UNIVERSIDADE FEDERAL DE JUIZ DE FORA
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA

Amanda de Oliveira Timotheo

Preenchimento de áreas de oclusão em domo tridimensional texturizado

Juiz de Fora

2021

Amanda de Oliveira Timotheo

Preenchimento de áreas de oclusão em domo tridimensional texturizado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Leonardo de Mello Honório

Coorientador: Prof. Dr. André Gustavo Scolari Conceição

Juiz de Fora

2021

Ficha catalográfica elaborada através do programa de geração
automática da Biblioteca Universitária da UFJF,
com os dados fornecidos pelo(a) autor(a)

Timotheo, Amanda de.
Preenchimento de áreas de oclusão em domo tridimensional texturizado / Amanda de Timotheo. -- 2021.
97 p.

Orientador: Leonardo de Mello Honório
Coorientador: André Gustavo Scolari Conceição
Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, Faculdade de Engenharia. Programa de Pós-Graduação em Engenharia Elétrica, 2021.

1. Inpainting. 2. Panorâmica equiretangular. 3. Domo texturizado. 4. Vídeo esférico. I. de Mello Honório, Leonardo, orient. II. Gustavo Scolari Conceição, André , coorient. III.Título.

Amanda de Oliveira Timotheo

Preenchimento de áreas de oclusão em domo tridimensional texturizado

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Juiz de Fora, na área de concentração em Sistemas de Energia, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Aprovada em 24 de março de 2021

BANCA EXAMINADORA

Digitally signed by Leonardo de Mello Honório
DN: C=BR, OU=Faculdade de Engenharia, O=Universidade Federal de Juiz de Fora, CN=Leonardo de Mello Honório, E=leonardo.honorio@ufjf.edu.br
Reason: I am approving this document
Location: Juiz de Fora
Date: 2021.06.18 10:01:45-03'00'
Foxit Reader Version: 10.1.3

Prof. Dr. Leonardo de Mello Honório - Orientador

Universidade Federal de Juiz de Fora, UFJF

Prof. Dr. André Gustavo Scolari Conceição - Coorientador

Universidade Federal da Bahia, UFBA

Prof. Dr. André Luis Marques Marcato

Universidade Federal de Juiz de Fora, UFJF

Prof. Dr. Carlos Henrique Valério de Moraes

Universidade Federal de Itajubá, UNIFEI

Dedico este trabalho aos meus pais, minha avó, minha irmã, meu namorado, amigos e professores que foram indispensáveis para tornar essa realização possível.

AGRADECIMENTOS

Agradeço a Deus por zelar por mim e pela minha família, concedendo-nos saúde e proteção neste período turbulento de pandemia, permitindo a defesa desta dissertação de mestrado.

Aos meus pais, Valéria e Gilmar, sou eternamente grata pelos esforços realizados em pró da minha educação, pelos ensinamentos, pelo apoio e pela confiança em mim depositada.

À minha avó, Maria Antônia e à minha irmã, Gabrielle, agradeço pelo carinho, pelo companheirismo e pela paciência diária. Ao meu namorado, Humberto, agradeço pelo amparo e pelo incentivo nos momentos em que mais precisei.

Aos meus amigos de laboratório, agradeço por tornarem mais leve e descontraído esse ano de tantas incertezas, que mesmo a distância tornaram essa experiência única e cheia de aprendizados.

Ao meu orientador, professor Leonardo Honório, agradeço imensamente pela paciência, pela atenção, pelos valiosos ensinamentos e por todas as oportunidades concedidas.

Agradeço ainda ao professor André Scolari, que com muita dedicação e comprometimento foi fundamental para a concretização desta caminhada.

RESUMO

Este trabalho tem por objetivo aplicar a técnica de *inpainting* sobre as disoclusões de uma panorâmica equiretangular e, texturizar o vídeo obtido em um domo tridimensional, de modo a gerar um vídeo esférico interativo com a aproximação uniforme dos objetos retratados na cena. A aplicação do *inpainting* tem a finalidade de reconstruir áreas de oclusão localizadas entre o plano frontal de um determinado cenário e sua região adjacente, completando informações ausentes em determinados pontos de vista. A reconstrução é feita a partir da estimação do mapa de profundidade da panorâmica de entrada, o qual será utilizado para a criação de uma estrutura em camadas e receberá a aplicação de uma rede neural convolucional. A etapa de texturização complementa a metodologia ao inserir o vídeo resultante sob a perspectiva tridimensional de um domo, obtendo assim uma visualização capaz de alcançar 360 graus de amplitude. A junção de ambos os processos, o *inpainting* e a texturização no domo tridimensional, geraram como resultado um vídeo projetado sobre uma esfera que proporciona ao observador a ampliação de objetos inseridos na panorâmica equiretangular, de forma homogênea e sem distorções aparentes. Os resultados obtidos apresentaram ainda uma forma mais realística de virtualização do ambiente através da fotogrametria.

Palavras-chave: *Inpainting*. Panorâmica equiretangular. Domo texturizado. Vídeo esférico.

ABSTRACT

This work aims to apply the inpainting technique on the disocclusions of an equi-rectangular panorama and texture the video obtained in a three-dimensional dome, in order to generate an interactive spherical video with a uniform approximation of the objects portrayed in the scene. The application of inpainting aims to reconstruct areas of occlusion located between the frontal plane of a given scenario and its adjacent region, completing information missing from certain points of view. The reconstruction is made from the estimation of the input panoramic depth map, which will be used to create a layered structure and will receive the application of a convolutional neural network. The texturing step complements the methodology by inserting the resulting video from the three-dimensional perspective of a dome, thus obtaining a view capable of reaching 360 degrees of amplitude. The combination of both processes, inpainting and texturing in the three-dimensional dome, generated as result a video projected on a sphere that provides the observer the enlargement of objects inserted in the equirectangular panorama, in a homogeneous way and without apparent distortions. The results obtained presented a more realistic way of virtualizing the environment through photogrammetry.

Keywords: Inpainting. Equirectangular panoramic. Textured dome. Spherical video.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O *inpainting* é uma técnica utilizada para a reconstrução de áreas danificadas ou ausentes em determinada estrutura, que teve sua origem no campo das artes, através do processo de restauração em pinturas antigas. A técnica tornou-se conhecida através do conservador alemão, Helmut Ruhemann (1), que durante uma Conferência Internacional de Estudos sobre a Preservação em Artes, ocorrida em 1930, sugeriu um método de restauração que mantivesse os traços confeccionados pelo pintor original.

Com o avanço da tecnologia, a técnica de *inpainting* foi se tornando mais abrangente e sofisticada, apresentando várias aplicações e metodologias. Atualmente utiliza-se esse procedimento não apenas em pinturas, mas também no meio digital, como em imagens (2), mapas de profundidade (3), estruturas tridimensionais (4) e vídeos (5). Entretanto, a abordagem utilizada para a mídia digital é diferente daquela adotada para restauração física, sendo desempenhada por s*oftwares* avançados e com embasamento matemático aprofundado.

A escolha da metodologia utilizada para o *inpainting* dependerá da finalidade do preenchimento e do tipo de entrada escolhida. Contudo, é válido ressaltar que as regiões a serem reconstruídas terão sempre o seu embasamento nos dados vizinhos circundantes, independente da metodologia adotada, isso porque as características necessárias para a realização do preenchimento são similares às informações disponíveis em suas vizinhanças (6). Desta maneira as abordagens se diferenciarão apenas entre análises globais e locais, variando os métodos de extração das características.

Este capítulo abrangerá as aplicações utilizando a técnica de *inpainting* para preenchimento em imagens, a definição da proposta deste trabalho e organização dos capítulos que serão apresentados.

1.1 APLICAÇÕES EM IMAGENS

Diversas são as finalidades da aplicação do *inpainting* sobre as imagens, uma abordagem recorrente nessa área de pesquisa é o preenchimento de imagens danificadas que sofreram perdas de dados ou interferência de ruídos em sua estrutura. O trabalho de Bertalmio et al. (7) é um exemplo de aplicação direcionada para solucionar lacunas em imagens, sua metodologia foi capaz de imitar o processo de restauração dos pintores, transferindo

informações da borda para o centro da imagem de forma sucessiva. Para Xie et al. (8) a finalidade do processo foi direcionada para a correção de imagens com ruídos, desenvolvendo uma metodologia para resolver casos complexos de interferência.

Outra abordagem é o preenchimento de regiões inexistentes na captura original, que podem ser originadas após a remoção de algum item da imagem ou pela paralaxe, que é a diferença da posição aparente de um objeto em virtude da movimentação do observador. As regiões de lacunas ocasionas pelo primeiro caso, são nomeadas de disoclusões, onde o objetivo do preenchimento realizado pelo *inpainting* é reconstruir o plano de fundo localizado atrás do objeto removido, de modo que seja imperceptível aos olhos humanos perceber a diferença. Um exemplo desta aplicação é o trabalho de Borole et al (9), que propõe um algoritmo onde a busca e o preenchimento são realizados automaticamente sobre as áreas de disoclusão, de modo simultâneo sobre diferentes regiões com estruturas e fundos distintos.

O segundo caso, originado pela paralaxe, resulta no campo de estudos das fotos 3D, que são vídeos tridimensionais capazes de fornecer movimento a imagem, a partir da utilização do preenchimento em áreas ocluídas localizadas entre o plano frontal e o plano de fundo da imagem. Para realizar esse procedimento é adotado a utilização de multi-planos de imagens, como a representação LDI (*Layered Depth Image*) (10), uma estrutura em camadas capaz de compactar dados de coloração e profundidade. Como exemplos da utilização desta abordagem, têm-se as pesquisas de Lu et al. (11), de Shih et al. (12) e também para a criação de fotos 3D na rede social Facebook.

As metodologias aplicadas para a extração de característica envolvem, em sua maioria, a utilização de redes neurais convolucionais, devido a sua capacidade de estimar e adaptar diferentes tipos de dados através da execução de treinamentos. A escolha da rede neural convolucional pode ser vista em diferentes trabalhos envolvendo a técnica de *inpainting*, como é o caso de Pathak et al. (13) que utilizou pela primeira vez uma *Generative Adversarial Network (GANs)* para o preenchimento de lacunas em imagens, e o de Laube et al. (14) que utilizou em sua pesquisa uma rede neural convolucional para a síntese de textura, de forma global e local.

Deste modo, percebe-se que a técnica de *inpainting* tem grande aplicabilidade para entradas com imagens, apresentando potenciais resultados que poderão ser utilizados para diversas finalidades.

1.2 OBJETIVOS

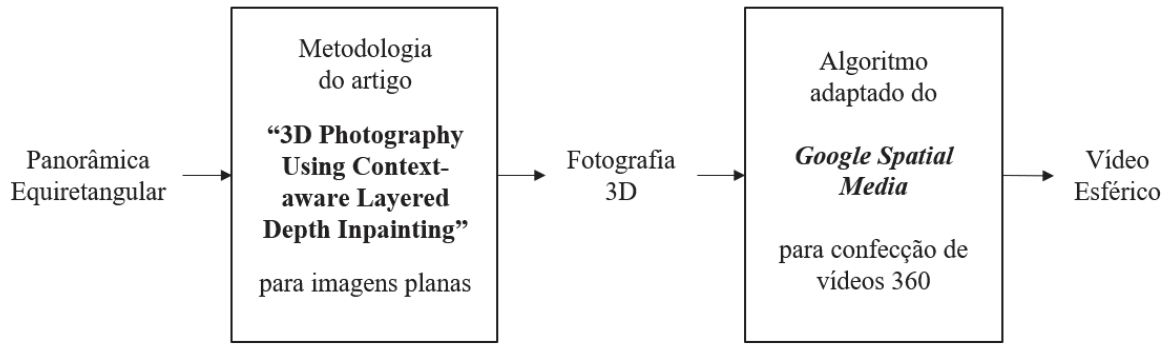
Nesta seção serão abordados os objetivos gerais e específicos desta dissertação.

1.2.1 Objetivos Gerais

Esse trabalho tem como objetivo gerar um vídeo esférico que apresente uma visualização a mais de profundidade, proporcionando ao observador movimentar-se pelo ambiente retratado, realizando aproximações do cenário sem causar distorções. Para isso, será aplicado a técnica de *inpainting* na região de oclusão presente entre os planos frontal e posterior de uma imagem panorâmica equiretangular. E em sequência, será realizada a texturização do vídeo resultante em um domo tridimensional.

Propõe-se então a utilização da metodologia de preenchimento em áreas de oclusão para imagens planas convencionais, desenvolvida por Shih et al (12), no artigo "*3D Photography Using Context-Aware Layered Depth Inpainting*". E para a projeção no domo, uma adaptação do algoritmo *Google Spatial Media*[1], utilizado pra gerar vídeos 360 graus (Figura 1).

Figura 1 - Fluxograma



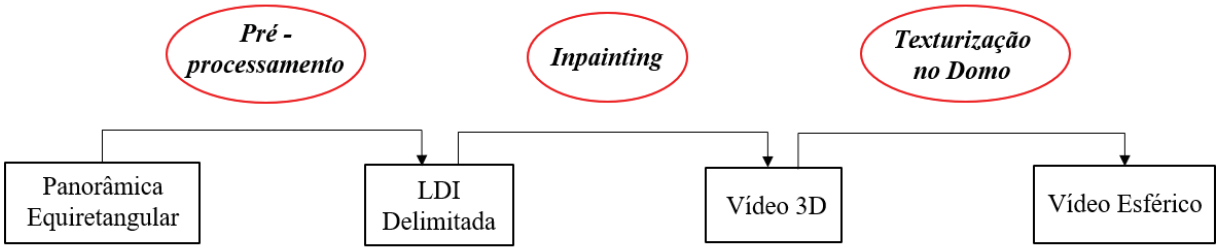
Fonte: Elaborado pela autora (2021)

A execução desta proposta é dividida em 3 processos: pré-processamento, *inpainting* e texturização no domo tridimensional. O primeiro processo será responsável pela extração de um mapa de profundidades e a criação da LDI (*Layered Depth Image*) (10) inicial, assim como a marcação dos segmentos que compõem a parte frontal da panorâmica equiretangular de

[1] *Google Spatial Media*: https://github.com/google/spatial-media

entrada. O segundo processo realizará o preenchimento sobre as áreas ocluídas através da utilização de uma rede neural convolucional. Por fim, o terceiro processo concretiza a proposta, ao texturizar a fotografia 3D em um domo, gerando um vídeo esférico. A Figura 2 apresenta um fluxograma que sintetiza os 3 processos citados acima, assim como as suas entradas e saídas.

Figura 2 - Processos



Fonte: Elaborado pela autora (2021)

1.2.1 Objetivos Específicos

Este trabalho apresenta os seguintes objetos específicos:

- Propor um novo tipo de vídeo esférico ao acrescentar informação de profundidade;
- Aplicar uma entrada não plana na metodologia de *inpainting* utilizada;
- Adaptar o algoritmo *Google Spatial Media.*

1.3 ESTRUTURA DO TRABALHO

Para concretizar a proposta de gerar um vídeo esférico utilizando uma panorâmica com áreas de oclusão preenchidas, este trabalho foi divido em 7 capítulos. O capítulo 2 é destinado a fundamentação teórica, onde será abordado a parte conceitual, como a enumeração de alguns tipos de panorâmicas existentes e o embasamento matemático relacionado a panorâmica do tipo equiretangular. Os capítulos 3, 4 e 5, apresentarão o desenvolvimento da metodologia sugerida neste trabalho, na ordem de execução dos processos. Deste modo, o capítulo 3 abordará as técnicas aplicadas na etapa de pré-processamento, como a extração do mapa de profundidade e a delimitação do plano frontal através da definição dos segmentos de pixel. O capítulo 4

discorrerá sobre a técnica de *inpainting*, apresentando o processo de preenchimento através de uma rede neural convolucional subdividida. E o capítulo 5 finaliza a metodologia ao detalhar o processo de texturização do vídeo resultante sobre o domo tridimensional. Os resultados obtidos utilizando diferentes panorâmicas equiretangulares serão apresentados no capítulo 6 e, as conclusões adquiridas, no capítulo 7, que evidenciará também as possibilidades de aperfeiçoamento para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho abordará o conceito e alguns tipos de panorâmicas, aprofundando na teoria relacionada a construção da panorâmica equiretangular. Esse conteúdo é essencial para compreender a escolha do tipo de entrada que será utilizada ao decorrer dos processos, uma vez que está decisão impactará diretamente no resultado final do vídeo esférico projetado.

2.1 PANORÂMICA

A palavra panorama é formada pelos radicais gregos “pan”, que significa “total”, e “orama”, que significa “vista”, concatenando ambos os morfemas têm-se o sentido de “vista total” ou vista completa, aquela que é capaz de capturar em amplitude máxima um determinado ambiente circundando em uma única localização. Ou seja, uma vista ampla que engloba áreas circunvizinhas em sua formação.

A fotografia panorâmica é um panorama capturado através de uma câmera, construída a partir de várias fotos em sequência que são emendadas para formar uma única imagem de até 360 graus. As primeiras versões foram construídas simplesmente pelo alinhamento de fotos impressas a partir do filme, eram rústicas e imperfeitas devido à dificuldade em conectá-las de forma homogênea. Atualmente, com a evolução da tecnologia, há distintos tipos de panorâmica, diferindo-se pela abrangência do número de imagens utilizadas em sua composição ou pelos critérios de projeção utilizados. Dentre os tipos existentes, há três modelos importantes a serem enunciados, são eles:

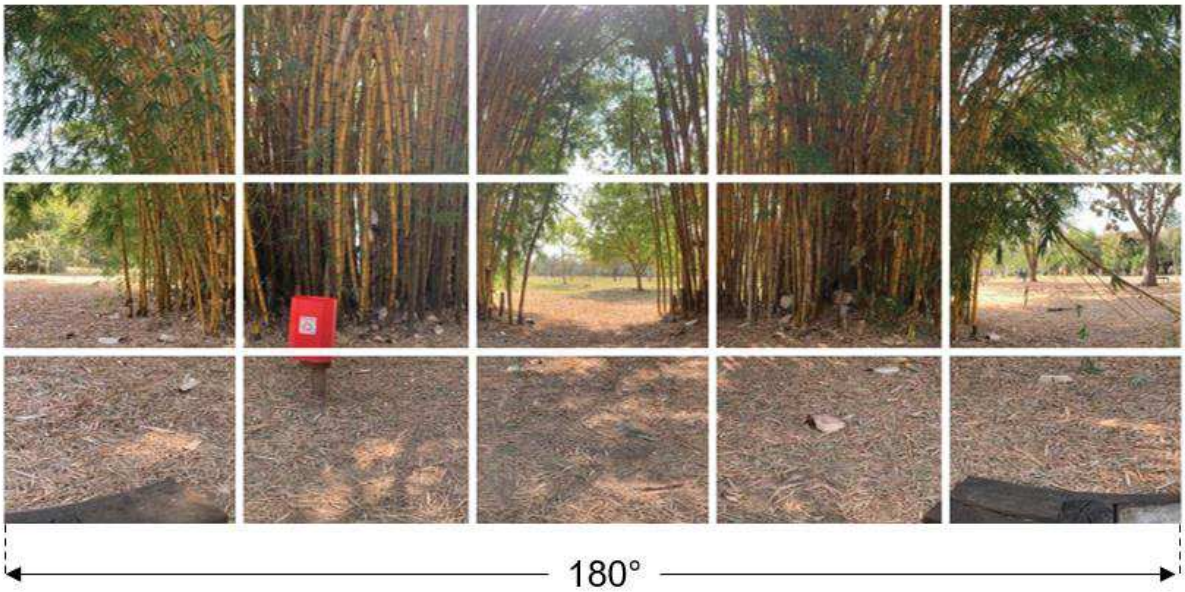
- Panorâmica parcial;
- Panorâmica 360 graus;
- Panorâmica equiretangular.

A seguir será apresentado uma breve explicação sobre cada um deles.

2.2 PANORÂMICA PARCIAL

A panorâmica parcial é a representação de algum cenário de maneira ampla, formada pela junção de um conjunto de n imagens, que juntas não atinjam 360 graus de amplitude horizontal. Este modelo pode ser dividido em três grupos: as que apresentam cenas inferiores a 180 graus de amplitude, as que apresentam exatamente 180 graus de abrangência e as que possuem angulação entre 180 e 360 graus (4).

Figura 3 – Panorâmica parcial



Fonte: Elaborada pela autora (2021)

As panorâmicas parciais com grande abrangência, ou seja, próximas a 360° de amplitude horizontal, apresentam a noção de completude. Entretanto, ao observar as suas extremidades, é possível perceber que não são complementares, (Figura 4), pois apresentam uma região onde há ausência de informação. Desta maneira, pode-se afirmar que ao unir as extremidades horizontais de uma panorâmica parcial, a representação resultante não irá formar um "todo" e evidenciará uma região de lacuna.

Figura 4 - Extremidades da panorâmica de grande abrangência

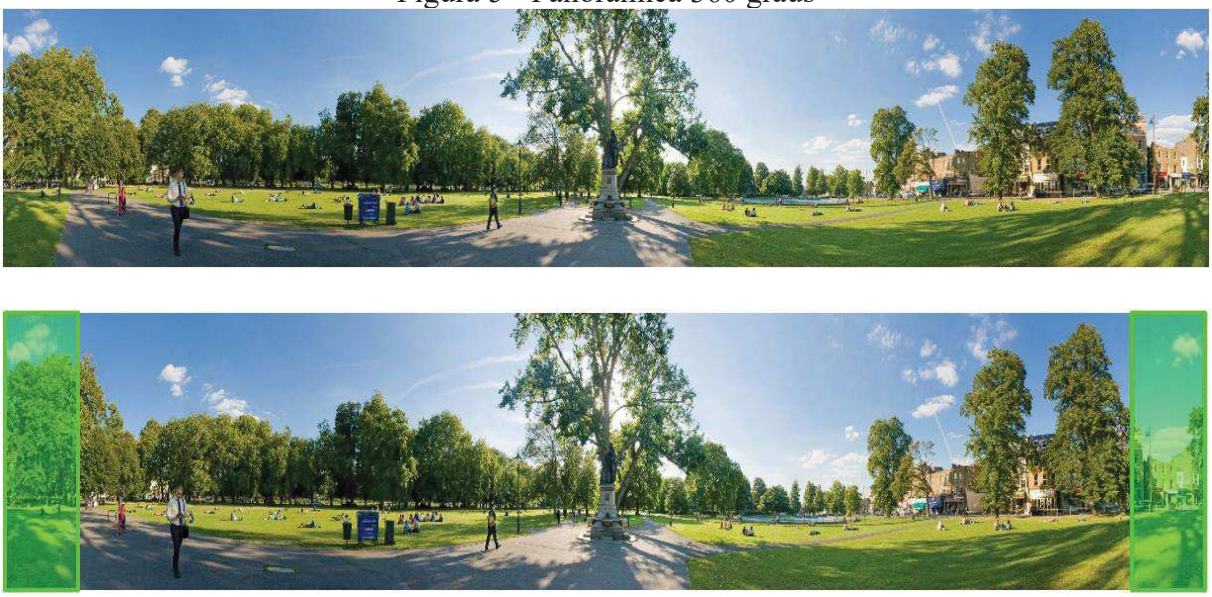
Fonte: Elaborada pela autora (2021)

Para alguns casos, este tipo de representação é considerado imperfeita, pois a mínima falta de informação nas extremidades a torna incompleta para determinadas aplicações.

2.3 PANORÂMICA 360 GRAUS

A panorâmica 360 graus captura a visão ao redor do observador formando 360 graus de abrangência. Desta maneira, a extremidade da direita complementa a extremidade da esquerda, e unidas compõem uma visão completa do horizonte circundando que forma o ambiente capturado. Porém apesar de completa no quesito amplitude horizontal, ainda não captura toda a angulação vertical presente no cenário. Desta forma, não são retratadas as informações relativas à parte superior e a parte inferior do ambiente, como por exemplo o céu e o chão (Figura 5).

Figura 5 - Panorâmica 360 graus



Fonte: Elaborada pela autora (2021)

Portanto, as panorâmicas 360º, apesar de avançadas, ainda são incompletas verticalmente, sendo constituídas por uma panorâmica parcial de grande abrangência acrescida de imagens “comuns” até que as duas extremidades da fotografia se complementem formando um todo. Este tipo satisfaz diversas aplicações, como por exemplo o registro de uma paisagem que circunda um determinado ponto de vista ou aquisições que promovam a ideia de amplitude horizontal máxima. Contudo, a falta de informações nas extremidades verticais impacta no desenvolvimento de processos que dependam destes dados, tornando esse tipo de panorâmica inválida para algumas aplicações, como é o caso deste trabalho.

2.4 PANORÂMICA EQUIRETANGULAR

Uma panorâmica equiretangular abrange todo o campo de visão em 360 graus horizontalmente e 180 graus verticalmente, a proporção 2:1 (360 x 180) forma um retângulo cuja altura é metade da largura. Os 180 graus representados na vertical são referentes aos 90 graus da perpendicular entre o ponto de visão central e o último ponto quando se olha para cima, somados aos 90 graus da perpendicular entre o ponto de visão central e o último ponto quando se olha para baixo.

Analisando este ambiente em termos tridimensionais, percebe-se que a amplitude de 360 graus horizontais e 180 graus verticais se assemelham a representação de uma estrutura esférica. Portanto, pode-se equiparar ambas representações, ou seja, a panorâmica equiretangular é uma representação bidimensional de uma esfera e, por isso, esta também é nomeada como imagem esférica. A subseção 2.3.1 discorrerá sobre essa projeção e suas peculiaridades.

Os demais modelos de panorâmicas mostrados anteriormente foram construídos a partir da concatenação de imagens “comuns”, para este modelo a mesma abordagem pode ser adotada,

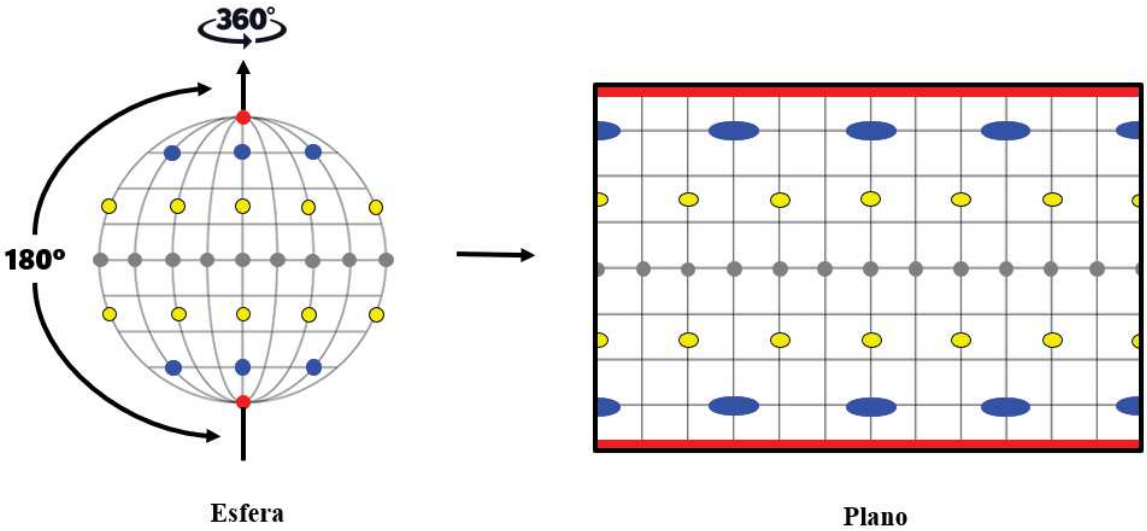
desde que o conjunto destas fotografias abranja todo o ambiente, tanto verticalmente, como horizontalmente. A subseção 2.3.2 mostrará o processo de construção de uma panorâmica equiretangular deste modo, embora existam atualmente tecnologias capazes de capturar todos estes dados simultaneamente, como é o caso das câmeras 360.

2.4.1 Projeção cilíndrica equidistante

A panorâmica equiretangular baseia-se na teoria da projeção cilíndrica equidistante, a qual realiza a representação de uma esfera em um plano cartesiano a partir de um cilindro que circunda essa esfera, tangenciando linearmente sua horizontal central. Desta maneira, respeita-se o espaçamento uniforme entre as retas verticais e as horizontais que a mapeiam, essa proporção cria uma grade com espaçamentos iguais.

Apesar de respeitar o distanciamento homogêneo nas linhas que compõem o mapeamento, esta proporção não é mantida quanto a representação das áreas. Desse modo, as regiões localizadas próximas as extremidades da esfera tendem a se alongar horizontalmente para preencher toda a extensão do plano e, de forma proporcional, tendem a reduzir a distorção quando se aproximam do centro. Portanto, as maiores distorções ocorrem exatamente nos extremos da esfera, pois na representação tridimensional ambos são representados apenas por um ponto, enquanto no plano bidimensional estes se deformam formando linhas (Figura 6).

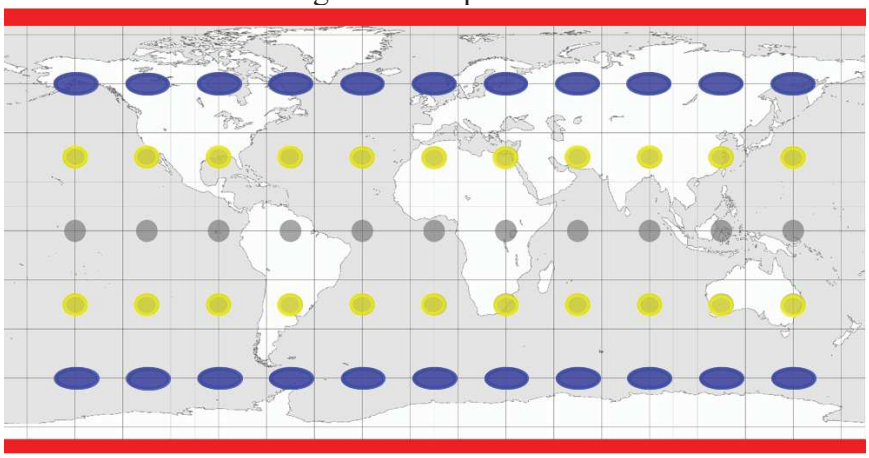
Figura 6 - Projeção cilíndrica equidistante



Fonte: Elaborada pela autora (2021)

Este processo é facilmente visualizado na transcrição do globo terrestre em um mapa-múndi (Figura 7). Os 360 graus horizontais são circundados pela linha do Equador enquanto os Meridianos representam os 180 graus percorridos. Percebe-se que na medida em que ocorre o afastamento no sentido dos polos, há distorções inversamente proporcionais à distância dos extremos superior e inferior, ou seja, quanto menor o distanciamento, maior será a distorção. Nesta projeção nota-se também que os polos norte e sul, representados por um ponto na esfera, quando transferidos para o plano bidimensional são representados por uma linha.

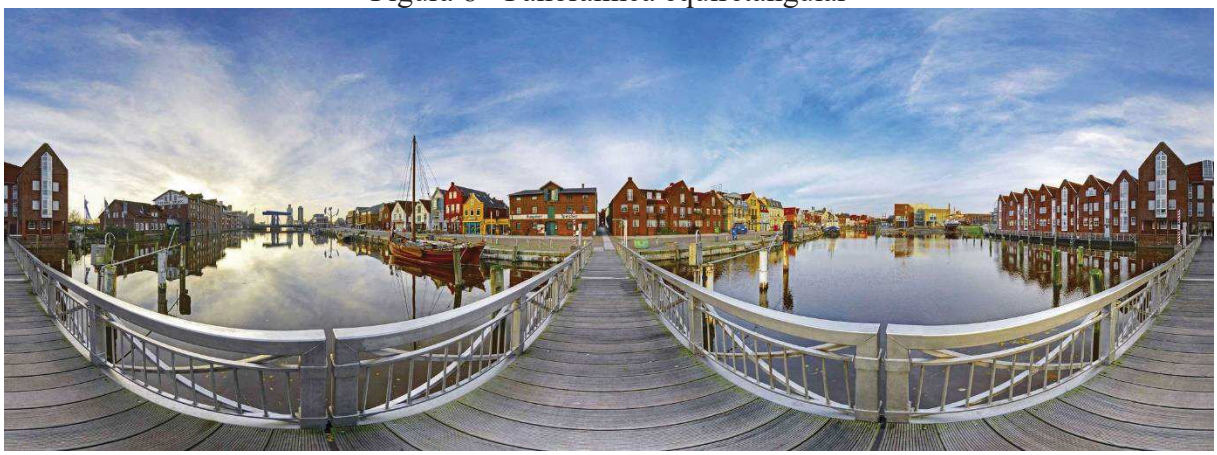
Figura 7 - Mapa múndi



Fonte: Elaborada pela autora (2021)

A fotografia panorâmica representada na Figura 8 apresenta todos os itens descritos anteriormente. O cenário está abrangido em 360 graus horizontais, pois é possível notar que as casas localizadas nos extremos da direita e da esquerda se complementam, e em 180 graus verticais devido a notável abrangência de informações relativas à parte superior e inferior da imagem. Há distorções pronunciadas nas regiões de extremidade, sendo possível percebe-las por exemplo, na ponte e no céu. De forma mais branda, estas aparecem ao caminhar em direção a região central da imagem, onde não há distorções visíveis.

Figura 8 - Panorâmica equiretangular

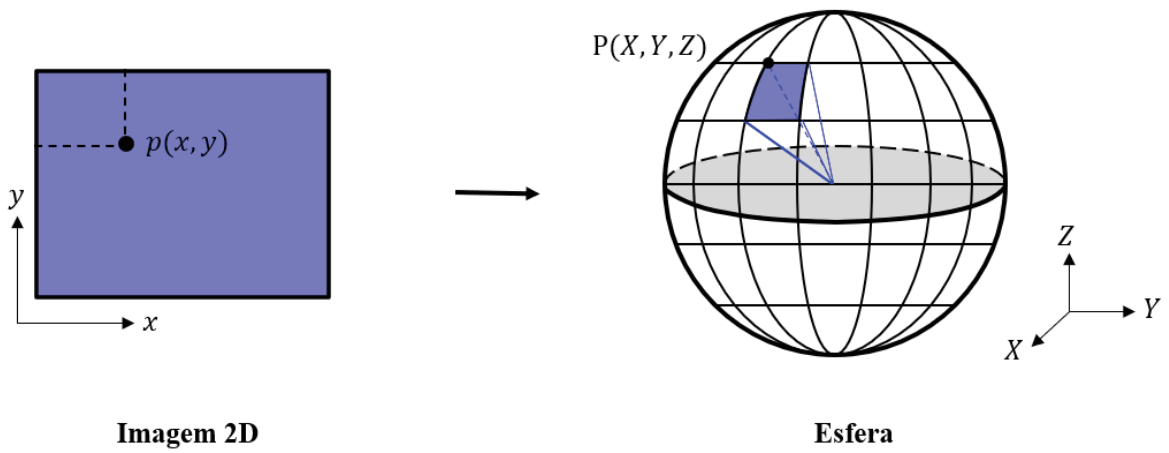


Fonte: (15)

2.4.2 Construção de uma panorâmica equiretangular

A construção de uma panorâmica equiretangular a partir da concatenação de diversas imagens "comuns" é realizada de forma diferenciada em relação as demais panorâmicas descritas anteriormente. Primeiramente é necessário a projeção individualizada de cada fotografia sobre uma esfera virtual de raio R (Figura 9), que converterá cada pixel (x, y) bidimensional para a estrutura tridimensional (X, Y, Z). Esse raio R pode ser definido arbitrariamente, porém para evitar a redução da resolução das fotografias originais, é sugerido que seja feito $R = f$, onde f é o comprimento focal da câmera, que deve se manter constante durante a aquisição das imagens isoladas.

Figura 9 - Projeção das imagens na esfera



Fonte: Elaborada pela autora (2021)

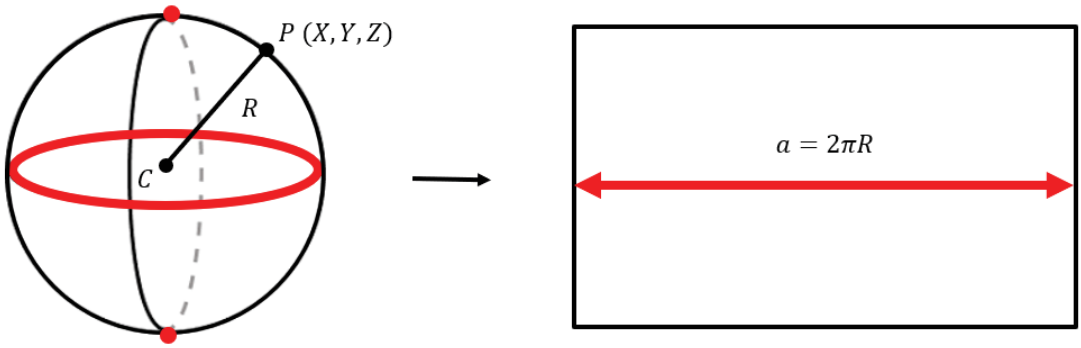
Após a projeção de cada imagem isolada obtém-se uma esfera completamente texturizada, onde cada ponto (X, Y, Z) corresponde a um pixel da imagem formada sobre a estrutura esférica. O próximo passo será então converter a texturização da esfera em uma imagem panorâmica equiretangular. Para isso, mapeia-se a esfera obtida para o plano bidimensional através da projeção cilíndrica equidistante.

Considere uma esfera de raio R, onde C é o centro da esfera e P é um ponto tridimensional de coordenadas (X, Y, Z) pertencente a esfera. Sabe-se que o perímetro da circunferência central da esfera é dado por:

$$Perímetro = 2\pi R \tag{1}$$

Deste modo, tem-se que a largura a da panorâmica equiretangular medirá $2\pi R$ ao ser mapeada para o plano bidimensional. Assim como os polos da esfera, uma vez que neste modelo de panorâmica, estes se deformam formando uma linha de extensão a (Figura 10).

Figura 10 - Largura da panorâmica equiretangular



Fonte: Elaborada pela autora (2021)

Cada ponto P_i da esfera, para $i \in \mathbb{N}$, representado pelas coordenadas $(X_i,\ Y_i,\ Z_i)$ pode ser calculado conforme as equações matemáticas a seguir.

$$X = R.\ sen\,\theta.\ cos\,\varphi \tag{2}$$

$$Y = R.\ cos\,\theta.\ sen\,\varphi \tag{3}$$

$$Z = R.\ cos\,\varphi \tag{4}$$

Onde os ângulos θ e φ são respectivamente longitude e latitude, em radianos, que podem ser deduzidos através de operações algébricas e trigonométricas como:

$$\theta = \arctan\frac{Y}{X} \tag{5}$$

$$\varphi = \arccos\frac{Z}{\sqrt{X^2 + Y^2 + Z^2}} \tag{6}$$

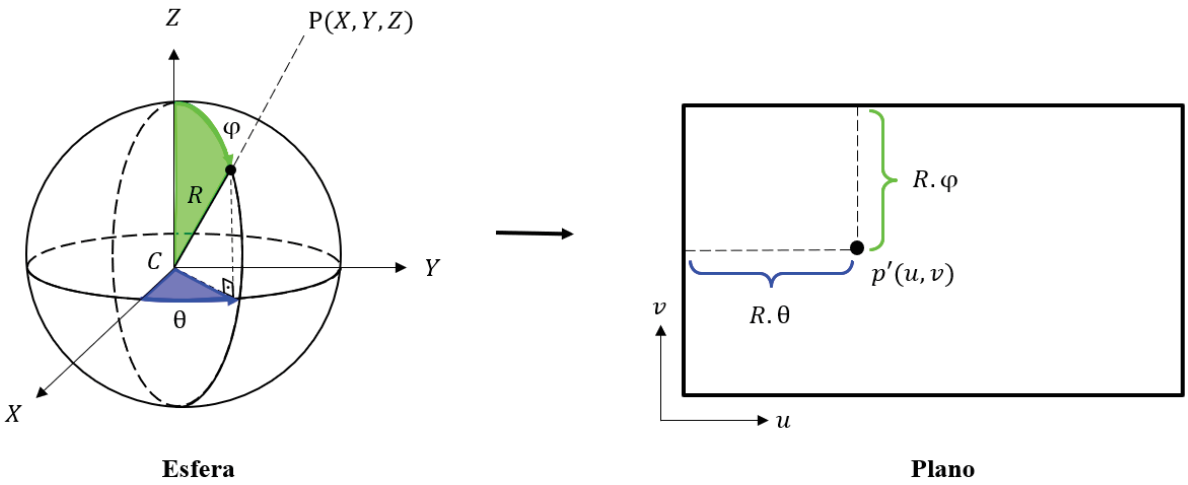
Conhecidos os valores de X, Y, Z, realiza-se a transformação da esfera para o plano tridimensional através da relação $R[\theta, \varphi]^T = [u, v]^T$ baseada nas direções em longitude e latitude. Esta relação é conhecida como mapeamento latitude- longitude, o qual pode ser calculado da seguinte maneira:

$$u = R.\varphi \tag{6}$$

$$v = R.\theta \tag{7}$$

Onde (u, v) são as coordenadas dos pixels na panorâmica. A figura a seguir sintetiza a conversão do sistema de coordenadas esférico para o sistema de coordenadas bidimensional u, v.

Figura 11 - Transformação para o sistema de coordenadas (u, v)



Fonte: Elaborada pela autora (2021)

Este trabalho utilizará apenas esse tipo de panorâmica, logo, toda menção posterior a palavra “panorâmica” estará se referindo a panorâmica equiretangular.

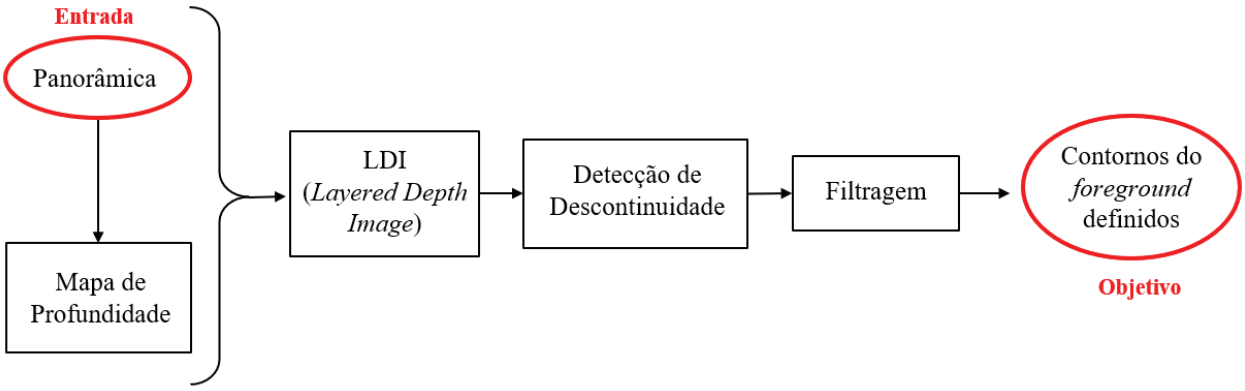
3 PRÉ-PROCESSAMENTO

Neste capítulo será abordada a etapa de pré-processamento da panorâmica equiretangular inserida na entrada. Este passo é fundamental para a preparação dos dados iniciais que serão inseridos no processo de *inpainting,* não apenas pela questão estrutural das informações, mas também para garantir um alto desempenho do algoritmo. As sub-etapas que serão descritas a seguir seguem a proposta do artigo *"3D Photography Using Context-Aware Layered Depth Inpainting"* escrito por M. -L. Shih et al. (12).

Ao ser inserida a fotografia panorâmica desejada, o primeiro passo é a obtenção de um mapa de profundidade que represente o ambiente, contendo a relação das distâncias dos objetos presentes na cena. Com ambos os dados, é possível construir uma estrutura denominada de LDI (*Layered Depth Image*), que é uma imagem em camadas produzida a partir dos dados de profundidade combinados com as informações dos pixels da imagem panorâmica de entrada. A partir da LDI detectam-se os limites de descontinuidades que se localizam entre a parte frontal e a parte remanescente da estrutura. O processo de detecção, porém, é imperfeito, apresentando alguns ruídos inconvenientes na delimitação do contorno dos objetos, por isso é acrescida uma etapa final de filtragem, capaz de gerar bordas homogêneas. Como resultado final do pré-processamento, pretende-se obter o delineamento das regiões que compõem o plano frontal da imagem, nomeado de *foreground*, diferenciando-o do plano de fundo, denominado de *background.*

A fim de facilitar o entendimento, a Figura 12 apresenta um fluxograma com as sub-etapas presentes neste tópico.

Figura 12 - Fluxograma



Fonte: Elaborada pela autora (2021)

As seções seguintes serão divididas para abranger os passos aqui descritos. A seção 3.1 discorrerá sobre a obtenção do mapa de profundidade, e a seção 3.2 será destinada a formação da LDI, enquanto as seções 3.3 e 3.4 abordarão a detecção de descontinuidades e a filtragem, respectivamente.

A fim de facilitar a exemplificação das etapas de pré-processamento, será utilizado como entrada a fotografia mostrada na Figura 13, uma imagem não panorâmica contendo apenas um objeto, que tornará possível a observação detalhada de cada parte do processo. O análogo ocorrerá quando for inserida uma fotografia contendo um panorama.

Figura 13 - Imagem exemplo

Fonte: Elaborada pela autora (2021)

3.1 EXTRAÇÃO DO MAPA DE PROFUNDIDADE

A extração do mapa de profundidade sobre a panorâmica equiretangular de entrada é realizada através do modelo MiDaS (*Monocular Depth Estimation*). Este modelo foi desenvolvido por K. Lasinger et al.(16), para calcular a profundidade inversa relativa a partir da inserção de uma imagem no padrão de cores RGB, do inglês, *Red, Green, Blue*. O algoritmo é capaz de processar imagens com diferentes dimensões, o que o tornou viável para a aplicação não apenas em imagens convencionais, mas também sobre panorâmicas.

Sua alta eficácia foi obtida através do treinamento aplicado sobre 10 conjuntos distintos de dados, ReDWeb, DIML, Movies, MegaDepth, WSVD, TartanAir, HRWSI, ApolloScape, BlendedMVS e IRS. Todas as aplicações foram baseadas na otimização multiobjetivo, com a finalidade de garantir a obtenção de resultados de alta qualidade.

A imagem de entrada em formato RGB possui 3 canais ($3\ x\ altura\ x\ largura$), onde cada canal é formado por uma cor (vermelho, verde ou azul) e a respectiva altura e largura da fotografia inserida. A fim de garantir o desempenho eficiente do MiDaS, sugere-se adotar os seguintes padrões de normalização para cada canal.

$$Média\ (\mu)\ = [\ 0.485, 0.456, 0.406\ ] \tag{8}$$

$$Desvio\ Padrão\ (\sigma)\ = [\ 0.229, 0.224, 0.225] \tag{9}$$

Desta forma, os dados inseridos respeitarão as condições de treinamento ao qual foram estipuladas para a criação do modelo. Além disso, como o algoritmo fornece ainda uma transformação personalizada para o redimensionamento da imagem, outra condição deve ser garantida para que a proporção da entrada original seja mantida. Portanto, a altura e a largura precisam ser divisíveis por 32 para garantir resultados ideais que produzam valores próximos de 384, isto porque este foi o valor utilizado para a resolução durante o treinamento.

Inserindo a fotografia mencionada no início do capítulo, obtém-se o mapa de profundidade da Figura 14, produzido pelo MiDaS.

Figura 14 -Mapa de profundidade

Fonte: Elaborada pela autora (2021)

O conjunto de pixels em tons mais escuros representam visualmente o *background*, enquanto os pixels mais claros referem-se ao *foreground*. Em outras palavras, tem-se que as partes destacadas em tons de branco representam menores profundidades, enquanto as porções em tons de preto apresentam maiores profundidades. A diferença dentro do mesmo subconjunto

de tonalidades também evidência distinções entre as profundidades de cada parte do objeto, resultando em uma proximidade maior conforme clareiam-se os tons e uma proximidade menor à medida que são escurecidos. O exemplo refere-se a um único objeto, porém, para múltiplos objetos, a estrutura se mantém.

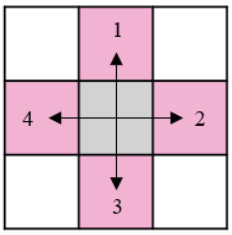
3.2 LAYERED DEPTH IMAGE – LDI

Com a panorâmica equiretangular em padrão RGB mais o mapa de profundidade gerado pelo modelo MiDaS é possível criar uma estrutura em camadas denominada de LDI (*Layered Depth Image*) (10). Esta estrutura é um tipo de representação compacta de imagens em multiplanos, que pode ser rígida ou flexível em relação ao número de camadas. Quando rígida, os pixels são alocados de maneira fixa e pré-determinada. Aqueles que apresentam menores profundidades dentro de um intervalo ε são colocados na primeira camada, e conforme aumentam as profundidades, as demais camadas são preenchidas até alcançar o maior valor quantitativo, de forma a não extrapolar o número de camadas definido. Entretanto, quando flexível, não possuem quantidade de camadas pré-definidas, ou seja, este tipo de estrutura se adapta conforme a complexidade dos dados de profundidade, montando suas camadas de forma arbitraria.

Todavia, para a detecção de descontinuidades, não é favorável a utilização de uma LDI rígida, devido a presença de várias camadas com quantidades diversificadas de pixels, as quais resultam mudanças abruptas sucessivas entre as camadas. Portanto, para essa aplicação, será adotada uma LDI flexível, onde cada camada poderá receber qualquer quantidade de pixels. Cada pixel LDI contém o valor de cor juntamente com o valor de profundidade, que serão localmente conectados no sistema *4-connected*.

A ligação *4-connected* aplicada sobre a LDI é realizada de forma similar a uma imagem convencional. Cada pixel armazena ponteiros em direção aos pixels vizinhos, dependendo da vizinhança, este pode não possuir ponteiros ou possuí-los variando de 1 até no máximo 4. As direções a serem apontadas se dividem em direita, esquerda, acima e abaixo, como mostra a Figura 15.

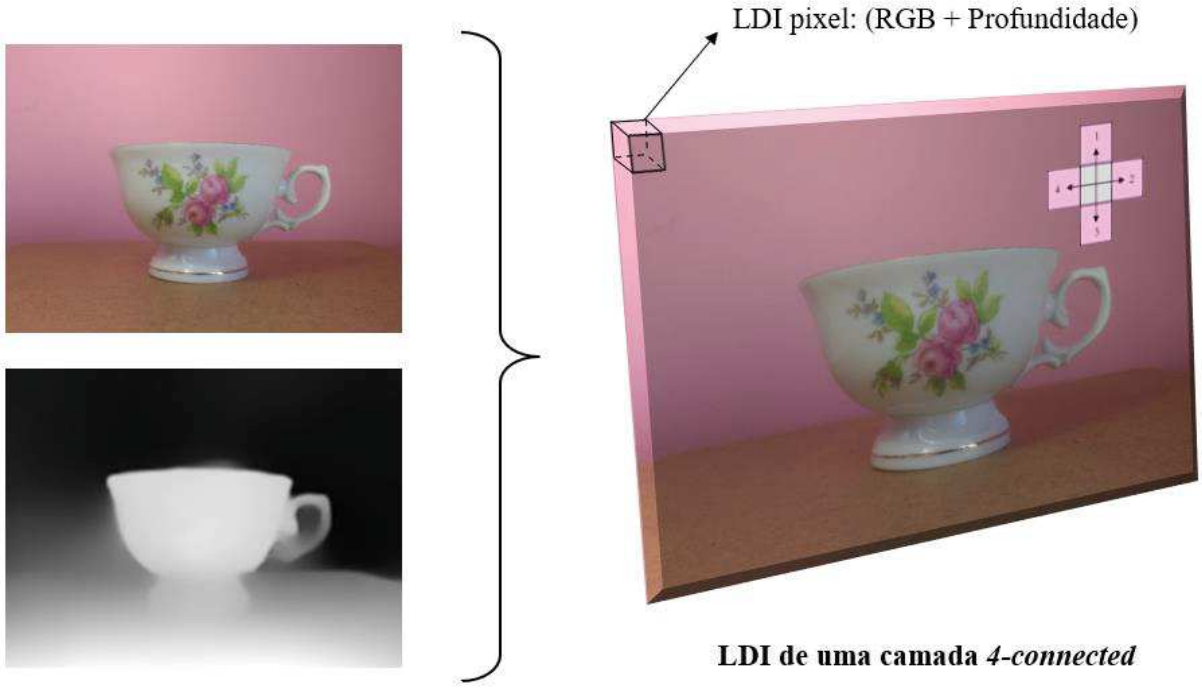
Figura 15 - Conectividade do tipo 4-connect



Fonte: Elaborada pela autora (2021)

Cria-se então uma LDI flexível, inicializada apenas com uma camada, a qual terá todos os seus pixels inteiramente conectados no modelo *4-connected.* As cores serão obtidas através dos pixels da panorâmica equiretangular RGB inserida enquanto as profundidades serão retiradas do mapa de profundidade normalizado. Essa normalização é realizada sobre o mapa de profundidade obtido do MiDaS, alterando os valores de mínima e máxima disparidade para 0 e 1 respectivamente, e modificando as demais profundidades para satisfazerem esse intervalo. A Figura 16 a seguir sintetiza o processo descrito.

Figura 16 - LDI com conectividade *4-connected*



Fonte: Elaborada pela autora (2021)

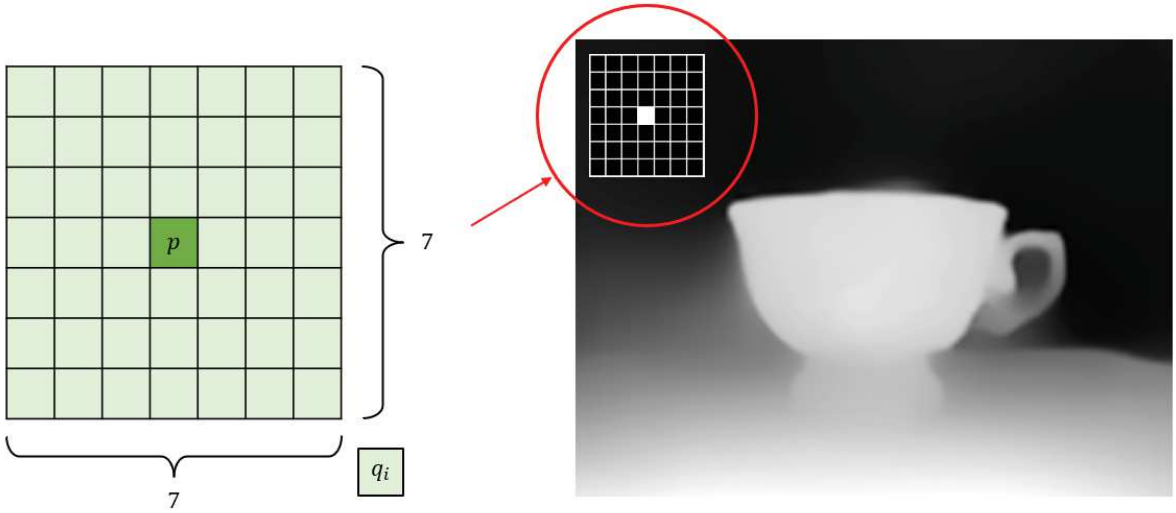
3.3 DETECÇÃO DE DESCONTINUIDADE

A partir da LDI inicial formada, deseja-se desmembrar a região que compõe o *foreground* do restante da imagem. Para isso é necessário delimitar os pontos de transição entre os pixels, detectando os momentos onde ocorrem descontinuidades na profundidade. Essa detecção é composta por duas etapas, a filtragem, responsável por lapidar o mapa de profundidade, e o cálculo da diferença de disparidade, que realizará a marcação dos contornos evidenciados.

O mapa de profundidade gerado pelo MiDaS é confeccionado suavizando as áreas de mudança de profundidade, o que dificulta a detecção de descontinuidades de maneira precisa e eficiente. Portanto, é necessário realizar uma filtragem que transforme essas regiões "borradas" em fronteiras de fácil identificação e, para isso, aplica-se então o Filtro Bilateral.

Esse método de filtragem reduz o ruído e preserva as bordas da imagem, substituindo a intensidade de cada pixel p por uma média ponderada dos q_i pixels vizinhos presentes em uma janela de nxn pré-estabelecida. Define-se então uma janela de dimensão $7x7$ para realizar a varredura.

Figura 17 - Janela $7x7$ do filtro Bilateral



Fonte: Elaborada pela autora (2021)

Considere S o conjunto formado pelas possíveis posições dos pixels vizinhos q_i . Deste modo a intensidade I do pixel p pode ser definida pela seguinte equação:

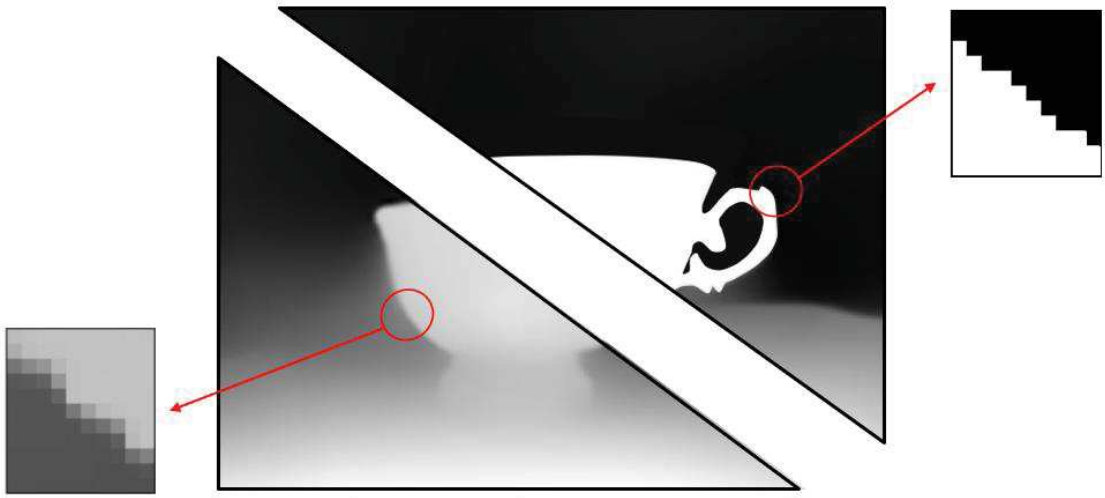
$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(||p - q||)\, G_{\sigma_r}(|I_p - I_q|)\, I_q \quad (10)$$

Onde o W_p é a média ponderada dos pixels vizinhos calculada por:

$$W_p = \sum_{q \in S} G_{\sigma_s}(||p - q||)\, G_{\sigma_r}(|I_p - I_q|) \quad (11)$$

A equação além de considerar a variação de intensidade $I_p - I_q$, responsável pela preservação da borda, ainda leva em consideração a diferença radial entre os pixels, representada por *p*-*q*. Além disso, utilizam-se os parâmetros σ_s e σ_r aplicados a uma distribuição Gaussiana para controlar a quantidade de filtragem a ser realizada. A componente espacial σ_s controla a influência do distanciamento entre os pixels sobre o peso W_p a ser atribuído, penalizando as maiores distâncias. Enquanto a componente de alcance σ_r controla a influência da diferença de intensidades entre os pixel*s*, penalizando os pixels vizinhos com variação de intensidade elevadas. Para este trabalho, utilizou-se $\sigma_s = 4{,}0$ e $\sigma_r = 0{,}5$.

Figura 18 - Resultado da filtragem no mapa de profundidade



Fonte: Elaborada pela autora (2021)

Após aprimorar o mapa de profundidade (Figura 18), encontram-se as descontinuidades calculando a diferença de disparidade entre os pixels vizinhos. Deste modo, dois pixels serão descontínuos se a diferença de disparidade entre eles for superior a 0,04, demarcando então transições que apresentam variações abruptas. Esse processo além de delinear as bordas,

também realiza falsas marcações, resultando em manchas isoladas e segmentos curtos avulsos ou “pendurados” (Figura 19).

Figura 19 - Resultado final da filtragem utilizando o Filtro Bilateral

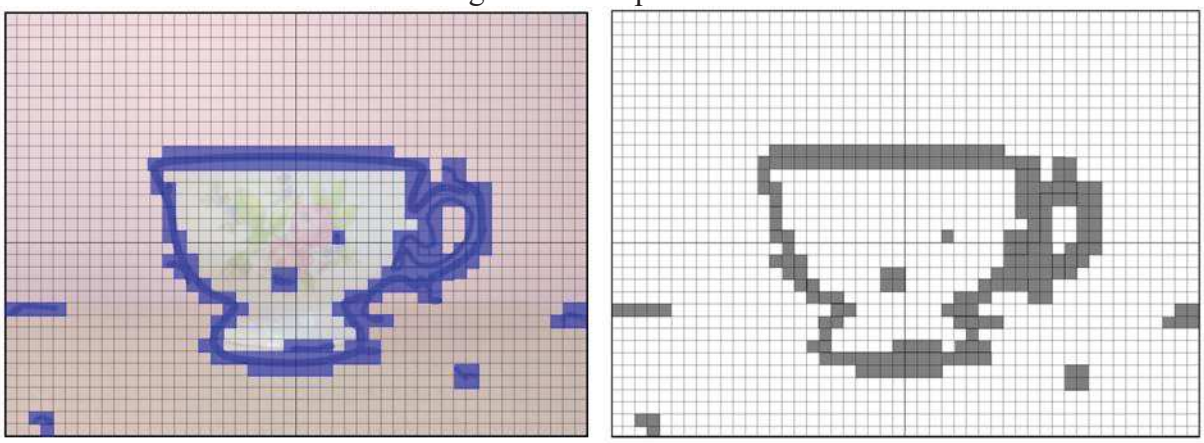
Fonte: Elaborada pela autora (2021)

3.4 FILTRAGEM

Esta última etapa do pré-processamento tem a finalidade de remover os contornos errôneos, de tal maneira que as marcações remanescentes representem coerentemente a silhueta do objeto a ser destacado. Para executar essa “limpeza” serão necessárias 3 fases: a criação de um mapa binário, a junção das descontinuidades selecionadas em grupos distintos e a extração de dados indesejados.

O mapa binário é construído a partir das marcas evidenciadas pela etapa de detecção de descontinuidades, atribuindo o valor 1 para os pixels marcados como descontinuidades e 0 para os demais pixels da imagem. Na Figura 20 é exemplificado o processo de criação deste mapa, sendo representados, em cinza, os pixels classificados com o valor 1 e, em branco, os pixels classificados com valor 0. O quadriculado que simula a divisão entre os pixels foi realizado de forma exagerada com o propósito de facilitar a compreensão, portanto, a aplicação original atinge precisões mais elevada, percebendo por exemplo os contornos que compõem a alça da xícara representados de forma separada.

Figura 20 - Mapa binário



Fonte: Elaborada pela autora (2021)

A partir do mapeamento, agrupam-se então os pixels em conjuntos separados, onde essa junção é realizada através dos vizinhos adjacentes, de modo que formem uma estrutura interligada capaz de representar uma partição da borda de descontinuidade, ou seja, segmentos de contorno. Entretanto, essa conectividade deverá ser realizada de maneira localizada para evitar a formação de um contorno único, isso porque a junção entre os conjuntos de bordas pode dificultar a identificação dos dados errôneos. Deste modo, separam-se essas junções conforme a conectividade local expressa pela estrutura LDI, dividindo a formação de segmentos por regiões de profundidade da imagem. A Figura 21 apresenta este processo de agrupamento, assim como a escolha por áreas realizadas através da LDI. Porém, é válido frisar que a representação é meramente ilustrativa, retratando a divisão de forma grosseira e imprecisa.

Figura 21 - Agrupamento dos pixels adjacentes

Fonte: Elaborada pela autora (2021)

Selecionados os segmentos, executa-se o estágio de remoção dos dados indesejados, onde considera-se apto para extração os conjuntos cujo tamanho seja inferior a 10 pixels, independentemente da sua localização na imagem. A definição do parâmetro de tamanho foi definida a partir de aplicações realizadas sobre 50 amostras aleatórias do conjunto de treinamento *"Real Estate 10K*".

Figura 22 - Definição dos segmentos de pixels

Fonte: Elaborada pela autora (2021)

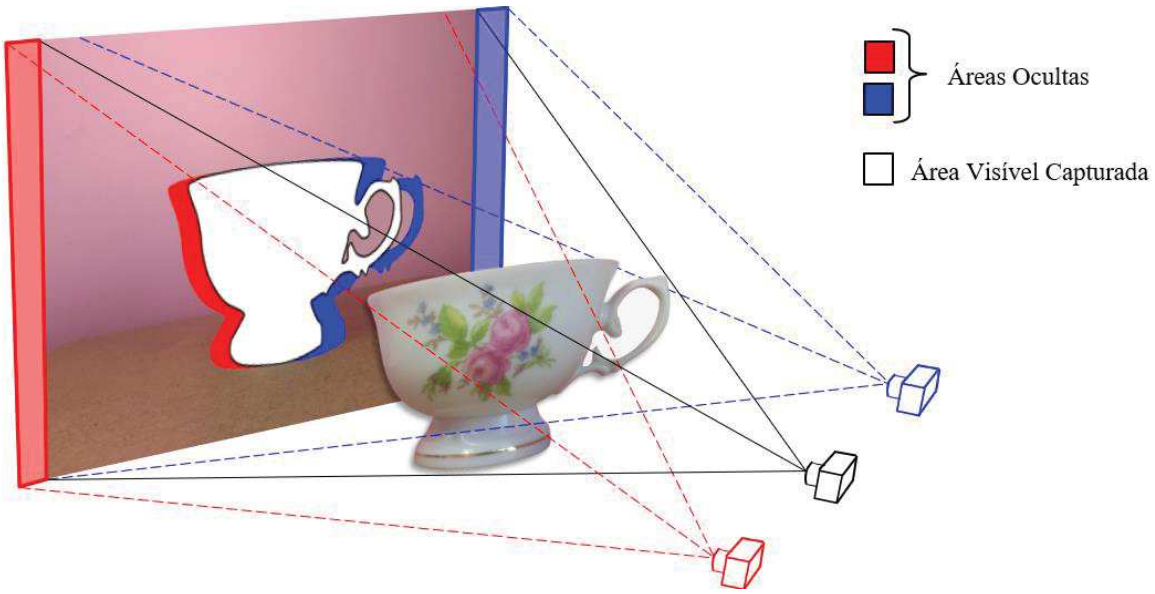
Logo, como resultado final da etapa de pré-processamento, têm-se contornos bem definidos (Figura 22), subdivididos em segmentos isolados que compõem a estrutura base para iniciar o processo ulterior de *inpainting*.

4 INPAINTING

A técnica que realiza o preenchimento de lacunas é denominada *inpainting,* nomenclatura advinda da língua inglesa, que se refere ao conceito de pintura interna, em outras palavras, a pintura aplicada como forma de conservação ou reparação. Seu objetivo principal é a reconstrução de dados danificados ou ausentes, de modo a simular de forma realística a retratação dos dados originais. Normalmente, esse processo se baseia nas informações fornecidas por dados vizinhos locais, onde é possível realizar médias e previsões para a inserção dos novos elementos na estrutura. A empregabilidade deste método é elevada, abrangendo campos além das pinturas, como é o caso das fotografias, digitais ou físicas, das estruturas tridimensionais e dos vídeos.

A implementação dessa técnica pode ser realizada por diversas metodologias com diferentes objetivos. Para este trabalho, o *inpainting* será aplicado com a finalidade de preencher as áreas de oclusão localizadas entre o *foreground* e o *background* de uma imagem. O surgimento das áreas ocultas ocorre com a mudança do centro óptico do observador, conforme demonstra a Figura 23. Suponha que a parte frontal da imagem tenha sido desmembrada do restante e aproximada do observador que permanece na mesma posição onde a fotografia foi capturada, neste caso, é possível identificar todos os detalhes antes visíveis na imagem. Porém, se o observador se move para a direita ou para a esquerda e realiza a mesma observação, não conseguirá visualizar todos os detalhes referentes ao *background* nas proximidades do objeto deslocado, uma vez que esses dados não foram retratados pela captura original.

Figura 23 - Áreas de oclusão



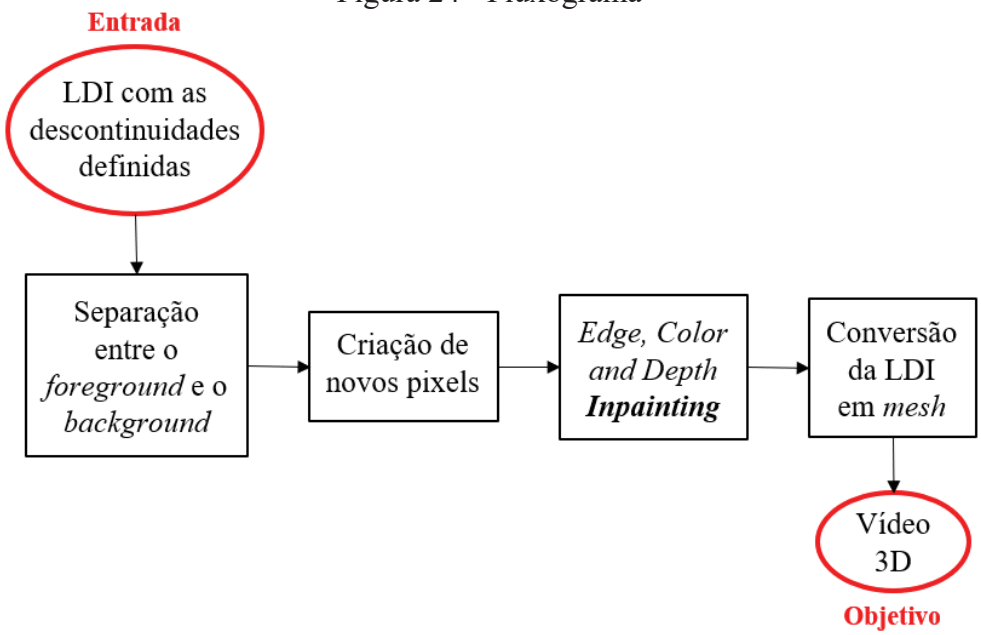
Fonte: Elaborada pela autora (2021)

A metodologia adotada para o preenchimento dessas lacunas é baseada em uma rede neural convolucional, também proposta pelo artigo *"3D Photography Using Context-Aware Layered Depth Inpainting"* de M. -L. Shih et al. (12). O método adotado realiza o processo de *inpainting* através de 3 sub-redes: *Edge Network*, *Color Network* e *Depth Network*, as quais são aplicadas respectivamente sobre o contorno, a cor e a profundidade do *background* da imagem.

Entretanto, antecedente a aplicar a rede neural sobre as áreas de oclusão há a execução de dois estágios: a separação do *foreground e* a criação de novos pixels no *background*. A seção 4.1 será dedicada ao primeiro estágio, que realizará o "corte" do plano frontal delimitado pelo contorno obtido ao final do pré-processamento. Enquanto a seção 4.2 abordará a complementação do *background* através de um processo iterativo baseado nas informações dos pixels vizinhos.

Deste modo a seção 4.3 será destinada aos conteúdos envolvendo a rede neural convolucional utilizada, apresentando detalhadamente cada uma das sub-redes que a compõem. Além disso esse capítulo ainda abordará, na seção 4.4, a construção do vídeo tridimensional a partir da conversão da LDI final em uma *mesh* levemente texturizada. O fluxograma da Figura 24 sintetiza as etapas que serão abrangidas neste tópico.

Figura 24 - Fluxograma

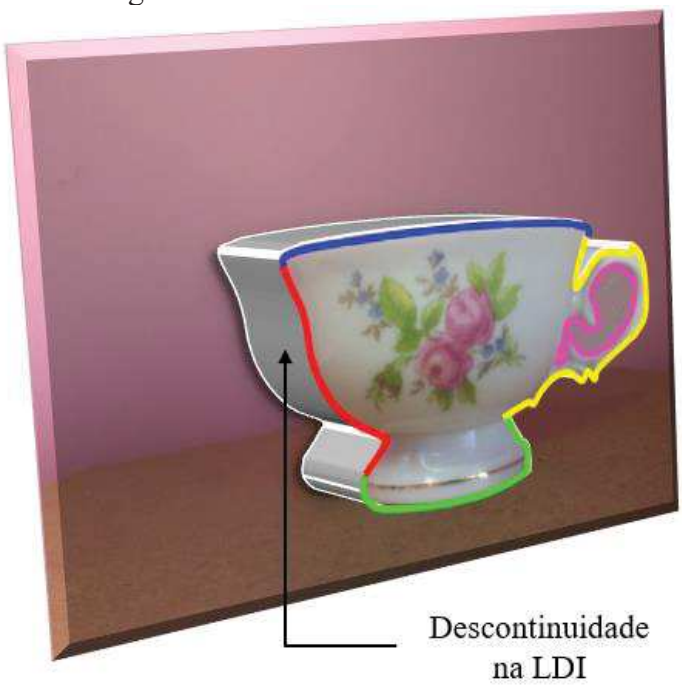


Fonte: Elaborada pela autora (2021)

4.1 SEPARAÇÃO ENTRE O FOREGROUND E O BACKGROUND

Ao aplicar os contornos obtido sobre a LDI inicial 4-connected, é possível definir os pontos que contenham descontinuidades. A Figura 25 representa esta evidenciação simbólica na estrutura.

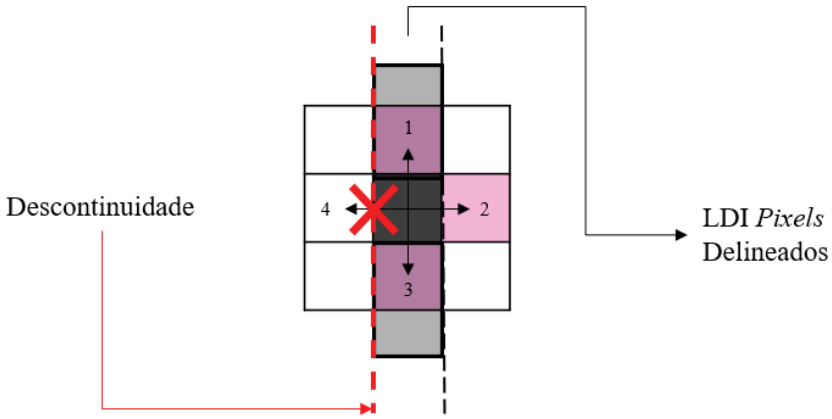
Figura 25 - Descontinuidades na LDI



Fonte: Elaborada pela autora (2021)

Para realizar a separação entre o *foreground* e o *background*, desconecta-se o elo entre o LDI pixel pertencente a fronteira demarcada e seu pixel vizinho cuja ligação esteja sobre a descontinuidade (Figura 26).

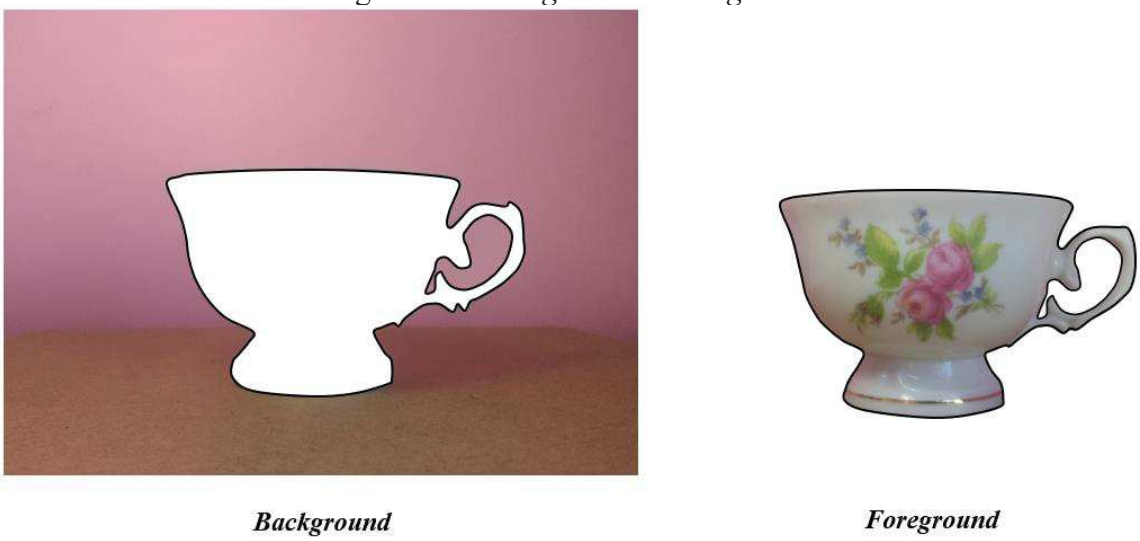
Figura 26 – Desconexão dos LDI pixels vizinhos



Fonte: Elaborada pela autora (2021)

Os LDI pixels que passam pelo processo de desconexão são nomeados de *Silhouette* Pixels, devido a silhueta delineada que formam ao redor de determinado objeto. Deste modo, após a realização do desmembramento entre os planos, existirão pixels de silhueta tanto no *foreground* quanto no *background,* pois os vizinhos que foram desconectados dos LDI pixels presentes na borda, também possuíram um vizinho a menos. A Figura 27 mostra a separação executada entre o *foreground,* a direita e o *background,* a esquerda, juntamente a marcação dos pixels de silhueta representados pelos contornos em cor preta.

Figura 27 - *Background* e *Foreground*



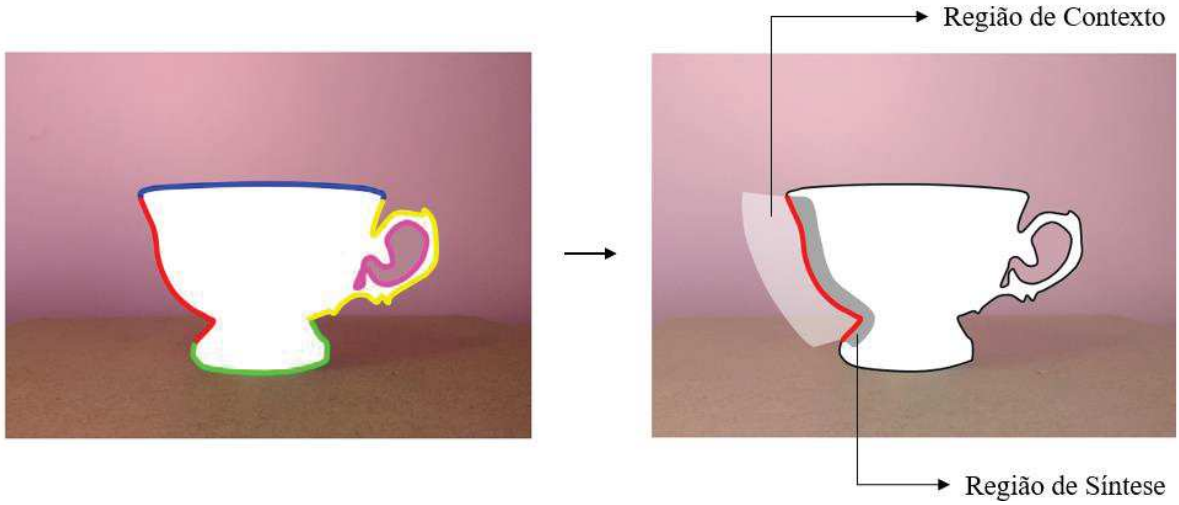
Fonte: Elaborada pela autora (2021)

4.2 CRIAÇÃO DE NOVOS PIXELS

A técnica de *inpainting* será aplicada apenas sobre o *background* da imagem, portanto a criação de novos pixels se estenderá através da área de oclusão partindo dos pixels de silhueta. Essa aplicação será dividida conforme os contornos estabelecidos pelos segmentos de descontinuidade, fornecendo a cada partição um preenchimento individualizado.

A construção desses novos pixels baseia-se na delimitação de duas regiões definidas no entorno dos pixels de silhueta, a região de contexto e a região de síntese. A primeira delas é formada pelos LDI pixels existentes no *background,* os quais fornecerão posteriormente os dados necessários para construir a segunda região, composta pelos novos LDI pixels. A Figura 28 apresenta as áreas de contornos sobre os pixels de silhueta, assim como a representação da região de contexto e da região de síntese.

Figura 28 - Região de contexto e região de síntese



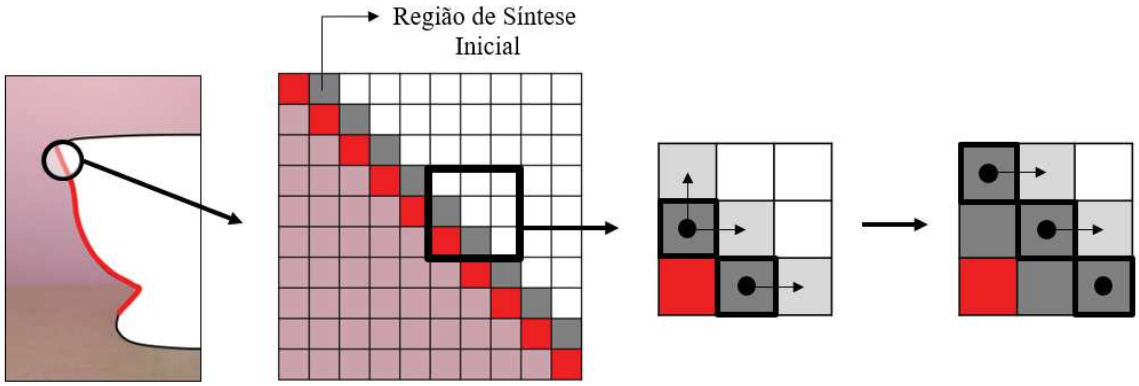
Fonte: Elaborada pela autora (2021)

Ambas as regiões são formadas a partir de algoritmos iterativos simples, os quais se baseiam na criação simultânea de pixels a partir dos criados na iteração anterior, simulando um efeito cascata. Embora utilizam a mesma abordagem de criação, as regiões de síntese e contexto, utilizam algoritmos diferentes, com número de iterações e campos de abrangência distintos.

O algoritmo responsável pela criação da região de síntese, inicia-se através da construção de uma região de síntese inicial, a qual é formada a partir dos pixels de silhueta estendendo-se apenas um passo em direção ao centro da região de oclusão. A partir dos pixels

definidos nessa área inicial, outros pixels são criados de forma iterativa, sempre adentrando a região de oclusão, jamais de forma retroativa. Desta maneira, durante 40 iterações, cada pixel criado pode se expandir para a direita, para a esquerda, para cima e para baixo, estabelecendo neste momento apenas as coordenadas 2D. A Figura 29 retrata este processo.

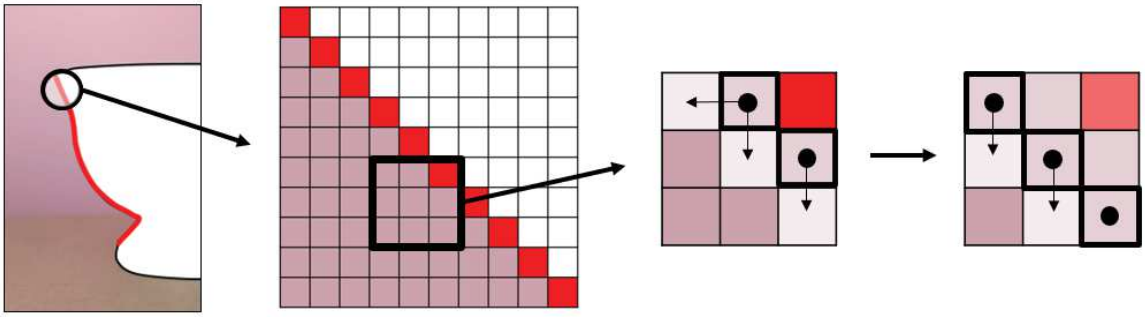
Figura 29 - Preenchimento iterativo da região de síntese



Fonte: Elaborada pela autora (2021)

A região de contexto, é definida por um algoritmo iterativo similar (Figura 30), que ao invés de fornecer novas localizações bidimensionais, seleciona os LDI pixels já existentes, com seus conteúdos e suas conectividades. Porém, altera-se o número de iterações, elevado de 40 para 100, pois o preenchimento da região de síntese apresenta melhores resultados quando baseados em regiões de contexto mais abrangentes.

Figura 30 - Preenchimento iterativo da região de contexto

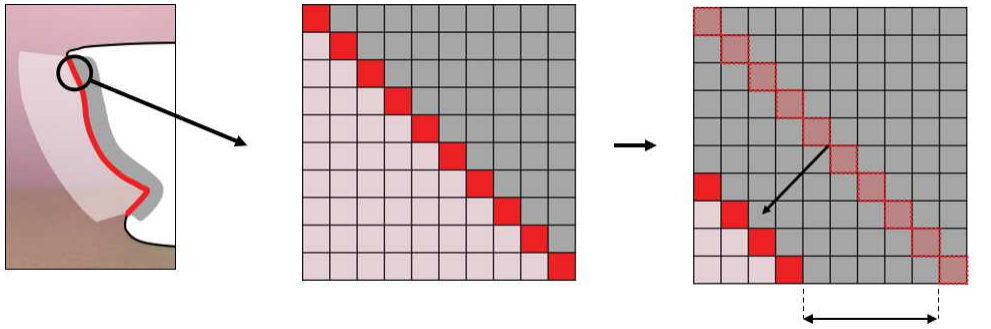


Fonte: Elaborada pela autora (2021)

Os algoritmos funcionam de forma conjunta, porém alternada, ora expande-se a região de contexto, ora expande-se a região de síntese. Desta maneira tem-se a garantia de que nenhum pixel pertencerá simultaneamente aos dois conjuntos. Finalizado esse processo de definição das regiões, realiza-se o deslocamento do segmento composto pelos pixels de silhueta em 5 pixels

na direção da região de contexto, aumentando assim a região de síntese (Figura 31). Essa alteração é necessária porque a demarcação desses contornos é construída com base em estimações imperfeitas de profundidades, o que fornece divergências em relação a realidade e atrapalha o conteúdo do preenchimento inicial da região de síntese, causando transições não homogêneas.

Figura 31 - Deslocamento em 5 pixels



Fonte: Elaborada pela autora (2021)

Analogamente, o mesmo processo de criação de novos LDI pixels se repete para cada uma das bordas definidas no pré-processamento, como mostra a Figura 32.

Figura 32 - Regiões de contexto e de síntese em outros segmentos

Fonte: Elaborada pela autora (2021)

4.3 EDGE, COLOR AND DEPTH INPAINTING

Esta etapa tem o objetivo de fornecer os valores de cor e profundidade para os LDI pixels definidos na região de síntese, utilizando como parâmetro de conteúdo a região de contexto demarcada. Para executar este preenchimento, será adotada a utilização de uma rede neural convolucional, em função da sua capacidade de prever dados desconhecidos.

Apesar da utilização de uma estrutura LDI, o conteúdo de seus pixels é constituído por formatos conhecidos de cor e de profundidade, deste modo, as regiões delimitadas serão formadas também por estruturas conhecidas, ou seja, uma matriz multidimensional RGB com 3 canais para representar a coloração e uma matriz bidimensional para representar a profundidade. Portanto, ao realizar uma análise local sobre a LDI, percebe-se que esta se assemelha a estrutura de uma imagem convencional, seja ela multidimensional ou bidimensional. Desta maneira, é possível então aplicar arquiteturas convencionais de redes neurais convolucionais voltadas para imagens.

Porém, a aplicação dessa rede neural convolucional deverá atender tanto o preenchimento de cor quanto o de profundidade, de forma alinhada e coerente, o que não seria possível se fossem realizadas de maneira individualizada. Portanto, como meio de atender a essa exigência, subdivide-se essa rede neural convolucional em 3 sub-redes, cada qual com a finalidade de atender a uma demanda especifica. Essas subdivisões são nomeadas como *Edge Network*, *Color Network* e *Depth Network*, e tratam, respectivamente, de informações relacionadas as bordas, a coloração e a profundidade.

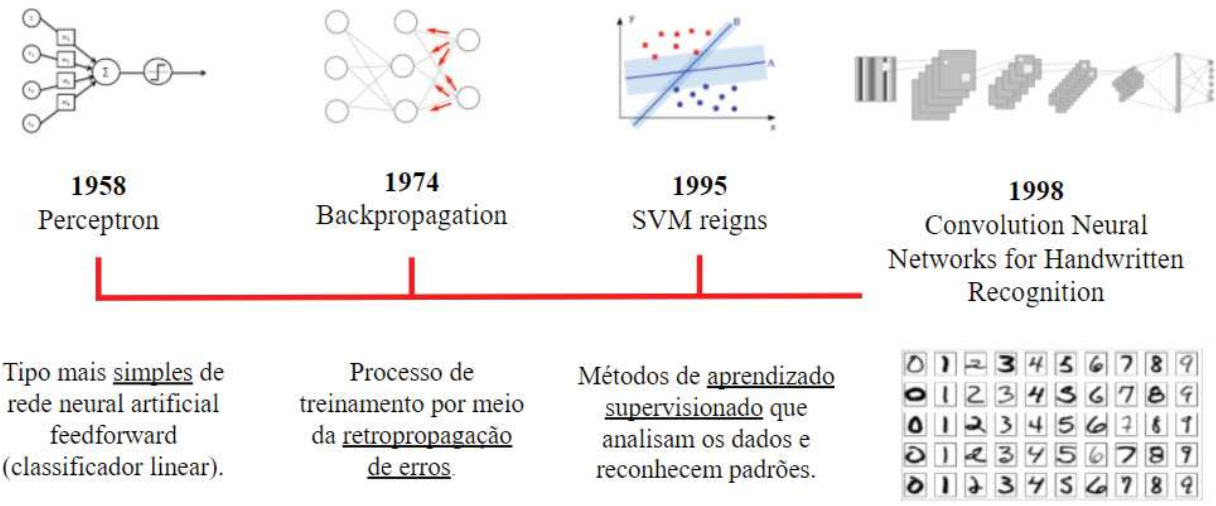
Esta seção será composta por 5 subseções para abordar a construção desta rede neural, a subseção 4.3.1 apresentará uma parte conceitual fundamental para o decorres das demais subseções, pois introduzirá a teoria relativa à rede neural convolucional. A subseção 4.3.2 discorrerá sobre a sub-rede aplicada aos contornos, responsável por interligar as outras duas sub-redes. A subseção 4.3.3, abordara as sub-redes de cor e a de profundidade, detalhando dados na arquitetura e a metodologia da integração em dois estágios. Por fim, a subseção 4.3.4 mostrará a integração entre as 3 sub-redes, enquanto a subseção 4.3.5 tratará sobre a necessidade de reaplicação da rede.

4.3.1 Rede Neural Convolucional

As redes neurais convolucionais, também nomeadas pela sigla CNN, do inglês *Convolution Neural Network,* são arquiteturas de aprendizado profundo construídas por meio de múltiplas camadas, com a finalidade de extrair características de um determinado dado inserido. Devido a análise realizada através da partição em conjuntos, essa arquitetura é capaz de lidar com uma enorme quantidade de dados, possuir várias camadas e um elevado número de parâmetros.

Essa arquitetura foi desenvolvida em 1998 para a diferenciação de algarismos escritos a mão, porém o primeiro esboço de uma rede neural é datado de 1958, com a criação do perceptron proposto por Frank Rosenblatt, que utilizava apenas a classificação linear denominada de *feedforward*. Assim como o perceptron, outras duas metodologias foram de extrema importância para a concepção das CNN: o *backpropagation* e os *SVM reigns.* O algoritmo de *backpropagation*, foi criado em 1974, propiciando o treinamento por retropropagação de erros, possibilitando à rede o aprendizado baseado nos erros obtidos durante o processo. E os métodos de aprendizado supervisionado, *SVM reigns*, datados de 1995, foram responsáveis por inserir a análise de dados e o reconhecimento de padrões. Uma linha do tempo abrangendo esses marcos é mostrada na Figura 33.

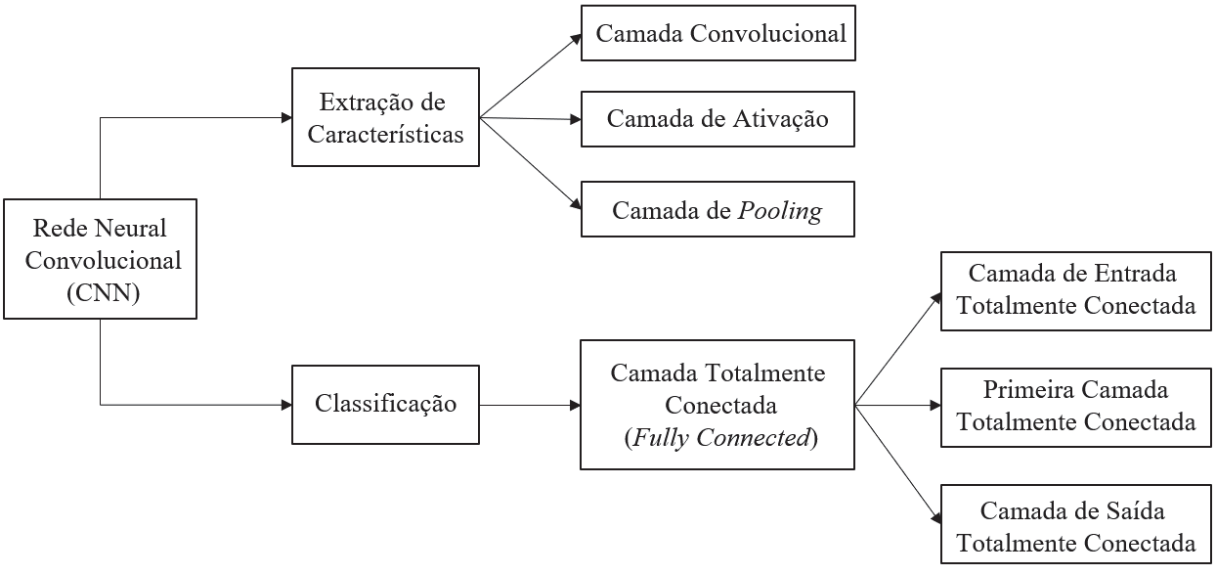
Figura 33 - Linha do tempo



Fonte: Elaborada pela autora (2021)

Uma CNN é dividida em duas grandes etapas: a extração de características e a classificação. A primeira etapa é responsável pela obtenção de características que auxiliem na distinção entre diferentes entradas inseridas e, é composta pelas seguintes camadas: Camada Convolucional, Camada de Ativação e Camada de *Pooling*. A segunda etapa, porém, é encarregada de analisar as características extraídas e classificá-las em diferentes saídas, sendo formada pelas Camadas Completamente Conectadas, também denominadas de *Fully Connected.* Essas camadas ainda podem ser divididas em Camada de Entrada Totalmente Conectada, Primeira Camada Totalmente Conectada e Camada de Saída Totalmente Conectada. Um resumo dessas divisões é apresentado no fluxograma da Figura 34.

Figura 34 - Fluxograma

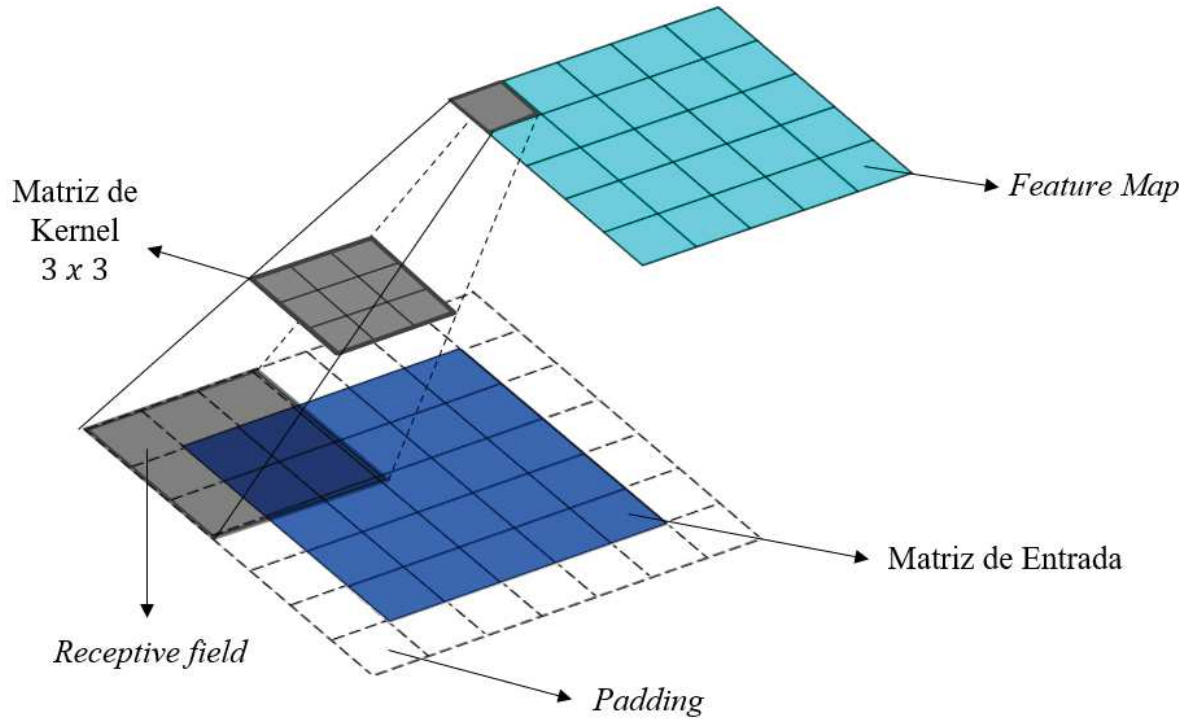


Fonte: Elaborada pela autora (2021)

Uma camada convolucional aplica uma operação de convolução sobre a entrada da rede ou sobre camadas anteriores, convertendo sub-regiões de dados (*receptive field*) em um dado único. Em outras palavras, as convoluções funcionam como filtros capazes de transformar um subconjunto de informações em um valor único. Esses filtros convolucionais são chamados de kernel e deslizam sobre toda a extensão da entrada formando um mapa de características, do inglês, *feature map* (Figura 35). O kernel pode variar em tamanho e em relação a composição dos pesos, podendo também possuir ou não o processo de *Padding,* o qual é caracterizado pela adição de alguns pixels ao redor da entrada antes da operação de convolução, normalmente

definidos com o valor zero (*zero-padding*), cuja finalidade é manter a dimensionalidade na saída resultante.

Figura 35 - Convolução

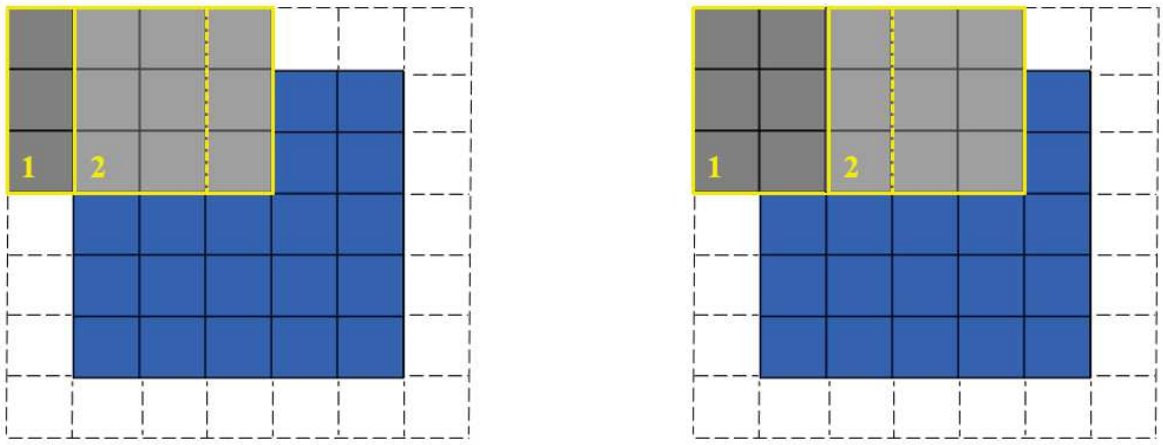


Fonte: Elaborada pela autora (2021)

Para uma matriz multidimensional como é o caso de uma imagem RGB, o processo é similar, cada canal de cor receberá a aplicação de uma matriz de kernel, que poderá ser igual ou distinta para cada canal. Deste modo, haverá a construção de três *features maps,* que deverão ser somados, elemento a elemento, para obter uma matriz bidimensional de características.

O kernel possuirá ainda modos diferentes de aplicação, variando quanto ao passo a ser percorrido durante a varredura e em relação a abrangência da área de cobertura utilizada pelo filtro. O parâmetro que define o passo a ser aplicado é nomeado de *stride,* deste modo um *stride* igual a 1 representa que o filtro aplicado se locomoverá avançando apenas um passo em direção ao próximo subconjunto. Deste modo, um *stride* igual a 2 representa dois passos que o filtro terá que percorrer para a próxima aplicação, tanto na vertical quanto na horizontal (Figura 36).

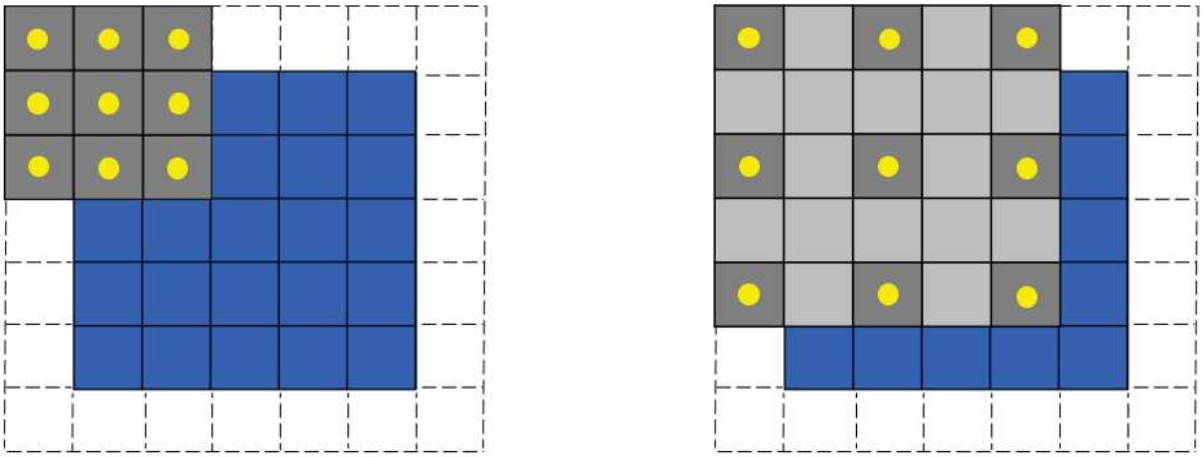
Figura 36 - *Stride*



Fonte: Elaborada pela autora (2021)

Quanto a abrangência da região de incidência do filtro, é definido o parâmetro de *dilatation* ou *dilatation rate*, que estabelecerá a taxa de dilatação permitida ao kernel para ampliar a sua cobertura durante o processo de varredura da entrada, o que permite a análise de um campo maior de informações com um custo computacional reduzido. A taxa de dilatação igual a 1, indica que não haverá dilatação na área de aplicação do filtro, portanto, um kernel de $3x3$ abrangerá uma área igualmente dimensionada de $3x3$. Para um valor de dilatação igual a 2, um kernel de $3x3$, abrangerá duas colunas extras e duas linhas extras, aumentando a *receptive field* para a dimensão $5x5$. Essas linhas e colunas serão inseridas de tal maneira que separem os dados que serão convolvidos pelo kernel. Deste modo, tanto o *receptive field* de 9 elementos quanto o de 25 elementos, terão apenas 9 elementos convoluidos pelo kernel. Portanto, uma taxa de dilatação igual a 2 resultará em um espaçamento igual a 1 entre os dados do *receptive field,* isso porque a dilatação igual a 1 representa que não ocorrerá dilatamento. Analogamente, uma taxa de dilatação igual a 3 resultará em um espaçamento igual 2 entre os dados e assim sucessivamente. A Figura 37 exemplifica a dilatação igual a 1 e 2 para a aplicação de um kernel de dimensão $3x3$.

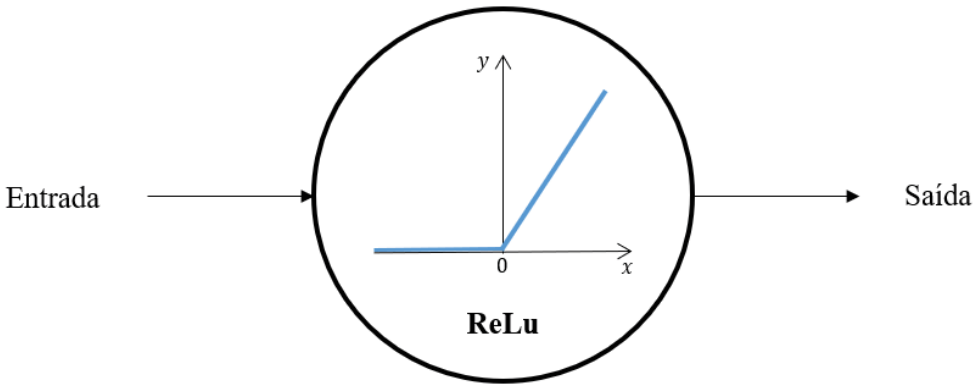
Figura 37 - Taxa de dilatação



Fonte: Elaborada pela autora (2021)

Na camada de ativação são utilizadas as chamadas funções de ativação, cujo propósito é fornecer não-linearidade a rede. Estas funções se localizam dentro de cada neurônio e são responsáveis por aplicar transformações sobre os dados recebidos (Figura 38). A mais utilizada delas é a de Unidade Linear Retificadora, comumente citada pela abreviatura ReLU, representada matematicamente por $y = \max(0, x)$. Essa equação garante que dados de entrada negativos sejam convertidos para zero e que valores positivos se mantenham inalterados.

Figura 38 - Função de ativação

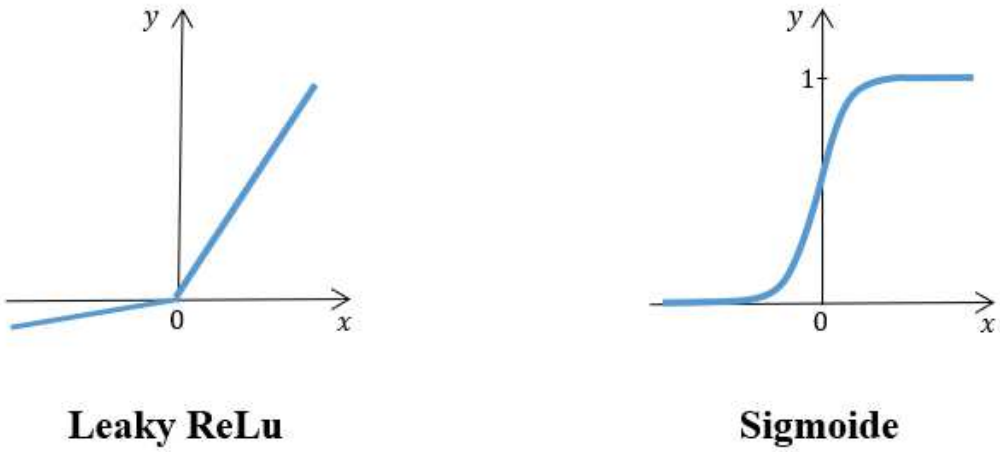


Neurônio

Fonte: Elaborada pela autora (2021)

Existem diversas funções de ativação além da ReLu, das quais é válido citar mais duas, que serão utilizadas nas seções posteriores, são elas: a Leaky ReLu e a Sigmoide. A Leaky ReLu é uma variação da função ReLu, matematicamente expressa por $y = \max(ax, x)$, para $0 < a < 1$, que ao invés de extinguir os valores negativos zerando-os, transformam estes em valores próximos de zero, porém mantendo a negatividade. A função Sigmoide, assim como a função ReLu, fornece apenas valores positivos na sua conversão, porém os mantem entre 0 e 1, podendo ser matematicamente expressa por $\sigma(x) = 1/(1 + e^{-x})$. Os gráficos referentes as duas funções de ativação podem ser observadas na Figura 39.

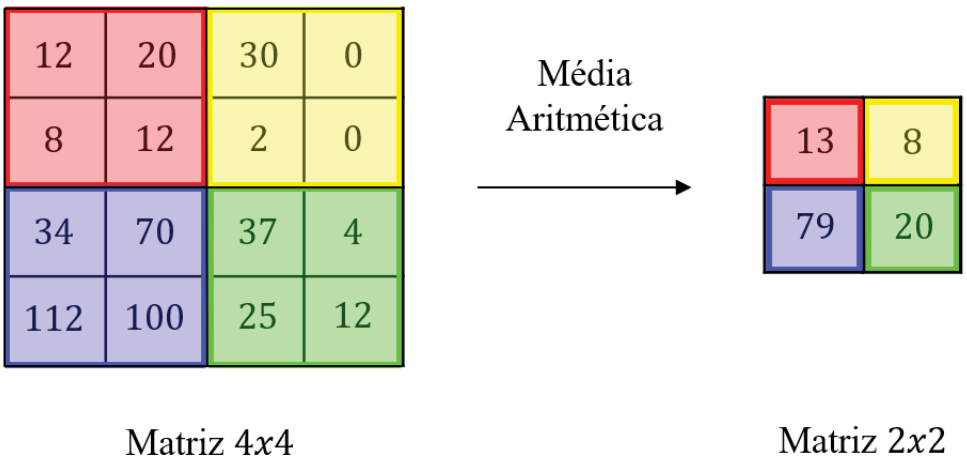
Figura 39 - Funções de ativação Leaky ReLu e Sigmoide



Fonte: Elaborada pela autora (2021)

A última camada que compõe a extração de características é a camada de *pooling (downsampling)*, sua função é reduzir a dimensionalidade dos dados na rede, produzindo uma espécie de resumo do mapa de características anterior. Essa redução é aplicada em subconjuntos e pode ser realizada de diferentes maneiras, como por exemplo, através da média (*avarage*), da mediana (*median*), do valor máximo (*max*) ou da norma do conjunto (*L2- pooling*). Para exemplificar o funcionamento desta camada, aplicou-se a média aritmética sobre uma matriz de dados aleatórios com dimensão $4x4$, em subconjuntos de 4 elementos (Figura 40).

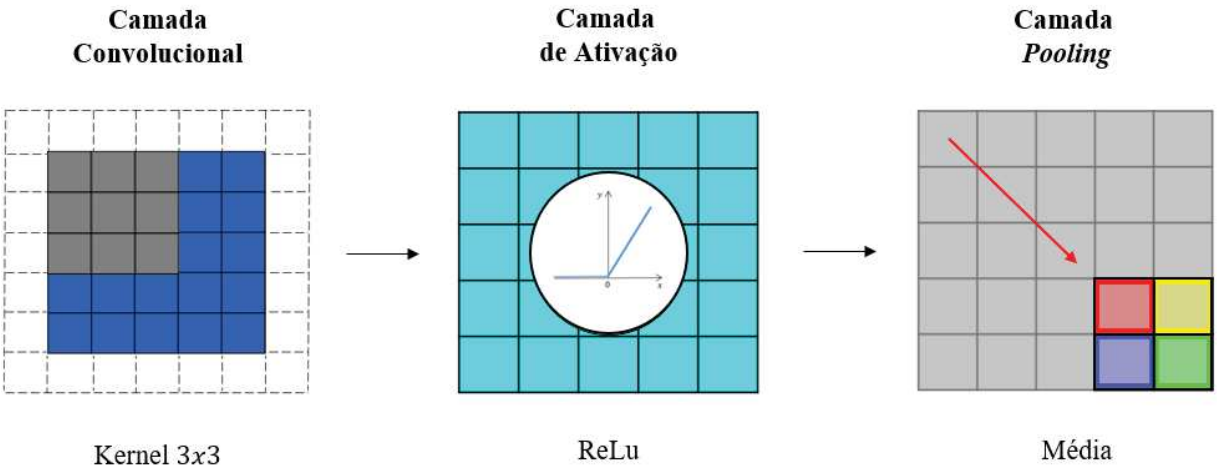
Figura 40 - *Pooling*



Fonte: Elaborada pela autora (2021)

Com a realização do *pooling*, encerra-se a etapa de extração de características, que pode ser resumida pela Figura 41, a qual apresenta a sequência de operações realizadas: a convolução, a aplicação da função de ativação, e o *downsampling*.

Figura 41 - Resumo da etapa de extração de características



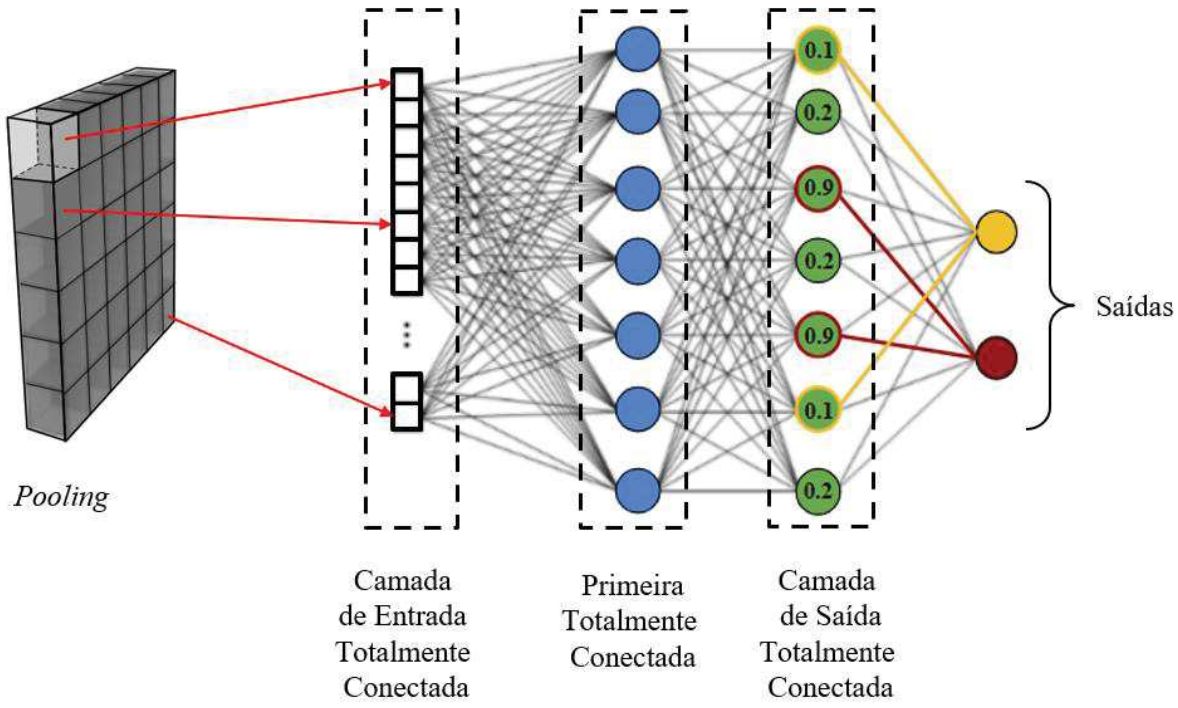
Fonte: Elaborada pela autora (2021)

Obtida as características, segue-se então a etapa de classificação, formada pela Camada Totalmente Conectada, a qual conecta todos os neurônios da camada anterior com os neurônios de saída que representam as classes a serem classificadas. Como dito anteriormente, essa camada é formada por mais 3 camadas que conduzirão esse processo de classificação, a primeira delas, nomeada de camada de entrada totalmente conectada, nivela a saída das

camadas anteriores e aloca os dados resultantes em um vetor. Cada valor que compõe esse vetor representa a probabilidade que um determinado recurso tem de pertencer a um rótulo específico.

A camada seguinte, chamada de primeira camada totalmente conectada, utiliza o vetor da camada anterior e aplica sobre ele diferentes pesos, com a finalidade de auxiliar no processo classificatório. Por fim, aplica-se a última das 3 camadas, a de saída totalmente conectada, que fornece as probabilidades finais para cada categoria previamente definida. A Figura 42 representa as 3 camadas que compõem a camada totalmente conectada da etapa de classificação descritas, assim como a ligação entre elas.

Figura 42 - Camada totalmente conectada



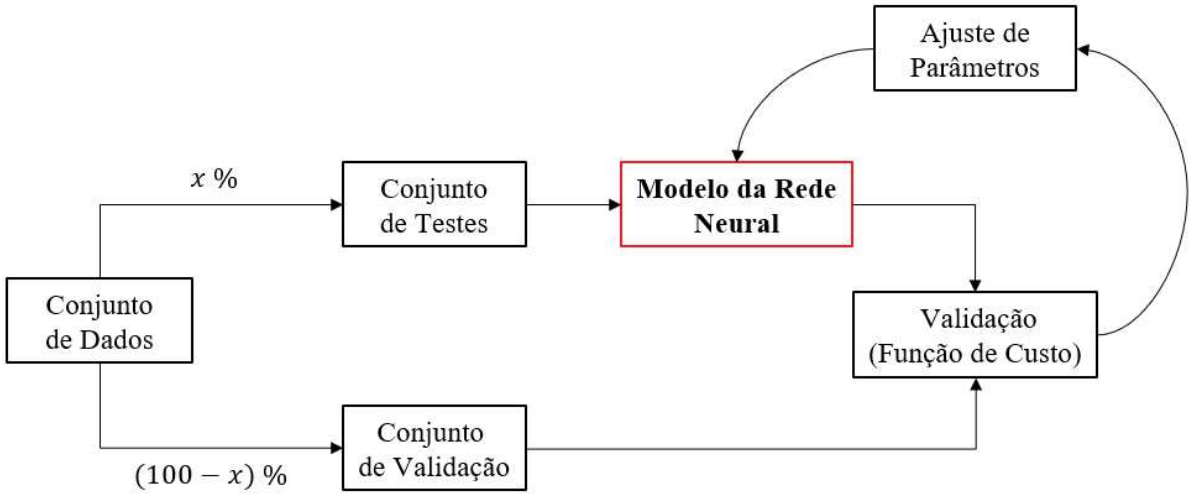
Fonte: Elaborada pela autora (2021)

Uma CNN pode apresentar mais de uma camada convolucional na etapa de extração, quando utiliza vários kernels, o que aumenta em igual quantidade os *features maps* produzidos. Esse conjunto de *features maps* produzidos em uma convolução é contabilizado como canais (*channel)*, por exemplo, uma camada convolucional que possua 64 *features maps,* possuirá 64 canais. Conforme a complexidade do problema podem haver também camadas convolucionais a partir de *features maps* anteriores, assim como distintas saídas na etapa de classificação, o que torna a CNN extremamente versátil.

Após a construção da rede é necessário a realização do treinamento como meio de garantir que a saída obtida será igual a saída esperada, para isso aplicam-se técnicas como o Gradiente Descendente e o *Backpropagation*, que terão a função de ajustar os parâmetros da rede para diminuir a diferença entre essas saídas. Essa disparidade entre o resultado esperado e o resultado obtido é calculado através de uma função de custo, que pode ser uma equação específica ou alguma metodologia conhecida, como por exemplo, o cálculo do erro pelo método dos mínimos quadrados.

Uma maneira de efetuar o treinamento é utilizando um banco de dados que contenha variadas entradas a serem classificadas. Desse conjunto escolhe-se então uma porcentagem x para formar o conjunto de testes e uma porcentagem $(100 - x)$ para o conjunto de validação. Normalmente adota-se para x valores próximos a 80. O primeiro subconjunto fornecerá a entrada no modelo de rede neural proposto, enquanto o restante servirá de base para calcular a função de custo e avaliar a disparidade da saída resultante. Realizada a validação, ou seja, calculado o valor do erro entre a saída e o conjunto de validação, ajustam-se então os parâmetros da rede para melhorar o desempenho no ciclo seguinte. Deste modo repete-se o treinamento até que os resultados sejam satisfatórios e, caso necessário, pode ocorrer a troca do conjunto de dados iniciais, assim como a mudança do valor de x e da função de custos (Figura 43).

Figura 43 – Fluxograma



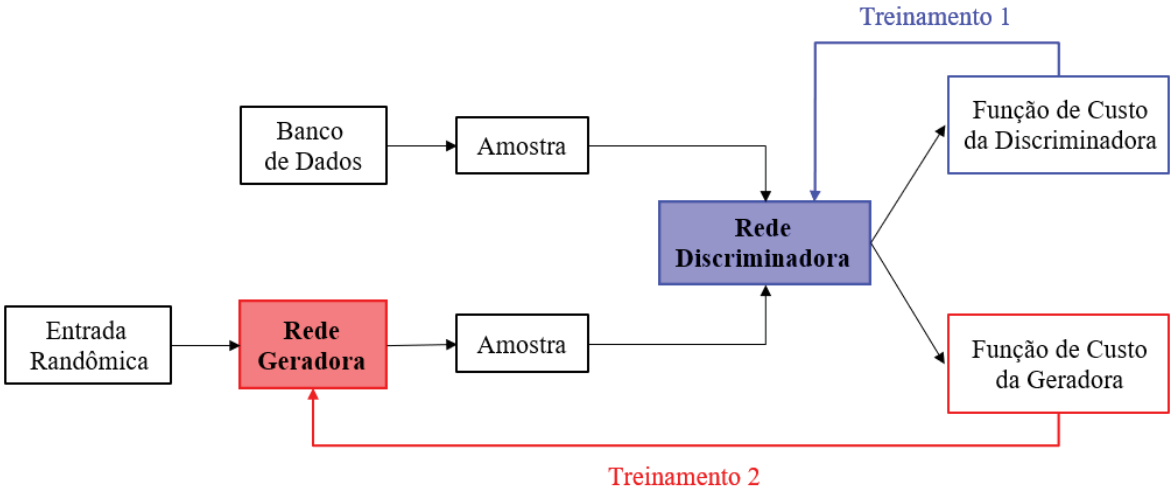
Fonte: Elaborada pela autora (2021)

4.3.2 *Edge Network*

A rede neural convolucional de profundidade, *Edge Network,* é baseada no modelo GAN, *Generative Adversarial Networks* proposta por Ian GoodFellow et al. (17), cuja construção demanda duas redes neurais, uma geradora (*generator*) e uma discriminadora (*discriminator*). Como o próprio nome do modelo sugere, ambas as redes funcionam como adversárias, cujo objetivo é realizar um aprendizado não supervisionado, capaz de gerar dados tão bons quanto os inseridos por um conjunto verídico.

A rede neural geradora é responsável por gerar os novos dados, enquanto a rede neural discriminadora classifica-os como verdadeiros ou falsos, deste modo a segunda rede supervisiona o aprendizado da primeira, criando uma espécie de jogo, onde a geradora tenta "enganar" a discriminadora. O treinamento é realizado de maneira alternada, podendo durar por uma ou mais épocas. Desta maneira, haverá dois treinamentos distintos, cada qual responsável por penalizar uma das redes. Considere o treinamento 1 voltado para a rede discriminadora, desta forma, caso a classificação seja errônea, somente esta rede será penalizada e sofrerá alterações de parâmetros, enquanto a rede geradora não é afetada. Durante o treinamento 2, porém, o comportamento é inverso, penaliza-se a rede geradora quando esta não for capaz de "enganar" a discriminadora. A penalização ocorre através da técnica de *backpropagation*, que utiliza como base os erros obtidos pelas funções de custo. A Figura 44 exemplifica esse processo de treinamento.

Figura 44 - Fluxograma do treinamento geradora-discriminadora



Fonte: Adaptada de (18)

É indicado realizar um treinamento individualizado da rede discriminadora antes de integra-la a geradora, como meio de melhorar os resultados obtidos. O treinamento termina quando rede discriminadora é incapaz de afirmar a veracidade das amostras, indicando que os dados fornecidos pela geradora são confiáveis o suficiente. Neste caso, pode-se retirar a rede classificadora e manter apenas a geradora.

Apesar de utilizar o modelo GAN, a arquitetura utilizada para a construção da *Edge Network* é a mesma proposta por K. Nazeri et al. (19), assim como os hiper parâmetros e o protocolo de treinamento. Os hiper parâmetros podem ser definidos como os valores usados para controlar o processo de aprendizado da rede, que influenciam fatores como a velocidade e a qualidade do processo. Um exemplo desse tipo de parâmetro é o tamanho da rede a ser utilizada.

A seguir é apresentada duas tabela que descreve os detalhes da arquitetura da *Edge Network*, baseadas no material complementar do artigo "*3D Photometry Using Context-Aware Layered Depth Inpainting*"(20). A Tabela 1 se refere a rede neural geradora, enquanto a Tabela 2 representa a rede neural discriminadora.

Tabela 1 - Rede Neural Convolucional Geradora

Rede Neural Convolucional Geradora

Módulo	Dimensão do Kernel	Canais	Taxa de Dilatação	*Stride*	Normalização	Não - Linearidade
Conv1	7 x 7	64		1		
Conv2	4 x 4	128	1	2	SN → IN	ReLu
Conv3	4 x 4	256		2		
BlocoResidual 4						
BlocoResidual 5						
BlocoResidual 6						
BlocoResidual 7	3 x 3	256	2	1	SN → IN	ReLu
BlocoResidual 8						
BlocoResidual 9						
BlocoResidual 10						
BlocoResidual 11						

ConvTransposta12 ConvTransposta13	4 x 4	128 64	1	2	SN → IN	ReLu
Conv14	7 x 7	1	1	1	SN → IN	Sigmoide

Fonte: Adaptada de (20)

Tabela 2 - Rede Neural Convolucional Discriminadora

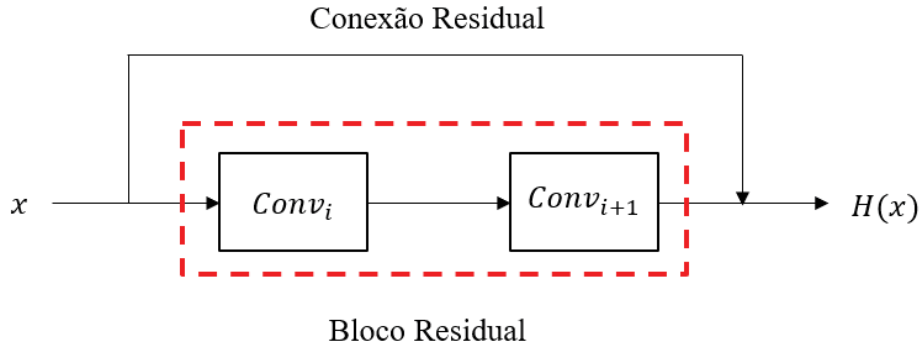
Rede Neural Convolucional Discriminadora

Módulo	Dimensão do Kernel	Canais	Taxa de Dilatação	*Stride*	Normalização	Não - Linearidade
Conv1		64		2		LeakyReLu(0.2)
Conv2		128		2		LeakyReLu(0.2)
Conv3	4 x 4	256	1	2	SN	LeakyReLu(0.2)
Conv4		512		1		LeakyReLu(0.2)
Conv5		1		1		Sigmoide

Fonte: Adaptada de (20)

A coluna Módulo, representa o processo de convolução adotado, onde Conv representa uma camada convolucional, ResnetBlock refere-se a compressão entre duas camadas convolucionais e ConvTransposta simboliza uma convolução transposta. A utilização do Bloco Residual permite um número maior de camadas, sem aumentar o custo computacional, isso porque o seu funcionamento é executado através de um atalho, chamado conexão residual, que liga a entrada das 2 camadas até a saída das mesmas. Suponha uma entrada x para um bloco residual em que se deseja aprender determinada distribuição $H(x)$ em sua saída. Seja ainda $R(x) = H(x) - x$ o caminho descrito entre x e $H(x)$, deste modo, as camadas do bloco serão treinadas para aprender o residual R(x) ao invés de $H(x)$, o que gera uma rede mais profunda e de fácil otimização (Figura 45).

Figura 45 - Bloco Residual



Fonte: Elaborada pela autora (2021)

Quanto a convolução transposta, essa é capaz de aumentar a dimensão de matrizes que tenham passado por *downsampling*, ao mesmo tempo em que realiza uma convolução. São de extrema utilidade em estruturas codificador-decodificador, como é o caso da rede neural geradora atribuída a *Edge Network*, que codifica um tipo de entrada a fim de gerar um novo dado de igual dimensão.

Analisando a tabela novamente, é possível obter o tamanho dos kernels e o número de *canais* gerados utilizados em cada convolução, assim como, a taxa de dilatação aplicada ao *receptive field* e o passo que será conduzida a varredura do filtro, na coluna *Stride*. Por fim, há ainda a especificação de qual processo de normalização foi utilizado sobre dados e as informações das funções de ativação utilizadas para fornecer não-linearidade a rede.

A etapa de normalização aparece em diversas redes neurais, e tem a função de equilibrar os dados, tornando o treinamento mais eficiente e com melhores resultados. Para essa rede utilizou-se duas normalizações distintas, a *spectral normalization* (SN) (21) e a *instance normalization* (IN), onde o símbolo SN→IN, significa a ordem em que foi aplicada na rede geradora. A SN controla a normalização através da constante de Lipschitz, ao restingir essa constante para valores menores que 1, a SN estabiliza o treinamento da rede substituindo as matrizes de pesos pelos seus maiores valores, evitando mudanças repentinas de parâmetros e valores de gradiente. A IN por outro lado, utiliza o cálculo baseado nas médias e nas variâncias calculadas sobre as dimensões de altura e largura em cada *feature map* de cada camada convolucional, podendo ser definida pelas seguintes equações:

$$\mu_{nc} = \frac{1}{HW} \sum_{j=1}^{H} \sum_{k=1}^{W} x_{ncjk} \tag{12}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W}(x_{ncjk} - \mu_{nc})^2 \tag{13}$$

$$\hat{x} = \frac{x - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \varepsilon}} \tag{14}$$

Onde n é a camada convolucional, c é o *feature map* ou canal, H é a altura e W é a largura. O ε é definido como um valor positivo próximo de zero, que tem a função de garantir que não haverá divisão por zero.

O protocolo de treinamento da *Edge Network* é realizado conforme proposto em (19), portanto, aplica-se um método de otimização capaz de adaptar taxas de aprendizagem individualizadas para cada peso da rede neural, nomeado de ADAM (22), *Adaptive Moment Estimation,* com a taxa de aprendizado inicial igual a 0.0001 e o momento $\beta = 0.9$. Quanto as funções de custo utilizadas, a rede geradora e a rede discriminadora se baseiam em uma mesma equação, porém com objetivos distintos, pois enquanto a rede geradora tenta minimiza-la, a rede discriminadora tenta maximizá-la. A função de custo que abrange essas duas redes é baseada em duas outras funções de custo: a *adversarial loss* (L_{adv}) e a *feature-matching loss* (L_{FM}). Considere então L_G e L_D como sendo as funções de custo referentes a rede neural geradora e a rede neural discriminadora, logo:

$$\mathbf{min}\, L_G = \mathbf{min}\ (\lambda_{adv}.L_{adv} + \lambda_{FM}.L_{FM}) \tag{15}$$

$$\mathbf{max}\, L_D = (\lambda_{\mathrm{adv}}\,.\mathbf{max}\ (L_{adv}) + \lambda_{FM}.L_{FM}) \tag{16}$$

Onde λ_{adv} e λ_{FM} são parâmetros de regularização definidos como 1 e 10, respectivamente. Deste modo, define-se a *adversarial loss* como:

$$L_{adv} = E_x[\log(D(x))] + E_z[\log(1 - D(G(z))] \tag{17}$$

A primeira parcela da equação apresenta $E_x[\log(D(x))]$, onde x é um dado real, $D(x)$ é a estimativa feita pela rede discriminadora de que um dado real seja classificado como real e E_x é o valor esperado por todas as amostras de dados reais. Na segunda parcela tem-se $G(z)$

representando a saída da rede geradora dada uma entrada z. $D(G(z))$ é a estimativa feita pela rede discriminadora de que um dado falso seja real e E_z é o valor esperado por todas as amostras falsas geradas. Deste modo, a rede geradora só afetará a segunda parcela da L_{adv}.

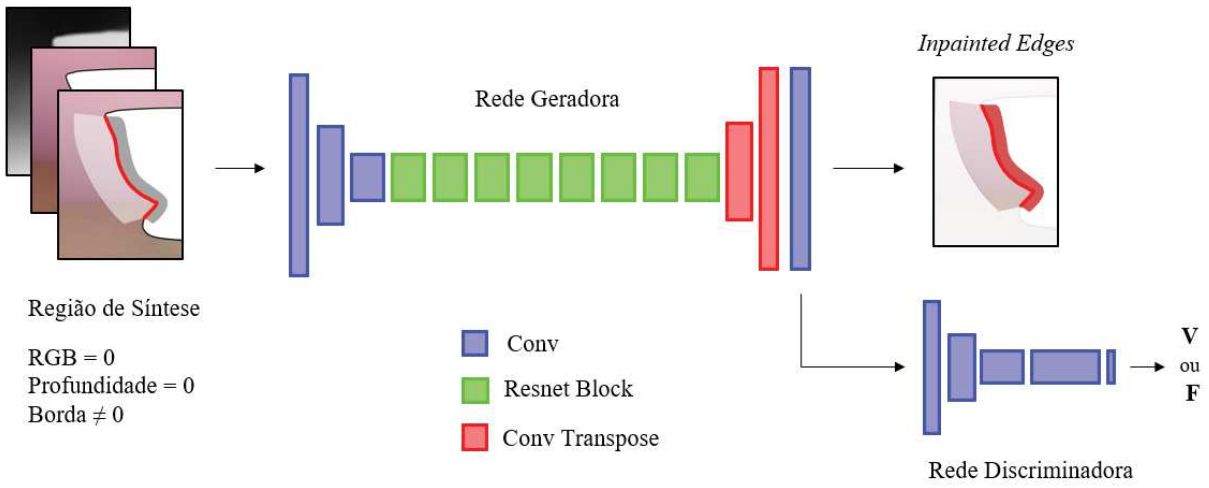
Para a *feature-matching loss* tem-se a seguinte equação:

$$L_{FM} = E\left[\sum_{i=1}^{K} \frac{1}{N_i} \|D_i(x) - D_i(z)\|\right] \tag{18}$$

Onde K é a última camada convolucional da discriminadora e N é o número de elementos na camada i de ativação.

O objetivo desta etapa do *inpainting* é, portanto, gerar a região de síntese a partir da entrada inserida, utilizando a rede geradora treinada. Tal abordagem foi inserida como meio de garantir o alinhamento entre a *Color Network* e a *Depth Network,* que serão acopladas posteriormente. Deste modo, a entrada da *Edge Network* será o *background* da imagem em RGB, o mapa de profundidade e um dos segmentos escolhidos, juntamente com a suas regiões de contexto e de síntese já demarcadas. A região de contexto conterá a informação de cor e profundidade, enquanto a região de síntese será inserida com valores nulos, com exceção da borda. Desta forma geram-se a delimitação da extensão que comporá a região de síntese. Uma representação da *Edge Network* pode ser observada na Figura 46.

Figura 46 - *Edge Network*



Fonte: Elaborada pela autora (2021)

4.3.3 *Color and Depth Network*

As redes neurais convolucionais *Color Network e Depth Network*, são definidas de forma similar, baseada na composição em dois estágios. O primeiro estágio é composto pela *Edge Network* que definirá a região a qual será aplicada tanto a cor e quanto a profundidade, enquanto o segundo estágio é responsável pelo preenchimento dos conteúdos de fato. Ambas as redes foram baseadas na arquitetura U-net (23), porém adaptadas conforme G. Liu et al. (24), utilizando o conceito de camadas convolucionais parciais e etapas de atualização de máscaras.

A camada convolucional parcial pode ser definida como um conjunto de operações, formado pelo processo de convolução parcial mais a atualização da máscara, que só ocorrerá se a soma da matriz M, relativa à máscara for diferente de zero. A convolução parcial pode ser definida como:

$$x' = \begin{cases} W^T \left(X \odot M_{ij}\right) \dfrac{soma\left(matriz[1]_{ij}\right)}{soma\left(matriz\, M_{ij}\right)} + b, & se\ a\ soma(M) > 0 \\ 0, & caso\ contrário \end{cases} \quad (19)$$

Onde W são os pesos que compõem o kernel e b o seu bias correspondente, que pode ser definida como um parâmetro adicionado a rede neural com a finalidade de realizar ajustes matemáticos. A matriz X representa o *receptive field*, enquanto o símbolo $\odot$ indica que a multiplicação entre X e M será realizada elemento a elemento. Para esta aplicação, M é considerada como a concatenação entre a região de contexto e a região de síntese, e é acrescentado um processo de *padding* pra garantir que a preservação das bordas.

A seguir é apresentada a Tabela 3 que descreve os detalhes da arquitetura de ambas as redes neurais, Color Network e Depth Network, também baseada no material complementar do artigo "*3D Photometry Using Context-Aware Layered Depth Inpainting*" (19). Onde a camada convolucional parcial é representada por PConv.

Tabela 3 - Redes neurais convolucionais de cor e de profundidade

Color Network / Depth Network

Módulo	Dimensão do Kernel	Canais	Taxa de Dilatação	*Stride*	Normalização	Não - Linearidade
PConv1	7 x 7	64	1	2	-	ReLu
PConv2	5 x 5	128			BN	
PConv3	5 x 5	256			BN	
PConv4	3 x 3	512			BN	
PConv5	3 x 3	512			BN	
PConv6	3 x 3	512			BN	
PConv7	3 x 3	512			BN	
PConv8	3 x 3	512			BN	
Conc. (8|7)	-	**1024**	-	-	-	-
PConv9	3 x 3	512	1	1	BN	LReLu(0.2)
Conc. (9|6)	-	**1024**	-	-	-	-
PConv10	3 x 3	512	1	1	BN	LReLu(0.2)
Conc. (10|5)	-	**1024**	-	-	-	-
PConv11	3 x 3	512	1	1	BN	LReLu(0.2)
Conc. (11|4)	-	**1024**	-	-	-	-
PConv12	3 x 3	512	1	1	BN	LReLu(0.2)
Conc. (12|3)	-	**768**	-	-	-	-
PConv13	3 x 3	256	1	1	BN	LReLu(0.2)
Conc. (13|2)	-	**384**	-	-	-	-
PConv14	3 x 3	128	1	1	BN	LReLu(0.2)
Conc. (14|1)	-	**192**	-	-	-	-
PConv15	3 x 3	64	1	1	BN	LReLu(0.2)
Conc. (15|Entrada)	-	68 / 70	-	-	-	-
PConv16	3 x 3	1 / 3	1	1	-	-

Fonte: Adaptada de (20)

As colunas que formam essa tabela, apresentam a mesma divisão de parâmetros citadas para as tabelas das redes geradora e discriminadora que compõem a *Edge Network.* Os itens nomeados como "**Conc.(*X*|*Y*)**" representam as informações da concatenação entre a camada

convolucional parcial PConv$X$ e a camada convolucional parcial PConv$Y$. Portanto, se a camada PConv$X$ é formada 512 canais e a camada PConv$Y$ também for composta 512 canais, a junção destas apresentará 1024 canais.

A partição destacada em amarelo, representa o único momento onde há diferença entre a *Color Network* e a *Depth Network,* deste modo o número de canais mudará conforme o tipo de entrada. A entrada da Color Network é formada por 6 canais, 3 relacionados ao RGB + 3 da saída obtida pela *Edge Network,* analogamente, a *Depth Network* é constituída por 4 canais, 1+3, onde o número 1 é relativo ao canal de profundidade. Deste modo, sabendo que a PConv 15 possui 64 canais, a Conv(15|Entrada), possuirá 68 canais para a rede de profundidade e 70 para a rede de coloração. Para a PConv16, tem-se então o resultante de 1 canal para a *Depth Network* e 3 para a *Color Network*, que representam, respectivamente, o canal de profundidade e os canais RGB.

Ainda relacionado a tabela, a normalização indicada pela sigla BN faz menção a *Batch Norm.* A BN é similar a *instance normalization* (IN), apresentada na subseção anterior, também baseada em médias e variâncias. Porém, ao invés de percorrer cada *feature map* de cada camada convolucional, esta percorrerá o primeiro *feature map* de cada camada convolucional, depois o segundo *feature map* de cada camada convolucional e assim sucessivamente. O equacionamento é mostrado a seguir:

$$\mu_c = \frac{1}{NHW}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W} x_{icjk} \tag{20}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W} (x_{icjk} - \mu_c)^2 \tag{21}$$

$$\hat{x} = \frac{x - \mu_c}{\sqrt{\sigma_c^2 + \varepsilon}} \tag{22}$$

Onde N é a camada convolucional, c é o *feature map* ou canal, H é a altura e W é a largura. O ε é definido como um valor positivo próximo de zero, que tem a função garantir que não haverá divisão por zero.

O treinamento da *Color Network* e da *Depth Network,* utiliza funções similares as propostas por G. Liu et al. (23), que baseia suas funções de custos em duas outras funções, uma referente a região de contexto ($L_{Contexto}$) e outra relativa a região de síntese ($L_{Síntese}$), que são descritas a seguir.

$$L_{Contexto} = \frac{1}{N} \left\| C \odot (I - I_r) \right\| \tag{23}$$

$$L_{Síntese} = \frac{1}{N} \left\| S \odot (I - I_r) \right\| \tag{24}$$

Os parâmetros C e S indicam as máscaras binárias atribuída a cada uma das regiões, a variável N simboliza o número total de pixels, I é o resultado do *inpainting* e I_r é o dado original. Deste modo, tem-se que a função de custo associada a *Depth Network* será:

$$L_D = L_{Contexto} + L_{Síntese} \tag{25}$$

Para a *Color Network*, porém, a função de custo L_D não é suficiente, sendo necessário abranger mais três funções além de L_C e L_S, são elas: a *perceptual loss* ($L_{Percentual}$), a *style loss* (L_{Style}) e a *total variation loss* (L_{TV}). A equação utilizada pra a função de custo na rede de coloração é dada a seguir.

$$L_C = L_{Contexto} + 6\, L_{Síntese} + 0.05\, L_{percentual} + 120\, L_{Style} + 0.01\, L_{TV} \tag{26}$$

Onde cada uma das funções adicionais é expressa por:

$$L_{Percentual} = \sum_{p}^{p-1} \frac{\left\| \psi_p(I) - \psi_p(I_r) \right\|}{N_{\psi_p}} \tag{27}$$

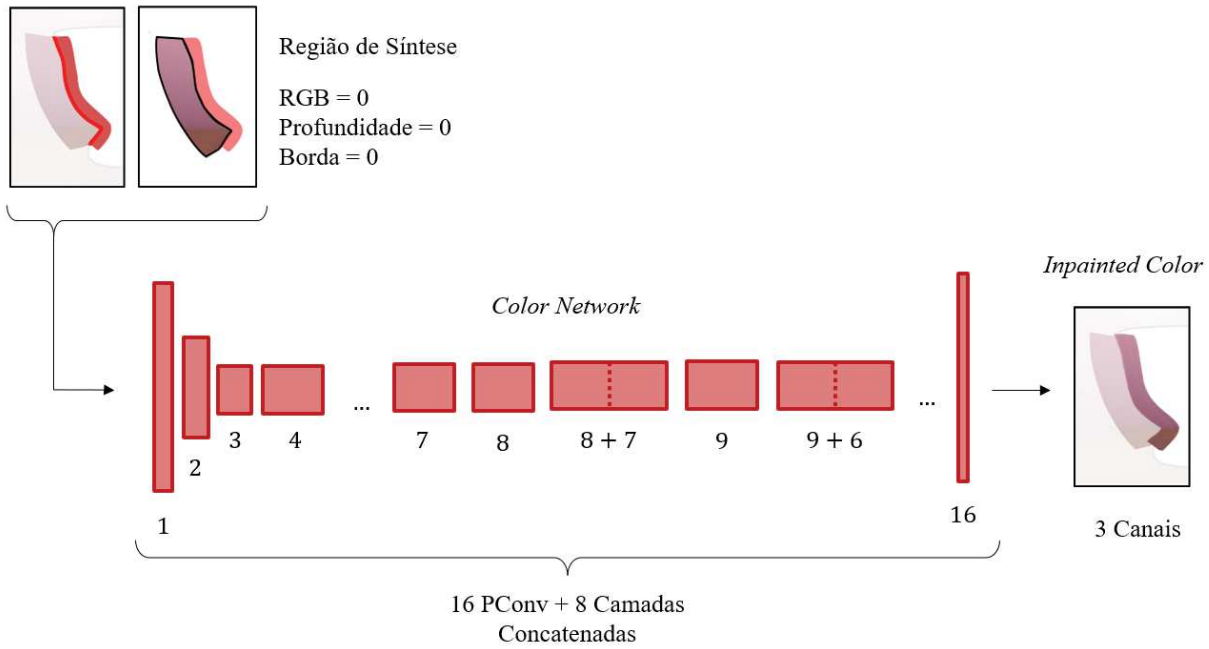
$$L_{Style} = \sum_{p}^{p-1} \frac{1}{C_p C_P} \left\| \frac{1}{C_p H_p W_p} \left[\left(\psi_p^I\right)^T \psi_p^I - \left(\psi_p^{I_r}\right)^T \psi_p^{I_r} \right] \right\| \tag{28}$$

$$L_{TV} = \sum_{(i,j) \in S,(i,j+1) \in S} \frac{\|I(i,j+1) - I(i,j)\|}{N} + \sum_{(i,j) \in S,(i+1,j) \in S} \frac{\|I(i+1,j) - I(i,j)\|}{N} \quad (29)$$

Os termos desconhecidos são encontrados apenas em $L_{Percentual}$ e L_{Style}. Portanto tem-se que as variáveis C_p, H_p e W_p são, respectivamente, o número de canais, a altura e a largura da saída $\psi_p()$. Onde $\psi_p()$ é a saída da camada p da VGG-16 proposta por K. Simonyan and A. Zisserman (25), onde p = {1; ... ; 16}e N_{ψ_p} é o número total de elementos de ψ_p.

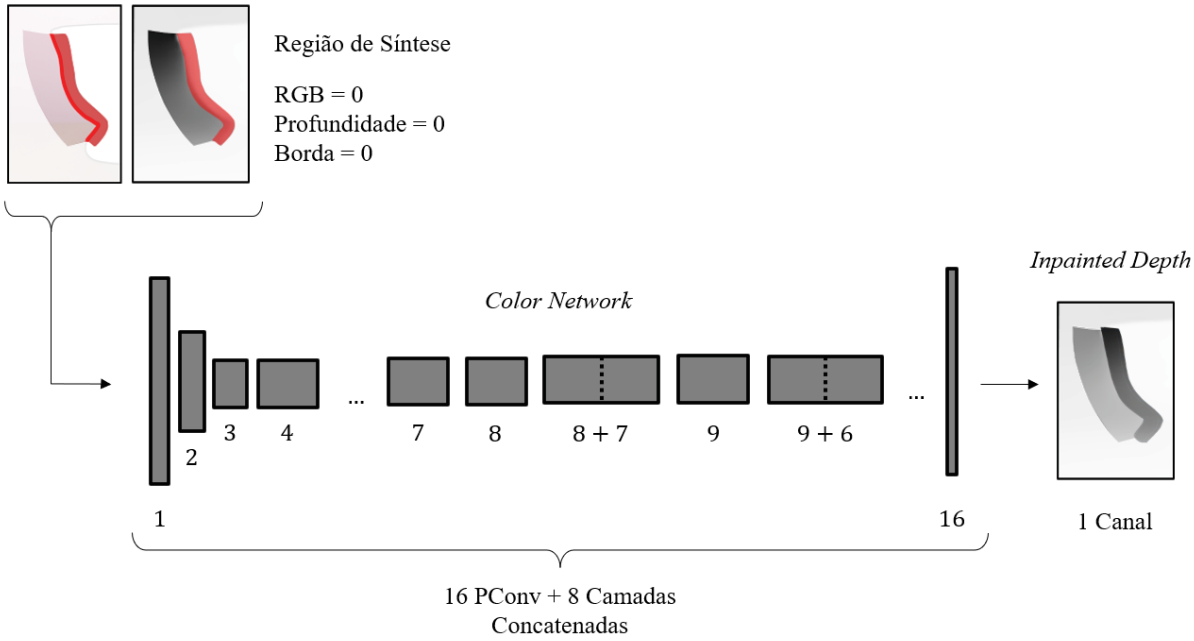
Como entrada a *Color Network* utiliza a imagem RGB do *background* mais a saída fornecida pela *Edge Network,* que fornece a região aonde deve ser utilizado o *inpainting*, assim como a região de contexto que deve ser utilizada na imagem RGB. Para a *Depth Network,* a entrada é análoga, porém trocando a imagem RGB pelo mapa de profundidade. A região de síntese inserida em ambas as redes é preenchida por zeros tanto na cor, quanto na profundidade, se estendendo também para as bordas. As representações dessas arquiteturas podem ser vistas nas Figura 47 e Figura 48.

Figura 47 - *Color Network*



Fonte: Elaborada pela autora (2021)

Figura 48 - *Depth Network*

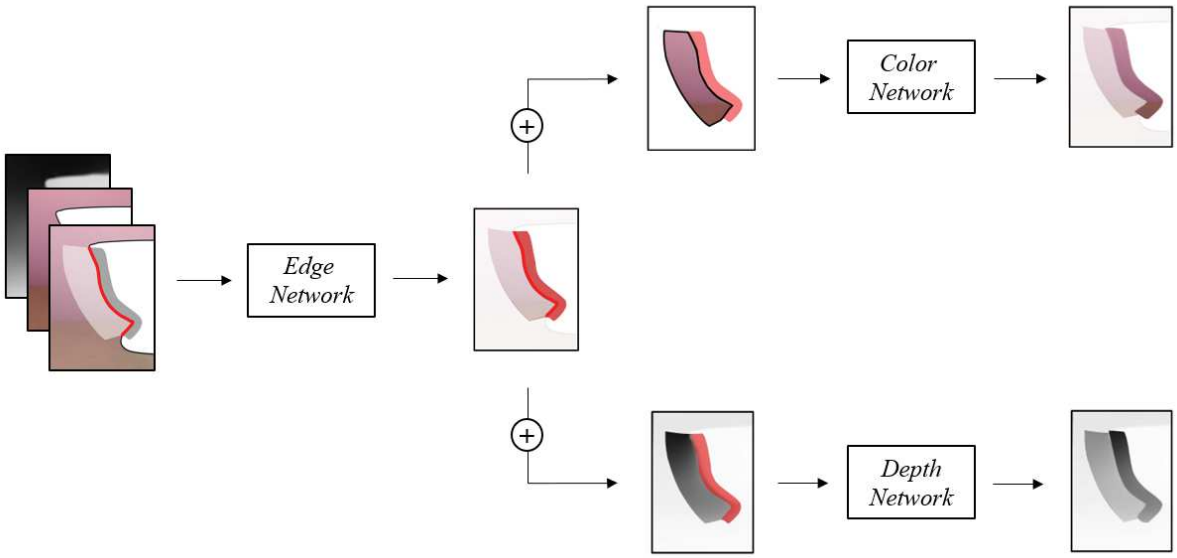


Fonte: Elaborada pela autora (2021)

4.3.4 Integração das sub-redes

A *Edge Network* compõe a base da rede neural de *inpainting,* sendo responsável pela parte estrutural, ou seja, fornecer a região a ser preenchida pelas redes posteriores. As redes neurais *Color Network* e *Depth Network* são destinadas ao preenchimento efetivo da cor e da profundidade, respectivamente. Como já citado anteriormente, as redes que preencherão o conteúdo utilizam a saída da *Edge Network* para realizar o processo de dois estágios, por isso há uma bifurcação a partir desta, como mostra a Figura 49.

Figura 49 - *Inpainting Network*



Fonte: Elaborada pela autora (2021)

Como entrada a rede neural convolucional tem o *background* da imagem em RGB, o mapa de profundidade, os segmentos de contornos definidos na etapa de pré-processamento, e as regiões de contexto e síntese fornecidas pelos algoritmos iterativos. A partir de todos os segmentos da entrada, sorteia-se apenas um para entrar na *Edge Network*. Deste modo, insere-se então o segmento sorteado, a informação RGB e a de profundidade, assim como a região de síntese e contexto correspondente. Neste momento, apenas a rede neural geradora treinada atuará sobre os dados, dispensando então a rede discriminadora da *Edge Network*.

Como saída tem-se a geração da extensão que irá compor a rede de síntese, porém desta vez obtida pela rede geradora ao invés do algoritmo de preenchimento iterativo, o que garantirá a sincronia das redes posteriores. A partir deste momento ocorre a bifurcação das redes que preencherão o conteúdo da região de síntese, que atuarão separadamente pra definir as cores e as profundidades que serão inseridas. A *Color Network* recebe então a saída da *Edge Network* mais a informação em RGB relativa à área de contexto delimitada, realizando a coloração da região de síntese com base nas cores presentes na região de contexto. A *Depth Network* funcionará de forma similar, porém, substituindo a informação RGB pelos dados de profundidade equivalentes.

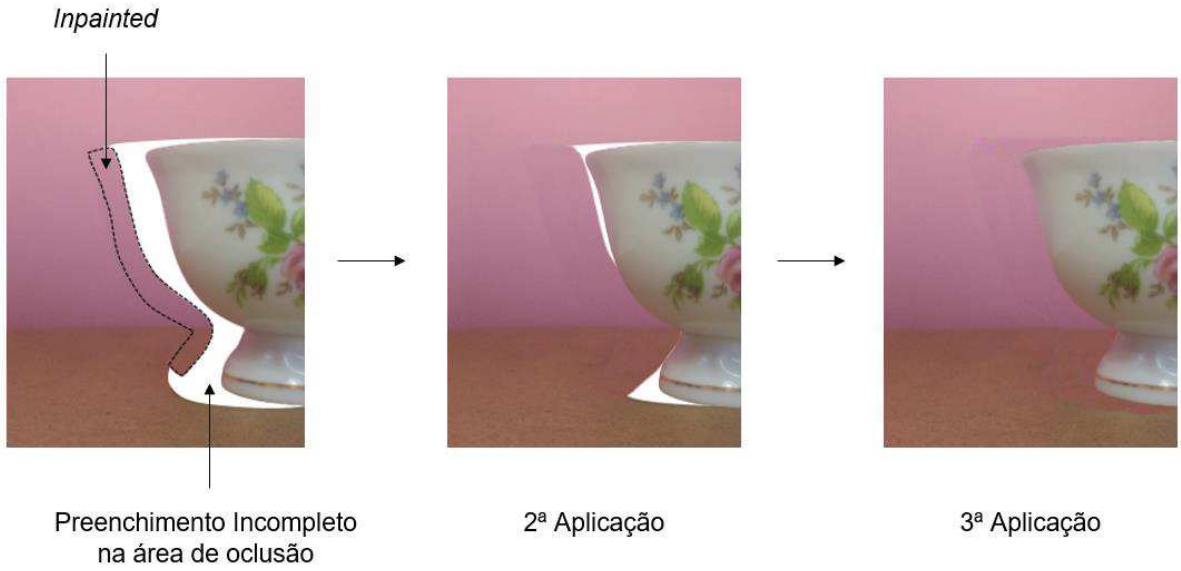
O banco de dados *Microsoft Common Objects in Context* (MSCOCO) (26) foi utilizado como base para gerar o banco de dados utilizado para treinar essa rede neural convolucional. A

partir dele gerou-se regiões de contexto e regiões de síntese, em pelo menos 3 regiões distintas de uma mesma imagem utilizando o MegaDepth (27). Aplicou-se então um par de região contexto e região de síntese, escolhidos aleatoriamente, sobre imagens do banco de dados *Common Objects in Context* (COCO)[2], inserindo-as para o treinamento.

4.3.5 Reaplicação da Rede

Uma única aplicação da rede neural convolucional de *inpainting* sobre um determinado segmento, em alguns casos, pode não ser suficiente. Nesses casos, as áreas de oclusão entre o *foreground* e o *background* apresentarão lacunas em determinados pontos de vista, isso porque a região de síntese preenchida não foi suficiente para cobrir toda a extensão necessária. Desta forma, é necessário a reaplicação da rede como meio de reparar essa falta de preenchimento. A Figura 50 expõe um exemplo da ocorrência deste problema.

Figura 50 - Reaplicação da rede



Fonte: Elaborada pela autora (2021)

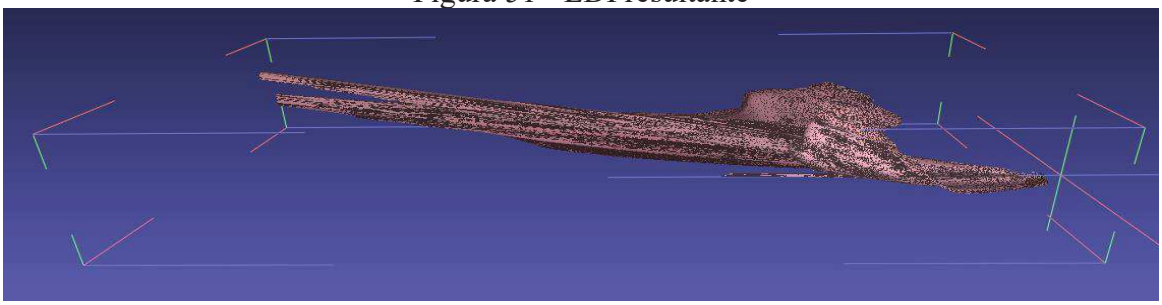
Portanto, para cada segmento sorteado, realiza-se o *inpainting* e reaplica-se a rede neural até que todas as lacunas presentes na área de oclusão desejada sejam preenchidas corretamente.

[2] COCO: https://cocodataset.org/#home

4.4 VÍDEO TRIDIMENSIONAL

Todos os pixels que receberam valores de coloração e profundidade na etapa de *inpainting* são alocados novamente sobre a LDI original, gerando diversas camadas conectadas que variam conforme profundidade. A capacidade de gerar um número arbitrário de camadas permite a LDI representar casos complexos de profundidade, que contenham diversos intervalos de variação. A Figura 51 mostra a LDI resultante para o exemplo da xícara utilizado durante o processo.

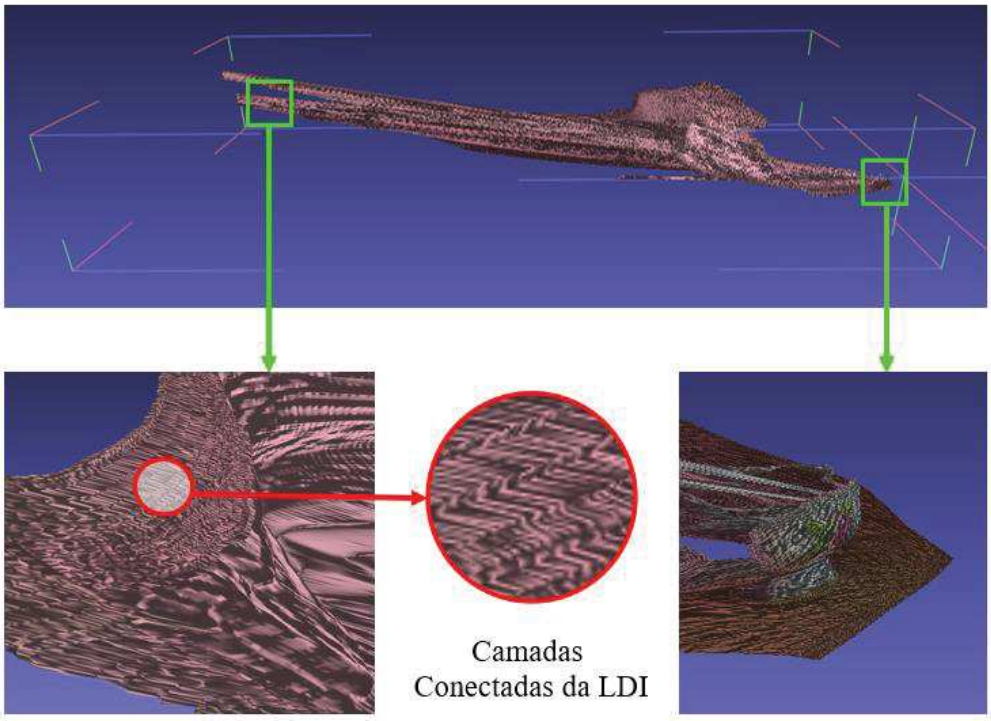
Figura 51 - LDI resultante



Fonte: Elaborada pela autora (2021)

Ao aproximar a estrutura resultante, é possível observar a composição em camadas conectadas e o *foreground* definido para a entrada, Figura 52.

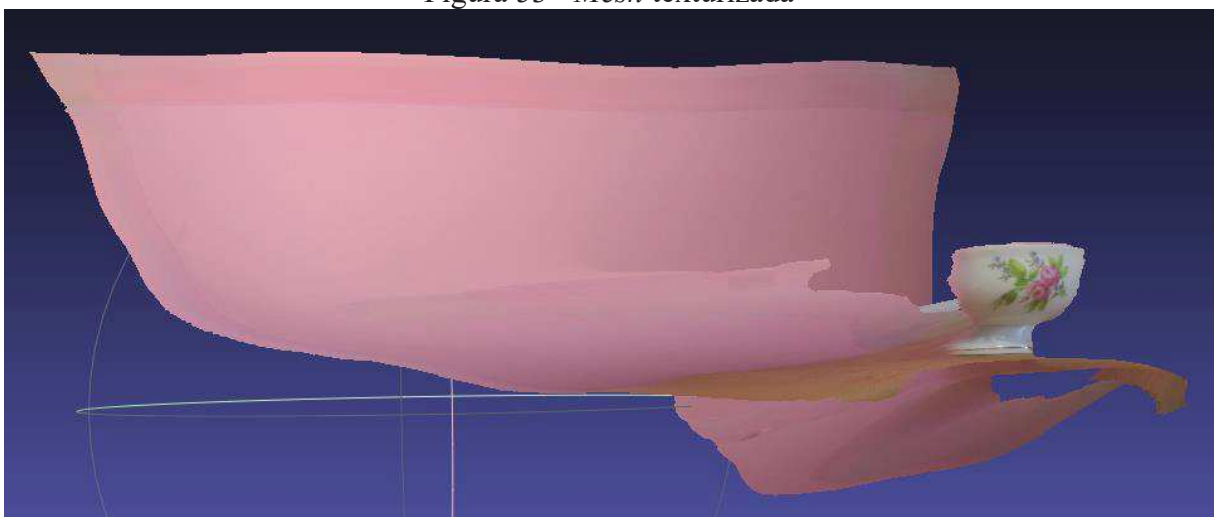
Figura 52 - Detalhes da LDI resultante



Fonte: Elaborada pela autora (2021)

Uma *mesh* texturizada pode ser gerada a partir de uma LDI, de maneira rápida e fácil, como mostra a Figura 53.

Figura 53 - *Mesh* texturizada



Os vídeos tridimensionais, também nomeados de fotos 3D, são gerados a partir da renderização da *mesh* texturizada. Considera-se que o objeto esteja no centro da imagem, e a partir disso desloca-se o ponto de vista conforme o desejado dando a sensação de mudança de posição aparente, o que se dá o nome de paralaxe, que em grego significa “alternância”. Para essa aplicação, é necessário apenas o movimento de *zoom-in*, que fornecerá a aproximação do *foreground*. A partir dessas definições é então gerado o vídeo tridimensional utilizando bibliotecas de desenvolvimento gráfico padrões, conhecidas pelo termo *graphics engines.* Em particular, para esse trabalho foi utilizada a biblioteca *cyNetworkX* para renderizar a *mesh e* a biblioteca *MoviePy* para gerar o vídeo, ambas em linguagem python.

5 TEXTURIZAÇÃO NO DOMO

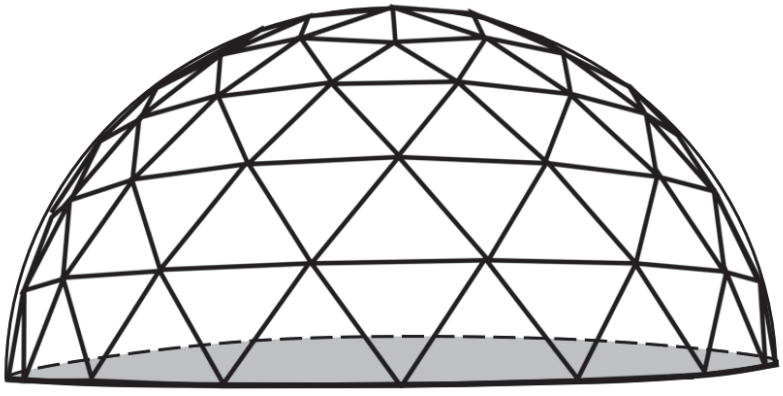
Este capítulo abordará o conceito e a metodologia utilizados para texturizar o vídeo tridimensional obtido na saída da etapa de *inpainting* para um domo tridimensional. Essa texturização fornecerá um vídeo esférico que promoverá a inserção do observador no centro do ambiente retratado pela panorâmica equiretangular. O objetivo é fornecer a aproximação dos objetos presentes no cenário captado, sem apresentar esticamento produzido pela utilização excessiva do *zoom* em um único ponto.

Para discorrer sobre o assunto, o capítulo foi dividido em 2 seções, a seção 5.1 tratará do conceito de domo tridimensional e da formulação matemática associada a projeção do vídeo tridimensional para vídeo esférico. Enquanto a seção 5.2 abordará a metodologia de implementação utilizada para concretizar essa conversão.

5.1 DOMO TRIDIMENSIONAL

Um domo tridimensional é uma estrutura espacial que representa metade de uma esfera (Figura 54). A partir de seu centro é possível mapear todo o ambiente circundante, o que proporciona ao observador uma visão em 360 graus do ambiente.

Figura 54 - Domo tridimensional



Fonte: Elaborada pela autora (2021)

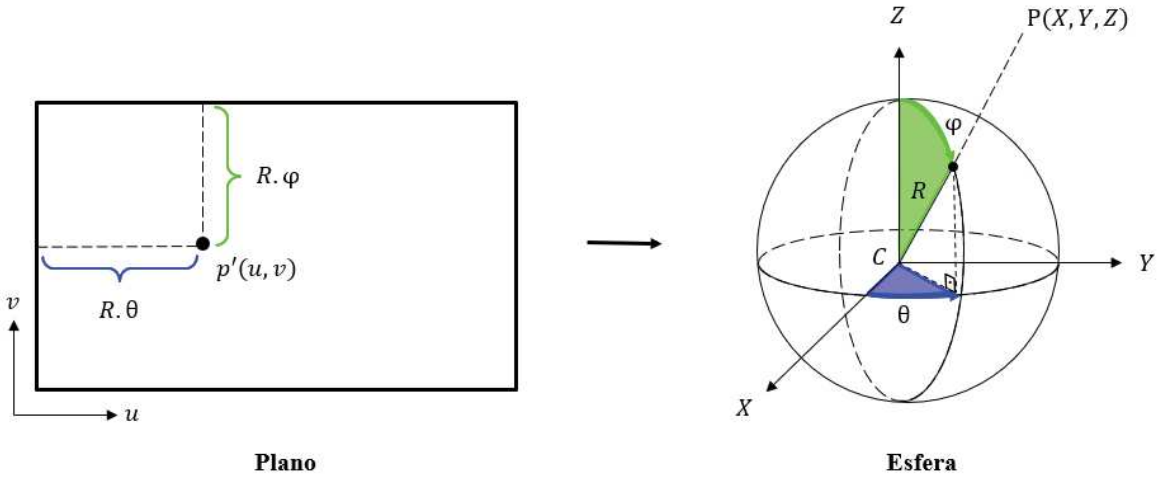
Um vídeo é formado por um conjunto de imagens mostradas em sequência, portanto, é correto afirmar que o vídeo tridimensional obtido na saída da metodologia de *inpainting* também seguirá essa composição. Porém, ao invés de imagens convencionais, este é constituído

por panorâmicas equiretangulares, cuja sequência mostra a aproximação dos componentes da cena juntamente com o preenchimento realizado sobre áreas de oclusão.

Logo, projetar o vídeo tridimensional em um domo é o mesmo que projetar sucessivas panorâmicas equiretangulares. Considerando que, em geral, as fotografias panorâmicas são aquisionadas perpendicularmente ao plano da superfície, é possível equipara a projeção em domo a uma projeção realizada sobre uma esfera, sem ocorrer a perda de informação. Essa adaptação proporciona um facilitamento matemático na conversão, isso porque a panorâmica equiretangular é construída baseada em uma estrutura esférica.

No capítulo 2 deste trabalho foi apresentado o processo de geração de uma panorâmica equiretangular a partir de uma projeção esférica, deseja-se neste momento realizar o caminho inverso, transformando uma panorâmica equiretangular em uma projeção esférica. Portanto o embasamento matemático será similar ao apresentado no capítulo 2, como mostra a Figura 55.

Figura 55 - Transformação para coordenadas esféricas



Fonte: Elaborada pela autora (2021)

A representação plana é expressa pelo sistema de coordenadas (u, v), e pode ser convertida para o sistema de coordenadas esféricas através da relação $[u, v]^T = R[\theta, \varphi]^T$, descrita como:

$$u = R.\varphi \tag{30}$$

$$v = R.\theta \tag{31}$$

Onde os ângulos θ e φ são respectivamente longitude e latitude, em radianos, e R é o raio da esfera, o qual pode ser definido arbitrariamente. As equações para θ e φ são deduzidas a partir das definições das coordenadas u e v citadas acima. Logo:

$$\varphi = \frac{u}{R} \tag{32}$$

$$\theta = \frac{v}{R} \tag{33}$$

Obtidos os valores dos ângulos, e considerando $R = f$, onde f é o comprimento focal da câmera, calculam-se as coordenadas $(X_i,\ Y_i,\ Z_i)$ dos pontos P_i da esfera conforme as equações matemáticas a seguir, onde $i \in \mathbb{N}$. Deste modo:

$$X = R.\ sen\,\theta.\ cos\,\varphi \tag{34}$$

$$Y = R.\ cos\,\theta.\ sen\,\varphi \tag{35}$$

$$Z = R.\ cos\,\varphi \tag{36}$$

5.2 TEXTURIZAÇÃO

A texturização do vídeo tridimensional é realizada ao projetar as sucessivas panorâmicas equiretangulares que o compõem uma esfera, o que resultará em um vídeo esférico. A implementação dessa conversão foi baseada na metodologia do algoritmo *Google Spatial Media*, adaptando sua entrada apenas para vídeos que não possuam informação de áudio, como é o caso dos vídeos tridimensionais produzidos no processo de *inpainting.*

O *Google Spatial Media* é um algoritmo formado por um conjunto de especificações e ferramentas para gerar um vídeo esférico 360 assim como o áudio espacial. Ele funciona baseado no sistema injetor de metadados, que retorna não só a conversão como a iteratividade no vídeo gerado, fornecendo um conjunto de controle que permite o usuário navegar no ambiente, com movimentos para a esquerda, para a direita, para cima e para baixo.

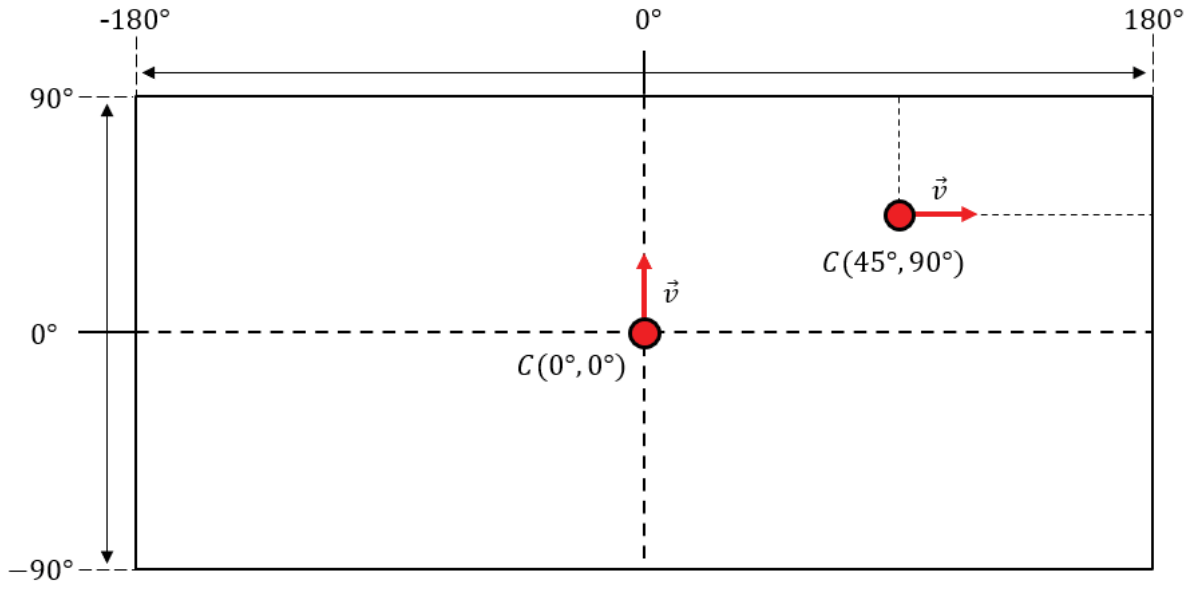
Metadados são dados que fornecem informações sobre outros dados, ou seja, dados descritivos que possuem informações sobre a forma e o conteúdo de determinado arquivo. Essa estrutura organizacional auxilia em quesitos como identificação, localização e classificação de informações. O tipo de organização de metadados utilizado neste trabalho é o XML (Extensible

Markup Language), uma estrutura em árvore composta por diversos comandos em linhas, chamados de *tags,* cuja finalidade é simplificar e uniformizar o processo de leitura. As *tags* obedecem a uma ordem hierárquica, sendo inicializadas pelo símbolo (<) e finalizada pelo símbolo (>), dispostas em sequência e apresentando também subelementos, as *subtags*.

Para representar as características de um vídeo esférico são necessários dois tipos de metadados: os globais e os locais. Os metadados globais são direcionados para dados gerais que compõem o arquivo, como por exemplo as dimensões de largura e altura do vídeo inserido, enquanto os metadados locais são responsáveis por informações pontuais, armazenadas em blocos, como por exemplo os valores dos ângulos de latitude e longitude.

A interatividade do vídeo é iniciada a partir de um ponto de vista padrão, definido pelo centro C sobreposto ao centro da panorâmica equiretangular. A partir de C é construído um modelo de rotação, estabelecendo horizontalmente, $180°$ para a direita e $-180°$ para a esquerda, e verticalmente, $90°$ para cima e $-90°$ para baixo. É definido também um vetor de direção $\vec{v}$, localizado inicialmente sobre no centro C apontado para cima. Deste modo, quando se executa uma rotação, por exemplo, $45°$ na vertical e $90°$ na horizontal, esse centro de percepção do observador muda para centralizar a imagem. O vetor $\vec{v}$ marcará a rotação ocorrida em relação a sua posição inicial, deste modo, um movimento de $45°$ na vertical seguido por um movimento de $90°$ na horizontal, indicará uma variação de $90°$ para o vetor $\vec{v}$ em relação a sua direção inicial. A Figura 56 exemplifica o modelo de rotação adotado.

Figura 56- Modelo de rotação



Fonte: Elaborada pela autora (2021)

Como resultado final deste processo, obtém-se um vídeo esférico interativo para o usuário, capaz de inserir o observador no cenário representado pela panorâmica equiretangular de entrada, aproximando os componentes da cena sem distorcer o seu formato (Figura 57).

Figura 57 - Vídeo esférico

Fonte: Elaborada pela autora (2021)

6 RESULTADOS

Neste capítulo serão mostrados os resultados obtidos neste trabalho. As seções serão divididas conforme a ordem citada através dos capítulos antecedentes, buscando mostrar os resultados parciais obtidos durante todo o processo. Na seção 6.1 serão apresentadas as panorâmicas escolhidas para demonstrar os resultados, suas características e dados relativos à estrutura. Na seção 6.2 serão analisados os mapas de profundidade obtidos pelo MiDaS, mostrando o impacto gerado pela diferença de conteúdo presente nas panorâmicas. Na seção 6.3 serão mostradas as *meshs* obtidas pelo algoritmo de *inpainting,* assim como a texturização das mesmas. Na seção 6.4 serão evidenciados os preenchimentos da área de oclusão nos vídeos tridimensionais gerados, mostrando as regiões de sucesso e também as regiões que apresentaram problemas. A seção 6.5 concluirá os resultados com as análises realizadas sobre o vídeo esférico final resultante.

6.1 PANORÂMICA

Para demonstrar os resultados foram escolhidas duas panorâmicas distintas, porém ambas equiretangulares. A primeira delas apresenta um ambiente externo (Figura 58), com poucos objetos e pouca variação de coloração, exibindo em sua maioria, tons de azul, verde, cinza e beje. Os componentes da cena se encontram mais espaçados e com uma quantidade de detalhes reduzidos.

Figura 58 – Panorâmica equiretangular 1

Fonte: (28)

Essa fotografia panorâmica possui uma resolução de 1024 x 512, ou seja, apenas 1024 pixels de largura por 512 pixels de altura. É considerada uma imagem com média qualidade, apresentando detalhes bem delineados, mas não extremamente definidos, o que é possível perceber com a aproximação dos detalhes da imagem utilizando a ferramenta de *zoom.* O arquivo possui apenas 82,5 KB de tamanho, o que resulta em um processamento relativamente rápido das etapas.

A segunda panorâmica equiretangular escolhida (Figura 59), retrata um ambiente interno, com uma grande quantidade de objetos em diferentes cores, como nas portas, nos sofás, nas mesas e nos acessórios. Contém ainda a representação de um conteúdo relativo a um espaço externo, que pode ser visualizado através das portas e das janelas.

Figura 59 - Panorâmica equiretangular 2

Fonte: (29)

Possui uma resolução de $6.336 \, x \, 3.168$, ou seja, 6336 pixels de largura por 3168 pixels de altura. É considerada uma imagem de alta qualidade, portanto, apresenta detalhes bem definidos, devido ao número de pixels utilizados para compor a cena. O arquivo possui 1.86 MB de tamanho e, devido a sua extensão, a execução de alguns dos processos torna-se mais demorada.

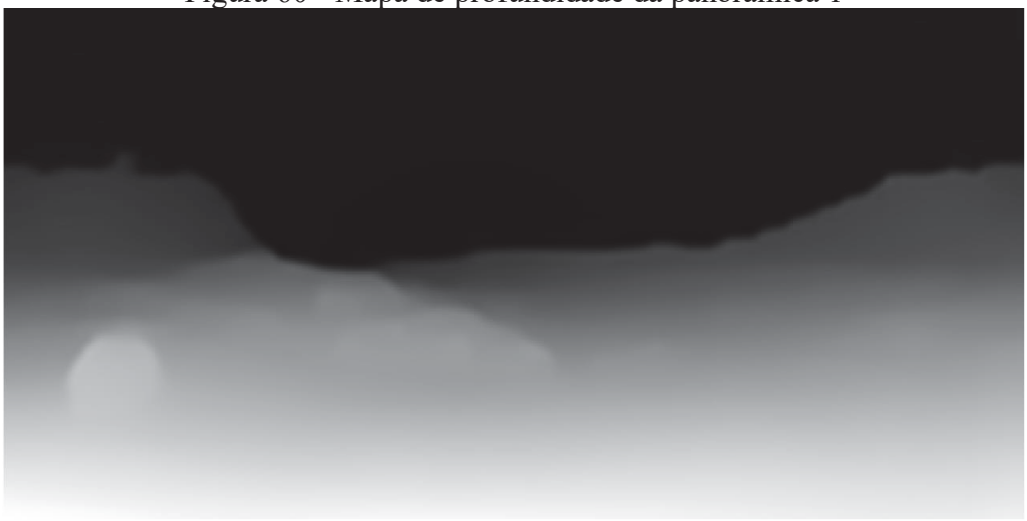
Ambas as imagens tem o formato .JPG, que é o único aceito pelo algoritmo, portanto, as imagens que não possuam esse formato devem ser convertidas externamente para serem processadas.

6.2 MAPA DE PROFUNDIDADE

Em ambas as entradas, os mapas de profundidade foram obtidos pelo modelo MiDaS, e alocados em uma pasta externa como uma das saídas do algoritmo. Caso a panorâmica também possua um mapa de profundidade aquisionado de maneira externa, o algoritmo permite a inserção deste nos formatos .NPY e .PNG, neste contexto, pode-se desativar a estimativa realizada pelo MiDaS.

A Figura 60 apresenta o mapa de profundidade obtido pelo MiDaS a partir da primeira panorâmica equiretangular. Em tons mais claros é possível observar a extensão de areia que forma a imagem, acrescida do formato de algumas pedras avulsas que compõem o cenário. Os tons intermediários são constituídos pelas regiões relacionadas ao mar e as vegetações, e os tons mais escuros são compostos completamente pela porção que faz representa o céu.

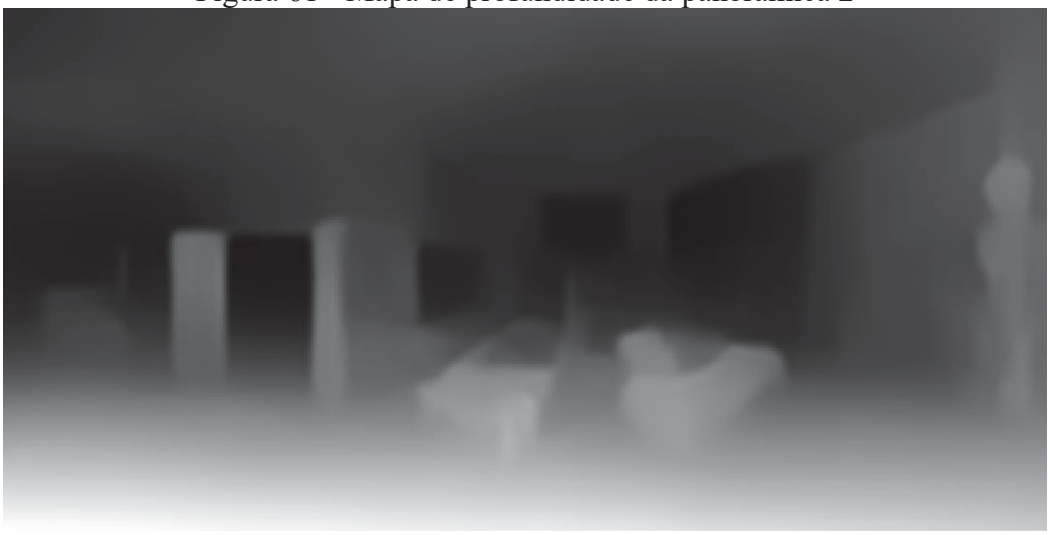
Figura 60 - Mapa de profundidade da panorâmica 1



Fonte: Elaborada pela autora (2021)

A segunda panorâmica equiretangular pode ser representada pelo mapa de profundidade da Figura 61. Pelo resultado é possível perceber que o modelo definiu como as partes mais próximas o sofá da direita, a pilastra de sustentação localizada próxima a escada em espiral, a mesa laranja e as portas, representados por tons mais claros de cinza. Em um segundo plano, com tons de cinza escuros, percebem-se a mesa de jantar e as paredes. E por fim, nas regiões em preto estão a área externa retratada na imagem e o fundo da sala.

Figura 61 - Mapa de profundidade da panorâmica 2



Fonte: Elaborada pela autora (2021)

Como dito anteriormente, os mapas de profundidade fornecidos pelo MiDaS, proveem aproximações similares a realidade, porém desprovidos de detalhes e com as transições suavizadas. Todavia, apesar das suas limitações, este apresentou um excelente desempenho em relação as imagens panorâmicas, uma vez que sua utilidade inicial era para apenas imagens convencionais.

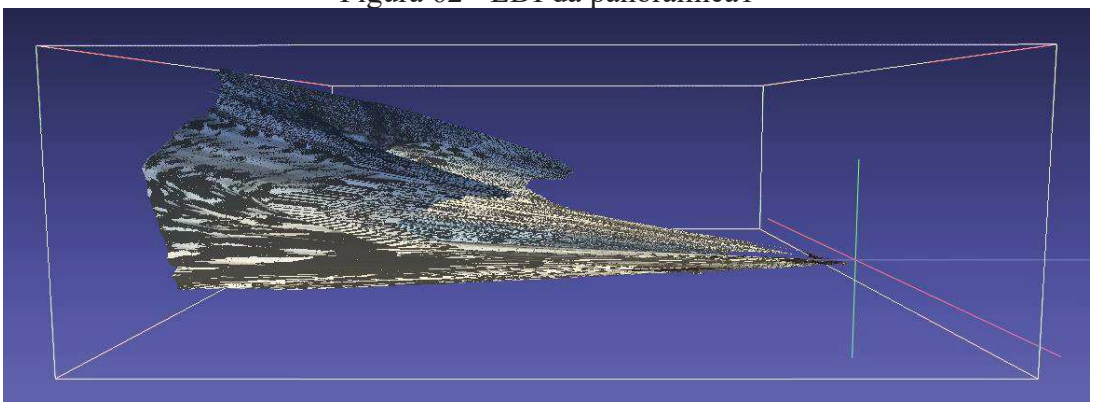
Quanto ao formato gerado, o MiDaS retorna um arquivo .PNG que possui resolução idêntica as panorâmicas equiretangulares inseridas, entretanto apresenta um tamanho de arquivo bem maior que o das imagens originais. As panorâmicas resultaram em mapas de profundidade com respectivamente, $3{,}24\ MB$ e $371\ KB$ de tamanho e, por isso, é necessário aplicar um redimensionamento sobre esses mapas, com a finalidade de diminuir o tempo de processamento. A escala adotada foi de $640/\,[max(altura, largura)]$, obtendo para a primeira entrada um mapa de profundidade de dimensão 640 x 320, com tamanho de arquivo igual a 260KB, a segunda panorâmica apresentou igual resolução, porém, com um arquivo de 199 KB de extensão.

6.3 *MESH*

A próxima saída do algoritmo são as LDIs em camadas fornecidas pelo *inpainting*, e consequentemente as *meshs* texturizadas, onde é possível analisar as áreas de oclusão que receberam o preenchimento de coloração e profundidade. As LDIs geradas variam conforme a complexidade da profundidade do problema, gerando formas e comprimentos distintos para cada caso.

A Figura 62 apresenta a LDI obtida para a primeira panorâmica equiretangular, mostrando o formato da estrutura e a composição das camadas. Como o exemplo da xícara mostrado no capítulo 4 deste trabalho, o formato de construção da LDI se mantém. Na ponta da estrutura são representados os componentes da imagem que apresentam menores profundidades enquanto na direção oposta são localizadas, as maiores, portanto, quanto mais distante da ponta, maior será a profundidade. É possível perceber ainda que a estrutura assumiu um formato côncavo e único, isso porque as profundidades ao redor da panorâmica apresentavam uma distribuição relativamente semelhante e homogênea.

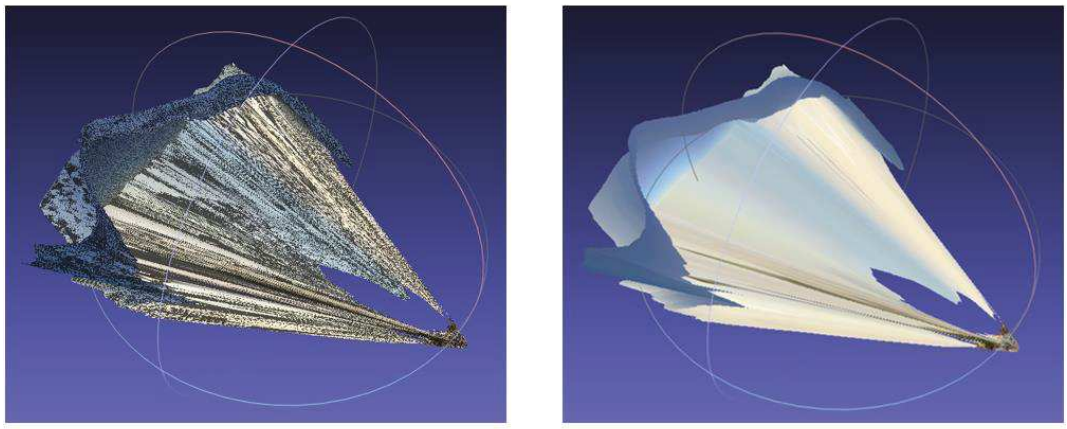
Figura 62 - LDI da panorâmica1



Fonte: Elaborada pela autora (2021)

Devido a sua estrutura, as LDIs podem ser facilmente convertidas em *meshs* texturizadas. A Figura 63 mostra esse processo realizado pelo programa de visualização 3D MeshLab[3].

Figura 63 - Comparação entre a LDI e a *mesh* da panorâmica 1



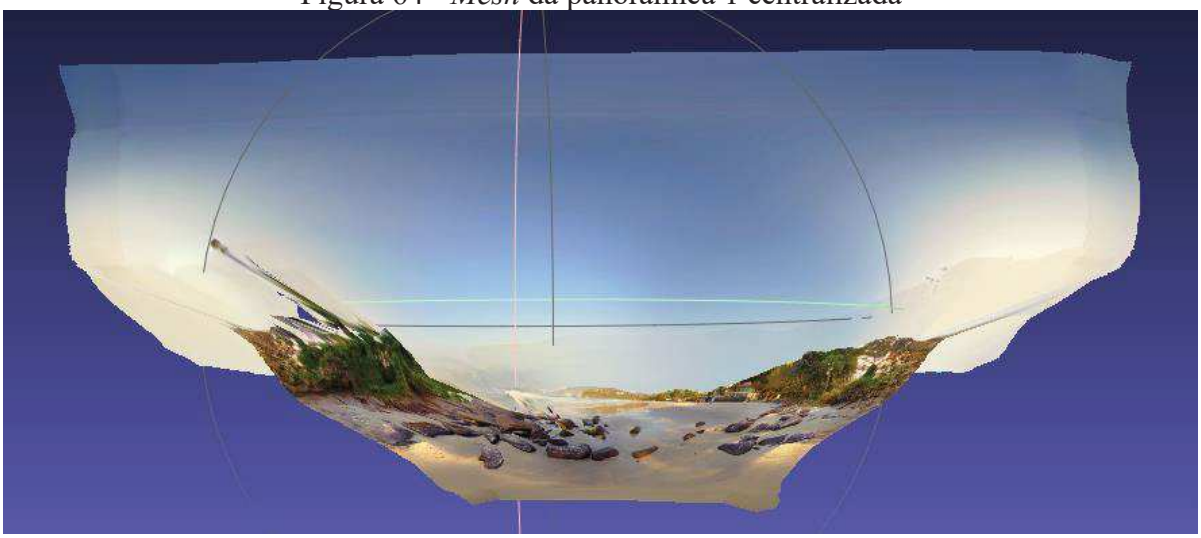
LDI em Camadas *Mesh* Texturizada

Fonte: Elaborada pela autora (2021)

[3] MeshLab: https://www.meshlab.net/

Através do processo de texturização, é possível perceber o desaparecimento das transições entre as camadas, proporcionando uma composição mais uniforme das cores presentes na estrutura, o que permite uma visualização nítida da curvatura da mesma. Ao rotacionar e centralizar a ponta da *mesh* para o centro referencial do MeshLab, observa-se que as profundidades são representadas a partir do centro da panorâmica equiretangular, se distorcendo nas regiões de extremidade lateral. Esse ponto de vista pode ser observado através da representação exposta pela Figura 64.

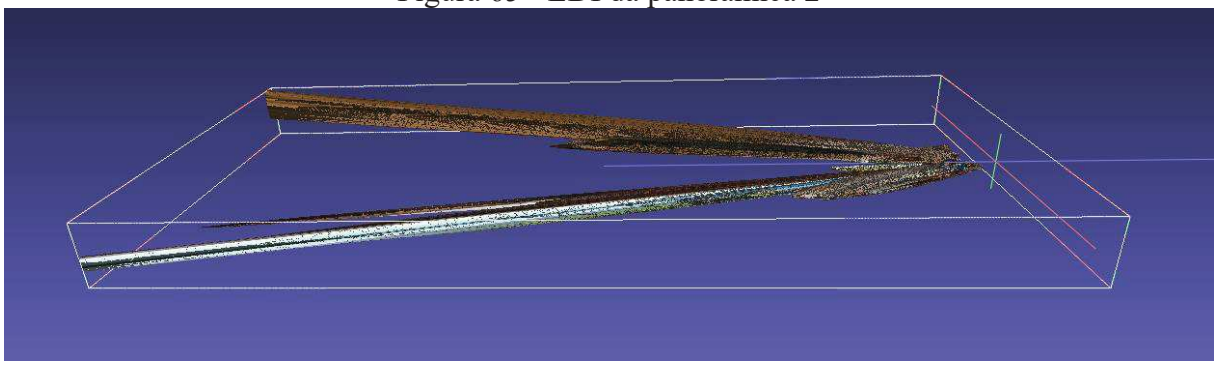
Figura 64 - *Mesh* da panorâmica 1 centralizada



Fonte: Elaborada pela autora (2021)

A LDI obtida para a segunda panorâmica é apresentado na Figura 65. Essa representação se difere bastante da estrutura obtida para panorâmica anterior, tanto em extensão quanto em formato. Por se tratar de um conteúdo de profundidade mais complexo, a densidade de informações e o comprimento desta LDI são maiores do que os apresentados pelo resultado anterior. Quanto ao formato, é possível perceber o surgimento de conjuntos de profundidade espaçados, não existentes no primeiro exemplo, os quais foram gerados pela área externa, captada através da porta e das janelas, e pela região relativa ao fundo da sala.

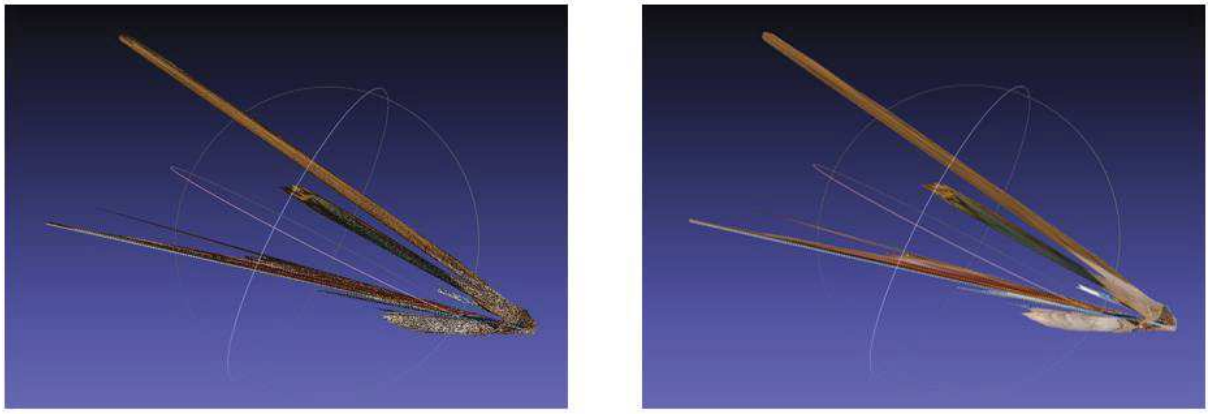
Figura 65 - LDI da panorâmica 2



Fonte: Elaborada pela autora (2021)

A conversão da LDI em camadas obtida pela segunda panorâmica equiretangular para a *mesh* texturizada é mostrada pela Figura 66.

Figura 66 - Comparação entre a LDI e a *mesh* da panorâmica 2



LDI em Camadas *Mesh* Texturizada

Fonte: Elaborada pela autora (2021)

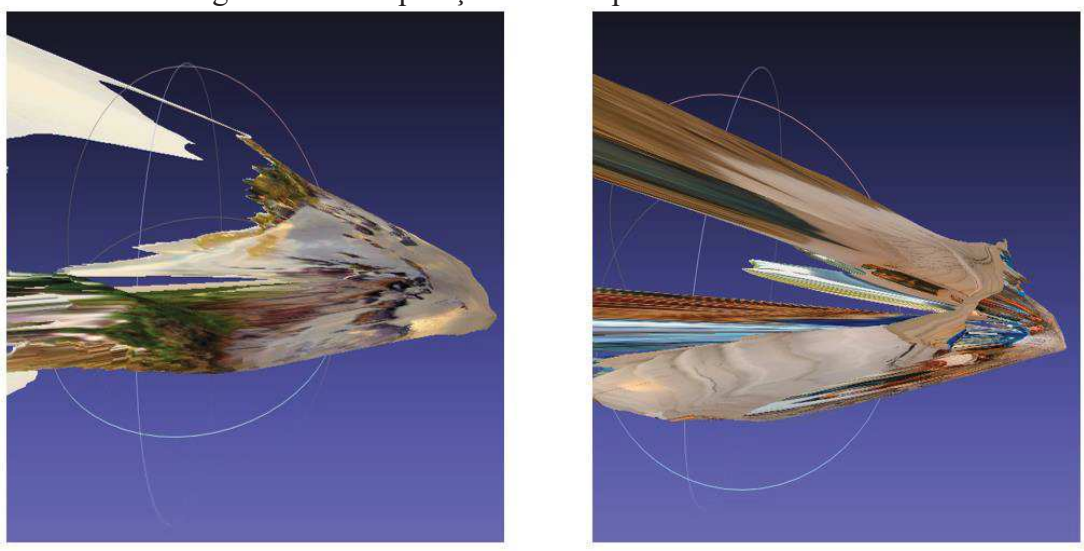
Assim como no primeiro caso, após a texturização é possível observar o desaparecimento das transições representadas pelas camadas da LDI, que proporciona uma continuidade entre as cores dos pixels. Ao rotacionar a estrutura pelo MeshLab direcionando o a visão central para a ponta da *mesh*, observa-se novamente as distorções das extremidades em relação ao centro (Figura 67).

Figura 67 - *Mesh* da panorâmica 2 centralizada

Fonte: Elaborada pela autora (2021)

Analisando as pontas das duas *mesh*, na Figura 68, observa-se uma diferença estrutural entre ambas, em virtude da complexidade da profundidade e da densidade do conteúdo de cada uma das entradas. A primeira panorâmica não possuía muitos detalhes nem muitas diferenças de profundidade, deste modo, o *foreground* produzido é visto destacado em relação ao resto da estrutura, isso porque há uma região de lacuna na área de oclusão que não houve a necessidade de ser preenchida. Este mesmo fato não ocorre para a *mesh* da segunda panorâmica, pois a grande quantidade de elementos gerou várias áreas de oclusão, que ao serem preenchidas não resultaram em um *foreground* delineado.

Figura 68 - Comparação entre as pontas das *meshs* 1 e 2



Mesh 1 *Mesh* 2

Fonte: Elaborada pela autora (2021)

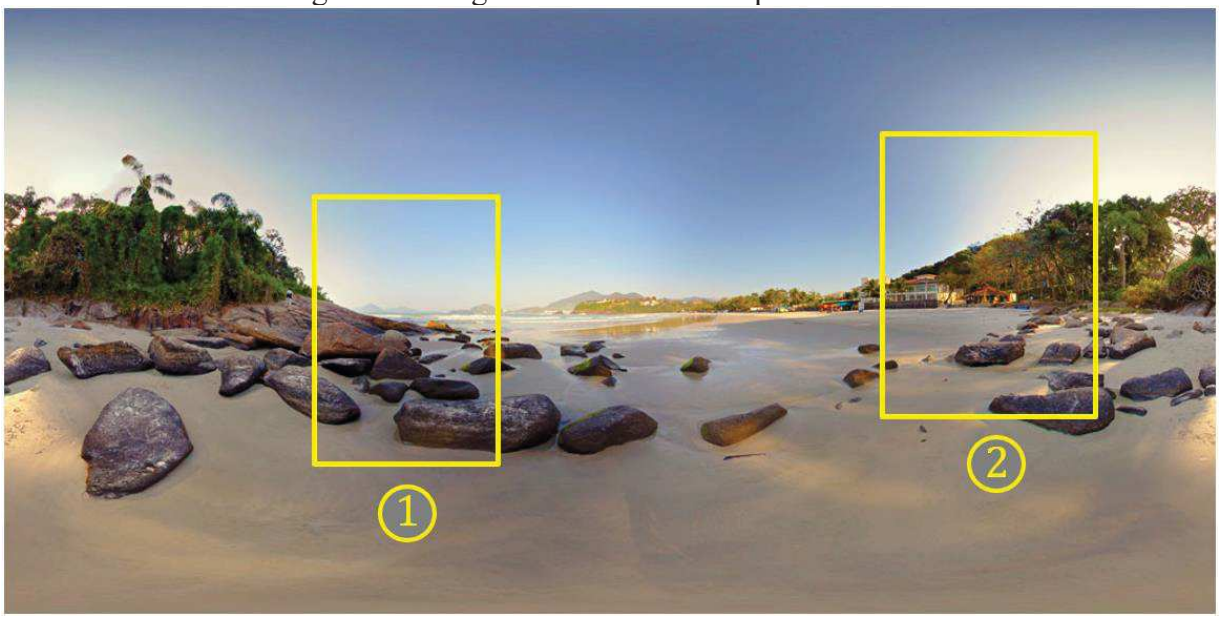
Quanto aos tamanhos dos arquivos gerados, a mesh obtida através da panorâmica 1 possui 406 MB, enquanto a mesh produzida através da panorâmica 2 tem 119 MB, portanto, quanto maior a resolução da panorâmica de entrada, maior será o tamanho da mesh de saída. Em relação ao formato, ambos os arquivos são fornecidos em formato. PLY.

6.4 VÍDEO TRIDIMENSIONAL

A última saída do processo de *inpainting* será o vídeo tridimensional, que mostrará as áreas de oclusão preenchidas à medida que os componentes da panorâmica se aproximam. Para demonstrar o desempenho e os resultados da aplicação dessa técnica foram selecionadas duas regiões em cada panorâmica de entrada, em seguida as mesmas regiões foram recortadas dos vídeos com *inpainting*.

As regiões selecionadas para a primeira panorâmica estão delimitadas na Figura 69, na qual a região numerada como 1 evidenciará o preenchimento das áreas de oclusão no entorno das pedras, enquanto a região 2 mostrará o *inpainting* ao redor da vegetação.

Figura 69 - Regiões selecionadas na panorâmica 1

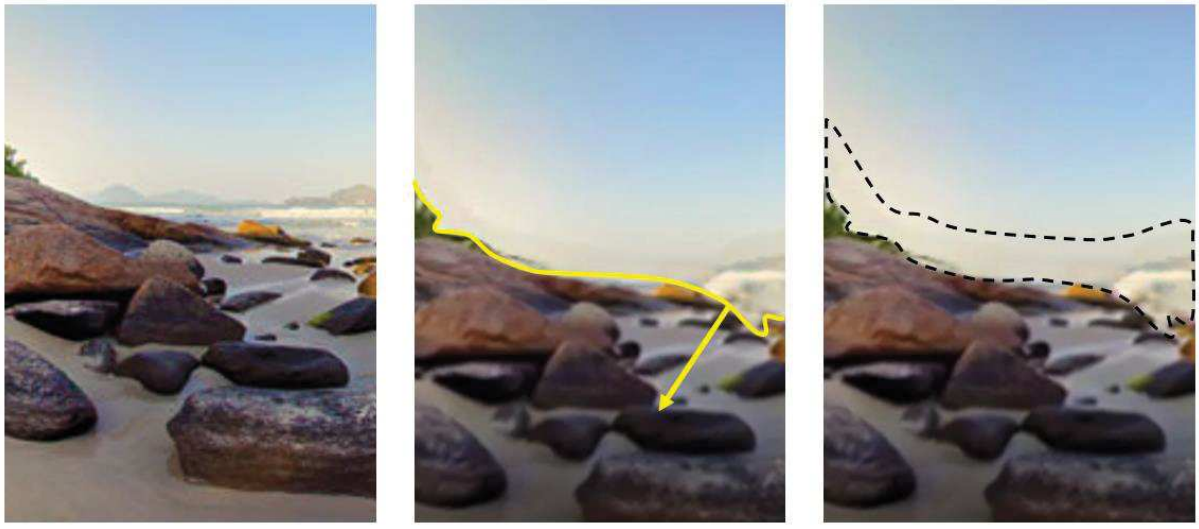


Fonte: Elaborada pela autora (2021)

As mudanças realizadas sobre a região 1 são mostradas pela Figura 70. No primeiro quadro é mostrado o recorte da imagem original, apenas ampliada, enquanto no segundo é evidenciado a mesma região, porém com o *inpainting* já realizado, demarcando com uma linha o momento de separação entre *foreground* do *background.* A seta acrescentada indica o

direcionamento do deslocamento desta porção da imagem ao realizar o efeito de aproximação da cena. E o terceiro quadro limita a região que sofreu o *inpainting,* a fim de facilitar a percepção em relação ao primeiro quadro.

Figura 70 - Região 1 da panorâmica 1



Fonte: Elaborada pela autora (2021)

Com a aproximação do plano frontal definido é possível perceber que nesta parcela da panorâmica ocorreu o preenchimento ao redor das pedras de modo uniforme em relação ao céu, surgindo um esboço no complemento da vegetação e uma extensão da onda do mar localizada a direita. As ilhas ao fundo do primeiro quadro desapareceram devido a aproximação do primeiro plano, que se sobrepôs a elas durante a movimentação.

Os efeitos incidentes sobre a região 2 podem ser observados através da Figura 71, que segue a mesma ordem de quadros da figura anterior. Neste exemplo é mais fácil de evidenciar o preenchimento ocorrido, visto que é perceptível uma leve transição entre o plano frontal e o fundo original da imagem. Pelo segundo quadro nota-se o a demarcação do morro, assim como a perda de alguns contornos da vegetação antes presentes no primeiro quadro. Na última representação percebe-se que o algoritmo adotou as pontas da árvore como parte do plano posterior, portanto ao realizar o *inpainting*, esses são estendidos formando duas linhas paralelas.

Figura 71 - Região 2 da panorâmica 1

Fonte: Elaborada pela autora (2021)

Para a segunda panorâmica foram definidas as regiões mostradas pela Figura 72, onde a primeira delas abrange uma porção da região externa da imagem situada na lateral da porta, enquanto a segunda faz a análise sobre um objeto situado no ambiente interno.

Figura 72 - Regiões selecionadas na panorâmica 2

Fonte: Elaborada pela autora (2021)

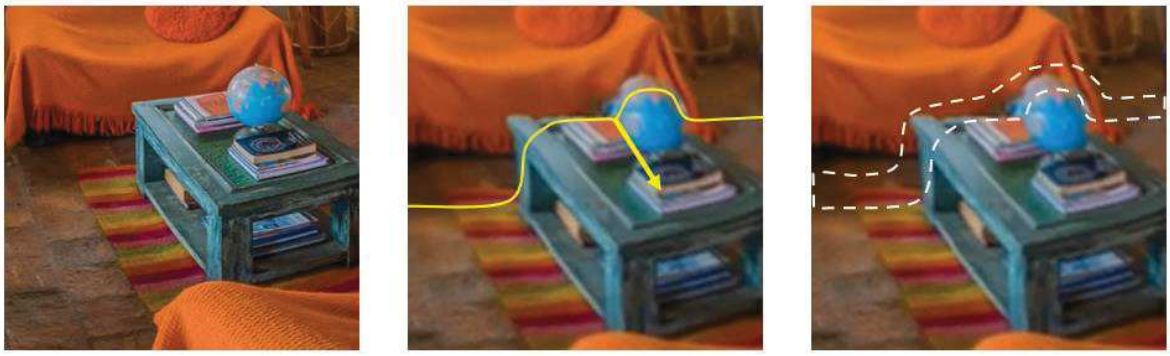
A região 1 abrange parte da porta azul e parte da área externa, o que representada o primeiro quadro da Figura 73. Assim como nos exemplos retirados da primeira panorâmica, o segundo quadro evidencia o limite do plano frontal e a direção em que a região será movida durante a aproximação. Através dessa marcação é possível notar que a barra lateral da porta não foi definida de forma alinhada, fato que se deve a imprecisão das bordas fornecidas pelo MiDaS. No último quadro é possível perceber o esticamento da região externa, que não completou devidamente apenas o acessório circular suspenso, isso porque as informações vizinhas não lhe permitiram completa-lo adequadamente.

Figura 73 - Região 1 da panorâmica 2

Fonte: Elaborada pela autora (2021)

A região 2 representa uma mesa de centro localizada no interior da sala, que pode ser visualizada pela Figura 74. Através dos três quadros é possível observar o preenchimento da área de oclusão localizada na parte posterior da mesa, gerando um aumento na parte esquerda da mesa e uma extensão no enfeite de globo.

Figura 74 - Região 2 da panorâmica 2



Sem *Inpainting*

Delimitação do *Foreground* + Direção da aproximação

Área Preenchida

Fonte: Elaborada pela autora (2021)

O processo completo de *inpainting* é demorado, podendo variar bastante a quantidade de minutos, dependendo da qualidade ajustada e da entrada inserida. Para a primeira panorâmica, foi ajustado uma largura de quadro de vídeo de 960 pixels, o que ocasionou uma demora de aproximadamente 9 minutos, e um tamanho de 214 KB. Já para a segunda panorâmica foi ajustada para uma largura de quadro de vídeo maior, com 2000 pixels, demorando 28 minutos para terminar o processamento e possuindo um tamanho de 2,38 MB. Essa diferença na largura do quadro do vídeo impacta diretamente na qualidade final da saída, deste modo, é possível perceber que as transições de *inpainting* apresentadas para a primeira panorâmica possuem menos detalhes do que as apresentadas para a segunda panorâmica.

Portanto, o aumento da qualidade da saída do *inpainting* impacta diretamente sobre as transições, tornando-as visíveis nas panorâmicas (Figura 75). Para corrigir esse problema foi necessário alterar o deslocamento do segmento que delimita o *foreground* do *background* em valores maiores de 5. Para o caso da segunda panorâmica, onde foi considerado uma largura de quadro maior, esse valor foi igualado a 10, enquanto que para a primeira foi mantido em 5.

Figura 75 - Deslocamento igual a 5

Fonte: Elaborada pela autora (2021)

Embora os preenchimentos sejam realizados, em sua maioria, com sucesso, o *inpainting* também produz alguns preenchimentos errôneos em virtude do erro de estimativa fornecido pelo mapa de profundidade, o que gera separações definidas de forma imprecisa, resultando em regiões de contexto delimitadas de forma incorreta. Esse erro pode ser visto no mesmo exemplo da porta da região 2 da segunda panorâmica, em que a aplicação do *inpainting* (Figura 76) gerou preenchimentos incorretos em virtude da delimitação incorreta da lateral da porta, deste modo, os detalhes em azul foram transferidos para a região de síntese, que se misturaram com as informações da área externa.

Figura 76 - Preenchimento incorreto

Fonte: Elaborada pela autora (2021)

Os vídeos resultantes possuem formato .MP4 e foram ajustados para apresentar uma duração de 12 segundos, porém esse valor pode ser alterado ajustando o número de frames por segundo, que para esse tempo determinado foi adotado com 40 fps.

6.5 VÍDEO ESFÉRICO

Com a saída do *inpainting*, realiza-se então a texturização na esfera, gerando um vídeo esférico. Nesta etapa a qualidade escolhida através da definição da largura de quadro do vídeo é de extrema importância, pois impacta diretamente na qualidade do vídeo esférico final. A Figura 77 apresenta recortes retirados dos vídeos esféricos gerados pela panorâmica equiretangular 1 e 2, onde é possível observar essa diferença da qualidade resultante.

Figura 77 – Comparação da qualidade dos vídeos esféricos

Fonte: Elaborada pela autora (2021)

Ao contrário do processo de *inpainting*, a conversão do vídeo tridimensional em vídeo esférico é feita de maneira rápida, fornecendo vídeos com o mesmo formato .MP4. O vídeo obtido através da panorâmica 1, tem $302\ KB$ de extensão e quadro de dimensão $960\ x\ 480$, enquanto o vídeo resultante para a imagem 2 tem $2{,}38\ MB$ de extensão e quadro com dimensão $2000\ x\ 1000$.

7 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo será apresentada, na seção 7.1, as conclusões obtidas sobre o trabalho, e na seção 7.2 serão propostas algumas melhorias para trabalhos futuros.

7.1 CONCLUSÃO

Neste trabalho foi apresentada uma aplicação inovadora a partir de uma panorâmica equiretangular, na qual foi aplicada a técnica de *inpainting* para o preenchimento de áreas de oclusões originadas do movimento de parallax, texturizando o vídeo tridimensional obtido sobre uma esfera. O resultado gerado foi um vídeo esférico capaz de fornecer ao observador a imersão interativa no ambiente retratado pela panorâmica, aproximando os objetos de modo a tornar imperceptíveis as transições ocorridas.

O processo foi realizado em 3 grandes etapas: pré-processamento, *inpainting* e texturização no domo tridimensional. Na primeira delas ocorreu a geração do mapa de profundidade gerado pelo modelo MiDaS, cujo resultado se mostrou válido não apenas para imagens planas, mas também para as panorâmicas equiretangulares. Apesar de mostrar um desempenho favorável, ainda é uma estimativa imperfeita, que gera profundidades aproximadas e transições suavizadas entre as regiões de maior e menor profundidade.

Ainda durante a etapa de pré-processamento, foi feita a formação da LDI inicial e a segmentação do plano frontal que forma o *foreground* da imagem. A utilização da estrutura LDI propiciou enormes vantagens em todo o decorrer do processo, como a junção entre os pixels de cor e de profundidade de forma compacta, a conectividade da estrutura e a capacidade desta de se ampliar em diversas camadas, gerando uma estrutura facilmente texturizavel. Quanto ao processo para se obter os segmentos, todas as filtragens apresentaram ótimos resultados, desde o filtro Bilateral para filtrar o mapa de profundidade, até a filtragem realizada pelo mapa binário para a extração das marcações indesejadas. A obtenção de segmentos separados ao invés de um único contorno permitiu que o *inpainting* fosse realizado de maneira localizada.

A etapa de *inpainting*, apresentou em sua parte inicial o rompimento da ligação entre os pixels localizados na fronteira de descontinuidade, que foi realizado com as informações dos segmentos obtidas baseadas no mapa de profundidade alcançado pelo MiDaS, portanto, algumas desconexões indevidas que ocorreram nesta etapa são em virtude do erro propagado

pela estimação de profundidade. Posterior à desconexão, foi feita a criação iterativa das regiões de síntese e da região de contexto, realizadas por algoritmos simples de preenchimento sucessivos, que forneceram a base importante para a entrada da *Edge Network*.

A rede neural convolucional utilizada, é extensa e subdivida em 3 sub-redes: *Edge Network*, *Color Network* e *Depth Network.* A *Edge Network* apresentou excelente desempenho em recriar a região de síntese a partir do treinamento entre rede geradora e rede discriminadora, principalmente pelo fato de possuir blocos residuais em sua composição, o que propiciou uma rede mais densa sem esforços extras. Essa redefinição da região de síntese, garante que as redes de coloração e de profundidade realizem seus preenchimentos de forma alinhadas. Caso a *Edge Network* delimitasse essas regiões de forma independente para as duas sub-redes, poderia ocorrer a geração de duas regiões de síntese levemente distintas, resultando em junções errôneas de cor e profundidade.

A *Color Network* e a *Depth Network,* apesar de possuírem entradas e saídas diferentes, são formadas pela mesma arquitetura de rede neural convolucional, o que gera o mesmo grau de efetividade. As saídas dessas redes são realocadas sobre a LDI, criando estruturas tridimensionais capazes de representar situações complexas de profundidade, variando a sua forma conforme a distribuição de profundidades na panorâmica. O processo de *inpainting* é bastante demorado, devido a utilização e a reaplicação da rede neural para garantir o preenchimento total das lacunas.

O vídeo tridimensional, criado a partir da LDI texturizada, apresentou resultados excelentes para a aplicação em panorâmicas, preenchendo áreas de oclusão em torno de diversos objetos da cena. Quanto maior a quantidade de objetos da cena, maior será o preenchimento das áreas de oclusão, isso porque haverá mais segmentos marcados no plano frontal para receber o *inpainting*. Em relação à qualidade dos vídeos gerados, quanto maior o quadro de largura do vídeo melhor será a resolução, porém, mais demorado será o processo de preenchimento e haverá a necessidade de mudar o parâmetro de deslocamento para evitar transições abruptas.

Finalizando as 3 grandes etapas, realizou-se a projeção do vídeo tridimensional sobre um domo, que na prática é executada sobre a esfera, em virtude da facilidade matemática e da semelhança no resultado final. O vídeo esférico foi gerado rapidamente e, resultou em uma ambientação do usuário no cenário representado pela panorâmica equiretangular de entrada, com a possibilidade de se movimentar ao redor da cena e de visualizar a aproximação de seus

componentes. Apesar de promissor, o resultado final, ainda precisa de ajustes, principalmente quanto a junção entre os extremos horizontais.

Conclui-se então que a aplicação proposta neste trabalho, abre campos para outras pesquisas relacionadas, onde almeja-se a projeção de itens de um ambiente panorâmico, com a finalidade de fornecer proximidade ao observador.

7.2 TRABALHOS FUTUROS

Para melhorar o desempenho desta aplicação, uma das possíveis adaptações é a substituir o mapa de estimação utilizado pelo MiDaS, porque apesar de eficiente este não é perfeito, e embora seja feita a etapa de pré-processamento, alguns erros ainda são propagados por essa etapa. Uma abordagem para solucionar esse problema seria aquisionar as imagens que compõem a panorâmica juntamente com o laser, de modo que a panorâmica equiretangular gerada, possua informações confiáveis de profundidade.

Outra adaptação é em relação ao referencial em que é realizado o *inpainting*, pois apenas o processo realizado a partir da região central da panorâmica, gera o problema das junções das extremidades no vídeo esférico final. Duas abordagens podem ser seguidas para tentar solucionar este problema. A primeira delas é ampliar o número de referências em que se aplica o processo, de forma a concatenar diferentes pontos de vista para o observador central, e a segunda opção seria realizar *inpainting* isolados, colocando-os em partes especificas de *tour* virtual, onde o usuário teria a opção de se aproximar ou não de uma área desejada.

REFERÊNCIAS

1 JESSELL, BETTINA. **Helmut Ruhemann's Inpainting Techniques.** 1977. Journal of the American Institute for Conservation, 17:1, 1-8, doi: 10.1179/019713677806029275

2 OLIVEIRA, MANUEL; BOWEN, BRIAN; McKENNA, RICHARD; CHANF, YU-SUNG. 2001. **Fast Digital Image Inpainting**. 261-266.

3 ZHANG, YINDA; FUNKHOUSER, THOMAS. **Deep Depth Completion of a Single RGB-D Image**. 2018. IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 175-185, doi: 10.1109/CVPR.2018.00026

4 FU, ZEQING; HU, WEU; GUO, ZONGMING. 2020. 3d Dynamic **Point Cloud Inpainting Via Temporal Consistency On Graphs**. 1-6. 10.1109/ICME46284.2020.9102861.

5 CHANG, YA-LIANG; YU LIU, ZHE; LEE, KUAN-YIANG; HS. 2019 **Free-Form Video Inpainting With 3D Gated Convolution and Temporal PatchGAN**. IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 9065-9074, doi: 10.1109/ICCV.2019.00916.

6 GUILLEMOT, CHRISTINE; LE MER, OLIVIER. 2014. **Image Inpainting: Overview and Recent Advances**. Signal Processing Magazine, IEEE. 31. 127-144. doi: 10.1109/MSP.2013.2273004.

7 BELARMÍO, MARCELO; SAPIRO, GUILLERMO; CASELLES, VICENT; BALLESTER, C.. 2000. **Image inpainting**. Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. 417-424.

8 XIE, JUNYUAN; XU, LINLI; CHEN, ENHONG. 2012. **Image Denoising and Inpainting with Deep Neural Networks**. Advances in Neural Information Processing Systems. 1.

9 BOROLE, RAJESH PANDURANG; BONDE, SANJIV VEDU. **Patch-Based Inpainting for Object Removal and Region Filling in Images**. 2013. Journal of Intelligent Systems, vol. 22, no. 3, pp. 335-350. https://doi.org/10.1515/jisys-2013-0031

10 SHADE, JONATHAN; GORTLER, STEVEN; HE LI-WEI; SZELISKI, RICHARD.1998. **Layered depth images**. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 231-242. ACM.

11 LU, SHAOPING; HANCA, JAN, MUNTEANU, ADRIAN; SCKELKENS, PETERS. **Depth-based view synthesis using pixel-level image inpainting**. 2013. 18th International Conference on Digital Signal Processing (DSP), Fira, Greece, 2013, pp. 1-6, doi: 10.1109/ICDSP.2013.6622773.

12 SHIH, MENG-LI; SU, SHIH-YANG; KOPH, JOHANNES, HUANG, JIA-BIN.2020. **3D Photography Using Context-Aware Layered Depth Inpainting**,2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 8025-8035, doi: 10.1109/CVPR42600.2020.00805.

13 PATHAK, DEEEPAK; KRAHENBUHL, PHILLIPP; DONAHUE, JEFF; DARRELL, TREVOR; EFROS, ALEXEI. 2016. **Context Encoders: Feature Learning by Inpainting**. 2536-2544. 10.1109/CVPR.2016.278.

14 ALESSANDRO. Diferentes tipos de fotos panorâmicas-Como elas podem ajudar? **Blog NPossibilidades.** 19 de agosto de 2019. Disponível em: http://npossibilidades.com.br/n/diferentes-tipo-de-fotos-panoramicas-como-elas-podem-ajudar/. Acesso 19 de março de 2021.

15 **CuteWallpaper.org**. Disponível em: https://cutewallpaper.org/download.php?file=/21/360-degree-wallpaper-free-download/Degree-seamless-panoramic-7768-Seamless-360-degree-.jpg. Acesso 19 de março de 2021.

16 LASINGER, KATRIN; RANFTL, RÉNE; SCHINDLER, KONRAD, KOLTUN, VLADLEN. **Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer.** 2019. arXiv preprint arXiv: 1907.01341.

17 GOODFELLOW, IAN; POUGET-ABADIE, JEAN; MIRZA, MEHDI; XU, BING; WARDE-FARLEY, DAVID; OZAIR, SHERJIL; COURVILLE AARON; BENGIO, Y. 2014. **Generative Adversarial Networks**. Advances in Neural Information Processing Systems. 3. 10.1145/3422622.

18 **GENERATIVE ADVERSARIAL NETWORKS**. Disponível em: https://developers.google.com/machine-learning/gan/gan_structure. Acesso 19 de março de 2021.

19 NAZERI, KAMYAR; NG, ERIC; JOSEPH, TONY; QURESHI, Z FAISAL; EBRANHIMI, MEHRAN. 2019.K. **Edgeconnect: Generative image inpainting with adversal edge learning**. *arXiv preprint*.

20 SHIH, MENG-LI; SU, SHIH-YANG; KOPH, JOHANNES, HUANG, JIA-BIN.2020. **3D Photography Using Context-Aware Layered Depth Inpainting -Supplementary Material**,2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 8025-8035, doi: 10.1109/CVPR42600.2020.00805.

21 MIYATO, TAKERU; KATAOKA, TOSHIKI; KOYAMA, MASANORI; YOSHIDA, YUICHI. 2018. **Spectral Normalization for Generative Adversarial Networks**. In International Conference on Learning Representation.

22 KINGMA, P DIEDERIK; BA, JIMMY. 2015.Adam: **A method for stochastic optimization**. In ICLR.

23 RONNEBERGER, OLAF; FISCHER, PHILIPP; BROX, THOMAS. 2015. **U-net: Convolutional networks for biomedical image segmentation.** In MICCAI.

24 LIU, GUILIN; REDA, FITSUM A; SHIH, KEVIN J, WANG, TING-CHUN; TAO, ANDREW; CATANZARO, BRYAN. 2018. **Image inpainting for irregular holes using partial convolutions.** In ECCV.

25 SIMONYAN, KAREN; ZISSERMAN, ANDREW. **Very deep convolutional networks for large-scale image recognition.** 2014. arXiv preprint arXiv: 1409.1556.

26 LIN, TSUNG; MAIRE, MICHEL; BELONGIE, SERGE; HAYS, JAMES; PERONA, PIETRO; RAMANAN, DEVA; DOLLÁR, PRIOR; ZITNICK, C... 2014. **Microsoft COCO: Common Objects in Context.**

27 LI, ZHENGQI; SNAVELY. **Megadepth: Learning single-view depth prediction from internet photos.** 2018. In CVPR.

28 GUERRA, MARCOS. Foto panorâmica 360 Graus enriquece a imagem da sua empresa. **Blog Ubatuba Cobra**. 9 de setembro de 2011. Disponível em: http://ubatubacobra.blogspot.com/2011/09/foto-panoramica-360-graus-enriquece.html. Acesso 19 de março de 2021.

29 FELIPE. I will stich panoramic 360 images in high quality. **fiverr.** Disponível em: https://www.fiverr.com/felipeafa89/stitch-panorama-360-images-in-high-quality. Acesso 19 de março de 2021.