



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

LUIZ FELIPE FEITOSA LEITE

EFEITOS DE SUBAMOSTRAGEM EM NUVENS DE PONTOS DE FACES 3D EM
ALGORITMOS PARA CORREÇÃO DE ORIENTAÇÃO, EXTRAÇÃO DE
ATRIBUTOS E CLASSIFICAÇÃO

FORTALEZA

2017

LUIZ FELIPE FEITOSA LEITE

EFEITOS DE SUBAMOSTRAGEM EM NUVENS DE PONTOS DE FACES 3D EM
ALGORITMOS PARA CORREÇÃO DE ORIENTAÇÃO, EXTRAÇÃO DE ATRIBUTOS E
CLASSIFICAÇÃO

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Computação do Centro de Tecnologia da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de bacharel em
Engenharia de Computação.

Orientador: Prof. Dr. José Marques Soares

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

L554e Leite, Luiz Felipe Feitosa.

Efeitos de Subamostragem em Nuvens de Pontos de Faces 3D em Algoritmos para Correção de Orientação, Extração de Atributos e Classificação / Luiz Felipe Feitosa Leite. – 2017.
41 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia de Computação, Fortaleza, 2017.

Orientação: Prof. Dr. José Marques Soares.

1. Subamostragem. 2. Pré-processamento. 3. Nuvem de pontos. 4. Reconhecimento facial. 5. 3D. I.
Título.

CDD 621.39

LUIZ FELIPE FEITOSA LEITE

EFEITOS DE SUBAMOSTRAGEM EM NUVENS DE PONTOS DE FACES 3D EM
ALGORITMOS PARA CORREÇÃO DE ORIENTAÇÃO, EXTRAÇÃO DE ATRIBUTOS E
CLASSIFICAÇÃO

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Computação do Centro de Tecnologia da
Universidade Federal do Ceará, como requisito
parcial à obtenção do grau de bacharel em
Engenharia de Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. José Marques Soares (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. George André Pereira Thé
Universidade Federal do Ceará (UFC)

Prof. Dr. Giovanni Cordeiro Barroso
Universidade Federal do Ceará (UFC)

Aos meus irmãos e ao Prof. Me. Alexandre
Moreira de Moraes (*in memorian*), que serviram
de motivação por toda a caminhada.

AGRADECIMENTOS

Ao Prof. Dr. José Marques Soares, pela orientação, apoio e confiança.

Aos amigos e companheiros de pesquisa Artur Rodrigues Rocha Neto e Gilderlane Ribeiro Alexandre por tanto me ensinarem e por compartilharem seus conhecimentos que contribuíram para o meu trabalho.

Ao Prof. Me. Robson da Silva Siqueira e ao Prof. Dr. George André Pereira Thé, pelo apoio na elaboração deste trabalho.

Ao meu irmão João Paulo Feitosa Rodrigues, por tanto me ensinar, apoiar e motivar.

A toda minha família, em especial meus pais, pelo apoio.

A amiga Marilene de Oliveira Lima, pelas incontáveis ajudas, pela paciência e pela compreensão durante todos os momentos desta jornada.

Aos amigos Dennys da Silva Simião, Gabriel Diogo de Sampaio e Omar Abdalalim Alrai, por servirem de inspiração e me auxiliarem.

A todos colegas, corpo docente, coordenação e diretoria do Curso de Engenharia de Computação da Universidade Federal do Ceará e a todos amigos que contribuíram direta ou indiretamente com minha formação.

"A dúvida é o princípio da sabedoria."

(Aristóteles)

RESUMO

A difusão de sensores de imagens de profundidade, hoje presentes em aparelhos celulares e dispositivos similares, tem popularizado o uso de imagens tridimensionais para reconhecimento facial. Apesar da insensibilidade a condições como luminosidade, que afetam imagens de duas dimensões, existem variáveis, como distância e ângulo de captura, durante a aquisição das imagens, que geram amostras com densidades de pontos por área distintas. Além do problema de possível divergência de densidade entre nuvens que representam um mesmo objeto ou cenário, as estruturas tridimensionais podem conter dados excessivos, tornando o processamento computacional custoso em memória e tempo. Este trabalho apresenta uma análise sobre os efeitos da diminuição de pontos de nuvens 3D nos métodos RANSAC, Jacobi, *Normal Estimation* e ICP de correção de posicionamento de face. Além disso, são avaliados momentos invariantes tridimensionais utilizados como atributos para identificação de indivíduos. As imagens utilizadas pertencem à base de dados Bosphorus. Os resultados dos testes mostram a sensibilidade ao número de pontos dos métodos de correção de posicionamento de face em tempo de execução e precisão. Verifica-se também a possibilidade de obterem-se melhores resultados em processos de identificação de face com a variação do número de pontos.

Palavras-chave: Subamostragem. Pré-processamento. Nuvem de pontos. Reconhecimento facial. 3D.

ABSTRACT

The dissemination of depth image sensors, which is now present in smartphones and similar devices, has popularized the use of three-dimensional images for facial recognition. Despite the insensitivity to conditions such as luminosity, which affect two-dimensional images, there are variables, such as distance and capture angle during image acquisition, which generate samples with different point densities per area. Three-dimensional structures could contain an excessive amount of data in addition to possibly presenting problems with density divergence between clouds representing the same object or scenario, which makes computational processing costly in regard to time and memory. This work presents an analysis of the effects of 3D point cloud reduction by using the RANSAC, Jacobi, Normal Estimation and ICP methods of face position correction and three-dimensional invariants moments, used as attributes for identification of subjects in the Bosphorus database. The tests presented show the sensitivity to the number of points of correction methods of face positioning in execution time and accuracy in their results, as well as the possibility of obtaining better results in face identification processes, by varying the number of points.

Keywords: Subsampling. pre-processing. Point cloud. Facial recognition. 3D.

LISTA DE ILUSTRAÇÕES

Figura 1 – Amostras de faces expressivas da base de dados Bosphorus	16
Figura 2 – <i>Voxel grid</i> em uma nuvem de pontos 3D	17
Figura 3 – Registro de um par de nuvens de pontos 3D	17
Figura 4 – Configurações de subamostragem	25
Figura 5 – <i>Pipeline</i> para os métodos <i>RANdom SAmple Consensus</i> (RANSAC), Jacobi e <i>Normal Estimation</i>	26
Figura 6 – Cortes a partir da ponta do nariz	27
Figura 7 – <i>Pipeline</i> para o métodos <i>Iterative Closest Point</i> (ICP)	27
Figura 8 – Boxplot de densidade de pontos por área	30
Figura 9 – Erro médio absoluto de correção estimada e média de tempo de execução para RANSAC	32
Figura 10 – Erro médio absoluto de correção estimada e média de tempo de execução para Jacobi	32
Figura 11 – Erro médio absoluto de correção estimada e média de tempo de execução para <i>Normal Estimation</i>	33
Figura 12 – Erro médio absoluto de correção estimada e média de tempo de execução para ICP	33

LISTA DE TABELAS

Tabela 1 – <i>Kernels</i> para SVM	22
Tabela 2 – Configurações de subamostragem	24
Tabela 3 – Classificações	28
Tabela 4 – Classificações: C_1 - Neutros	35
Tabela 5 – Classificações: C_2 - Neutros	35
Tabela 6 – Classificações: C_2 - Não Neutros	36
Tabela 7 – Incidência de melhores e piores resultados de classificação para subamostragens	37

LISTA DE ABREVIATURAS E SIGLAS

ALS	<i>Airbone Laser Scanning</i>
AU	<i>Action Units</i>
CAU	<i>Combination Action Unit</i>
I3D	<i>Invariants 3D</i>
ICP	<i>Iterative Closest Point</i>
KNN	<i>K-Nearest Neighbors</i>
KNNE	<i>K-Nearest Neighbors Euclidian</i>
KNNM	<i>K-Nearest Neighbors Manhattan</i>
LFAU	<i>Lower Face Action Unit</i>
PCA	<i>Principal Component Analysis</i>
PCL	<i>Point Cloud Library</i>
RANSAC	<i>RANdom SAmple Consensus</i>
SVM	<i>Support Vector Machine</i>
ToF	<i>Time-of-Flight</i>
UFAU	<i>Upper Face Action Unit</i>

LISTA DE SÍMBOLOS

x	Coordenada no eixo das abscissas
y	Coordenada no eixo das ordenadas
X	Eixo das abscissas
Y	Eixo das ordenadas
Z	Eixo das cotas
N	Número de iterações
p, u, v	Probabilidade
R, S	Nuvem de pontos
A	Matriz simétrica
U	Matriz de transformação
D	Matriz diagonal
a_{ij}	Elemento ij de matriz A
d_e	Distância Euclidiana
d_m	Distância Manhattan
d	Densidade de pontos por área
P	Número de pontos
A_n	Área de nuvem de pontos
A_t	Área de triângulo
N_t	Número de triângulos
C_i	Conjunto de nuvens

SUMÁRIO

1	INTRODUÇÃO	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Imagens Tridimensionais	15
2.2	Bosphorus	15
2.3	Voxel Grid	16
2.4	RANSAC	17
2.5	ICP	18
2.6	Jacobi	19
2.7	Momentos Invariantes	20
2.8	Classificadores	20
2.8.1	<i>Classificador KNN</i>	<i>21</i>
2.8.2	<i>Classificador SVM</i>	<i>21</i>
2.9	Considerações relevantes	22
3	METODOLOGIA	24
3.1	Hardware e Software	24
3.2	Subamostragem	24
3.3	Análises	25
3.3.1	<i>Densidade</i>	<i>25</i>
3.3.2	<i>Correção de face</i>	<i>26</i>
3.3.3	<i>Extração de Atributos</i>	<i>28</i>
4	RESULTADOS	30
4.1	Densidade	30
4.2	Correção de face	31
4.3	Extração de Atributos	34
5	CONCLUSÕES	38
	REFERÊNCIAS	40

1 INTRODUÇÃO

O uso de imagens de profundidade ganhou popularidade nos últimos anos. A tecnologia tem sido usada para gerar dados para reconhecimento facial e está presente em diversos sensores, inclusive embarcados em aparelhos celulares e dispositivos similares.

Existem condições variáveis no momento da aquisição de imagens de profundidade que geram amostras com densidades de pontos por área distintas. Sensores diferentes, ou um mesmo sensor em momentos diferentes, distância e ângulo de captura, podem gerar nuvens de pontos que representam um mesmo objeto, ou cenário, com quantidade de pontos e/ou áreas de representação divergentes. A falta de congruência entre as nuvens pode aumentar a distância entre atributos para classificação — como momentos — de objetos e cenários, dificultando esse processo.

O número de pontos é proporcional ao custo de processamento — memória e tempo. Considerando reconhecimento facial como escopo de estudo, são utilizados os métodos para corrigir e detectar posição da face *Normal Estimation*, algoritmo de Jacobi, RANSAC e ICP, que precisam varrer e coletar dados de todos os pontos na nuvem, e podem ser demorados se aplicados sobre nuvens com muitos pontos, tornando inviável, por exemplo, aplicações de identificação de indivíduos em tempo real. A redução de dados da nuvem de pontos, se feita corretamente - mantendo as informações essenciais - pode facilitar no processamento futuro dos dados (BOEHLER *et al.*, 2002).

O presente trabalho tem como objetivo analisar efeitos da manipulação de densidade de pontos por área, através de redução no número de pontos. Atua-se na aproximação de densidades distintas, devido a condições de aquisição, e na mitigação dos custos de processamento em imagens de profundidade, a fim de verificar se esse pré-processamento facilita no procedimento correção de posicionamento e classificação de imagens de face em três dimensões.

2 FUNDAMENTAÇÃO TEÓRICA

Com sensores 3D de baixo custo, a percepção através de estruturas de dados tridimensionais vem ganhando maior importância (RUSU; COUSINS, 2011). Uma das formas de representação de imagens 3D é o uso de nuvens de pontos em três dimensões. O presente trabalho foi embasado nos conceitos de subamostragem, técnicas de correção do posicionamento, momentos e classificadores, todos aplicados sobre imagens tridimensionais em formatos de nuvens de pontos, com suporte da biblioteca *Point Cloud Library* (PCL). O objetivo deste capítulo é apresentar tais conceitos.

2.1 Imagens Tridimensionais

Uma imagem bidimensional, ou imagem 2D, é definida por Gonzalez e Woods (2000) como uma função $f(x,y)$, em que x e y são coordenadas espaciais e f é a amplitude ou intensidade da imagem naquela coordenada. Quando x , y e $f(x,y)$ são valores finitos e discretos, nos referimos à imagem como digital. Imagens digitais 2D são, portanto, projeções no espaço bidimensional de objetos tridimensionais. A notação usada para imagens digitais 2D é I_{xy} .

Uma outra forma de descrever objetos é por meio de nuvens de pontos. Nuvens de pontos são uma estrutura de dados utilizada para representar pontos multidimensionais - mais comumente tridimensionais. Cada ponto da nuvem de pontos 3D corresponde ao conjunto de coordenadas geométricas X , Y e Z de uma certa amostra de superfície (RUSU; COUSINS, 2011). Uma nuvem de pontos representa, portanto, o que chamamos de imagem tridimensional ou imagem 3D e pode ser capturada por meio de dispositivos como câmeras estéreo, sensores 3D e câmeras *Time-of-Flight* (ToF).

A crescente disponibilidade de dispositivos de aquisição 3D e as vantagens das imagens tridimensionais em relação às imagens 2D (independência das condições de iluminação e de *viewpoint*) justificam a popularização de técnicas de reconhecimento de indivíduos e de expressões faciais (SAVRAN *et al.*, 2008).

2.2 Bosphorus

Bosphorus (SAVRAN *et al.*, 2008) é uma base de dados formada por imagens de faces tridimensionais, em formato de nuvens de pontos, com 105 indivíduos, entre 31 e 54 amostras de cada indivíduo, dispondo um total de 4666 nuvens.

As aquisições de imagens são feitas utilizando o Inspeck Mega Capturor II 3D, com uma resolução nas dimensões X , Y e Z (profundidade) de 0.3mm, 0.3mm e 0.4mm respectivamente. Os indivíduos são posicionados a uma distância de 1.5 metros do sensor. A Figura 1 ilustra amostras presentes na base de dados.

Figura 1 – Amostras de faces expressivas da base de dados Bosphorus



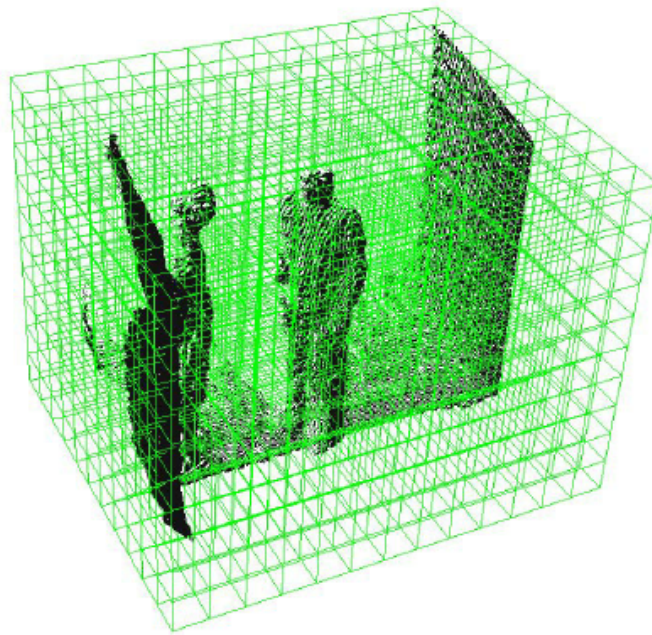
Fonte: Savran *et al.* (2008)

A base possui expressões de dois tipos: emoções e *Action Units* (AU). As emoções podem ser divididas em seis: *happiness*, *surprise*, *fear*, *sadness*, *anger* e *disgust*. As expressões podem ser *Upper Face Action Unit* (UFAU), *Lower Face Action Unit* (LFAU) e *Combination Action Unit* (CAU), que representam respectivamente expressões na região superior da face, expressões na região inferior da face e expressões nas regiões superior e inferior da face combinadas. Existem também na base de dados nuvens tridimensionais com oclusões da face por mão, cabelo e óculos, e faces com até 14 posições divergentes, com rotações em um ou mais eixos.

2.3 Voxel Grid

Como artifício para manipulação de densidade de pontos por área, foi utilizada a redução do número de pontos. O algoritmo *voxel grid*, funciona como um filtro, criando no espaço tridimensional uma grade de blocos (*voxels*), como apresenta a Figura 2 em que o *voxel grid* é representado em verde e uma nuvem de pontos com pontos pretos, e substitui os pontos da nuvem pelos centróides destes. A posição dos *voxels* é definida pela nuvem onde o algoritmo é aplicado, sendo o tamanho dos lados um parâmetro de entrada (ZYGMENT, 2013).

Figura 2 – *Voxel grid* em uma nuvem de pontos 3D

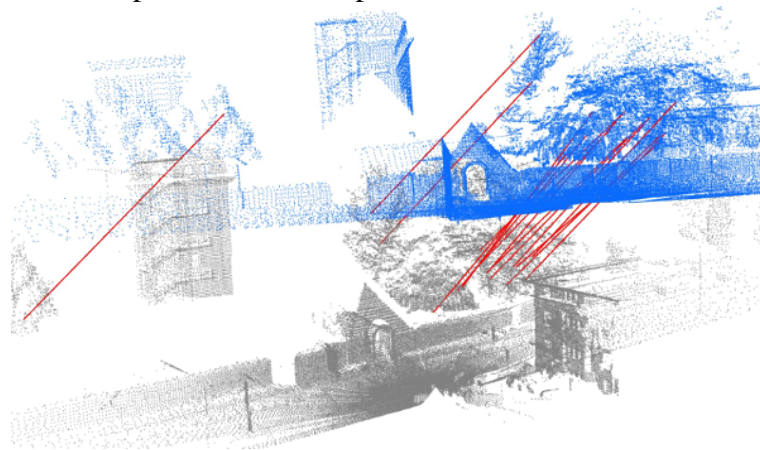


Fonte: Wang *et al.* (2012)

2.4 RANSAC

O algoritmo RANSAC foi apresentado por Fischler e Bolles (1981) com o objetivo de identificar um conjunto de pontos-chave correspondentes para um par de imagens — que podem representar um mesmo objeto — a serem unidas, como ilustrado na Figura 3. Este algoritmo gera uma matriz homográfica que mapeia todos os pontos e retas da primeira imagem na segunda, contendo parâmetros de rotação, translação e escala, necessários para juntar duas imagens.

Figura 3 – Registro de um par de nuvens de pontos 3D



Fonte: Open Perception Foundation (2017)

Utilizando, como uma das imagens, um modelo conhecido, o RANSAC tem sido usado para apoiar o reconhecimento de objetos. O algoritmo pode ser dividido em quatro etapas, como apresenta (DERPANIS, 2010):

1. Seleção aleatória dos pontos (mínimos necessários) para determinar os parâmetros do modelo.
2. Resolução dos parâmetros do modelo.
3. Determinação de quantos pontos do conjunto completo se encaixam com uma tolerância predefinida ε .
4. Se a fração do número de inliers sobre o número total de pontos no conjunto exceder um limite predefinido τ , são estimados novamente os parâmetros do modelo usando todos os inliers identificados e terminado. Caso contrário, são repetidos os passos 1 a 4 (máximo de N vezes).

Segundo Derpanis (2010), deve ser escolhido um número de iterações, N , que seja suficientemente grande para garantir, com uma probabilidade p (normalmente definida como 0.99), que em pelo menos um dos conjuntos de amostras aleatórias não existam pontos incorretos (*outliers*). Assumindo u a probabilidade de um ponto qualquer selecionado estar correto (*inlier*), $v = 1 - u$ ser a probabilidade de ser um *outlier* e m o número mínimo de pontos para estimar os parâmetros do modelo, então N é definido por:

$$N = \frac{\log(1 - p)}{\log(1 - (1 - v)^m)}. \quad (2.1)$$

2.5 ICP

O algoritmo ICP foi apresentado inicialmente por Besl *et al.* (1992), Chen e Medioni (1991), com o objetivo de, dadas duas nuvens R e S (referência), estimar parâmetros de transformação (rotação e translação) entre os pares de nuvens, tal que a nuvem R consiga alcançar a nuvem de referência, S , alinhando seus pontos correspondentes. De maneira semelhante ao RANSAC, ICP busca realizar um registro entre um par de nuvens, como o ilustrado na Figura 3.

O algoritmo é amplamente utilizado para alinhamento geométrico de modelos tridimensionais, quando uma estimativa inicial relativa da pose for conhecida (RUSINKIEWICZ; LEVOY, 2001). Começando com uma estimativa relativa de transformação rígida, de maneira iterativa, o algoritmo refina suas transformações, gerando repetidamente pares de pontos corres-

pondentes e minimizando erros. O algoritmo possui seis estágios que podem ser modificados e gerar variantes (RUSINKIEWICZ; LEVOY, 2001):

1. Seleção de conjunto de pontos em uma ou duas imagens.
2. Correspondência dos pontos selecionados com amostras da outra imagem.
3. Ponderação de pares correspondentes apropriados.
4. Rejeição de pares baseada em cada par individualmente ou considerando o conjunto completo de pares.
5. Minimização da métrica de erro.

2.6 Jacobi

Método de Jacobi é um algoritmo iterativo utilizado para solucionar sistemas de equações lineares e serve para determinar auto-valores e auto-vetores (JÚDICE; PATRÍCIO, 1996; FRANCO, 2006) de uma dada matriz simétrica A , aplicando uma série de transformações do tipo:

$$A_{k+1} = U_k^{-1} A_k U_k. \quad (2.2)$$

Na Equação 2.2, k é um inteiro positivo, $A_1 = A$ e U_k é a matriz de rotação. As matrizes A_1, A_2, \dots convergem para uma matriz diagonal. Se após um número de $m + 1$ iterações do método for encontrada uma matriz $A_{m+1} \approx D$, onde D é a matriz diagonal, então:

$$D \approx A_{m+1} = U_m^{-1} \dots U_2^{-1} U_1^{-1} A_1 U_1 U_2 \dots U_m. \quad (2.3)$$

A equação 2.3 traz então A_{m+1} como uma aproximação para os auto-valores da matriz A e as colunas $V = U_1 U_2 \dots U_m$ são aproximações para os auto-vetores, como mostra Franco (2006). A escolha da matriz de transformação U_k , demonstrada por Franco (2006), é tal que faça o elemento a_{pq} , da matriz A_k , zerar. O elemento a_{pq} é o elemento não diagonal com maior valor absoluto:

$$a_{pq} = \max_{i \neq j} (a_{ij}). \quad (2.4)$$

2.7 Momentos Invariantes

Momentos invariantes estão entre as técnicas de reconhecimento de padrões mais utilizadas para classificação de entidades. Chamam-se invariantes pois uma de suas propriedades principais é a invariância às transformações básicas de rotação, translação e escala. Momentos são valores escalares e podem ser vistos como projeções de uma função numa base polinomial (ponto de vista matemático) ou como o acúmulo de energia em uma certa região do espaço (ponto de vista físico).

2.8 Classificadores

A classificação é a etapa final de todo processo de reconhecimento de padrões. O problema de classificação é responsável por atribuir uma categoria (ou classe) a um elemento de um conjunto desconhecido (conjunto de teste) em função dos padrões modelados com base em um conjunto montado previamente (conjunto de treinamento). Por isso, a classificação se enquadra, no contexto de aprendizado de máquina, como uma técnica de aprendizagem supervisionada. Outras técnicas possíveis seriam a aprendizagem não-supervisionada e a aprendizagem semi-supervisionada. O contexto do presente trabalho aborda apenas a aprendizagem supervisionada. A aprendizagem supervisionada é dividida em duas fases: treinamento e teste. O elemento principal da etapa de classificação é o classificador.

No treinamento, um conjunto de amostras rotuladas por classe são apresentadas ao classificador. Ao final da etapa de treinamento, o classificador terá montado um conjunto de regras capazes de discriminar as classes entre si.

Na etapa de teste, um novo conjunto de amostras, dessa vez não rotuladas, será passado ao classificador e este será responsável por atribuir classes a essas amostras com base no modelo aprendido. As classes calculadas pelo classificador são comparadas às classes reais das amostras do conjunto de teste, permitindo assim avaliar a qualidade da classificação.

Se necessário, as fases de treinamento e de teste podem ser repetidas, modificando-se parâmetros particulares do classificador e/ou escolhendo um novo conjunto de treinamento que possua maior grau de descritibilidade entre as classes.

2.8.1 Classificador KNN

O classificador *K-Nearest Neighbors* (KNN) é um dos métodos mais clássicos e simples usado em reconhecimento de padrões. Foi introduzido por Fix e Jr (1951) e depois sofreu várias melhorias (FIX; JR, 1952; COVER; HART, 1967) até ser padronizado como conhecemos hoje por Patrick e Fischer (1970). O KNN baseia-se em métricas de distância para atribuir classes aos elementos desconhecidos.

O conjunto de treinamento é formado por vetores n-dimensionais, os quais são mapeados em um ponto no espaço também n-dimensional. Os elementos mais semelhantes tendem a se agrupar próximos entre si nesse espaço, formando as classes. Para atribuir uma classe a uma amostra de teste, o KNN busca os k elementos no espaço n-dimensional mais próximos (ou seja, aqueles com a menor distância) à amostra de teste. Desses k elementos encontrados, verifica-se a classe mais frequente, que será então atribuída à amostra de teste.

Os parâmetros do classificador KNN são o valor k de elementos formadores do conjunto de vizinhança e a métrica de distância a ser usada. Considerando-se dois pontos (x_a, y_a) e (x_b, y_b) , as métricas escolhidas para esse trabalho foram: distância Euclidiana (Equação 2.5) e distância Manhattan (Equação 2.6).

$$d_e = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2} \quad (2.5)$$

$$d_m = |x_b - x_a| + |y_b - y_a| \quad (2.6)$$

2.8.2 Classificador SVM

Support Vector Machine (SVM) é uma técnica de classificação baseada em separação espacial proposta inicialmente por Vapnik e Lerner (1963) como um classificador de natureza linear. O SVM funciona mapeando um vetor de entrada em um espaço de maior dimensão. Nesse espaço, um hiperplano é construído de forma a separar as amostras em dois conjuntos bem distintos. Por isso, o SVM é considerado um classificador binário.

Alguns padrões podem ser separados linearmente, ou seja, traçando-se uma reta que divide dois conjuntos distintos. Tais padrões enquadram-se nos chamados problemas linearmente separáveis. Entretanto, os conjuntos de dados da maioria das aplicações reais não

conseguem ser separadas linearmente, constituindo assim chamados problemas não linearmente separáveis. Boser *et al.* (1992) sugeriu um aperfeiçoamento ao SVM baseado em funções de *kernel*, possibilitando a modelagem de um classificador não linear.

O problema de separação pode ser resolvido através de:

$$g(x) = \sum_{i \in VS} \alpha_i y_i K(x_i, y_i) + w_0 \quad (2.7)$$

em que:

- $i = 1, 2, 3, \dots, n$;
- α_i são multiplicadores de Lagrange que satisfazem $0 \leq \alpha_i \leq C$;
- C é o parâmetro de regularização de erro;
- $y_i = \pm 1$ é o indicador das classes;
- $K(x_i, y_i)$ é a função de *kernel* adotada;
- $w_0 = y_i - \alpha_i y_i \phi^T(x_i) \phi(x_i)$ é o termo independente para ϕ uma função não linear.

As funções kernel mais comuns estão mostradas na tabela 1.

Tabela 1 – *Kernels* para SVM

Tipo	$K(x, y)$
Polinômial (grau d)	$(1 + xy)^d$
Gaussiano	$\exp(-\frac{ x - y ^2}{2\sigma^2})$
Perceptron Multi-Camada	$\tanh(xy - \Theta)$

Fonte: Silva *et al.* (2014)

2.9 Considerações relevantes

Boehler *et al.* (2002) expõem em seu trabalho requisitos de ferramentas de software necessárias para digitalização de imagens 3D. Dentre os módulos de softwares apresentados como necessários, está o de tratamento de nuvens de pontos, que compreende uma série de funcionalidades para remoção de pontos, com intuito de realizar limpezas nas nuvens de pontos, remover outliers, e gerar modelos 3D mais representativos dos objetos de interesse.

É feita uma comparação de 8 filtros para imagens em 3D por Sithole e Vosselman (2003). As imagens analisadas são de topografias, rurais e urbanas, geradas por *Airbone Laser Scanning* (ALS). Os testes apresentados levam em consideração características sobre os filtros comparados e realiza uma análise sobre a capacidade de remoção de outliers sobre regiões das imagens topográficas em 3D, como vegetações, construções e objetos.

Complementar ao que mostram Boehler *et al.* (2002) e Sithole e Vosselman (2003), destacando a importância de filtrar nuvens de pontos 3D, o presente trabalho tem como objetivo analisar efeitos de subamostragem com o filtro *voxel grid* em nuvens de pontos 3D de faces, sem o objetivo de remover outliers, e mostrar seus impactos sobre métodos de correção de posicionamento e atributos para identificação de faces.

3 METODOLOGIA

3.1 Hardware e Software

A fim de realizar comparações justas, todos os testes foram realizados com os mesmos hardware e software.

Hardware: Intel Core i5-7200U 2.5GHz, NVIDIA GeForce 940MX (2GB vram), 8GB DDR4.

Software: Ubuntu 16.04.3 LTS, GCC 5.4.0.

Bibliotecas: CUDA 8.0, PCL 1.8 (compilado com CUDA), OpenCV 3.3.0 (compilado com CUDA).

3.2 Subamostragem

Como pré processamento, todas as nuvens de pontos da base sofreram uma remoção de *outliers*, fazendo uso de funções com esse propósito específico, disponibilizadas pela biblioteca PCL. A biblioteca PCL disponibiliza também uma implementação do filtro *voxel grid*. Todos os procedimentos apresentados são realizados para as nuvens da base Bosphorus, em 14 configurações de subamostragens diferentes, dispostas na Tabela 2 e ilustradas na Figura 4, utilizando *voxel grid*.

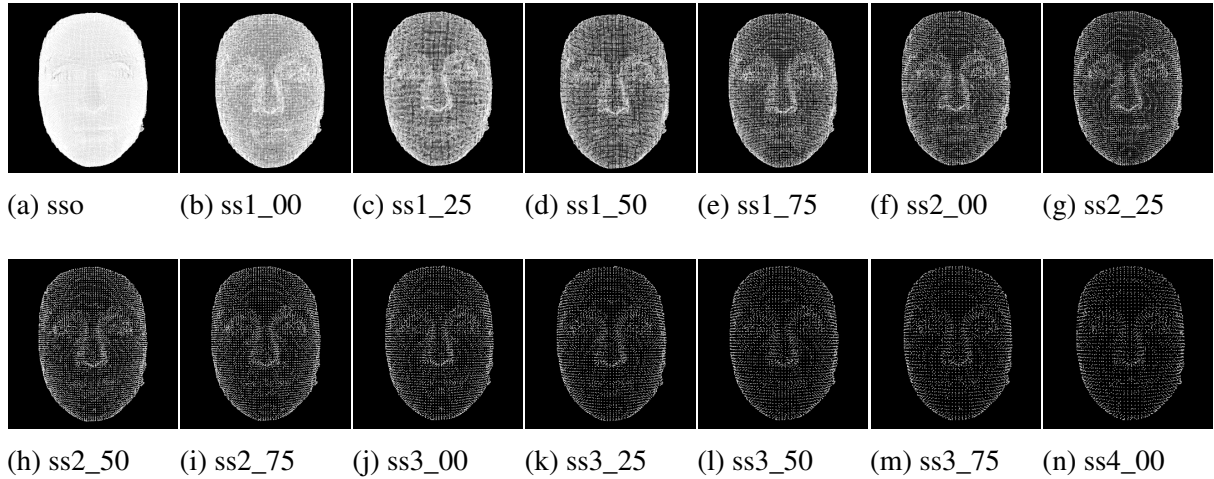
Tabela 2 – Configurações de subamostragem

Nome	Lado do <i>voxel</i> (mm)	Média de pontos
sso	(sem subamostragem)	36756.21
ss1_00	1.00	27157.50
ss1_25	1.25	20469.63
ss1_50	1.50	15856.63
ss1_75	1.75	12538.89
ss2_00	2.00	10134.12
ss2_25	2.25	8344.58
ss2_50	2.50	6973.93
ss2_75	2.75	5909.79
ss3_00	3.00	5069.81
ss3_25	3.25	4393.25
ss3_50	3.50	3842.54
ss3_75	3.75	3388.36
ss4_00	4.00	3009.81

Fonte: elaborado pelo autor.

Na Tabela 2 são apresentadas as 14 configurações de subamostragens. Elas divergem apenas no tamanho do lado do *voxel* cúbico utilizado na técnica de subamostragem *voxel grid*.

Figura 4 – Configurações de subamostragem



Fonte: elaborado pelo autor.

A Figura 4 apresenta uma mesma face sob o efeito das 14 configurações de subamostragem definidas na Tabela 2.

3.3 Análises

No presente trabalho, foram feitas análises sobre densidade, algoritmos de correção de ângulo e extração de atributos. Para as análises, foram realizados 4 tipos de procedimentos:

1. Cálculo de área e aquisição de densidades.
2. Execução de algoritmos de correção de face.
3. Extração de atributos.
4. Classificação — utilizando os atributos extraídos — de indivíduos.

3.3.1 Densidade

A densidade, d , de pontos, P , por área, A_n , da nuvem pode ser calculada, tendo seus dois parâmetros, como mostra a equação 3.1.

$$d = \frac{P}{A_n} \quad (3.1)$$

Primeiramente foi estimada uma malha para nuvem por triangulação, depois calculada a área de cada triângulo, A_t . Considerando N_t o número de triângulos utilizados para representar a malha da nuvem, é feito o somatório das áreas dos N_t triângulos e essa é a área estimada para superfície da nuvem. Com a área e número de pontos foi possível estimar qual a

densidade de cada nuvem (equação 3.2).

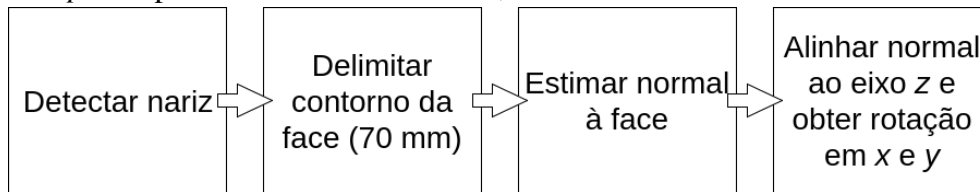
$$d = \frac{P}{\sum_{i=1}^{N_t} A_{ti}} \quad (3.2)$$

3.3.2 Correção de face

Foram analisados o tempo de execução e ângulo definido para correção de cada um dos algoritmos. Os algoritmos estudados são RANSAC, Jacobi e ICP. Além dos três algoritmos definidos, é utilizado para fins comparativos o método *Normal Estimation*, implementado pela biblioteca PCL, que estima a normal de uma superfície por meio de *Principal Component Analysis* (PCA) de uma matriz de covariância gerada pelos vizinhos mais próximos do ponto consultado.

Para o algoritmo RANSAC, são utilizadas como nuvens de entrada a face que se deseja corrigir e um plano. São encontradas a posição na qual o plano melhor representa a face e a reta normal a esse plano. O RANSAC, deste modo, é utilizado para estimar um vetor normal a nuvem desejada. Assim como *Normal Estimation*, o método de Jacobi é utilizado para estimar uma normal a nuvem desejada utilizando PCA. A figura 5 apresenta o *pipeline* utilizado para se analisar os métodos RANSAC, Jacobi e *Normal Estimation*.

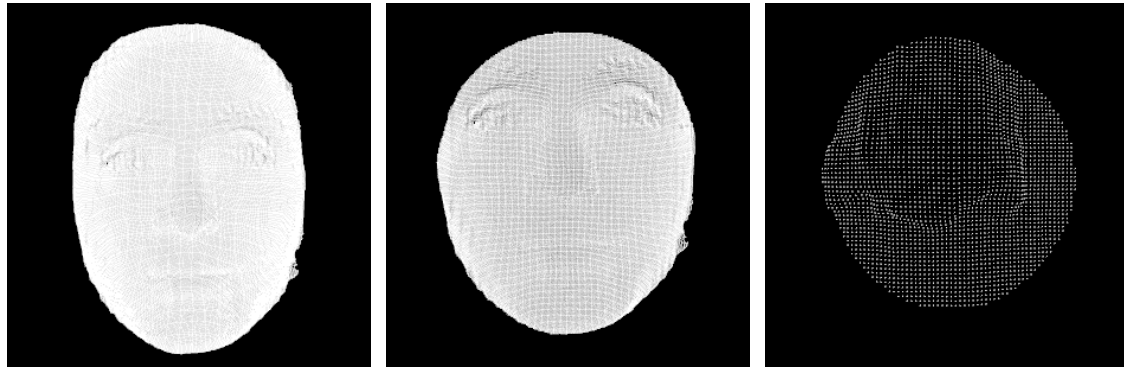
Figura 5 – *Pipeline* para os métodos RANSAC, Jacobi e *Normal Estimation*



Fonte: elaborado pelo autor.

Antes de se aplicar os métodos *Normal Estimation*, RANSAC e de Jacobi, que estimam normais à superfície, foi determinada a ponta do nariz de cada nuvem e removidos pontos a uma distância euclidiana maior que 70 mm, como é apresentado na Figura 6b. Isso é feito com o intuito de remover regiões que não são comuns a todas as nuvens e de realizar uma estimativa de reta normal mais representativa da região de interesse da face.

Figura 6 – Corte a partir da ponta do nariz



(a) Sem cortes

(b) Corte de 70 mm

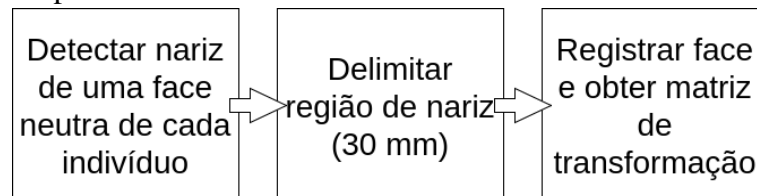
(c) Corte de 30 mm

Fonte: elaborado pelo autor.

Com os vetores normais à superfície estimados, foram verificadas quais rotações nos eixos X e Y (nessa ordem) foram necessárias para alinhar a superfície ao eixo Z , que representa a profundidade. Seguindo essa metodologia, não foram feitas rotações para correção no eixo Z . A base não apresenta posições da face com rotações em torno do eixo Z , mas elas podem existir devido as condições de aquisições de imagens dos indivíduos.

As correções utilizando o ICP seguem o *pipeline* apresentado na Figura 7, sem fazer uso das estimativas de normal à superfície da face.

Figura 7 – *Pipeline* para o métodos ICP



Fonte: elaborado pelo autor.

Para correções fazendo uso do ICP, foi feita a detecção da ponta do nariz apenas de uma face neutra (sem expressões e rotações) de cada indivíduo. Para essas faces com a ponta do nariz detectada, são delimitadas regiões de 30 mm em torno desse ponto, como mostra a figura 6c. Essa região foi escolhida devido à baixa variação e foi utilizada como nuvem de referência para o método ICP.

O ICP foi, então, aplicado para realizar o registro sobre cada nuvem, utilizando como referência a nuvem com a região do nariz do mesmo indivíduo. O resultado é uma matriz de transformação que, dentre outras informações, traz dados sobre as rotações necessárias nos eixos X , Y e Z . Desta forma, o ICP, faz uma correção relativa por indivíduo e não para base toda,

como é feito com os métodos que buscam retas normais à superfície.

3.3.3 Extração de Atributos

Como atributos para classificação das imagens foram escolhidos três tipos de momentos: os momentos *Invariants 3D* (I3D), os momentos de Hu e os momentos de Zernike. Os momentos I3D são implementados pela biblioteca PCL e são compostos por 3 momentos. Os momentos de Hu, consistem em 7 momentos, e são comumente utilizados como atributos para classificação de imagens bidimensionais (HU, 1962). Zernike possui número de momentos variável, em função dos parâmetros de ordem e repetição, com ênfase na tarefa de detecção de contornos em imagens 2D (TEAGUE, 1980). Para o presente trabalho, serão utilizadas adaptações dos momentos de Hu e de Zernike para estruturas tridimensionais (em fase de elaboração)¹, composta por 21 — 7 momentos para cada um dos planos *XY*, *XZ* e *YZ* — e 36 momentos respectivamente.

Para melhor analisar a qualidade dos vetores de atributos gerados para as diferentes configurações foram feitas identificações dos indivíduos com os classificadores SVM e KNN, sendo o segundo utilizando as distâncias Euclidiana — *K-Nearest Neighbors Euclidian* (KNNE) — e Manhattan — *K-Nearest Neighbors Manhattan* (KNNM). As classificações foram executadas, para todas as configurações de subamostragem e todos três classificadores (SVM, KNNE e KNNM), para as condições expostas na Tabela 3, usando os conjuntos de faces neutro (sem expressões ou rotações), não neutro (com expressões), C_1 , conjunto formada por uma face neutra de cada indivíduo, e C_2 , o conjunto de 6 faces de cada indivíduo, podendo ser 2 ou 3 faces neutras e as demais não neutras.

Tabela 3 – Classificações

Conjunto de treinamento	Conjunto de teste	Atributos
C_1	Neutros	Momentos de Hu
C_1	Neutros	Momentos de Zernike
C_1	Neutros	Momentos I3D
C_2	Neutros	Momentos de Hu
C_2	Neutros	Momentos de Zernike
C_2	Neutros	Momentos I3D
C_2	Não neutros	Momentos de Hu
C_2	Não neutros	Momentos de Zernike
C_2	Não neutros	Momentos I3D

Fonte: elaborado pelo autor.

¹ ROCHA NETO, A. R. **Extração de Atributos de Faces em Nuvens de Pontos Usando Momentos Invariantes Adaptados**. Trabalho de Conclusão de Curso a ser apresentado ao curso de graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará em 2017.

Como é mostrado na Tabela 3, foram feitas classificações em 9 condições diferentes, com o objetivo de identificar indivíduos. Como o presente trabalho tem como foco analisar os efeitos de subamostragem sobre as nuvens tridimensionais, não foram incluídas faces com rotações ou oclusões, pois exigem técnicas de extração de atributos, pré-processamento ou pós-processamento mais adequadas, o que foge ao escopo deste trabalho.

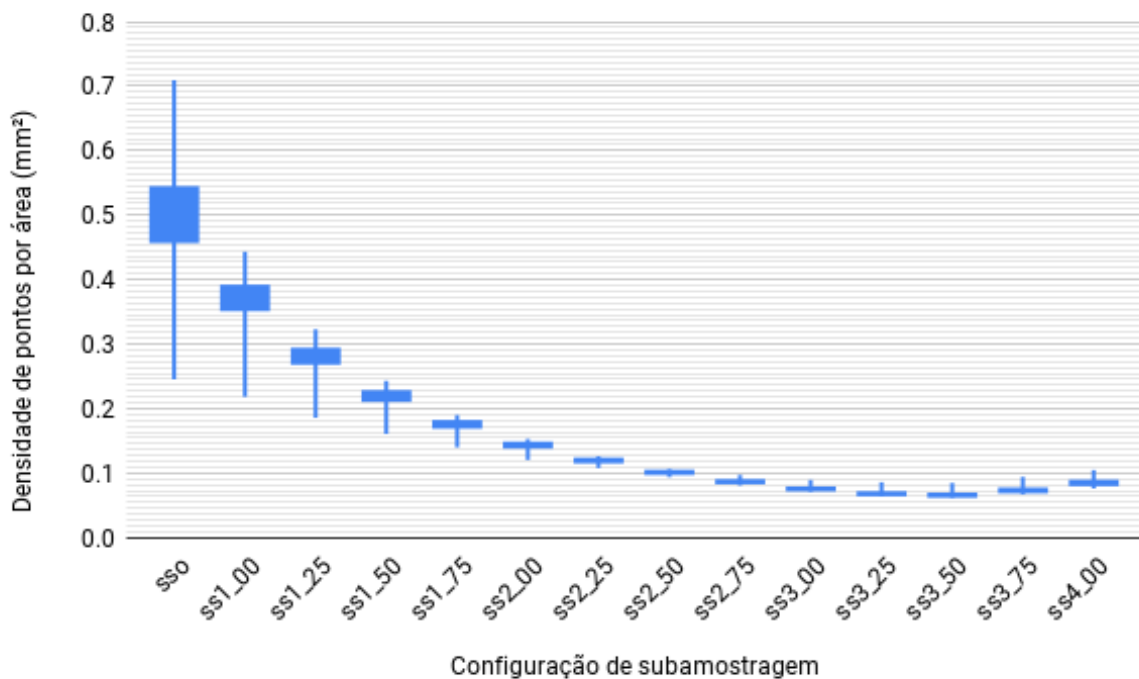
4 RESULTADOS

Após a execução dos testes propostos, foram coletados os dados para análise sobre os efeitos da subamostragem em nuvens de pontos 3D de faces da base de dados Bosphorus para a densidade de pontos por área, métodos de correção de posicionamento, extração e classificação de atributos. Os resultados são apresentados ao longo deste capítulo.

4.1 Densidade

A Figura 8 é um gráfico do tipo *boxplot* de densidade de pontos por área para cada configuração de subamostragem. Para a configuração sso — referente as nuvens sem subamostragem — é possível verificar que existe uma alta variação, intervalo de 0.25 à 0.71 pontos por milímetro quadrado, das nuvens pertencentes à base Bosphorus.

Figura 8 – Boxplot de densidade de pontos por área



Fonte: elaborado pelo autor.

Com a diminuição do número de pontos, mostra-se uma tendência de redução na variação da densidades das nuvens da base, assim como a densidade média. Com a aplicação do *voxel grid*, apenas o número de pontos deve diminuir e a área da superfície deve ser mantida, mas isso não ocorre devido a forma de aquisição da área, que sofre também com a redução de

pontos e descaracterização da nuvem, o que implica em uma estimativa menos precisa com a alteração no número de pontos. As configurações ss3_50, ss3_75 e ss4_00, como se pode ver na Figura 8, mostram uma quebra na tendência de redução da média e da variação a partir da subamostragem, indicando limitação no método para redução e normalização da densidade do conjunto de nuvens de pontos.

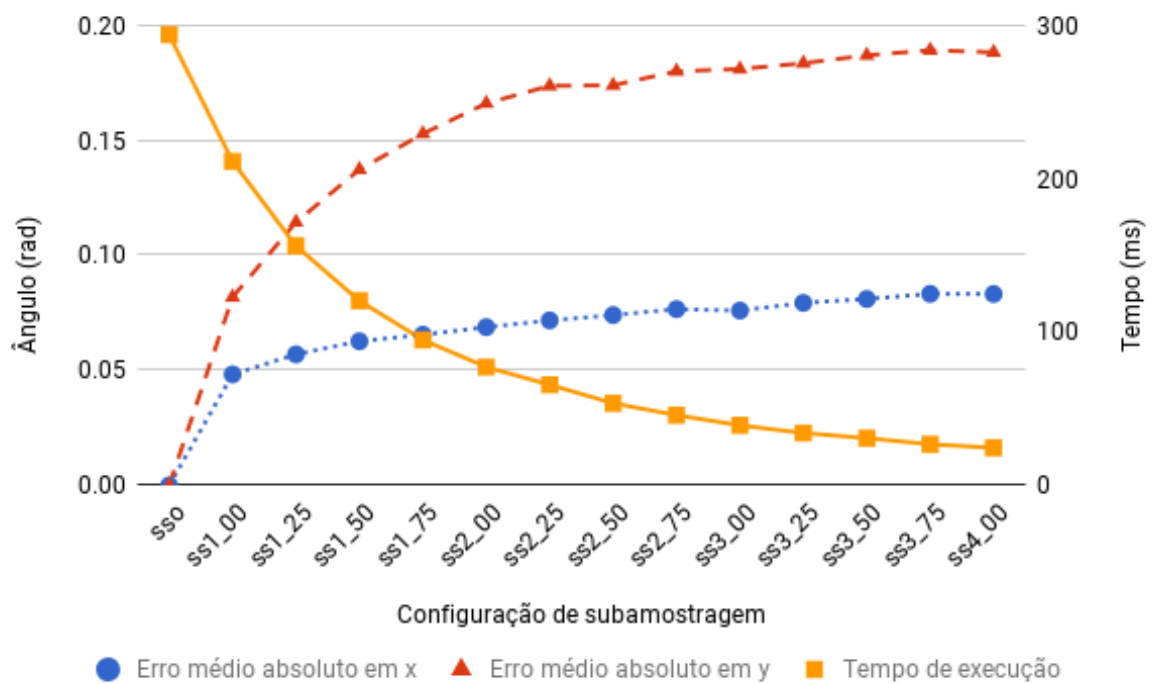
Na Figura 8 é também possível identificar um limiar, para a base de dados Bosphorus, em que a subamostragem, feita por meio de filtro *voxel grid*, ajuda na padronização na densidade das nuvens. Os melhores resultados são encontrados para a configuração de subamostragem entre ss2_25 e ss3_00. Assim, se o objetivo é aproximar as densidades das nuvens, a configuração ss2_25 é possivelmente a melhor escolha. Após esta configuração, não existem grandes alterações no desvio das médias de densidade, apenas deslocam a média para um valor mais baixo. Além disso, aumenta-se a possibilidade de descaracterizar a nuvem, devido ao menor número de pontos.

4.2 Correção de face

Para os métodos de correção do posicionamento não existe uma referência que sirva de verdade terrestre (padrão ouro) que permita orientar todas as faces da base para frente (alinhada com eixo Z), que é o objetivo dos métodos RANSAC, Jacobi e *Normal Estimation*. Da mesma maneira, não existe uma pose de referência para um alinhamento relativo por indivíduo, que é o que se busca com o alinhamento pelo ICP. Por isso, para validar os resultados das correções de posicionamento, as rotações efetuadas nas nuvens sem subamostragem foram usadas como base referencial para a comparação dos resultados da aplicação dos métodos estudados sobre as nuvens subamostradas. Além dos ângulos de rotações estimados, foram também verificados o tempo de execução médio para cada um dos métodos.

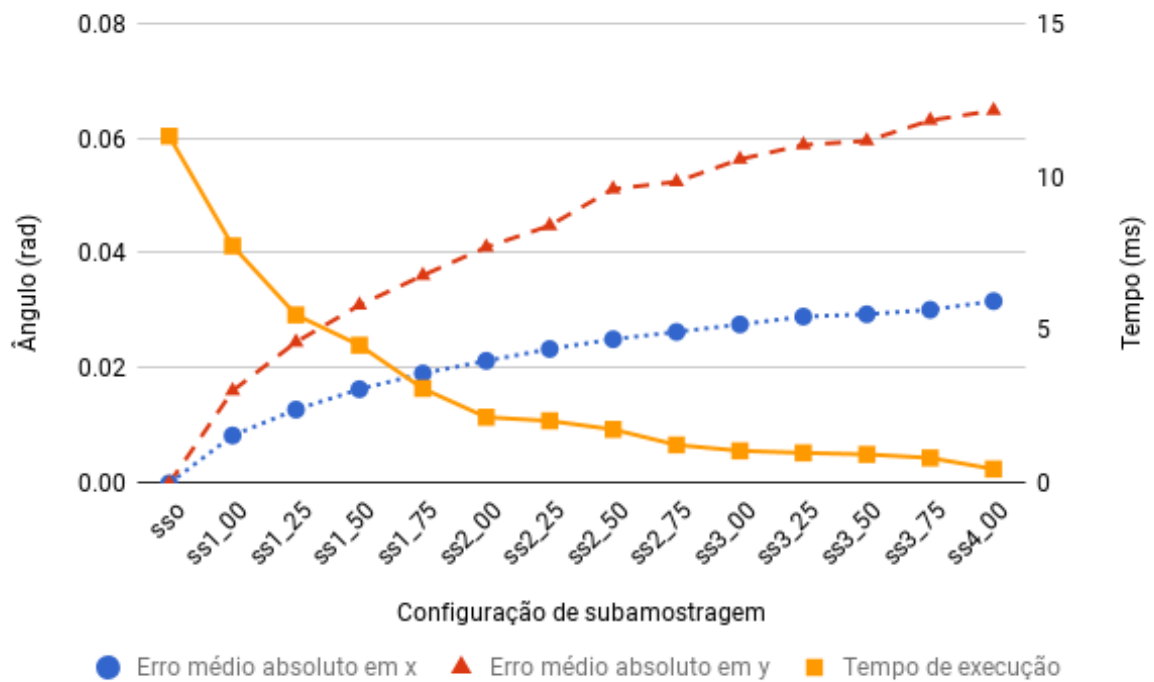
Nas Figuras 9, 10, 11 e 12 são apresentados, para os métodos RANSAC, Jacobi, *Normal Estimation* e ICP, respectivamente, a média do tempo de execução para cada uma das configurações, assim como o erro médio absoluto da rotação estimada nos eixos X e Y, para os três primeiros métodos, e X, Y e Z, para o último, nas 14 configurações de subamostragem, referentes as rotações estimadas na configuração sso.

Figura 9 – Erro médio absoluto de correção estimada e média de tempo de execução para RANSAC



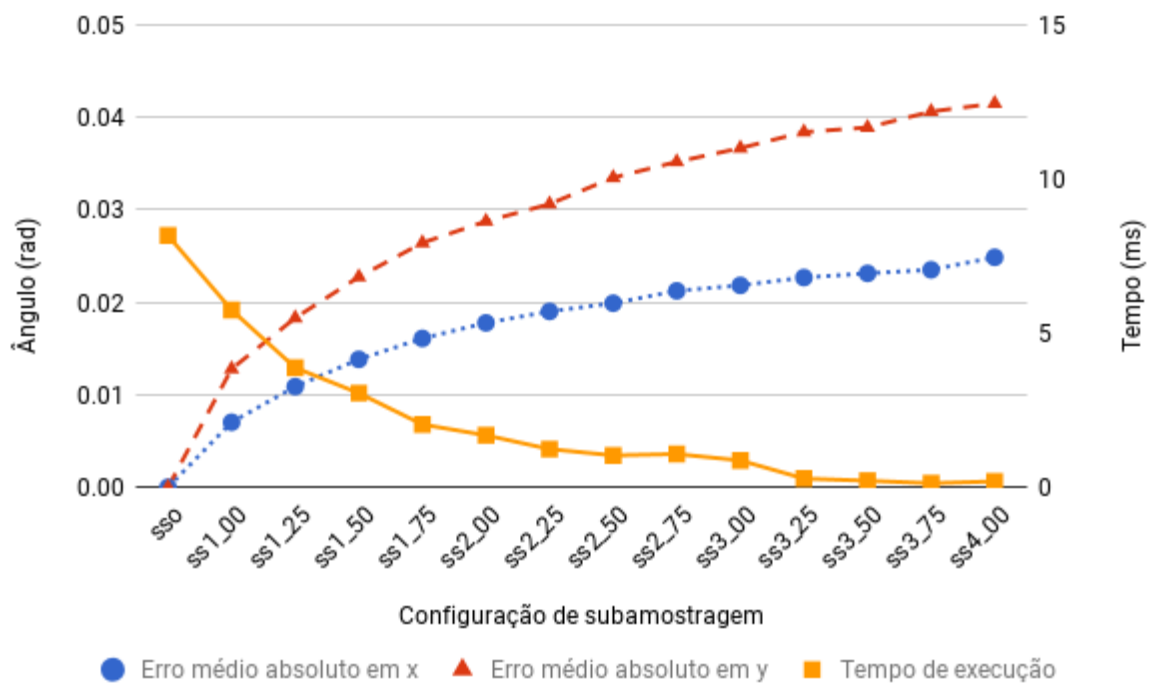
Fonte: elaborado pelo autor.

Figura 10 – Erro médio absoluto de correção estimada e média de tempo de execução para Jacobi



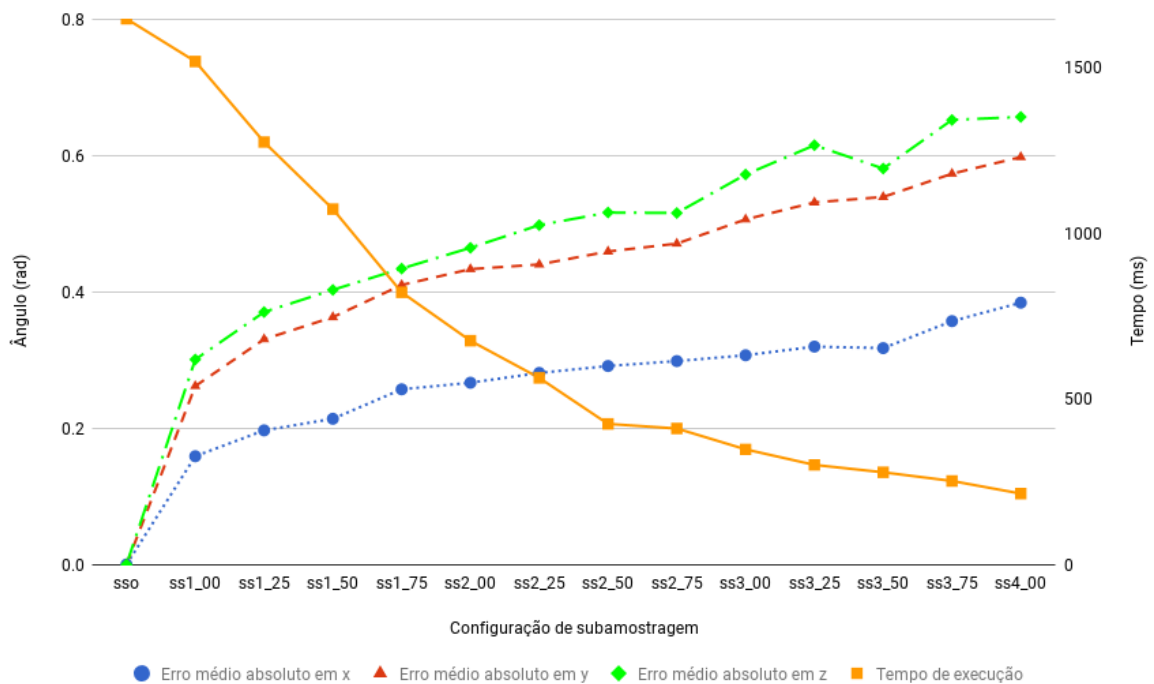
Fonte: elaborado pelo autor.

Figura 11 – Erro médio absoluto de correção estimada e média de tempo de execução para *Normal Estimation*



Fonte: elaborado pelo autor.

Figura 12 – Erro médio absoluto de correção estimada e média de tempo de execução para ICP



Fonte: elaborado pelo autor.

O método *Normal Estimation* apresenta melhor desempenho, executando em menor tempo que os métodos RANSAC, Jacobi e ICP, além de possuir os menores erros nos ângulos de rotações para correção. Com relação aos ângulos estimados para correção, o *Normal Estimation*, quando comparado aos métodos RANSAC, Jacobi e ICP, é menos sensível à subamostragem. RANSAC apresentou maior sensibilidade, tendo erros médios absolutos nos eixos X e Y , na configuração ss1_00, maiores que os erros apresentados pelo *Normal Estimation* na configuração ss4_00. O método RANSAC, após a configuração de subamostragem ss2_25, mostra-se mais estável e resiliente à diminuição de pontos. Em ordem decrescente de precisão na estimativa de rotação com relação às nuvens sem subamostragens, os métodos de correção de posicionamento são organizados na sequência: *Normal Estimation*, *Jacobi*, ICP e RANSAC.

O método *Normal Estimation*, por exemplo, reduz seu tempo de aproximadamente 8.2 ms para 5.8 ms, gerando um erro na rotação, relativo às nuvens sem subamostragem, abaixo de 0.013 radianos para os dois eixos. Assim, é necessário analisar o *trade-off* entre tempo de execução e precisão na transformação estimada, o que dependerá sempre do tipo de aplicação onde os métodos podem ser utilizados.

4.3 Extração de Atributos

Foram feitas 9 classificações, que são apresentadas na Tabela 3, com cada um dos classificadores KNNE, KNNM e SVM, em todas as 14 configurações de subamostragem (apresentadas na Tabela 2). As Tabelas 4, 5 e 6 mostram, respectivamente, os resultados para as classificações dos conjuntos de treino e teste C_1 - Neutros, C_2 - Neutros e C_2 - Não neutros, considerando apenas a taxa de resultados verdadeiros positivos das classificações. Cada linha dessas tabelas é referente a uma configuração de subamostragem. A coluna define os tipos de atributo (Hu, Zernike e I3D) para cada classificador. Os valores destacados em verde representam os maiores valores das colunas, em vermelho os menores valores das colunas e as colunas sem valores destacados são as que não possuem variações.

Tabela 4 – Classificações: C_1 - Neutros

	Hu			Zernike			I3D		
	KNNE	KNNM	SVM	KNNE	KNNM	SVM	KNNE	KNNM	SVM
sso	8,05	7,05	35,91	8,39	7,72	52,01	6,04	6,04	35,91
ss1_00	8,05	9,06	35,91	10,07	10,74	50,67	7,05	7,05	35,91
ss1_25	7,38	7,72	35,91	9,40	8,05	54,03	7,38	7,38	35,91
ss1_50	8,39	8,39	35,91	7,05	8,05	54,70	7,05	7,05	35,91
ss1_75	8,39	8,39	35,91	7,05	7,05	56,71	7,72	7,72	35,91
ss2_00	8,39	6,71	35,91	8,39	8,39	55,37	7,72	7,72	35,91
ss2_25	7,72	7,38	35,91	7,05	8,39	58,05	7,05	7,05	35,91
ss2_50	6,04	6,04	35,91	10,07	10,07	57,72	6,04	6,04	35,91
ss2_75	7,38	6,71	35,91	8,72	7,38	58,05	6,71	6,71	35,91
ss3_00	8,05	7,38	35,91	8,72	8,39	56,04	7,38	7,38	35,91
ss3_25	7,05	7,38	35,91	7,38	8,39	55,03	7,72	7,72	35,91
ss3_50	6,38	7,05	35,91	7,05	7,05	56,04	8,05	8,05	35,91
ss3_75	7,05	7,05	35,91	7,05	8,05	53,02	7,38	7,38	35,91
ss4_00	7,38	7,72	35,91	7,72	8,05	53,69	8,05	8,05	35,91

Fonte: elaborado pelo autor.

Tabela 5 – Classificações: C_2 - Neutros

	Hu			Zernike			I3D		
	KNNE	KNNM	SVM	KNNE	KNNM	SVM	KNNE	KNNM	SVM
sso	21,48	23,15	80,87	45,97	41,95	26,51	18,46	18,46	80,87
ss1_00	20,13	19,80	80,87	46,64	47,32	32,55	21,81	21,81	80,87
ss1_25	18,12	19,80	80,87	50,00	52,01	35,91	21,14	21,14	80,87
ss1_50	19,46	19,80	80,87	53,69	53,36	36,24	23,49	23,49	80,87
ss1_75	22,48	22,82	80,87	55,37	57,38	36,58	22,48	22,48	80,87
ss2_00	22,15	20,47	80,87	50,34	52,01	36,91	21,48	21,48	80,87
ss2_25	19,13	18,46	80,87	57,38	55,7	36,91	23,49	23,49	80,87
ss2_50	18,79	21,48	80,87	53,36	55,03	37,92	18,46	18,46	80,87
ss2_75	18,12	17,79	80,87	52,35	51,01	35,57	18,12	18,12	80,87
ss3_00	20,81	20,81	80,87	56,04	55,7	38,26	19,13	19,13	80,87
ss3_25	20,13	20,13	80,87	51,68	50,67	35,57	21,48	21,48	80,87
ss3_50	20,13	23,49	80,87	55,37	53,36	35,57	22,82	22,82	80,87
ss3_75	19,80	19,46	80,87	53,36	50,34	36,91	19,46	19,46	80,87
ss4_00	17,79	20,47	80,87	50,67	52,35	38,93	18,46	18,46	80,87

Fonte: elaborado pelo autor.

Tabela 6 – Classificações: C_2 - Não Neutros

	Hu			Zernike			I3D		
	KNNE	KNNM	SVM	KNNE	KNNM	SVM	KNNE	KNNM	SVM
sso	6,99	7,76	13,18	21,71	20,52	10,38	7,26	7,26	13,18
ss1_00	6,76	6,96	13,18	24,83	25,37	12,41	7,30	7,30	13,18
ss1_25	6,73	7,11	13,18	24,40	24,60	13,45	7,38	7,38	13,18
ss1_50	7,26	7,53	13,18	24,21	24,21	13,53	7,61	7,61	13,18
ss1_75	7,42	7,42	13,18	25,75	25,13	14,64	6,80	6,80	13,18
ss2_00	6,76	7,88	13,18	23,94	24,25	14,53	6,88	6,88	13,18
ss2_25	7,92	8,72	13,18	24,56	24,25	14,95	7,30	7,30	13,18
ss2_50	6,46	6,50	13,18	23,10	23,25	14,72	7,03	7,03	13,18
ss2_75	6,80	6,84	13,18	24,17	23,90	14,87	7,46	7,46	13,18
ss3_00	7,69	7,23	13,18	24,56	24,21	14,8	7,15	7,15	13,18
ss3_25	6,76	7,03	13,18	23,79	24,17	14,45	6,96	6,96	13,18
ss3_50	6,34	6,38	13,18	23,79	23,48	14,26	7,46	7,46	13,18
ss3_75	6,92	7,57	13,18	23,94	23,83	14,57	8,22	8,22	13,18
ss4_00	6,73	6,73	13,18	23,06	22,98	14,37	7,61	7,61	13,18

Fonte: elaborado pelo autor.

Nota-se que os momentos de Hu e I3D são menos sensíveis à subamostragem, sobretudo com o classificador SVM, que não apresenta alterações. Com os classificadores KNNE e KNNM é possível identificar as variações nas taxas de acertos em todas as três tabelas (Tabelas 4, 5, 6). Os momentos de Zernike mostram-se mais sensíveis e geram resultados mais distantes entre uma classificação e outra para todos os classificadores.

Em todos os casos testados, com exceção daqueles em que não existem variações, há uma configuração de subamostragem para as nuvens que gera resultados melhores do que em com as nuvens originais (sem subamostragem). A Tabela 7 mostra a quantidade de vezes em que cada uma das configurações apresentou os melhores e os piores resultados de classificação.

Tabela 7 – Incidência de melhores e piores resultados de classificação para subamostragens

Subamostragem	Incidência de:	
	pior classificação	melhor classificação
sso	8	0
ss1_00	1	4
ss1_25	0	0
ss1_50	1	3
ss1_75	4	4
ss2_00	0	1
ss2_25	0	7
ss2_50	0	1
ss2_75	3	1
ss3_00	0	0
ss3_25	0	0
ss3_50	4	3
ss3_75	1	2
ss4_00	0	3

Fonte: elaborado pelo autor.

A configuração original (sem subamostragem) apresenta o pior resultado para as classificações por 8 vezes e nunca consegue se destacar e alcançar o melhor resultado. A configuração de subamostragem ss2_25, por outro lado, não possui nenhuma ocorrência com pior resultado de classificação, destacando-se por ter sido o melhor resultado em 7 situações. Os resultados mostram, assim, que a subamostragem para a base de dados Bosphorus, com uso dos atributos considerados neste trabalho, podem melhorar os resultados de classificação. Nos casos testados, a subamostragem por *voxel grid* cúbico de lado 2,25 mm apresenta os melhores resultados, sendo portanto a recomendada para trabalhos dessa natureza.

5 CONCLUSÕES

Neste trabalho foram analisados os efeitos da redução da densidade de pontos por área em métodos de correção de posicionamento, extração de momentos invariantes como atributos e classificações de faces tridimensionais. Os experimentos foram realizados na base de dados Bosphorus.

Os testes realizados sobre os métodos de correção do posicionamento de faces permitiram identificar uma relação da subamostragem das nuvens de pontos com o tempo de execução e ângulos de rotações estimados para cada eixo. Não foi possível identificar qual configuração de subamostragem estima a melhor correção, pois não existem referências exatas para as correções esperadas de cada face. Assim, as análises das correções foram comparadas àquelas estimadas pelas correções das nuvens não subamostradas (originais), apresentando a sensibilidade na estimativa e no tempo de execução, para cada método. Dentre os métodos analisados, *Normal Estimation* apresentou estimativas de correções de nuvens subamostradas mais próximas às nuvens originais. Porém outros métodos como RANSAC e ICP mostraram também vantagens, com maior redução do tempo de execução. Os resultados dessa pesquisa podem auxiliar na determinação de qual o melhor método e configuração de subamostragem em função dos requisitos de precisão e de custo computacional definidos, com possibilidade de diminuição no tempo de execução com erros de estimativa de ângulos menores que 0,015 graus radianos, como é o caso do *Normal Estimation*. Embora os experimentos tenham sido realizados apenas na base de dados Bosphorus, estima-se um comportamento semelhante da subamostragem em outras bases, com eventual modificação de limiares devido a diferentes resoluções e qualidade dos sensores utilizados.

Os valores de momentos calculados para nuvens de pontos distintas, mas que representam um mesmo objeto, variam quando as densidades de pontos dessas nuvens divergem umas das outras. Isso se explica devido ao próprio significado dos momentos, que são grandezas escalares que representam o acúmulo de energia (pontos) em uma certa região (espaço XYZ). Isso pode limitar o uso de momentos como atributos para classificação de tais objetos, mas o problema pode ser mitigado com o nivelamento da densidade de suas amostras. Assim, neste trabalho, investigou-se também a subamostragem como técnica para identificar a taxa de redução ideal, a ser aplicada sobre todas as amostras, de maneira a deixá-las com densidades semelhantes. Foram analisados momentos invariantes como atributos para classificação de faces de indivíduos. Em todos os casos de classificações, as nuvens originais (sem subamostragem)

geram resultados iguais ou piores aos resultados do processamento de todas as configurações com subamostragem. A configuração de subamostragem com *voxel grid* de lado 2,25 mm se mostrou a melhor padronização de densidade, tendo sido a que mostrou os melhores resultados para classificação na maioria dos casos. A análise mostra que, para os atributos e classificadores testados e outros de natureza semelhante, a subamostragem pode gerar melhores resultados de classificação, como é o caso do experimento com os momentos de Zernike e classificador SVM, apresentado na Tabela 4, em que a taxa de acerto (verdadeiro positivo), sem subamostragem, é de 52,01% e alcança 58,05% com subamostragem.

Trabalhos futuros podem ser conduzidos para confirmar os resultados apresentados em outras bases de imagens 3D, permitindo a identificação dos níveis ideais de densidade para a subamostragem de objetos escaneados com diferentes sensores. Além disso, pode ser investigada a resiliência de outros momentos invariantes à subamostragem enquanto atributos usados para a classificação de objetos. Por último, dada a importância de preservar as características de densidade para o uso de momentos, pode-se investigar também o efeito de interpolação de pontos em regiões da face com densidade diferente de outras regiões, o que pode ocorrer em caso de falhas ou devido a condições específicas de posicionamento durante a captura.

REFERÊNCIAS

- BESL, P. J.; MCKAY, N. D. *et al.* A method for registration of 3-d shapes. **IEEE Transactions on pattern analysis and machine intelligence**, v. 14, n. 2, p. 239–256, 1992.
- BOEHLER, W.; HEINZ, G.; MARBS, A.; SIEBOLD, M. 3d scanning software: an introduction. In: **CIPA Heritage Documentation, International Workshop on Scanning for Cultural Heritage Recording**. [S.l.: s.n.], 2002. p. 47–51.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: ACM. **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.], 1992. p. 144–152.
- CHEN, Y.; MEDIONI, G. Object modeling by registration of multiple range images. In: IEEE. **Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on**. [S.l.], 1991. p. 2724–2729.
- COVER, T.; HART, P. Nearest neighbor pattern classification. **IEEE transactions on information theory**, IEEE, v. 13, n. 1, p. 21–27, 1967.
- DERPANIS, K. G. Overview of the ransac algorithm. **Image Rochester NY**, v. 4, n. 1, p. 2–3, 2010.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, ACM, v. 24, n. 6, p. 381–395, 1981.
- FIX, E.; JR, J. L. H. **Discriminatory analysis-nonparametric discrimination: consistency properties**. [S.l.], 1951.
- FIX, E.; JR, J. L. H. **Discriminatory analysis-nonparametric discrimination: Small sample performance**. [S.l.], 1952.
- FRANCO, N. B. **Cálculo numérico**. [S.l.]: Pearson, 2006.
- GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Edgard Blucher, 2000.
- HU, M.-K. Visual pattern recognition by moment invariants. **IRE transactions on information theory**, IEEE, v. 8, n. 2, p. 179–187, 1962.
- JÚDICE, J. J.; PATRÍCIO, J. M. Sistemas de equações lineares. **Coimbra: Departamento de Matemática da Universidade de Coimbra**, 1996.
- OPEN PERCEPTION FOUNDATION. **Point Cloud Library (PCL):Module registration**. 2017. Disponível em: <http://docs.pointclouds.org/trunk/group__registration.html>. Acesso em: 18 nov. 2017.
- PATRICK, E. A.; FISCHER, F. P. A generalized k-nearest neighbor rule. **Information and control**, Elsevier, v. 16, n. 2, p. 128–152, 1970.
- RUSINKIEWICZ, S.; LEVOY, M. Efficient variants of the icp algorithm. In: IEEE. **3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on**. [S.l.], 2001. p. 145–152.

RUSU, R. B.; COUSINS, S. 3d is here: Point cloud library (pcl). In: IEEE. **Robotics and automation (ICRA), 2011 IEEE International Conference on**. [S.l.], 2011. p. 1–4.

SAVRAN, A.; ALYÜZ, N.; DİBEKLİOĞLU, H.; ÇELİKTUTAN, O.; GÖKBERK, B.; SANKUR, B.; AKARUN, L. Bosphorus database for 3d face analysis. **Biometrics and identity management**, Springer, p. 47–56, 2008.

SILVA, R. D. C. d. *et al.* **Um Estudo sobre a extração de características e a classificação de imagens invariantes à rotação extraídas de um sensor industrial 3D**. Dissertação (Mestrado), 2014.

SITHOLE, G.; VOSSelman, G. Report: Isprs comparison of filters. **ISPRS commission III, working group**, v. 3, 2003.

TEAGUE, M. R. Image analysis via the general theory of moments. **JOSA**, Optical Society of America, v. 70, n. 8, p. 920–930, 1980.

VAPNIK, V.; LERNER, A. Y. Recognition of patterns with help of generalized portraits. **Avtomat. i Telemekh**, v. 24, n. 6, p. 774–780, 1963.

WANG, Z.; LIU, H.; QIAN, Y.; XU, T. Real-time plane segmentation and obstacle detection of 3d point clouds for indoor scenes. In: SPRINGER. **Computer Vision–ECCV 2012. Workshops and Demonstrations**. [S.l.], 2012. p. 22–31.

ZYGMUNT, M. The testing of pcl: an open-source library for point cloud processing. **Geomatics, Landmanagement and Landscape**, n. 3, p. 105–115, 2013.