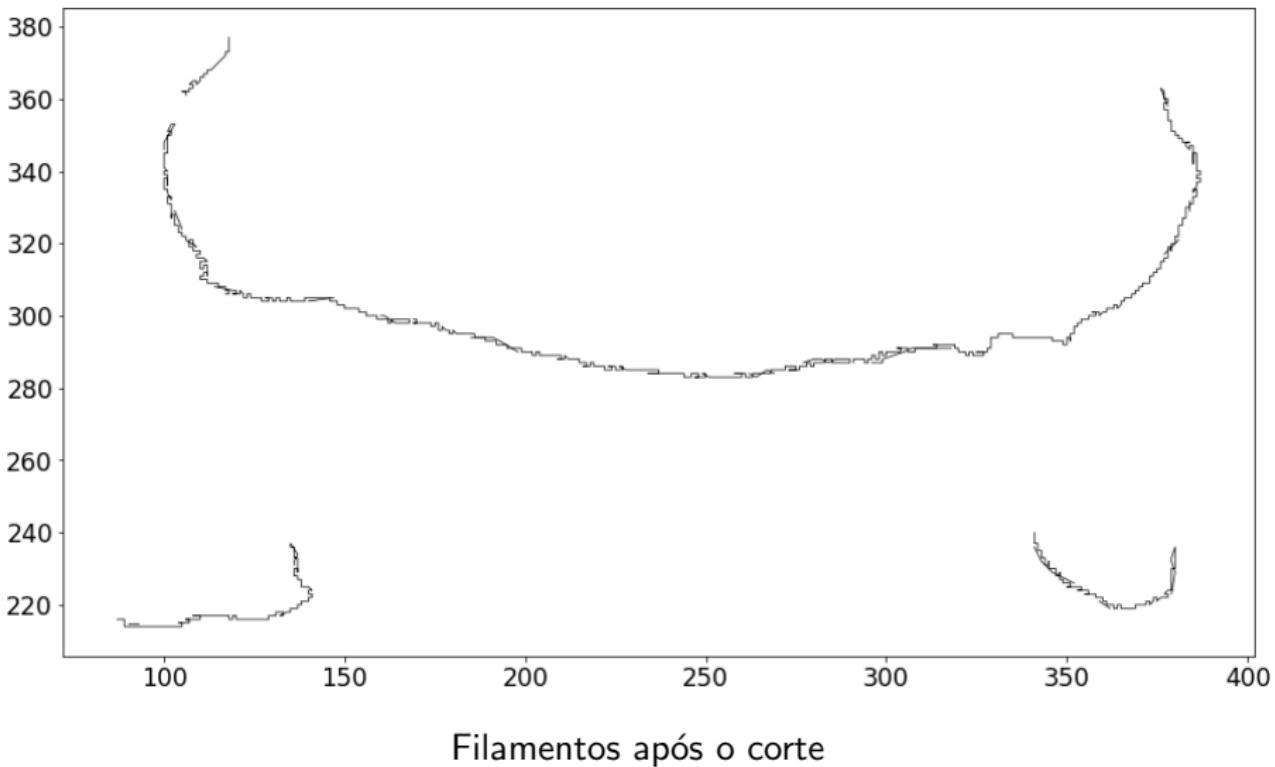
- Divisão em filamentos
  - ▶ Corte em camadas
  - ▶ Geração de um caminho
  - ▶ Corte em filamentos: limiar de distância de corte
- Codificação dos filamentos
  - ▶ Escolha do codec
  - ▶ Codificação diferencial
  - ▶ Codificação aritmética



Point cloud para o exemplo



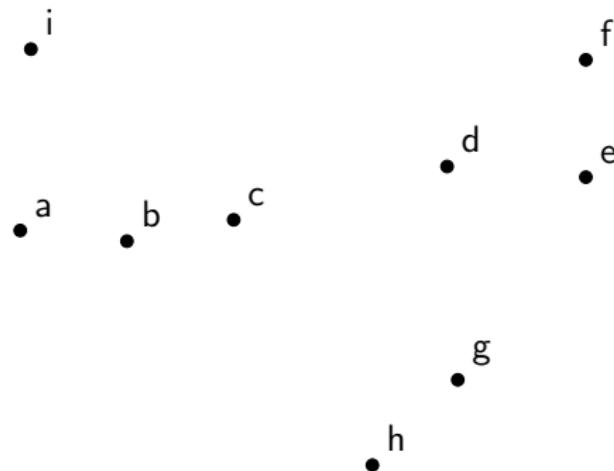




Filamentos após o corte

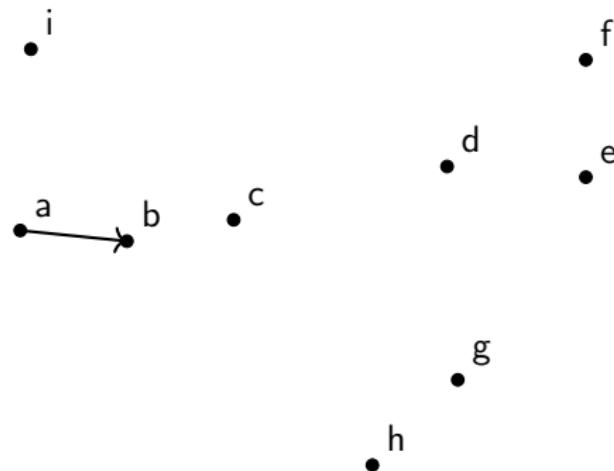
# Gerando um caminho

- Pular para o vizinho mais próximo.



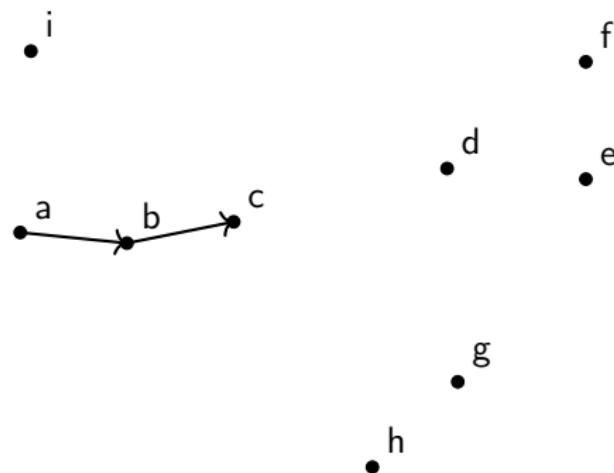
# Gerando um caminho

- Pular para o vizinho mais próximo.



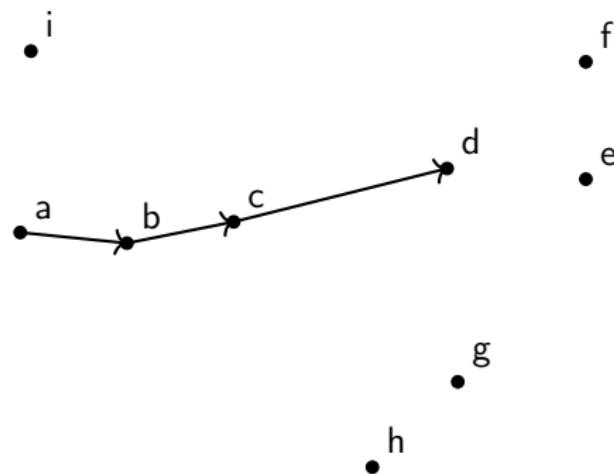
# Gerando um caminho

- Pular para o vizinho mais próximo.



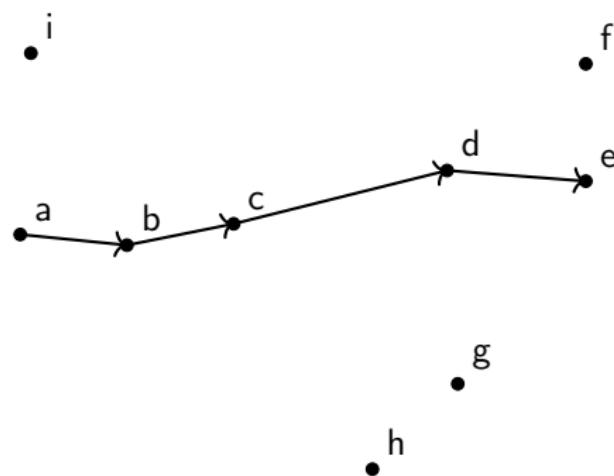
# Gerando um caminho

- Pular para o vizinho mais próximo.



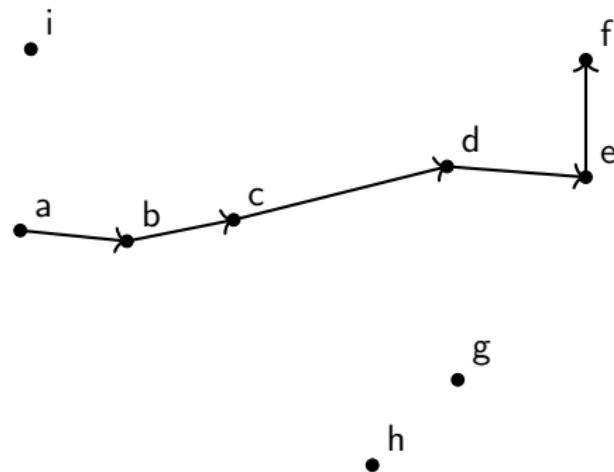
# Gerando um caminho

- Pular para o vizinho mais próximo.



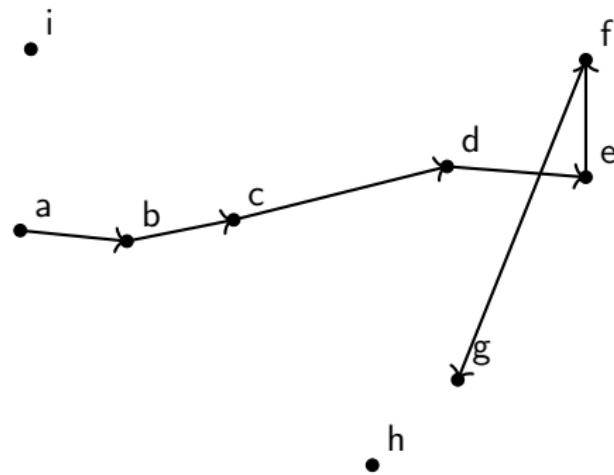
# Gerando um caminho

- Pular para o vizinho mais próximo.



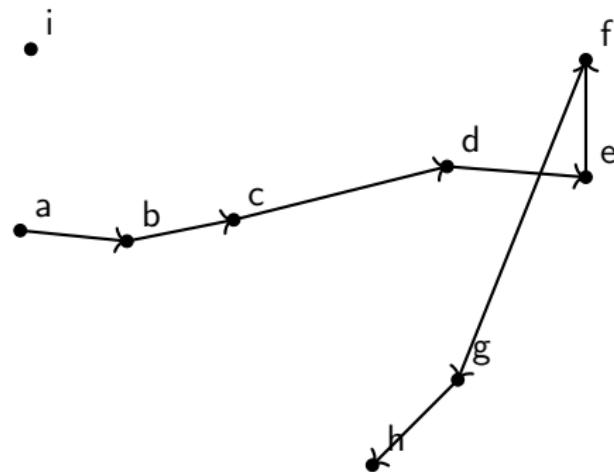
# Gerando um caminho

- Pular para o vizinho mais próximo.



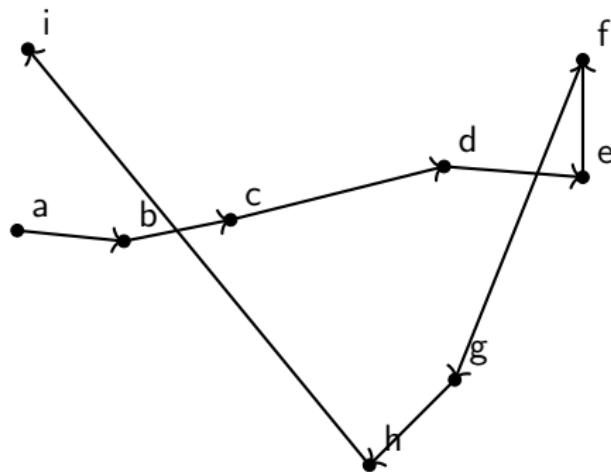
# Gerando um caminho

- Pular para o vizinho mais próximo.



# Gerando um caminho

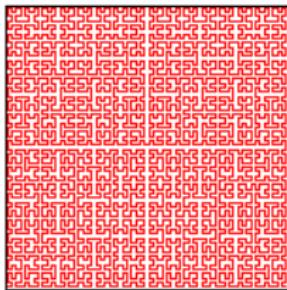
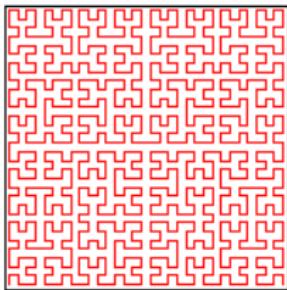
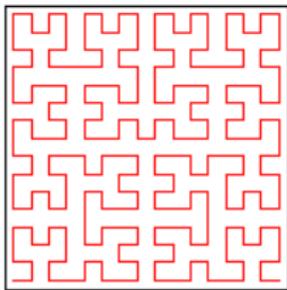
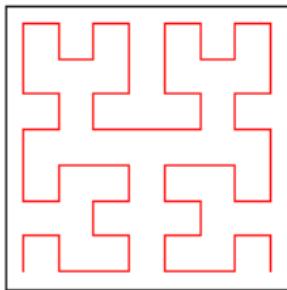
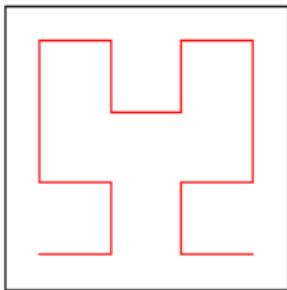
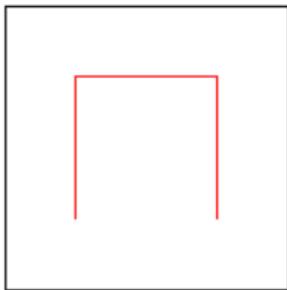
- Pular para o vizinho mais próximo.



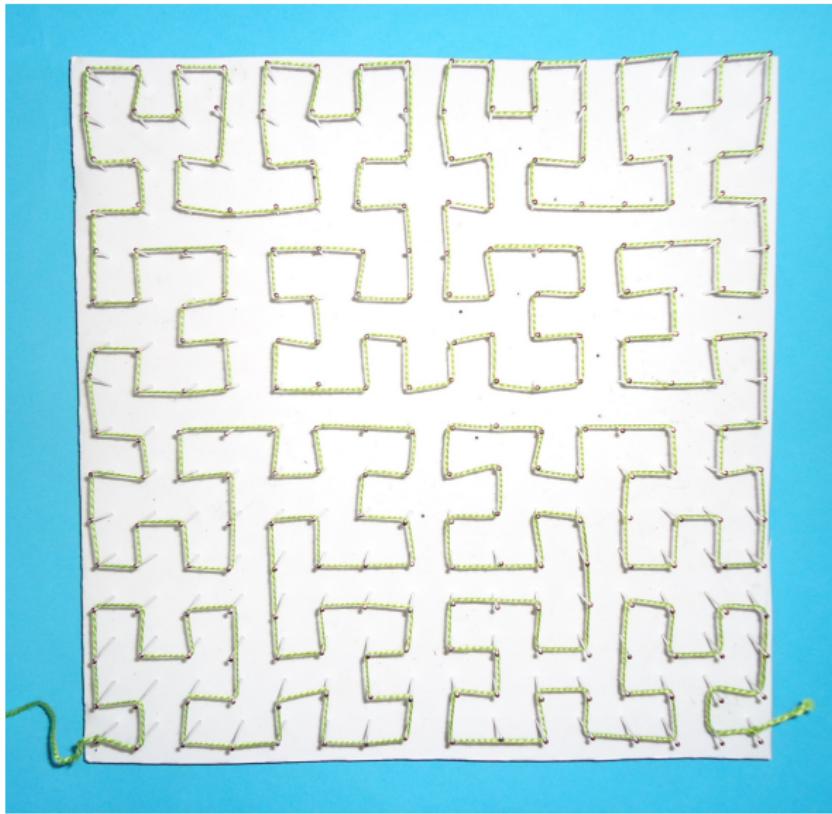
- Dois jeitos de encontrar o vizinho mais próximo:
  - ▶ Busca exaustiva
  - ▶ FLANN: Fast library for approximate nearest neighbours

- Dois jeitos de encontrar o vizinho mais próximo:
  - ▶ Busca exaustiva
  - ▶ FLANN: Fast library for approximate nearest neighbours
- Ambos são lentos: outro jeito?

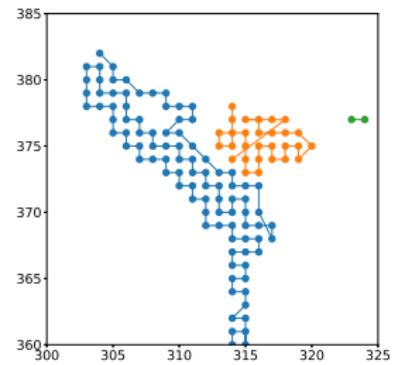
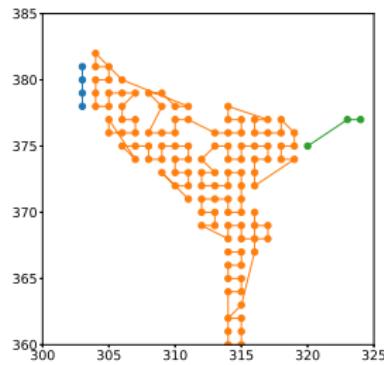
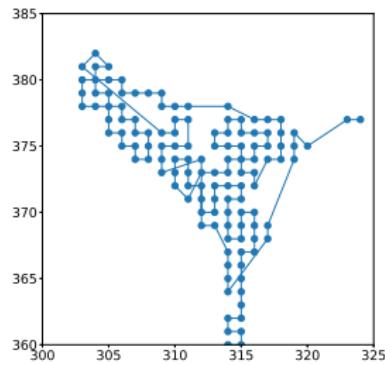
# A curva de Hilbert



# A curva de Hilbert



# Comparando os métodos



## **Lista de filamentos da point cloud**

Codificação de cada filamento

*Escolha do codec*

Filamentos curtos

Filamentos longos

Transmissão como  
informação lateral

Codificação diferencial

Codificação aritmética

# Codificação aritmética

- Obter a função de probabilidade acumulada.

# Codificação aritmética

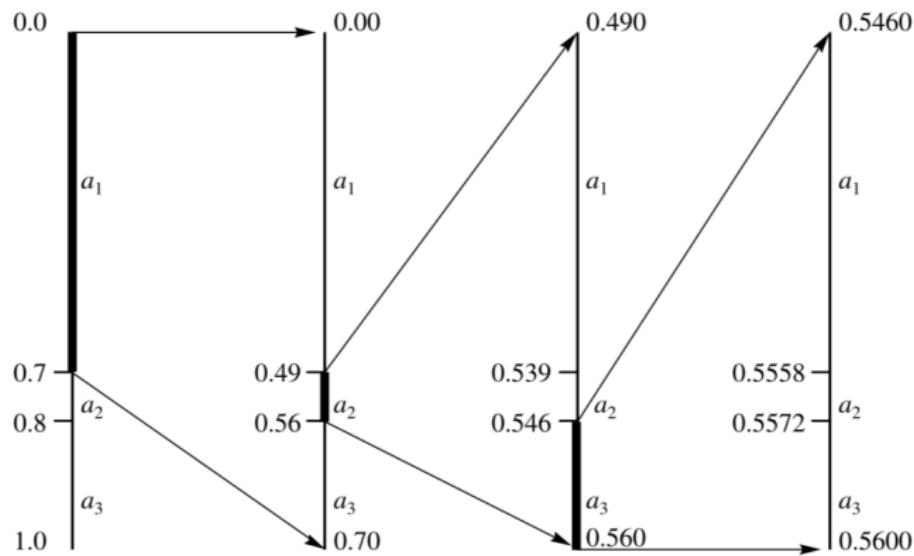
- Obter a função de probabilidade acumulada.

$$\mathcal{A} = \{a_1, a_2, a_3\}$$

$$P(a_1) = 0.7 ; P(a_2) = 0.1 ; P(a_3) = 0.2$$

$$F(a_1) = 0.7 ; F(a_2) = 0.8 ; F(a_3) = 1$$

- Para cada símbolo, escolher um subintervalo e dividi-lo de novo.
- Tag*: qualquer número do último intervalo.



# Codificação diferencial

## Definição

- Gera uma estimativa e transmite o erro.

# Codificação diferencial

## Definição

- Gera uma estimativa e transmite o erro.

$$\hat{x}_i = f(x_0, x_1, \dots, x_{i-1})$$

$$e_i = \hat{x}_i - x_i$$

$$x_i = \hat{x}_i - e_i$$

# Codificação diferencial

## Definição

- Gera uma estimativa e transmite o erro.

$$\hat{x}_i = f(x_0, x_1, \dots, x_{i-1})$$

$$e_i = \hat{x}_i - x_i$$

$$x_i = \hat{x}_i - e_i$$

## Exemplo

- Usando o último valor como estimativa:

$$\hat{x}_i = x_{i-1}$$

$$e_i = x_{i-1} - x_i$$

$$x_i = x_{i-1} - e_i$$

## Exemplo

- Usando o último valor como estimativa:

$$\hat{x}_i = x_{i-1}$$

$$e_i = x_{i-1} - x_i$$

$$x_i = x_i - 1 - e_i$$

- Aplicando:

$$x = (0, 1, 3, 4, 5, 4)$$

$$e = (1, 2, 1, 1, -1)$$

# Table of Contents

1 Motivação

2 Teoria

3 Metodologia

4 Resultados

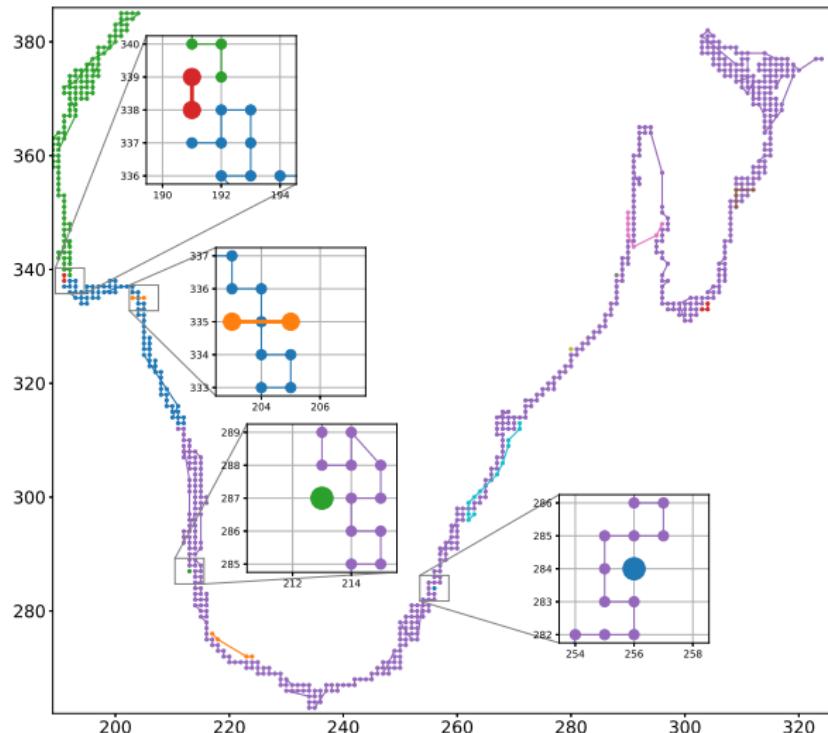
5 Conclusão

# Comparação do tempo de execução

	<b>Exato</b>	<b>FLANN</b>	<b>Hilbert</b>
Point cloud 1	47.34	32.37	0.13
Point cloud 2	106.51	80.29	0.21
Point cloud 3	72.00	48.66	0.20
Point cloud 4	63.61	43.28	0.18
Point cloud 5	112.85	78.88	0.24

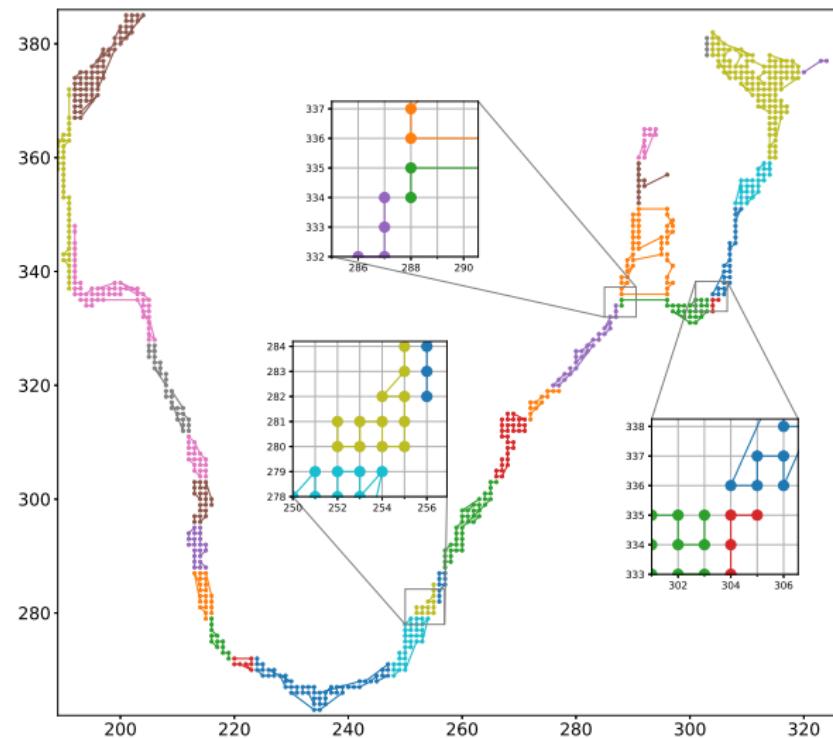
**Tabela:** Tempo de execução em segundos

# Comparação dos algoritmos de geração de filamentos



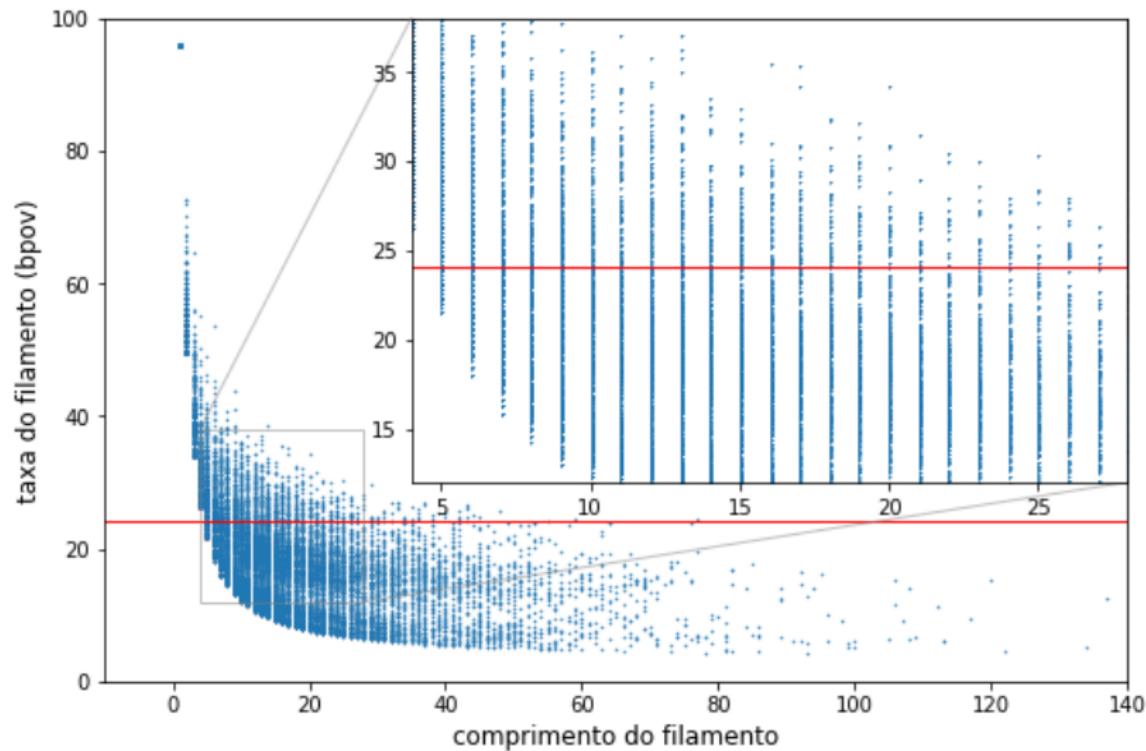
Filamentos pequenos gerados pelo algoritmo “exato”

# Comparação dos algoritmos de geração de filamentos



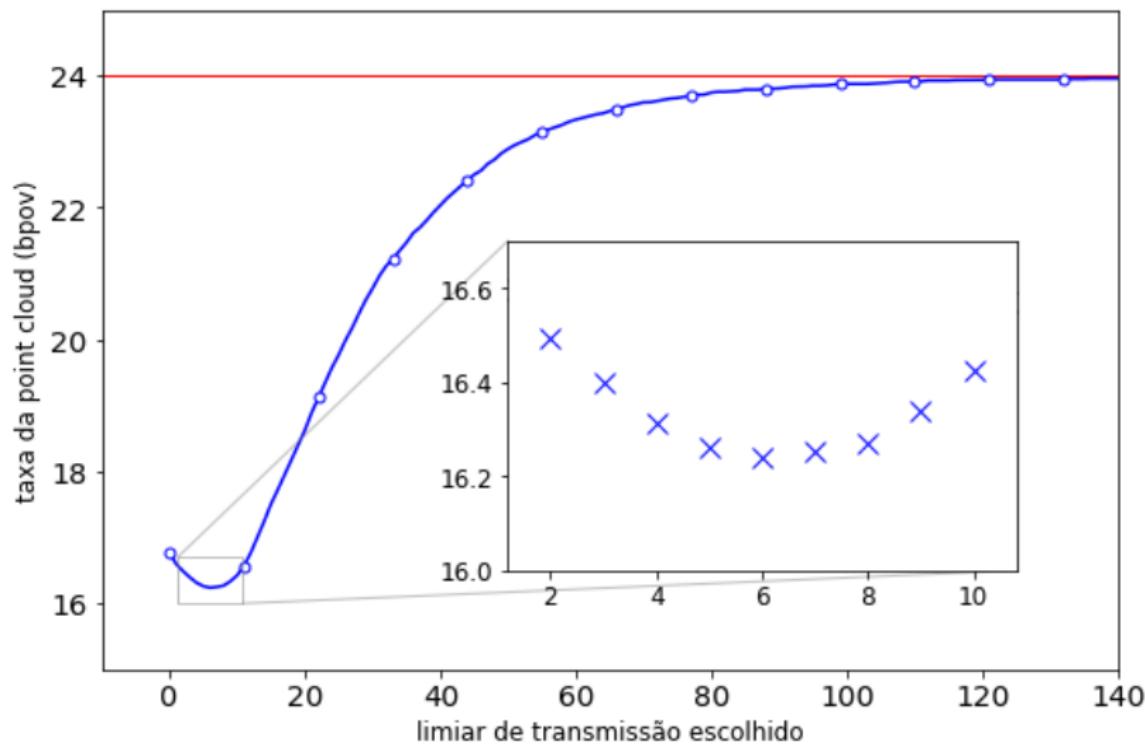
Fraturas geradas pelo algoritmo “Hilbert”

# Escolha do codec: otimizando limiar de transmissão



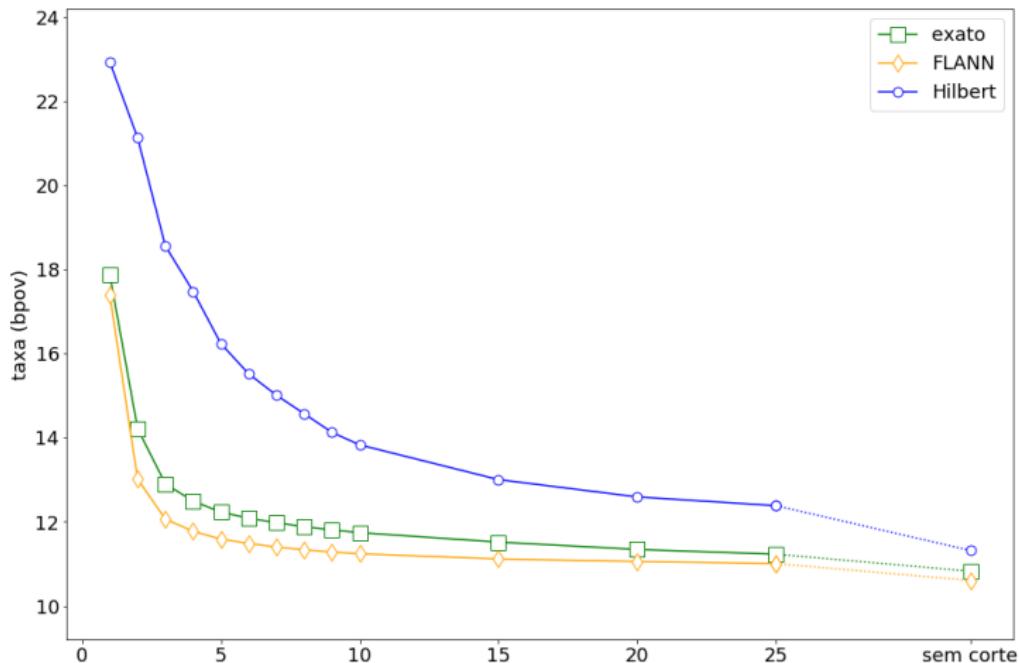
Taxa dos filamentos em função do comprimento

## Escolha do codec: otimizando limiar de transmissão



Taxa em função do limiar de transmissão

# Escolha do limiar de corte do filamento



Melhor taxa em função do limiar de distância de corte

# Resultados da compressão

Tabela: Comparação com os resultados de trabalhos anteriores

Arquivo	GPCC	MTDC	“exato”	“FLANN”	“Hilbert”
ricardo9/frame0000	5.8	13.8	9.9	9.7	10.3
ricardo9/frame0091	6.1	14.6	10.7	10.4	11.1
phil9/frame0116	9.7	22.3	15.3	14.9	15.8
andrew9/frame0263	10.5	24.6	16.7	16.5	17.0
sarah9/frame0146	5.7	16.5	10.8	10.6	11.3

# Table of Contents

1 Motivação

2 Teoria

3 Metodologia

4 Resultados

5 Conclusão

- Avanço em relação ao algoritmo anterior
- Longe do estado da arte
- Ainda há melhorias possíveis

Obrigado