



**MÉTODO DE COMPRESSÃO DE NUVENS DE
PONTOS COM PERDAS A TAXA
DETERMINADA BASEADO NO
CODIFICADOR SILHOUETTE 3D**

PEDRO HENRIQUES NOGUEIRA

**TRABALHO DE CONCLUSÃO DE CURSO
EM ENGENHARIA DA COMPUTAÇÃO**

**DEPARTAMENTO DE ENGENHARIA ELÉTRICA/CIÊNCIA DA
COMPUTAÇÃO**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica/Ciência da Computação

Método de Compressão de Nuvens de Pontos com Perdas a Taxa
Determinada Baseado no Codificador Silhouette 3D

Pedro Henriques Nogueira

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDA AO DEPARTAMENTO
DE ENGENHARIA ELÉTRICA DA UNIVERSIDADE DE BRASÍLIA COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU
DE ENGENHEIRO DA COMPUTAÇÃO.

APROVADA POR:

Prof. Bruno Luigi Macchiavello Espinoza, Dr., CiC/UnB
(Orientador)

Prof. Eduardo Peixoto Fernandes da Silva, Ph.D, FT/UnB
(Co-orientador)

Nilson Donizete Guerin Júnior, Me., Grupo Stefanini
(Examinador Interno)

Brasília/DF, dezembro de 2023.

FICHA CATALOGRÁFICA

NOGUEIRA, PEDRO

Método de Compressão de Nuvens de Pontos com Perdas a Taxa Determinada Baseado no Codificador Silhouette 3D. [Brasília/DF] 2023.

vi, 36p., 210 x 297 mm (ENE/FT/UnB, Engenheiro da Computação, Trabalho de Conclusão de Curso, 2023).

Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica/Ciência da Computação.

Departamento de Engenharia Elétrica/Ciência da Computação

1. Nuvem de pontos 2. Compressão com perdas de sinais

3. Silhouette 3D

I. ENE/FT/UnB

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

NOGUEIRA, PEDRO (2023). Método de Compressão de Nuvens de Pontos com Perdas a Taxa Determinada Baseado no Codificador Silhouette 3D. Trabalho de Conclusão de Curso, Departamento de Engenharia Elétrica/Ciência da Computação, Universidade de Brasília, Brasília, DF, 36p.

CESSÃO DE DIREITOS

AUTOR: Pedro Nogueira

TÍTULO: Método de Compressão de Nuvens de Pontos com Perdas a Taxa Determinada Baseado no Codificador Silhouette 3D.

GRAU: Engenheiro da Computação

ANO: 2023

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Conclusão de Curso e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho de conclusão de curso pode ser reproduzido sem autorização por escrito do autor.

Pedro Nogueira

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

Faculdade de Tecnologia - FT

Departamento de Engenharia Elétrica/Ciência da Computação (ENE)

Brasília - DF CEP 70919-970

Dedico esse trabalho aos meus pais, Giuseppe e Lillian, aos meus vários amigos e professores de curso na UnB, e à minha companheira de vida, Ana.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, Giuseppe e Lilian, por terem me colocado nesse caminho de muitas conquistas desde minha infância. Todos os meus anos de universitário culminam nesse projeto, e espero que tenham orgulho dessa minha trajetória como um todo.

Agradeço também aos meus professores Bruno e Eduardo, Café e Queiroz, Castanho e Genáina, dentre vários outros. Vocês são o motivo para essa universidade funcionar dessa forma excelente, e já por isso o mundo será um lugar muito melhor.

Obrigado a todos os meus amigos e colegas que encontrei nessa minha jornada no meu curso de graduação. Desde colegas que acompanhei em matérias, até amigos com quem fiz estágios e confraternizações. Que todos consigam sucesso em suas vidas acadêmicas!

Os departamentos da UnB responsáveis pelo meu curso - o Departamento de Matemática, o Instituto de Física, o Departamento de Engenharia Elétrica, e o Departamento de Ciência da Computação, todos merecem meus mais sinceros agradecimentos.

E à minha amada, Ana, companheira mais próxima de mim, que me acompanhou pelos meus últimos passos no curso - que esse seja motivo para você se orgulhar de mim!

RESUMO

Nuvens de pontos, ou *point clouds*, são formas de representação de imagens 3D com grande potencial nas aplicações contemporâneas. Uma forma de codificar a geometria dessas nuvens de pontos se dá pelo procedimento *Silhouette 3D*, e este já apresenta diversas melhorias sobre seu antecessor, o método Octree, tanto em quesito de representação quanto de compressão do objeto. Avistou-se agora a necessidade de se comprimir mais ainda a representação da geometria no *Silhouette 3D* por meio de compressão com perdas de dados, para que fosse possibilitada então a utilização desse método para aplicações mais limitadas em armazenamento ou transmissão de bits. O projeto então sugere formas possíveis de se comprimir uma nuvem de pontos visando o menor impacto possível em sua coerência. O projeto foi realizado em MATLAB e apresenta como resultados os gráficos de eficiência de cada método aplicado em diversas nuvens de pontos diferentes.

Palavras-chave: Nuvem de pontos (point cloud); Compressão com perdas de sinais; Silhouette 3D.

ABSTRACT

Point clouds are a way of representing 3D images that have great potential in modern applications. A way to encode a point cloud's geometry is by using the Silhouette 3D method, a method which shows enhancements over its predecessor, the Octree method, for its form of representation as well as for its compression capabilities of the point cloud. In the Silhouette 3D method, a need of compressing the point cloud even further arose for the use of more limited applications in terms of data storage and transmission, thus lossy compression methods were made desirable. This project suggests different methods for lossy compressions of point clouds while still maintaining the most cohesion possible. The project was developed in MATLAB e presents as results efficiency graphs for each method applied in different point clouds.

Keywords: Point cloud; Lossy signal compression; Silhouette 3D.

SUMÁRIO

Sumário	i
Lista de figuras	iii
Lista de tabelas	v
Glossário	vi
Capítulo 1 – Introdução	1
1.1 Motivação	1
1.2 Descrição do problema	1
1.3 Organização	2
Capítulo 2 – Fundamentação Teórica	3
2.1 Nuvens de pontos	3
2.1.1 Voxelização de uma Nuvem de Pontos	4
2.2 Compressão com perdas/sem perdas	4
2.3 Métricas de distorção de sinais	6
2.4 Técnicas de compressão de nuvens de pontos	7
2.4.1 Octree	7
2.4.2 Silhouette 3D	9
Capítulo 3 – Método Proposto	14
3.1 Banco de dados de nuvens de pontos utilizadas	15
3.2 Métodos de compressão	18
3.2.1 Adiciona nada	18
3.2.2 Adiciona pai uma vez	19
3.2.3 Adiciona pai duas vezes	20
3.3 Adaptação do código existente para o projeto atual	20
3.4 Método de escolha de folhas para compressão	21
3.4.1 Escolha estática	22

3.4.2	Escolha por quantidade de bits	22
3.4.3	Escolha por quantidade de bits em função da MSE	23
Capítulo 4 – Resultados		25
4.1	Métodos de compressão	25
4.2	Método de escolha de folhas para compressão	28
Capítulo 5 – Conclusão		34
5.1	Código desenvolvido	34
5.2	Trabalho futuro	35
Referências		36

LISTA DE FIGURAS

2.1	Nuvem de pontos de um objeto simples do dia-a-dia.	4
2.2	Exemplo de compressão com perdas de uma imagem.	6
2.3	Exemplo de quadtree.	8
2.4	Primeiro nível da divisão octree de um cubo.	8
2.5	Árvore gerada por divisão octree.	9
2.6	Exemplo de nuvem de pontos.	10
2.7	Voxels de cada folha da nuvem de pontos representados por quadrados com a mesma cor que o objeto.	10
2.8	Silhueta resultado do achatamento do objeto exemplo inteiro pelo eixo Z.	11
2.9	Silhuetas das duas metades e de cada folha da nuvem de pontos.	11
2.10	Silhuetas das folhas enfatizadas nas regiões de codificação.	12
2.11	Codificação das silhuetas da nuvem de pontos exemplo.	12
3.1	Fluxograma do algoritmo proposto.	15
3.2	Nuvem de pontos de busto queiroz.	16
3.3	Nuvem de pontos de busto phil.	17
3.4	Nuvem de pontos de corpo completo soldier.	17
3.5	Nuvem de pontos de corpo completo redandblack.	18
3.6	Uso do método adiciona nada no exemplo enfatizado em vermelho.	19
3.7	Uso do método adiciona pai uma vez no exemplo enfatizado em vermelho.	19
3.8	Uso do método adiciona pai duas vezes no exemplo enfatizado em vermelho.	20

4.1	Avaliação das PSNRs das folhas da nuvem de ponto queiroz0030 no eixo X. . . .	26
4.2	Avaliação das PSNRs das folhas da nuvem de ponto queiroz0077 no eixo X. . . .	26
4.3	Avaliação das PSNRs das folhas da nuvem de ponto queiroz0203 no eixo Y. . . .	27
4.4	Avaliação das PSNRs das folhas da nuvem de ponto redandblack1450 no eixo Y. . . .	27
4.5	Avaliação das PSNRs das folhas da nuvem de ponto soldier0679 no eixo Z. . . .	28
4.6	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em phil0000.	29
4.7	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em phil0024.	30
4.8	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em queiroz0000.	30
4.9	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em queiroz0032.	31
4.10	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em soldier0618.	31
4.11	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em soldier0679.	32
4.12	Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em redandblack1450.	32

LISTA DE TABELAS

3.1	Caso de exemplo de duplas de folhas de árvores de nuvens de pontos e suas quantidades de bits e MSE quando retirados.	23
-----	---	----

GLOSSARY

Nuvem de pontos dinâmica	Junção de várias nuvens de pontos em quadros.
Silhouette 3D	Algoritmo que codifica e decodifica nuvens de pontos em silhuetas.
Árvore de silhuetas	Representação em silhuetas após decomposição diádica.
Silhueta	Seção da nuvem de pontos como uma imagem binária 2D.
Folha	Nós dos níveis mais baixos de uma árvore de silhuetas.
MSE	Erro quadrático médio.
PSNR	Relação sinal-ruído de pico.

Esse capítulo apresenta a motivação para o projeto feito, assim como os objetivos que se deseja alcançar com seus resultados.

1.1 MOTIVAÇÃO

O uso de imagens 3D no mundo atual está cada vez mais em alta, e diversas pesquisas são feitas para a melhor forma de representá-las. Um método gerado para representação de espaços 3D no computador é a nuvem de pontos. Nuvens de pontos, dentre várias outras potenciais aplicações, são usadas em diversas áreas de conhecimento, como a área de mapeamento ou até mesmo de entretenimento.

Uma forma de se guardar as informações de uma nuvem de pontos é a Silhouette3D, e, por mais que esta tenha vantagens e desvantagens sobre sua precedente a Octree, ainda não havia sido expandida a área de compressão com perdas sobre esta técnica. A compressão de um sinal é uma técnica muito importante para que este sinal possa ser guardado ou transmitido mais eficientemente, e a compressão com perdas garante usos do sinal que demandam um menor tamanho deste, mesmo que este seja distorcido para ser guardado ou transmitido.

Este trabalho abre as possibilidades para que essa metodologia de representação de nuvens de pontos tenha técnicas de compressão com perdas, e faz isso apresentando algumas técnicas estudadas.

1.2 DESCRIÇÃO DO PROBLEMA

Na motivação de se ter nuvens de pontos que sejam pequenas o suficiente em quantidade de bits para que sejam armazenadas ou transmitidas em específicas aplicações, o problema consiste

na formulação de um método que diminua essa quantidade de bits. Mas, por esta se tratar de uma compressão com perdas de informações, também é desejado que esta formulação distorça a nuvem de pontos minimamente.

Por isso, por meio de técnicas de compressão, foram propostas formas de se diminuir a quantidade de bits de nuvens de pontos. Mas, usando métricas de distorção, foram propostas formas de se cortar os bits certos da representação da nuvem de pontos de forma que a compressão alcance a quantidade de bits exigida ao passo que distorça a nuvem de pontos minimamente.

À luz do problema, foi determinada uma lista de pesquisas necessárias para se conseguir atingir o objetivo final:

- Encontrar as melhores formas de comprimir a nuvem de pontos;
- Determinar as melhores partes da nuvem de pontos a serem comprimidas;
- Aplicar no código já existente de representação por silhuetas o algoritmo proposto para estas soluções.

1.3 ORGANIZAÇÃO

Este documento relata as soluções propostas para os problemas presentes acima. Este está estruturado em capítulos que encapsulam seus objetivos a seguir:

- Fundamentação Teórica - Capítulo responsável pela contextualização técnica de trabalhos passados usados neste trabalho;
- Metodologia - Apresentação dos métodos sugeridos para o desenvolvimento do projeto;
- Resultados - Produto gerado da aplicação da metodologia proposta sobre nuvens de pontos usadas de teste;
- Conclusão - Análise dos resultados alcançados.

O próximo capítulo então introduz os conceitos usados para a teorização da metodologia.

FUNDAMENTAÇÃO TEÓRICA

Esse capítulo explica a base de conhecimento necessária para se entender o trabalho feito nesse experimento. É explicado o que é uma nuvem de pontos e o que significa uma nuvem de pontos ser voxelizada, compressão de sinais com perdas e sem perdas, métricas de compressão de sinais, e técnicas de compressão de nuvens de ponto.

2.1 NUVENS DE PONTOS

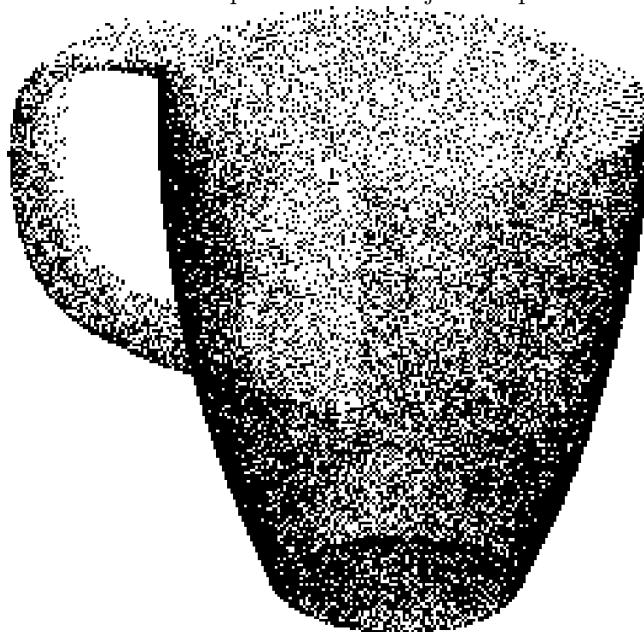
Nuvens de pontos, ou *point clouds*, são geometrias tridimensionais compostas por diversos pontos presentes em um plano cartesiano tridimensional $[x \ y \ z]$. Uma maneira de se obter uma nuvem de pontos é por meio de câmeras stereo de profundidade capazes de mapear o posicionamento 3D, a cor, e a refletância de diversos pontos da superfície de um objeto, ou usando a tecnologia do *LiDAR*. A forma externa do objeto é construída a partir da junção desses pontos, havendo então a possibilidade do estudo do tal objeto de diversas formas que a simples representação 2D deste não proporcionaria. Quanto melhor a qualidade de captura das câmeras utilizadas, maior resolução a nuvem de pontos terá, significando ter mais pontos mais próximos um do outro na superfície do objeto. Uma nuvem de pontos é classificada como estática quando é capturada em um único instante de tempo ou como dinâmica quando é constituída por uma sequência de nuvens de pontos estáticas, assim como são formados os vídeos 2D, com quadros de imagens 2D em sequência.

As aplicações dessa tecnologia já abrangem várias áreas da ciência e ainda têm a possibilidade de se expandir ainda mais, podendo ser a solução para muitas aplicações do dia a dia ou até mesmo de grandes empresas. A Microsoft já usou essa tecnologia no desenvolvimento de jogos eletrônicos diferenciados no uso do Kinect, e a tecnologia do LiDAR (OCEANIC; ADMINISTRATION, 2021) lidera no campo de mapeamento e topografia para carros autônomos.

2.1.1 Voxelização de uma Nuvem de Pontos

Assim como o ponto atômico de uma imagem 2D é chamado de píxel (*picture element*), foi dado o nome a um desses pontos de uma nuvem de pontos de voxel (*volume element*). As posições dos pontos de uma nuvem de pontos são obtidas da câmera com coordenadas geométricas de ponto flutuante, o que significa que sua posição 3D $[x \ y \ z]$ é representada por números racionais, e não simplesmente inteiros. Para a melhor manipulação desses pontos, são feitas suas voxelizações, o que significa que cada voxel recebe sua posição 3D com simples coordenadas inteiras, dessa forma posicionando-os em uma grade tridimensional. Nuvens de pontos voxelizadas são muito mais flexíveis para os seus usos, assim tornando mais fácil a análise do objeto em questão. A figura 2.1 mostra um objeto do dia-a-dia na representação de nuvem de pontos voxelizada.

Figura 2.1. Nuvem de pontos de um objeto simples do dia-a-dia.



Fonte: Autodesk (AUTODESK, 2018).

2.2 COMPRESSÃO COM PERDAS/SEM PERDAS

A compressão é definida como o processo de representar um sinal utilizando menos dados que o próprio sinal proporciona, o que é útil para seu armazenamento e para sua transmissão por tornar o sinal menor em quantidade de bits. Há duas formas de se comprimir um sinal

- a compressão sem perdas e a compressão com perdas. Como os próprios nomes já indicam, a compressão sem perdas de um sinal garante que há pelo menos um procedimento pelo qual o sinal original por completo pode ser reconstruído, enquanto a compressão com perdas não possibilita isso.

A compressão sem perdas de um sinal tem sua importância no fato de que o sinal original não é perdido ou corrompido, o que é muitas vezes desejado para arquivos executáveis, textos, dentre outros exemplos que precisam ser transmitidos ou guardados por completo. Por mais que seja contra intuitiva a perda de dados de um sinal, há também importância na compressão com perdas. Essa compressão geralmente é usada em sinais que, mesmo sofrendo leves alterações, ainda são similares o suficiente para um ser humano, mas que recebem grande mudança em suas capacidades computacionais. Um exemplo disso é uma pequena diferença dos valores de um píxel de uma imagem que geralmente é imperceptível para uma pessoa que a vê, mas que pode ser uma diminuição drástica em seu tamanho em bits no computador, ou até mesmo uma forma que simplifica um processamento futuro dessa imagem.

Com essas informações sobre os diferentes tipos de compressão, deve-se calcular uma métrica de desempenho sobre sinais comprimidos. A compressão sem perdas calcula seu desempenho simplesmente a partir da taxa de compressão alcançada, taxa de compressão calculada na equação 2.1, enquanto a compressão com perdas deve levar em consideração não só a taxa de compressão, mas também a distorção infligida no sinal. Essas duas características que então estabelecem a efetividade da compressão sobre o sinal.

$$CR = \frac{S_O}{S_C} \quad (2.1)$$

A taxa máxima de compressão de uma compressão sem perdas de sinais é a entropia do sinal, que é a definição da medida de previsibilidade de ocorrência de dados do sinal feita por Claude Elwood Shannon (SHANNON, 1948) pela equação 2.2.

$$H = - \sum p_i I(p_i) \quad (2.2)$$

Onde:

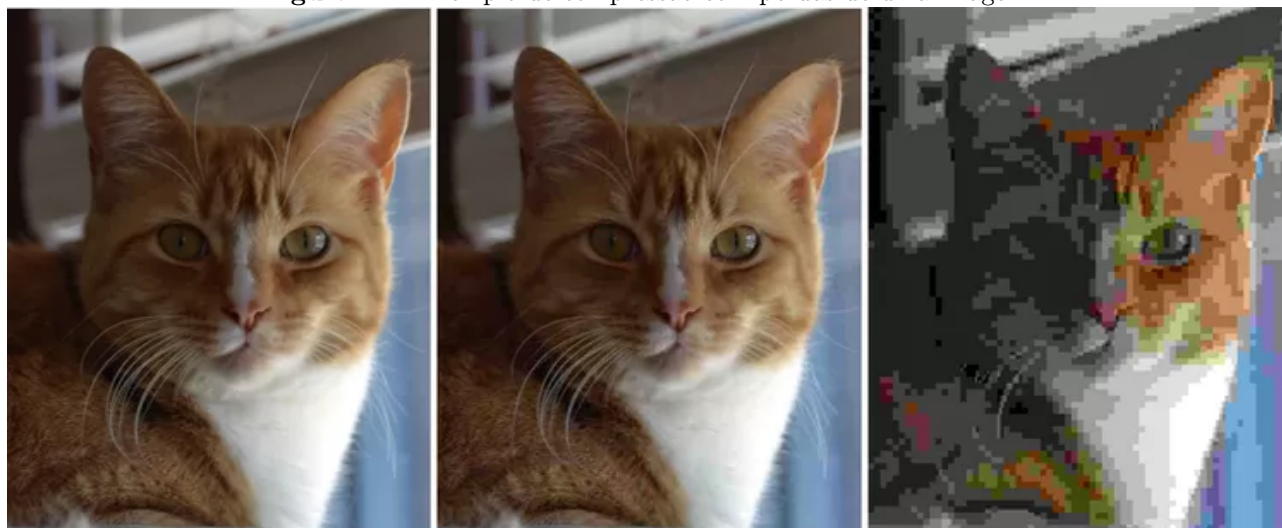
H = entropia;

p_i = probabilidade de ocorrência do símbolo i ;

I = redundância do evento;

A entropia de um sinal é definida pela quantidade média de números necessária para se representar o sinal.

Figura 2.2. Exemplo de compressão com perdas de uma imagem.



Fonte: Lifewire (PLUMRIDGE, 2020).

2.3 MÉTRICAS DE DISTORÇÃO DE SINAIS

A compressão de sinais sem perdas não consegue alcançar uma taxa de compressão maior que a entropia do próprio sinal, então não há sentido uma compressão com perdas ter uma taxa de compressão menor que a entropia do sinal, ou pelo menos menor que a maior taxa de compressão sem perdas alcançada para o sinal. A compressão com perdas tem a possibilidade de ter uma taxa de compressão menor até que a entropia, e por isso seu desempenho é calculado pela relação entre a taxa de compressão e a distorção feita no sinal.

A métrica de distorção do sinal pode ser obtida de forma subjetiva ou objetiva a partir dos dados distorcidos. Uma forma de conseguir uma métrica subjetiva pode ser uma simples pergunta do que um usuário acha melhor dentre diferentes formas de distorção do sinal, o que, mesmo soando banal, pode ser a melhor solução para a funcionalidade em questão. Para uma métrica objetiva, que mais provavelmente avaliará sinais que serão utilizados por computadores

e máquinas, há a medida de erro quadrático médio, a MSE.

A MSE de geometria para nuvens de pontos estabelecida da MPEG PCC (*point cloud processing*) (MEKURIA *et al.*, 2016) é o cálculo das diferenças entre as geometrias dos voxels da nuvem de pontos original e a nuvem de pontos modificada, contendo resultados de valores menores para nuvens de pontos pouco modificadas e valores elevados para nuvens muito diferentes uma da outra. Ambas as equações para a MSE (equação 2.3) e PSNR (equação 2.5) são como a seguir:

$$d_{symmetric_rms}(V_o, V_d) = \max(d_{rms}(V_o, V_d), d_{rms}(V_d, V_o)) \quad (2.3)$$

$$BBwidth = \max((x_{max} - x_{min}), (y_{max} - y_{min}), (z_{max} - z_{min})) \quad (2.4)$$

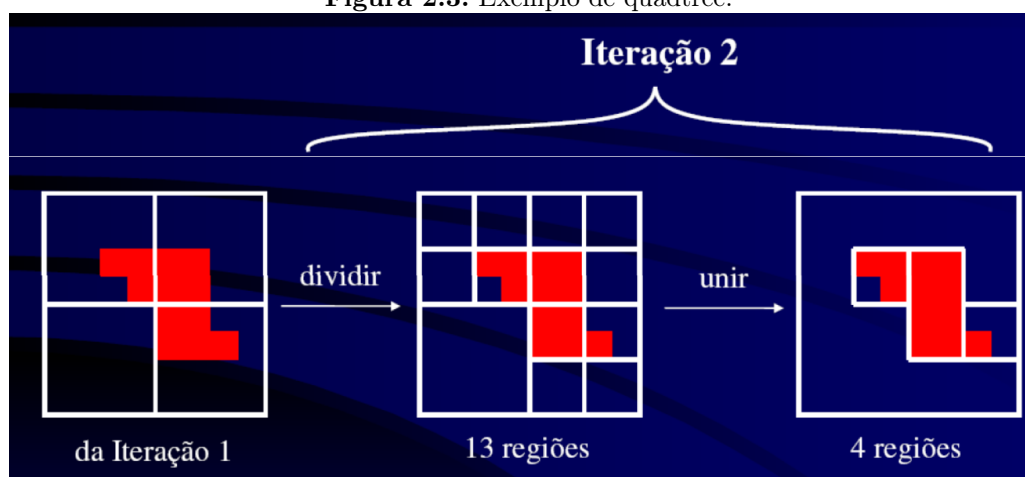
$$psnr_{geom} = 10 \log_{10}(|BBwidth|/d_{symmetric_rms}(V)^2) \quad (2.5)$$

2.4 TÉCNICAS DE COMPRESSÃO DE NUVENS DE PONTOS

Nuvens de pontos dinâmicas podem ser comprimidas por inter coders (VPCC - video point cloud compression) ou intra coders (GPCC - geometry point cloud compression), onde os algoritmos intra coders codificam cada um dos quadros da nuvem de pontos individualmente e os inter coders codificam a nuvem de pontos dinâmica como um todo, visando procurar as melhores formas de comprimir todos os quadros no aspecto temporal. A compressão de nuvens de pontos estáticas procura maneiras de comprimir os voxels em seus vários aspectos, como suas posições geométricas e suas cores. O foco dos estudos presentes se volta à compressão da geometria dos voxels de nuvens de pontos estáticas.

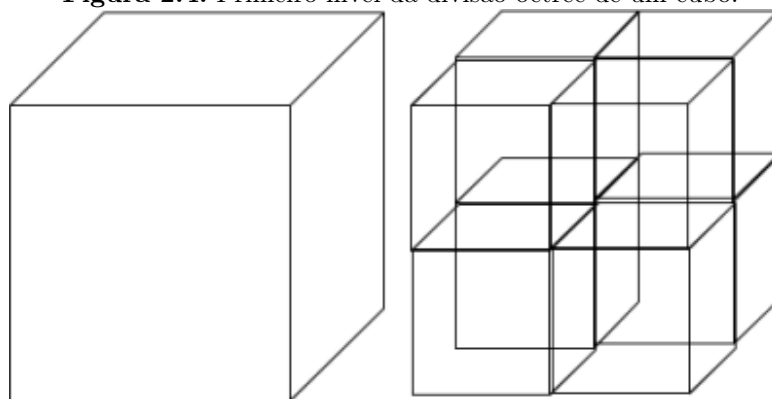
2.4.1 Octree

O método de compressão Octree consiste em dividir a nuvem de pontos ao nível atômico de voxels por meio de uma divisão de quadrantes do espaço. Esse método é conhecido como divisão Quadtree na dimensão das imagens 2D, e os procedimentos de ambos são muito parecidos.

Figura 2.3. Exemplo de quadtree.

Fonte: (MACCHIAVELLO, 2023).

O método Octree (DASKALAKI *et al.*, 2010), diferente do quadtree, corta o espaço no meio de seus três eixos (eixo vertical, horizontal, e de profundidade), e forma 8 sub-espacos com metade das 3 dimensões originais. Conferem-se então quais dos 8 cubos contêm parte da nuvem de pontos, e então o procedimento é repetido para cada um dos cubos encontrados até a resolução desejada. O resultado disso são vários cubinhos no espaço 3D, cada um representante de um diferente voxel da nuvem de pontos.

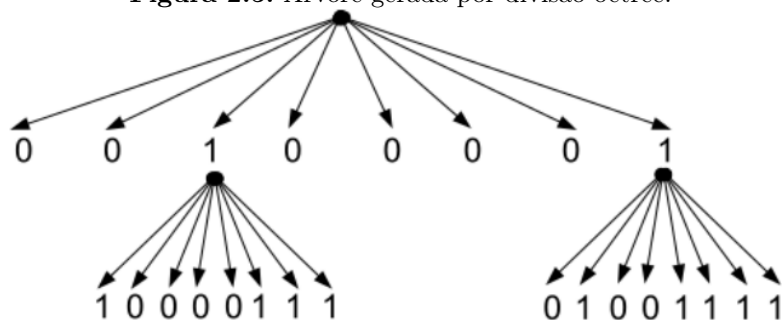
Figura 2.4. Primeiro nível da divisão octree de um cubo.

Fonte: Codificação de octrees (QUEIROZ; CHOU, 2016).

Essa técnica resume a nuvem de pontos a uma árvore de valores binários, onde cada cubo que contém dados da nuvem de pontos é representado por um 1 e todos seus cubos contidos são suas folhas, então uma árvore de no máximo oito folhas por nó, cada nó tendo o valor de 0 ou 1, ou seja, a informação de que a figura está presente ou não nesse quadrante formado.

Essa codificação diminui da nuvem de pontos a necessidade da representação de espaços vazios, codificando somente pontos presentes na nuvem. Esses pontos também podem ser reduzidos ao se modificar a resolução desejada, não cortando os quadrantes a tamanhos muito pequenos, diminuindo assim a quantidade de quadrantes usados no final.

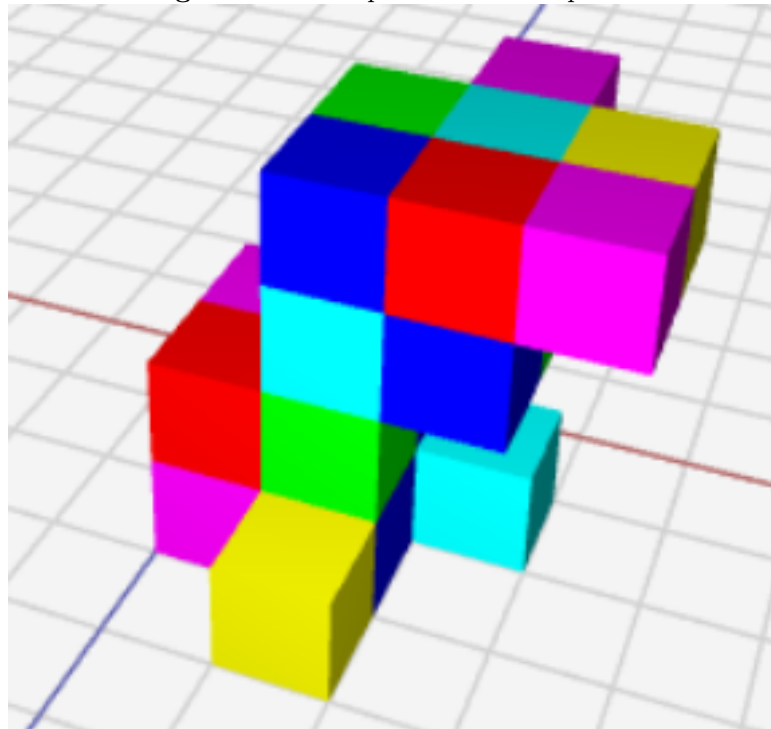
Figura 2.5. Árvore gerada por divisão octree.



Fonte: Codificação de octrees (QUEIROZ; CHOU, 2016).

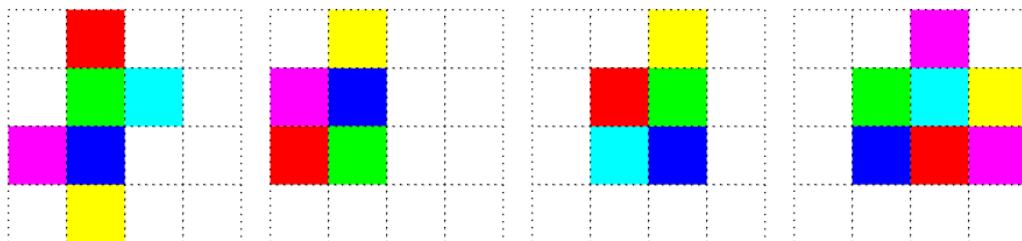
2.4.2 Silhouette 3D

O método de decomposição da nuvem de pontos em silhuetas (PEIXOTO, 2020) consiste na quebra da nuvem de pontos de dimensões $N \times N \times N$ em N folhas bidimensionais de dimensões $N \times N$. Para fins desse trabalho que se concerne somente com as posições geométricas dos voxels, as informações que as folhas terão serão de simplesmente voxels presentes ou não presentes em cada posição da folha. Para exemplificar o procedimento de quebra de uma nuvem de pontos em silhuetas, foi criado um objeto 3D de dimensões $4 \times 4 \times 4$ composto por voxels representados por cubos de dimensões $1 \times 1 \times 1$.

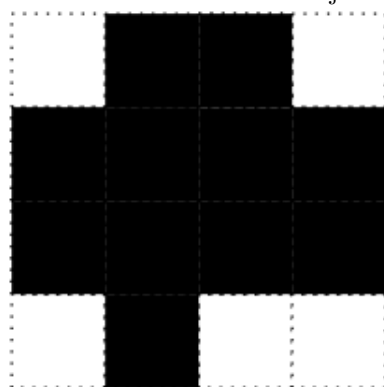
Figura 2.6. Exemplo de nuvem de pontos.

Fonte: Autoria própria.

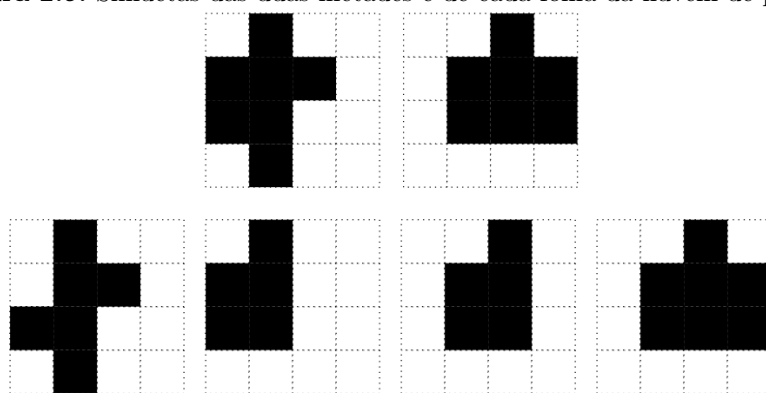
O procedimento Silhouette 3D começa pela seleção de um eixo pela qual a nuvem de pontos será dividida. Então é feita uma operação OU de todos os voxels da imagem visíveis a partir da perspectiva do eixo escolhido, obtendo-se assim a silhueta do objeto por completo. Este procedimento é chamado de achatamento. No exemplo, o eixo da altura Z foi escolhido para a perspectiva.

Figura 2.7. Voxels de cada folha da nuvem de pontos representados por quadrados com a mesma cor que o objeto.

Fonte: Autoria própria.

Figura 2.8. Silhueta resultado do achatamento do objeto exemplo inteiro pelo eixo Z.**Fonte:** Autoria própria.

Após a obtenção dessa silhueta, deve-se cortar o objeto no meio na orientação escolhida e aplicar o achatamento de ambas as metades para se conseguir mais duas silhuetas. Esse procedimento deve ser aplicado recursivamente até que as últimas silhuetas obtidas sejam as próprias folhas de voxels da nuvem de pontos.

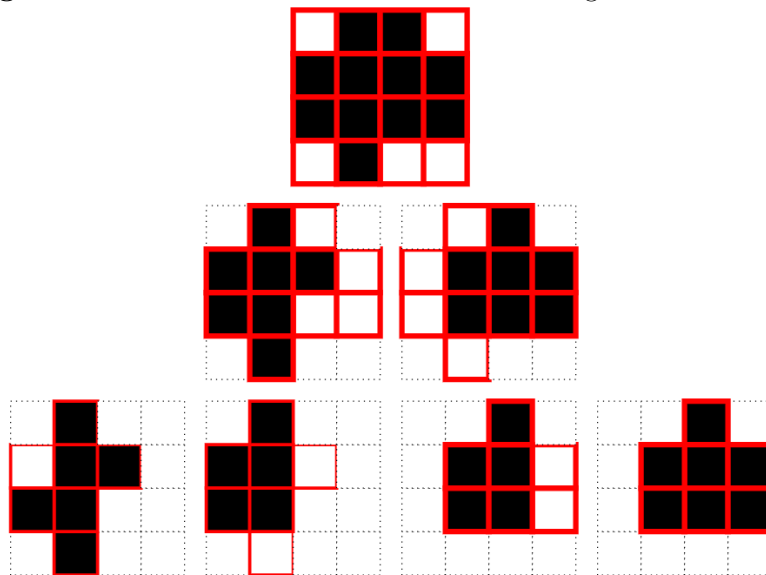
Figura 2.9. Silhuetas das duas metades e de cada folha da nuvem de pontos.**Fonte:** Autoria própria.

Essa forma de tratamento da nuvem de pontos proporciona um grande teor de redundância, e essa redundância é usada para a compressão da nuvem de pontos.

A primeira silhueta, obtida pelo achatamento da nuvem de pontos por completo, deve ser inteiramente codificada sempre pois cada informação de voxel presente ou não na silhueta é nova para o codificador. O próximo nível de codificação, que codifica a silhueta das duas metades da nuvem de pontos, somente precisa codificar as posições já existentes na silhueta pai pois não há possibilidade de existir voxel presente na silhueta filho onde não há voxel na silhueta pai.

Cada nível de silhuetas é codificado com o mesmo princípio. A figura 2.10 faz o procedimento de codificação da nuvem de pontos exemplo, codificando das silhuetas somente os espaços de voxels de margens vermelhas.

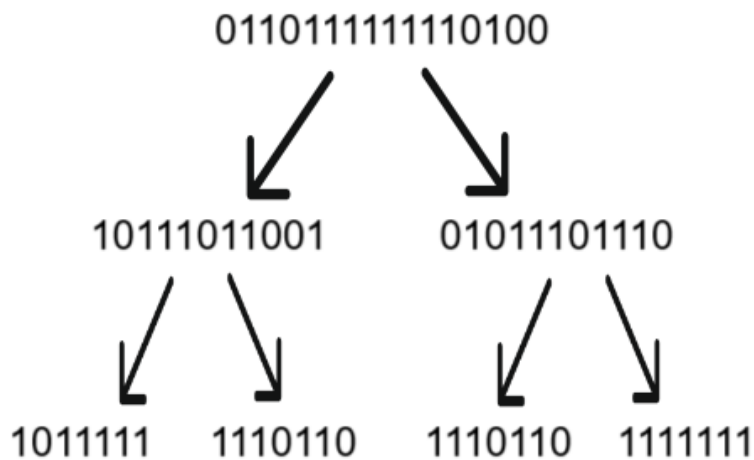
Figura 2.10. Silhuetas das folhas enfatizadas nas regiões de codificação.



Fonte: Autoria própria.

A codificação das silhuetas da nuvem de pontos do exemplo, ao se codificar espaços vazios com 0 e espaços com voxels com 1, e codificando primeiramente da esquerda para a direita e então de cima para baixo, resulta na figura 2.11.

Figura 2.11. Codificação das silhuetas da nuvem de pontos exemplo.



Fonte: Autoria própria.

A metodologia completa de codificação e decodificação de nuvens de pontos por silhuetas se

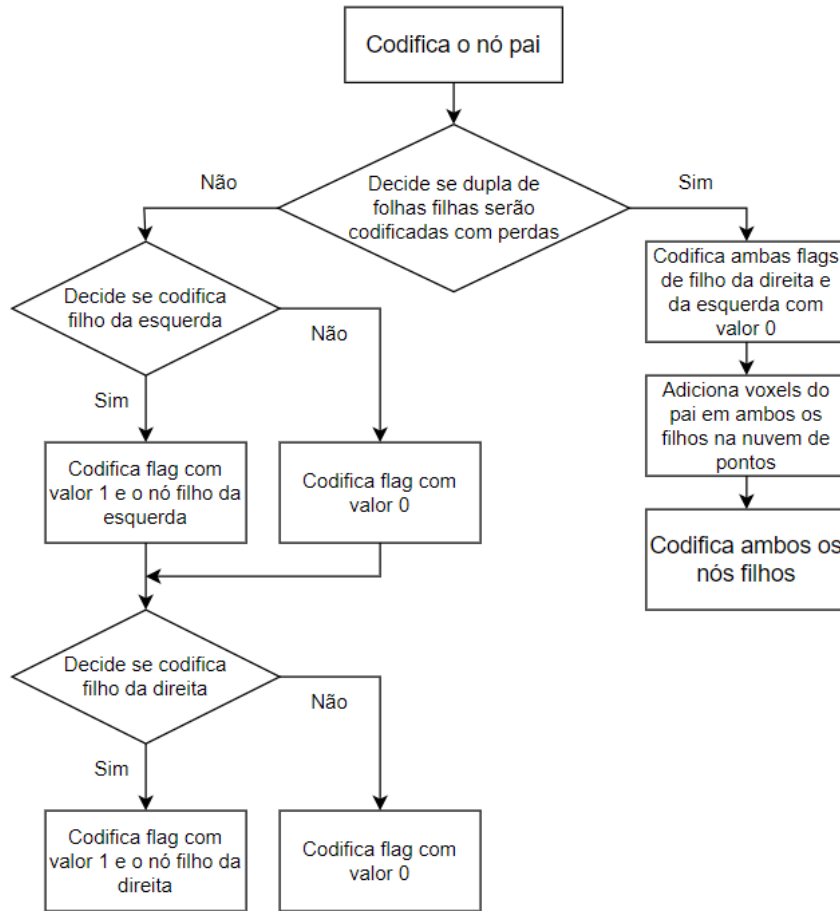
encontra no código em MATLAB no repositório do site <https://github.com/bananahell/GeometryCoder>, S3D. Essa codificação utiliza de contextos de geometria 2D para cada uma das folhas da nuvem de pontos e de contextos 3D para se aplicar a codificação aritmética, usando o nó da esquerda na árvore de silhuetas de cada nó como contexto de sua codificação aritmética.

O próximo capítulo usa da fundamentação teórica fornecida por este capítulo para explicar o trabalho proposto para esse projeto.

MÉTODO PROPOSTO

Para a viabilização do uso de nuvens de pontos em aplicações mais leves, foi notada a necessidade de comprimi-las até uma taxa de bits máxima estabelecida pela aplicação. Para isto é necessário ser feita a compressão de nuvens de pontos a uma taxa de compressão menor que sua taxa mínima sem perdas. Foi então feito o estudo sobre quais folhas da nuvem de pontos seriam as melhores para se fazer uma compressão com perdas, e sobre qual procedimento seria o melhor em termos de diminuição de bits e perdas de informações para se comprimir tais folhas.

O fluxograma a seguir explica o procedimento que o loop principal usa para a codificação nova proposta de cada folha da árvore de silhuetas. A metodologia proposta se aplica somente para os nós pais de nós de folhas, significando que somente folhas são codificadas com o método com perdas. Também, a primeira decisão e o lado direito do fluxograma são as mudanças feitas pelo metodologia com perdas, sendo que a decisão é explicada na seção 3.4 desse capítulo, o motivo para o primeiro, segundo, e terceiros retângulos do lado direito do fluxograma explicados na seção 3.3, e o modo de codificação com perdas explicado na seção 3.2.3.

Figura 3.1. Fluxograma do algoritmo proposto.

Fonte: Autoria própria.

Esse capítulo explica esses procedimentos, adentrando nos métodos de compressão propostos, explicando as mudanças feitas no código já existente de Silhouette 3D, e mostrando os métodos usados para escolher quais folhas de uma nuvem de pontos são as melhores para comprimir a nuvem de pontos.

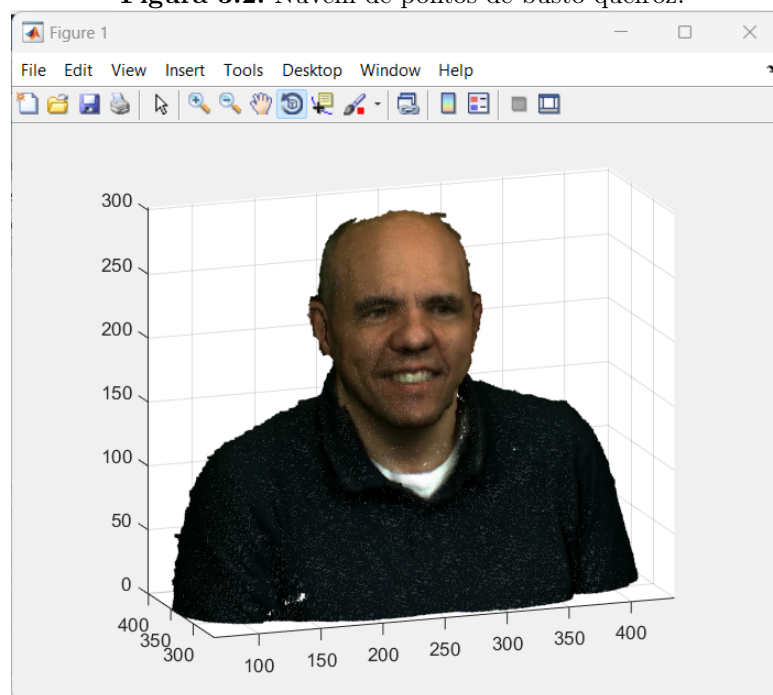
3.1 BANCO DE DADOS DE NUVENS DE PONTOS UTILIZADAS

Para a aplicação das metodologias propostas, primeiramente é necessário ter posse de um banco de dados contendo nuvens de pontos a serem testadas. O banco de dados utilizado foi o JPEG Pleno Database (JPEG...,), no site <https://plenodb.jpeg.org/>, base de dados que possui nuvens de pontos de diversos objetos e localidades, tanto estáticas quanto dinâmicas.

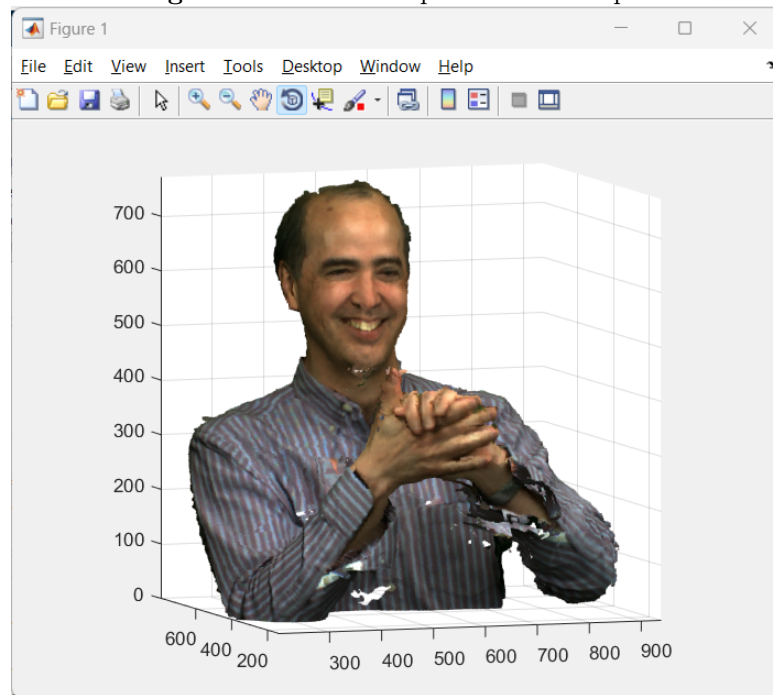
As nuvens de pontos escolhidas para os testes do projeto foram as nuvens de pontos de

corpos completo: redandblack e soldier; e de bustos: phil e queiroz. Como todas estas são nuvens de pontos dinâmicas, ou seja, nuvens de pontos que funcionam como vídeos se utilizando de diversas nuvens de pontos estáticas como quadros para diferentes instantes de tempo, foram usados alguns quadros dessas nuvens de pontos, quadros com nomes contendo o nome da nuvem de pontos dinâmica e o número do quadro em que se encontra.

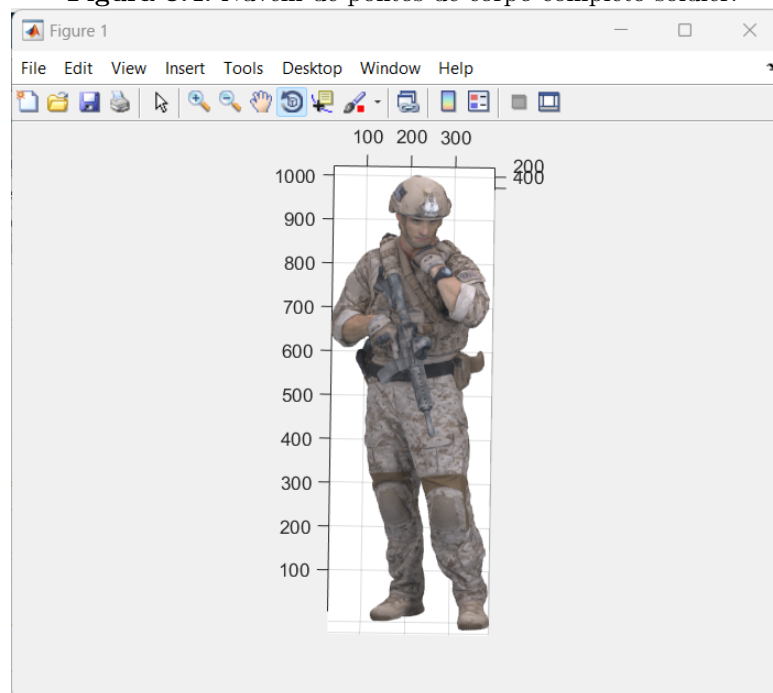
Figura 3.2. Nuvem de pontos de busto queiroz.



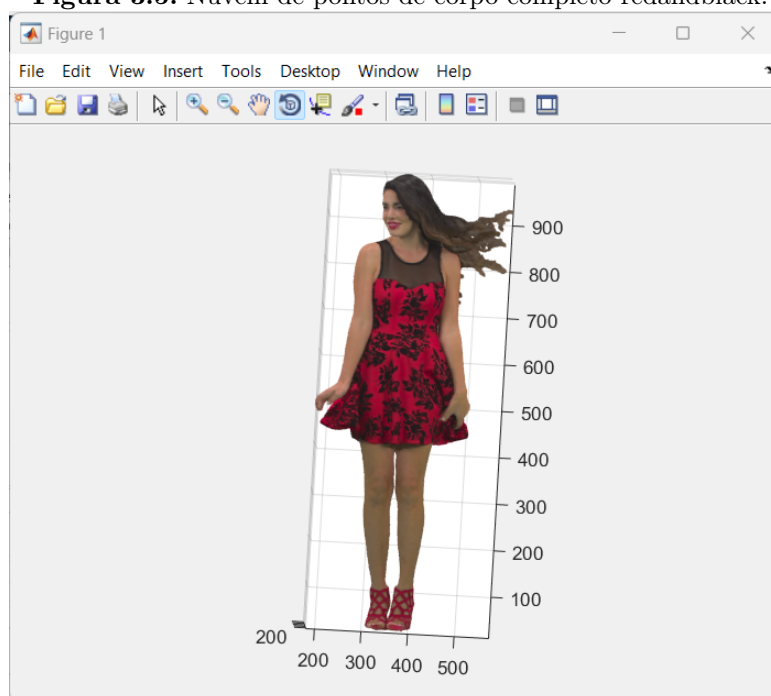
Fonte: Base de dados JPEG Pleno (JPEG...).

Figura 3.3. Nuvem de pontos de busto phil.

Fonte: Base de dados JPEG Pleno (JPEG...,).

Figura 3.4. Nuvem de pontos de corpo completo soldier.

Fonte: Base de dados JPEG Pleno (JPEG...,).

Figura 3.5. Nuvem de pontos de corpo completo redandblack.

Fonte: Base de dados JPEG Pleno (JPEG...).

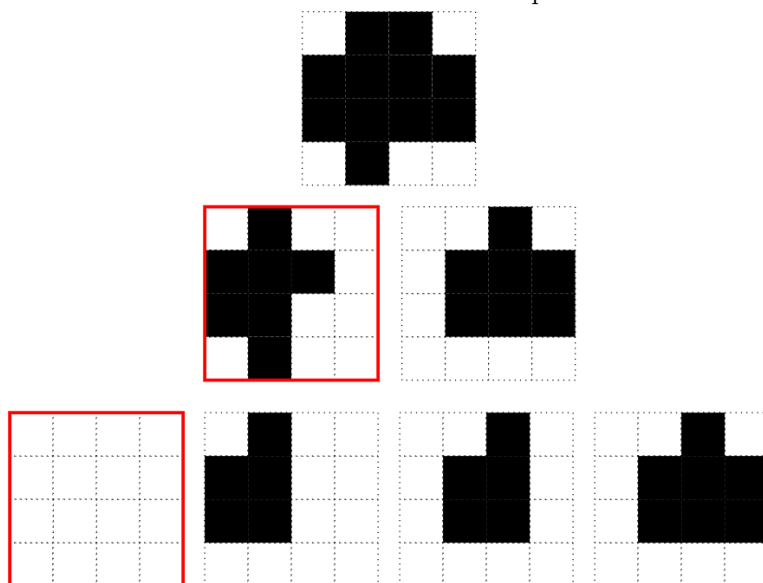
3.2 MÉTODOS DE COMPRESSÃO

Como explicado na seção 2.4.2 da fundamentação teórica, cada nó gerado pelo procedimento do Silhouette 3D (PEIXOTO, 2020) é codificado, tanto os nós das folhas da árvore quanto todos os outros nós também. Com isso, foi estudada a possibilidade da manipulação dos voxels nos níveis mais baixos da árvore de silhuetas na esperança de que fosse diminuída a quantidade total de bits da nuvem de pontos sem perdas muito significativas das informações da nuvem de pontos. Foram propostos três métodos:

3.2.1 Adiciona nada

Na codificação das silhuetas da nuvem de pontos, o método de adicionar nada simplesmente ignoraria as silhuetas folhas propostas na compressão codificando uma folha com valor zero em todas as suas posições, esperando que o método de decodificação da nuvem de pontos se valesse do nó pai dessa folha e a outra folha abaixo desse pai para expressar suas informações.

Figura 3.6. Uso do método adiciona nada no exemplo enfatizado em vermelho.

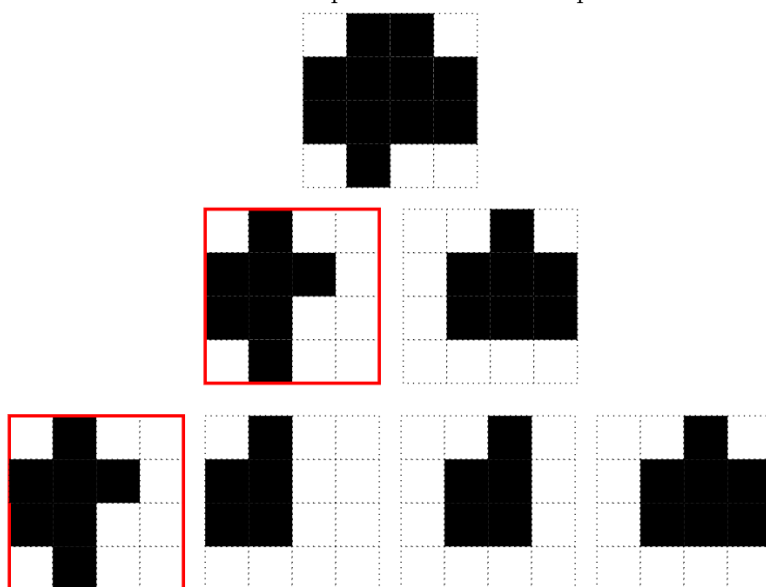


Fonte: Autoria própria.

3.2.2 Adiciona pai uma vez

O método de adicionar pai uma vez ignora a informação da folha escolhida e passa as informações de seu pai para a codificação final. Dessa forma, a informação passada fica extremamente redundante para o codificador, o que leva sua codificação a ser muito pequena.

Figura 3.7. Uso do método adiciona pai uma vez no exemplo enfatizado em vermelho.

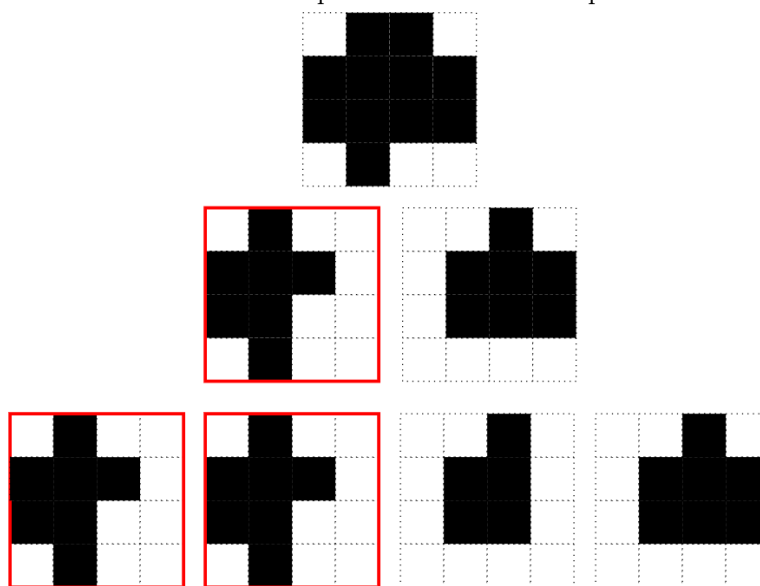


Fonte: Autoria própria.

3.2.3 Adiciona pai duas vezes

Da mesma forma que o método de adicionar o pai uma vez, este método adiciona o pai às suas duas folhas abaixo. Dessa forma, as informações das duas folhas passadas para o codificador são tão redundantes que é quase que efetivamente passado somente o equivalente ao pai, e como este já é passado para a codificação de qualquer forma, é como se nada mais fosse codificado.

Figura 3.8. Uso do método adiciona pai duas vezes no exemplo enfatizado em vermelho.



Fonte: Autoria própria.

3.3 ADAPTAÇÃO DO CÓDIGO EXISTENTE PARA O PROJETO ATUAL

O projeto do *Silhouette 3D* de nuvens de pontos por silhuetas já utilizava de codificação aritmética em contextos 3D para a codificação dos nós da árvore de silhuetas. Por esse motivo, a aplicação de compressão dos métodos propostos acima exige algumas mudanças nas funções do código já existente.

Primeiramente, no codificador, foi necessário passar uma nova combinação de *flags* para codificação das folhas da árvore de silhuetas. Um nó pai mandar uma *flag* de valor 1 para a codificação de um nó filho significava habilitar a codificação desse filho por nele haver pelo menos algum voxel a ser codificado. As possibilidades contempladas por essas *flags* eram as de

codificar ambas as folhas de um nó pai com os valores de 1 e 1, ou de codificar somente uma das folhas com os valores de 1 e 0 ou de 0 e 1. Não havia a codificação 0 e 0 no algoritmo porque isto significaria que o nó pai não contém nenhum voxel em nenhum de seus dois filhos, e por isso seu nó pai nunca o codificaria.

Para indicar o novo comportamento de codificação com perdas de adicionar o pai duas vezes, é passado no novo código duas *flags* com valores 0 e 0. Essa combinação de *flags* não indica mais que nenhuma das duas folhas devem ser codificadas, mas sim que o método com perdas escolhido para codificação será aplicado em ambas. Com essa solução, deve-se satisfazer o contexto 3D utilizado na codificação aritmética, então os voxels do nó pai são adicionados em ambos os filhos, o que para o contexto 3D do codificador aritmético efetivamente significa que ambas as folhas são similares, contendo todos os valores de voxels do nó pai.

O decodificador deve refletir esse novo comportamento, e por isso as *flags* 0 e 0 agora são recebidas pelo codificador como indicativo do novo comportamento, decodificando os nós filhos da nuvem de pontos selecionados como o pai destes.

3.4 MÉTODO DE ESCOLHA DE FOLHAS PARA COMPRESSÃO

O objetivo final desse estudo, independente do método de compressão escolhido, ainda é alcançar uma taxa de bits máxima proposta. Para isto, devem ser comprimidos vários nós da árvore de silhuetas até o ponto em que essa taxa é alcançada, mas quais nós são escolhidos para isso determinam o quanto as informações da nuvem de pontos final se desviam das informações da original. A métrica de distorção usada para a avaliação da perda de informações da nuvem de pontos foi a MSE, como explicada na seção 2.3, e cada vez que a nuvem de pontos é comprimida, a MSE entre a nuvem de pontos comprimida e a original é calculada e guardada para comparação com outros métodos aplicados. Esse trabalho aplicou esses métodos de escolha somente para as folhas da árvore, sendo então limitada a taxa máxima de bits que se pode alcançar. As escolhas de folhas foram todas feitas com o método de adição de pai duas vezes explicado na seção 3.2.3 acima.

3.4.1 Escolha estática

O procedimento de escolha de folhas para processamento com perdas visa gerar um vetor com um espaço para cada uma das folhas da árvore que tenha como valor 0 as posições das folhas que serão replicadas sem qualquer processamento adicional e valor 1 as folhas que devem ser processadas. Neste caso, como o método escolhido é o de adição de pai duas vezes, cada um dos valores do vetor representam na verdade duplas de folhas a serem comprimidas em vez de posições de folhas individuais.

A escolha estática nada mais é que a escolha manual de quais valores do vetor seriam 1 ou 0. Esse procedimento tem nenhum método lógico estabelecido, servindo somente para testar a escolha das duplas de folhas que seriam processadas ou não. Esse procedimento também não chega a uma taxa de bits escolhida pois não se sabe nem que duplas de folhas são processadas, muito menos seus tamanhos em bits, mas só suas posições.

3.4.2 Escolha por quantidade de bits

Uma forma mais simples de se estabelecer quais duplas de folhas devem ser processadas ou não é pela escolha de quais duplas têm os maiores tamanhos em bits e, começando pelas maiores duplas, selecionar cada dupla de folhas para processamento até que se encontre a taxa estabelecida.

Essa é uma forma simples de se encontrar a taxa de bits desejada, e inclusive é a forma que processa a menor quantidade de duplas de folhas na nuvem de pontos por sempre processar primeiramente as duplas de maior tamanho. Entretanto, isto não necessariamente significa que seja a forma que faça o menor impacto nas informações da nuvem de pontos.

Para efeito de exemplo, eis uma situação hipotética de uma nuvem de pontos de 1000 bits em que foram calculados a quantidade de bits e a MSE de 8 duplas de folhas comprimidas de uma nuvem de pontos nomeadas de A a H representadas na tabela 3.1. Neste exemplo, a dupla de folhas B quando comprimida pelo método de adição do pai duas vezes gera uma nuvem de pontos com 25 bits menos e MSE 4 em relação à nuvem de pontos original.

Tabela 3.1. Caso de exemplo de duplas de folhas de árvores de nuvens de pontos e suas quantidades de bits e MSE quando retirados.

Método	A	B	C	D	E	F	G	H
Nº de Bits	10	20	25	20	50	15	35	20
MSE	5	4	2	10	20	30	10	8
Nº de Bits pela MSE	2	5	12.5	2	2.5	0.5	3.5	2.5

Fonte: Autoria própria.

O método de escolha de folhas pela quantidade de bits procura neste exemplo as folhas que geram maior diferença de bits primeiro. Para uma taxa de, por exemplo, 90% do tamanho original em bits da nuvem de pontos, esse método escolheria para comprimir as duplas de folhas E, G, e C, pois estas têm as maiores quantidades de bits em ordem e somadas retiram 110 bits, alcançando a taxa máxima de 90%.

3.4.3 Escolha por quantidade de bits em função da MSE

O objetivo desse trabalho é alcançar a taxa de bits máxima, o que por meio desse método de escolha por quantidade de bits em função da MSE é um objetivo possível, mas, diferente da simples escolha das duplas com as maiores quantidades de bits, foi estipulado que essa forma de selecionar mais metodicamente as duplas de folhas faça menos impacto nas informações da nuvem de pontos. O cálculo desse impacto em cada uma das duplas de folhas é a MSE entre a nuvem de pontos original e a nuvem de pontos com somente essa dupla comprimida.

Com as informações da quantidade de bits de cada dupla de folhas e da MSE entre a nuvem de pontos original e a nuvem de pontos com cada dupla comprimida, foram primeiramente selecionadas as duplas com os maiores valores do cálculo da divisão entre a quantidade de bits e a MSE de cada dupla até que a taxa de bits fosse encontrada.

$$taxa(i) = \frac{nBits(i)}{MSE(i)} \quad (3.1)$$

Onde:

taxa é o valor de preferência para se comprimir uma dupla de folhas i ;

nBits é quantos bits se perde ao se comprimir somente a dupla de folhas i ;

MSE é o erro quadrático médio da nuvem de pontos original com a nuvem de pontos comprimida na dupla de folhas i .

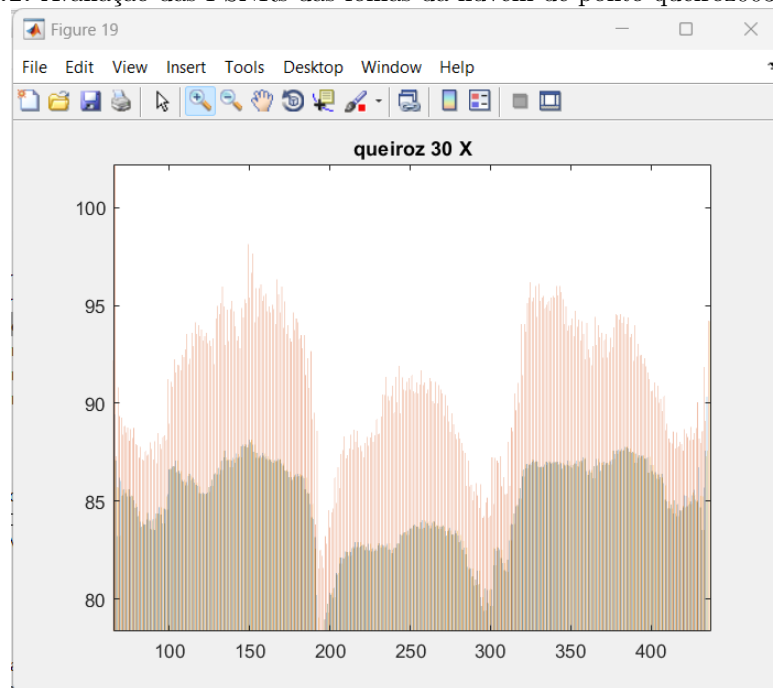
Usando o mesmo exemplo proposto no método da seção 3.4.2 de escolha de duplas de folhas por quantidades de bits, a ordem de escolha das duplas de folhas para esse método se dá crescentemente pela quantidade de bits perdida dividida pela MSE entre as duas nuvens de pontos. Por isso, diferente do último método proposto, as duplas de folhas selecionadas para compressão para alcançar a taxa máxima de 90% seriam as C, B G, e E respectivamente. A soma do impacto em bits que essas 4 duplas de folhas faz sobre a nuvem de pontos original é de 130 bits.

O próximo capítulo serve para mostrar os resultados da aplicação de cada um dos métodos propostos, possibilitando então o julgamento do melhor método para ser usado na compressão de uma nuvem de pontos.

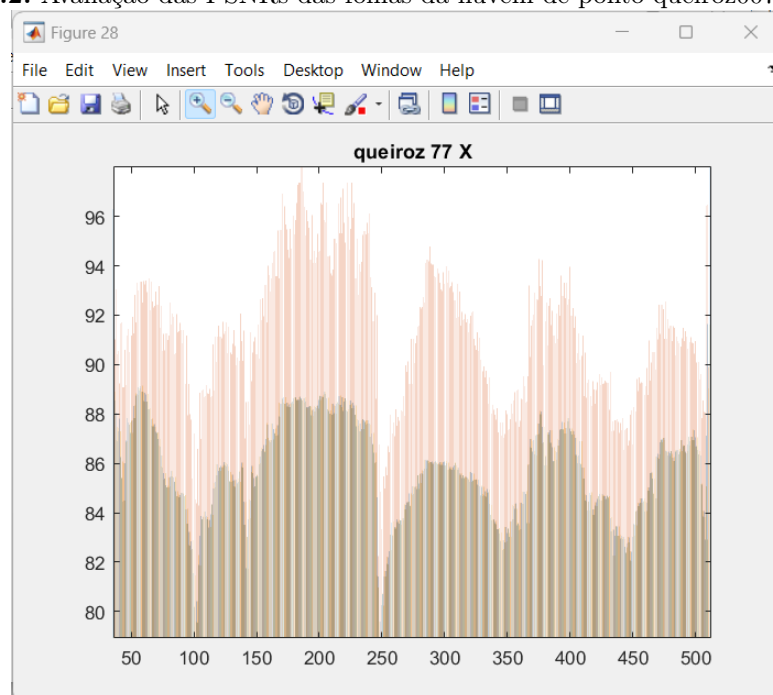
Os métodos propostos foram todos aplicados, e a eficiência de cada um destes foi documentada de forma que se possa tomar a conclusão de qual combinação de métodos é a melhor para o banco de dados selecionado. O capítulo a seguir lista os resultados dos métodos executados no trabalho.

4.1 MÉTODOS DE COMPRESSÃO

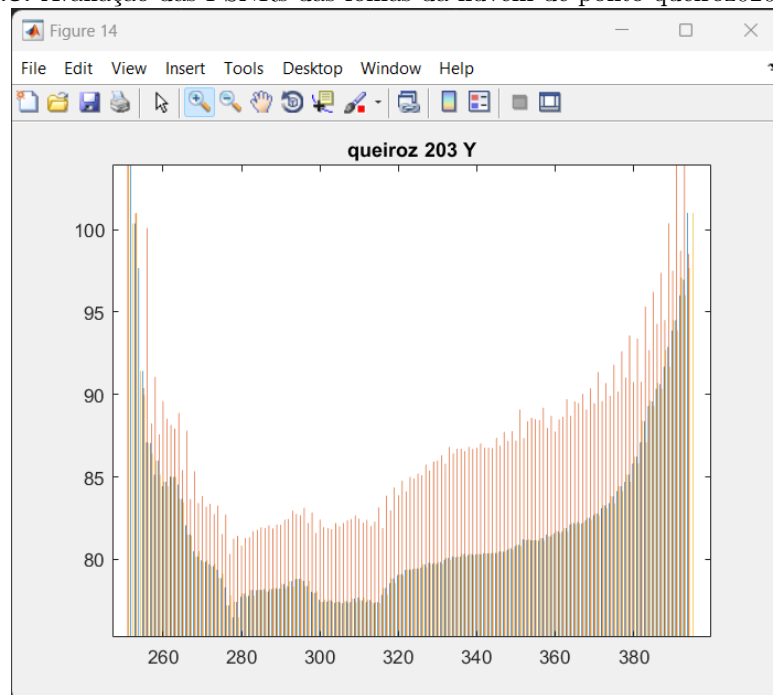
Os três métodos propostos de adicionar nada, adicionar o pai uma vez, e adicionar o pai duas vezes foram todos testados com um banco de dados de nuvens de pontos em várias orientações. Pelo MATLAB então foram criados gráficos de barra representativos da PSNR da nuvem de pontos original com a nuvem de pontos comprimida em cada uma das folhas separadamente. As barras dos resultados são representadas em trios, sendo a barra azul da esquerda de cada trio a que representa a PSNR da nuvem de pontos original com a nuvem de pontos que teve a folha correspondente comprimida pelo método de adicionar nada, a barra vermelha a PSNR obtida pelo método adicionar pai uma vez, e a barra amarela a PSNR obtida pelo método adicionar pai duas vezes. Como explicado na seção 2.3 da fundamentação teórica, quanto maior o valor da PSNR, mais próxima do sinal original é o sinal modificado, e esses gráficos de barra apresentam os métodos de compressão das folhas de uma nuvem de pontos lado a lado um do outro para facilitar a comparação. As nuvens de pontos usadas para testar os métodos de compressões foram: queiroz0000, queiroz0024, queiroz0030, queiroz0032, queiroz0066, soldier0618, soldier0679, queiroz0077, queiroz0103, queiroz0119, queiroz0171, e queiroz0203.

Figura 4.1. Avaliação das PSNRs das folhas da nuvem de ponto queiroz0030 no eixo X.

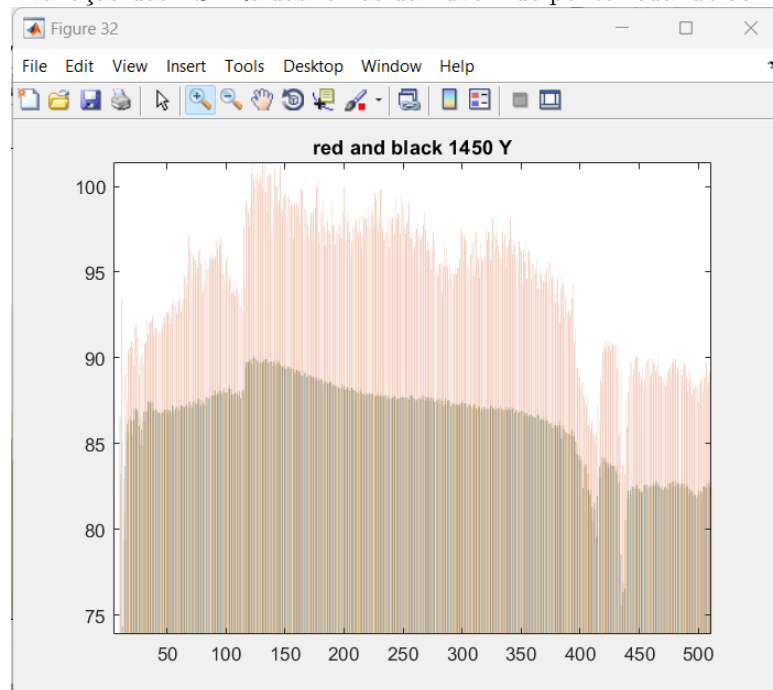
Fonte: Autoria própria.

Figura 4.2. Avaliação das PSNRs das folhas da nuvem de ponto queiroz0077 no eixo X.

Fonte: Autoria própria.

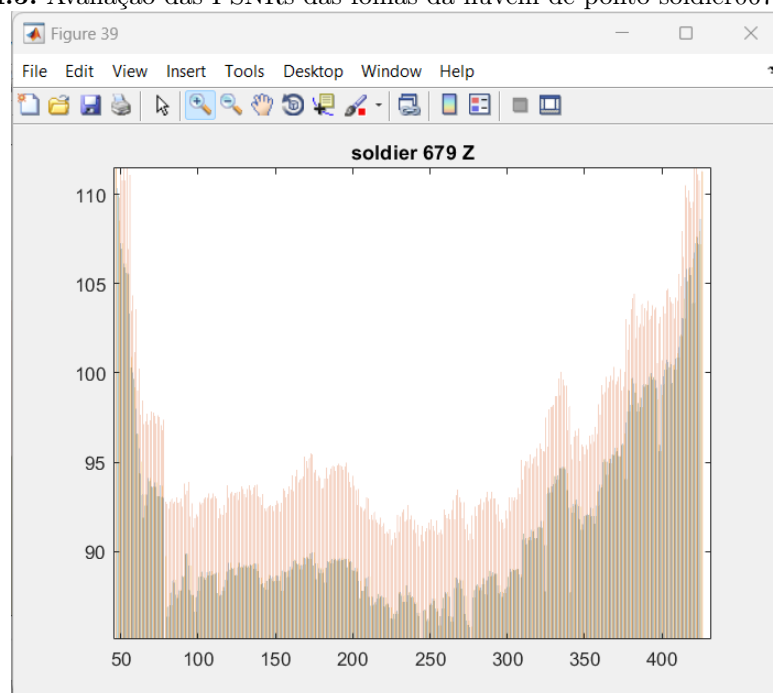
Figura 4.3. Avaliação das PSNRs das folhas da nuvem de ponto queiroz0203 no eixo Y.

Fonte: Autoria própria.

Figura 4.4. Avaliação das PSNRs das folhas da nuvem de ponto redandblack1450 no eixo Y.

Fonte: Autoria própria.

Figura 4.5. Avaliação das PSNRs das folhas da nuvem de ponto soldier0679 no eixo Z.



Fonte: Autoria própria.

Os valores calculados das PSNRs em cada uma das situações testadas apresentou uma grande tendência dos valores do método de adicionar pai uma vez terem valores maiores que os outros dois métodos sugeridos. Essa tendência fica clara nas barras vermelhas que representam esse método por estas estarem homogeneamente maiores que as outras duas barras.

Dos métodos sugeridos para compressão das folhas da nuvem de pontos, a melhor escolha avaliada para o banco de dados selecionado é o método de adicionar o pai uma vez.

4.2 MÉTODO DE ESCOLHA DE FOLHAS PARA COMPRESSÃO

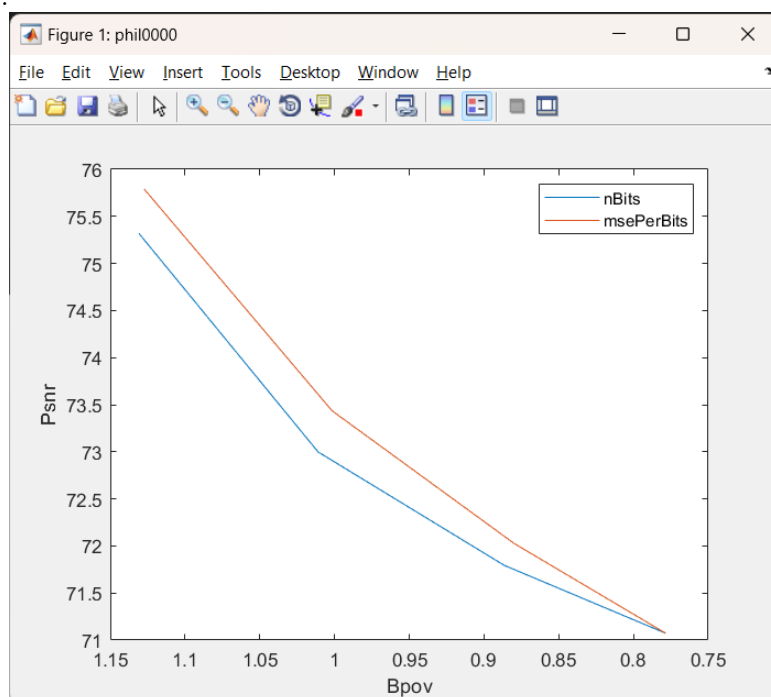
Feita a análise dos métodos de compressão, pode ser feita a compressão de várias folhas da nuvem de pontos até que uma taxa de compressão fosse alcançada. Por mais que a última seção leve à conclusão de que o método de compressão com maior PSNR na sua aplicação é o de adição do pai uma vez, o método de adição do pai duas vezes foi utilizado para os testes de escolhas de folhas para compressão da nuvem de pontos para a eficácia do estudo.

A métrica utilizada para a análise da eficiência dos métodos de compressão aplicados nas nuvens de pontos foi a MSE junto da quantidade de bits codificados pela quantidade de voxels

ocupados (bpov). Primeiramente, com o intuito de se testar o alcance de diferentes taxas de compressão de bits, foram escolhidas as quantidades máximas de 90%, 80%, 70%, 60%, e 50% da quantidade total de bits por voxel ocupado da nuvem de pontos original. Para cada uma dessas taxas estabelecidas, foram testados os dois métodos de escolha de folhas para compressão em cada uma das nuvens de pontos escolhidas para se fazer a comparação das MSEs entre cada um dos resultados e suas nuvens de pontos originais. Com posse dessa informação, pode-se afirmar qual método é mais eficiente em termos de número de bits comprimidos por perda de informações da nuvem de pontos original. As nuvens de pontos utilizadas para esses testes foram: phil0000, phil0024, queiroz0000, queiroz0032, redandblack1450, soldier0618, e soldier0679.

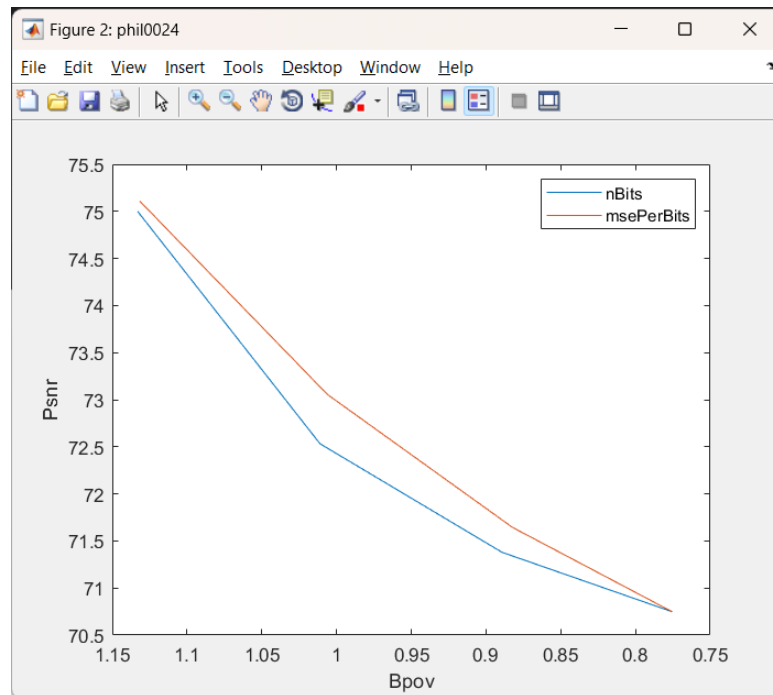
Para comparação da eficiência do método de retirar duplas de folhas por quantidades de bits com a do método de retirar duplas de folhas por quantidade de bits em função da MSE resultante, foram feitos gráficos da MSE das nuvens de pontos resultantes em função das bpovs atingidas nas taxas de 90%, 80%, 70%, 60%, e 50%. Todos os resultados atingidos apresentam valores de MSEs menores para o uso do método que leva a MSE em consideração.

Figura 4.6. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em phil0000.



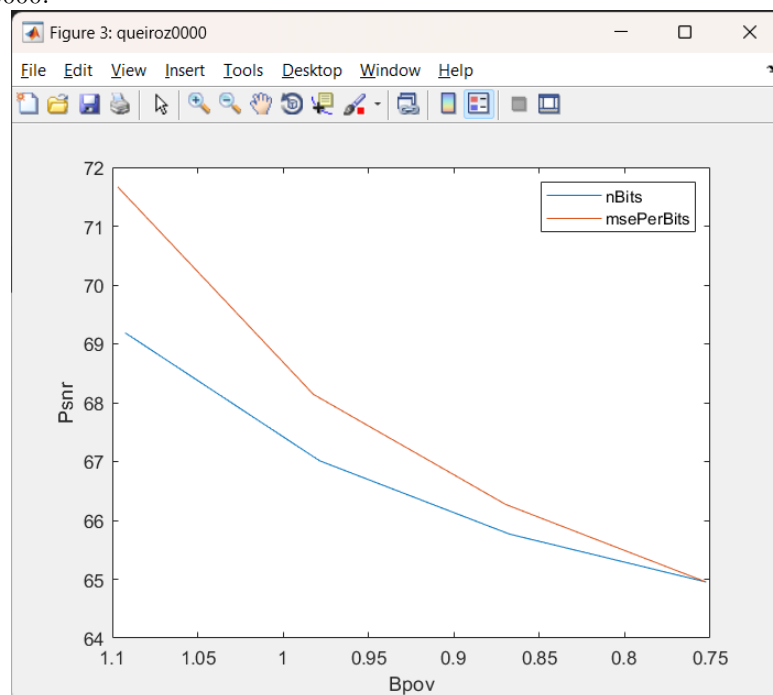
Fonte: Autoria própria.

Figura 4.7. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em phil0024.



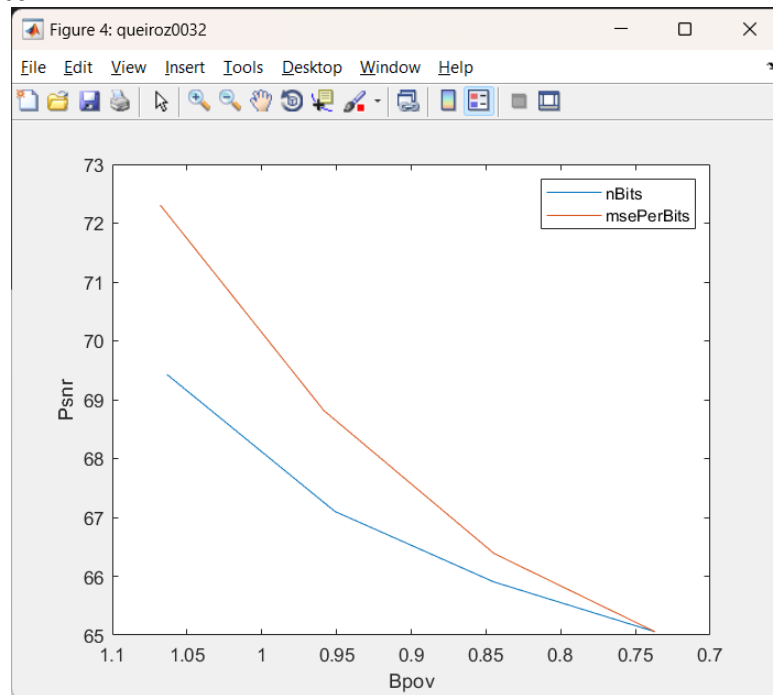
Fonte: Autoria própria.

Figura 4.8. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em queiroz0000.



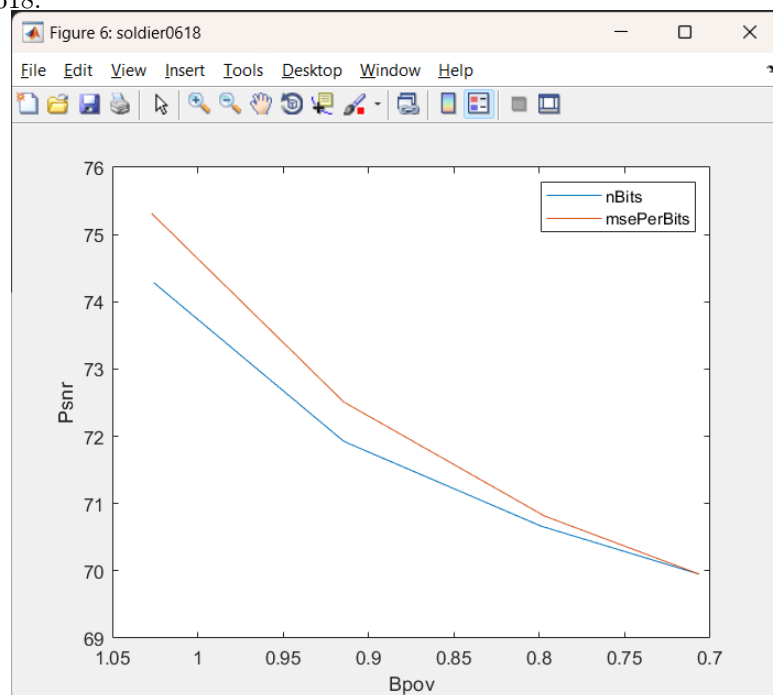
Fonte: Autoria própria.

Figura 4.9. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em queiroz0032.



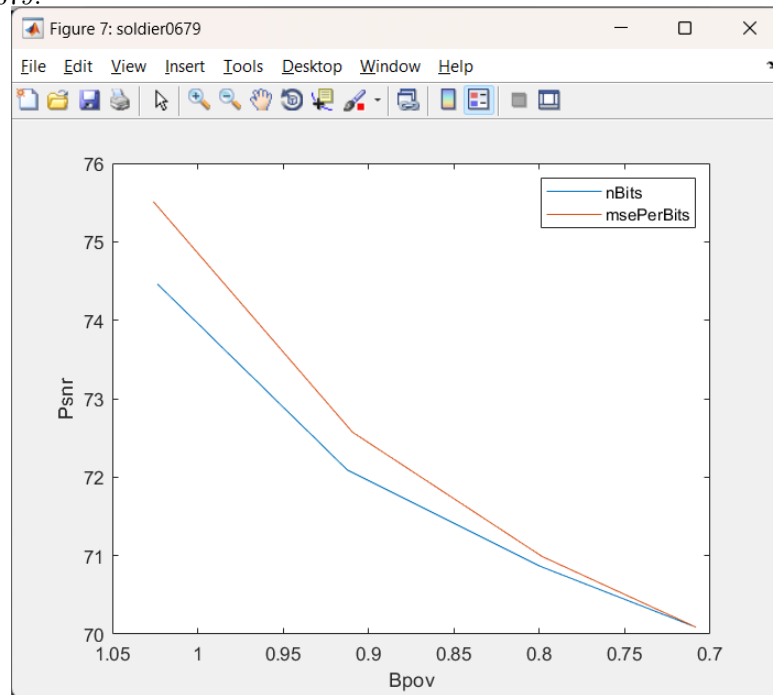
Fonte: Autoria própria.

Figura 4.10. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em soldier0618.



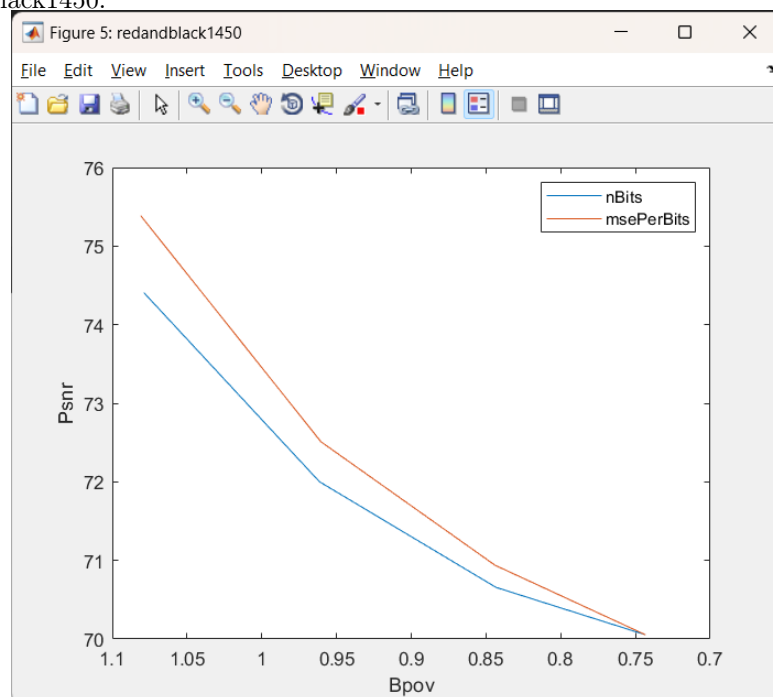
Fonte: Autoria própria.

Figura 4.11. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em soldier0679.



Fonte: Autoria própria.

Figura 4.12. Resultados da aplicação dos métodos de escolha de por quantidade de bits e de escolha em função da MSE em redandblack1450.



Fonte: Autoria própria.

Os dados adquiridos no trabalho são suficientes para se tirar conclusões relevantes para a

realização da metodologia proposta.

CONCLUSÃO

Os métodos propostos todos visavam o desenvolvimento de uma nova forma de se codificar nuvens de pontos por silhuetas, codificação que deve encontrar uma taxa máxima de bits por meio de compressão com perdas. Os métodos propostos para tal desenvolvimento foram todos aplicados, e a eficiência de cada um destes foi documentada de forma que se possa tomar a conclusão de qual combinação de métodos é a melhor para o banco de dados selecionado.

Dentre os métodos propostos, a compressão de adição de pai duas vezes em folhas selecionadas por suas quantidades de bits em função de suas MSEs foi a junção de métodos testada. A forma de compressão de folhas escolhida, por mais que não tenha sido a forma com melhor desempenho testada na metodologia, ainda assim alcançou resultados satisfatórios para a necessidade de compressão de nuvens de pontos até uma taxa máxima de bits.

A conclusão que pode ser tirada desse trabalho é que o que foi proposto tem valor para o ramo de compressão de nuvens de pontos. A utilização do método Silhouette3D se mostra cada vez mais favorável em suas aplicações sobre seu método anterior, o Octree. A abertura do ramo de compressões com perdas de nuvens de pontos expande seus usos ainda mais sobre esse método anterior.

5.1 CÓDIGO DESENVOLVIDO

O código completo do trabalho, incluindo os trabalhos paralelo sobre o método Silhouette3D, se encontram no Github no repositório abaixo. Esse trabalho especificamente se encontra na *branch* especificada na URL.

<https://github.com/bananahell/GeometryCoder/tree/tcc-pedro>

5.2 TRABALHO FUTURO

O trabalho feito alcançou o objetivo de comprimir uma nuvem de pontos até uma taxa máxima de bits, mas somente até uma certa taxa, porque os únicos nós da árvore de silhuetas que são comprimidos são as folhas da árvore, e isso mostrou alcançar só taxas de mais ou menos 60% do tamanho original. Por isso, uma melhoria no método proposto seria a de possibilitar a compressão de qualquer nó da árvore de silhuetas, checando primeiramente os melhores nós a serem comprimidos, tendo em mente a possibilidade da compressão de nós filhos ser melhor que a compressão do nó pai ou vice-versa.

Mais uma proposta de melhoria para o projeto seria a do desenvolvimento de melhores técnicas de compressão, tanto no método de compressão desenvolvido quanto na escolha dos nós a serem comprimidos. Os métodos de compressão propostos foram decididos a partir do conhecimento sobre a codificação Silhouette3D, mas mais pesquisa sobre a metodologia pode se provar benéfica, e a escolha dos nós a serem comprimidos também foi desenvolvida de forma imperativa, podendo ter melhorias em sua escolha, como algum coeficiente multiplicativo do cálculo de decisão na equação 3.1, ou até mesmo um método diferente.

REFERÊNCIAS BIBLIOGRÁFICAS

AUTODESK. *About point cloud*. 2018. Citado na página 4.

DASKALAKI, A.; LOBOS, C.; PAYAN, Y.; HITSCHFELD, N. Techniques for the generation of 3d finite element meshes of human organs. p. 126–158, 2010. Citado na página 8.

JPEG Pleno Database. <<https://plenodb.jpeg.org/>>. Citado 4 vezes nas páginas 15, 16, 17, and 18.

MACCHIAVELLO, B. *Notas de aula sobre Segmentação de Imagens 3*. [S.l.]: Departamento de Ciência da Computação da Universidade de Brasília, 2023. Citado na página 8.

MEKURIA, R.; LI, Z.; TULVAN, C.; CHOU, P. Evaluation criteria for pcc (point cloud compression). 2016. Citado na página 7.

OCEANIC, N.; ADMINISTRATION, A. *What is LIDAR*. 2021. Citado na página 3.

PEIXOTO, E. Intra-frame compression of point cloud geometry using dyadic decomposition. *IEEE SIGNAL PROCESSING LETTERS, VOL. 27*, 2020. Citado 2 vezes nas páginas 9 and 18.

PLUMRIDGE, J. *Compression in Digital Photography*. 2020. <<https://www.lifewire.com/the-effect-of-compression-on-photographs-493726>>. Citado na página 6.

QUEIROZ, R. L. d.; CHOU, P. A. *Compression of 3d point clouds using a region-adaptive hierarchical transform*. 2016. Citado 2 vezes nas páginas 8 and 9.

SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 1948. Citado na página 5.