

UNIVERSIDADE FEDERAL DE ITAJUBÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
CIÊNCIA E TECNOLOGIA DA COMPUTAÇÃO

Metodologia de desenvolvimento de  
*Digital Twin* para o campus Unifei -  
Itabira

Alexandre Leles Sousa

Agosto de 2024  
Itajubá

Trabalho de dissertação de mestrado submetido ao Programa de Pós-Graduação em Ciência e Tecnologia Da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciências da Computação.

Área de Concentração: Matemática Da Computação

Orientador: Dr. Giovani Bernardes Vitor

Coorientador: Dr. Érick Oliveira Rodrigues

Agosto de 2024

Itajubá

Alexandre Leles Sousa

**Metodologia de desenvolvimento de *Digital Twin*  
para o campus UNIFEI - Itabira**

Trabalho de qualificação submetido ao Programa de Pós-Graduação em Ciência e Tecnologia da Computação como parte dos requisitos para obtenção do Título de Mestre em Ciências da Computação.

Área de concentração: Matemática da computação.

***Banca Examinadora:***

Prof. Dr. Giovani Bernardes Vitor

Dr. Erick Oliveira Rodrigues

Prof. Dr. Rodrigo Aparecido da Silva Braga

Prof. Dr. Roberto Silva Netto

Itajubá, Agosto de 2024



# Agradecimentos

*Meus mais sinceros agradecimentos a Deus e a todos os serem que me ajudaram  
nesta jornada.*

*"The price of freedom is steep. Embrace your dreams and, whatever happens,  
protect your honor as SOLDIER!"*

*Zack Fair*

# Resumo

Esta dissertação apresenta a metodologia utilizada para a criação de um *digital twin* para parte do campus da UNIFEI - Itabira. Para uma melhor compreensão deste processo também é apresentada uma breve explicação sobre os conceitos e processos necessários para a criação de modelos 3D até a etapa de texturização. O cenário 3D e todos os projetos desenvolvidos durante a execução deste trabalho foram disponibilizados a fim de que sirvam tanto como um instrumento de estudo para aqueles interessados em modelagem 3D, quanto como um ponto de partida para pesquisas futuras relacionadas ao campus Itabira. Além disso, foi realizado o treinamento da rede *cyclegan* para fazer a transferência de estilo das imagens capturadas no campus virtual para se assemelharem ainda mais às imagens capturadas no campus real. O modelo obtido cumpriu o seu papel em representar virtualmente a região desejada do campus e o treinamento da *cyclegan* apresentou bons resultados deixando o caminho aberto para pesquisas futuras relacionadas ao treinamento de veículos autônomos neste ambiente virtual.

**Palavras-chave:** Fotogrametria, Modelagem 3D, Digital twins, Transferência de estilo.

# Abstract

This dissertation presents the methodology used to create a digital twin for part of the UNIFEI - Itabira campus. For a better understanding of this process, a brief explanation of the concepts and processes necessary to create 3D models until the texturing stage is also presented. The 3D scenario and all projects developed during the execution of this work were made available in order to serve both as a study tool for those interested in 3D modeling and as a starting point for future research related to the Itabira campus. In addition, the cycleGAN network was trained to transfer the style of images captured in the virtual campus to more closely resemble the images captured in the real campus. The model obtained fulfilled its role in virtually representing the desired region of the campus and the cycleGAN training presented good results, leaving an opened way for future research related to the training of autonomous vehicles in this virtual environment.

**Keywords:** Photogrammetry, 3D modeling, Digital twins, Style transfer.

# Lista de ilustrações

Figura 1 – Aprendizado contrastivo para tradução imagem-a-imagem não pareadas. Imagem retirada de [1]. . . . .	18
Figura 2 – Processos básicos da criação de <i>assets</i> 3D . . . . .	21
Figura 3 – Exemplo da utilização de UDIM para aplicação de textura . . .	23
Figura 4 – Exemplo de Mapa UV de um cubo e projeção de sua textura. . .	24
Figura 5 – Exemplos dos principais mapas de texturas para materiais PBR	26
Figura 6 – Representação das UDIMs no <i>UV editor</i> do Blender . . . . .	27
Figura 7 – Exemplo de Mapa UV de um cubo e projeção de sua textura . .	28
Figura 8 – Representação das UDIMs no <i>UV editor</i> do Blender. . . . .	29
Figura 9 – Exemplo de texturas <i>trim sheet</i> . . . . .	30
Figura 10 – Imagem de satélite com RI do Campus Itabira destacada. . . .	31
Figura 11 – Representação da RI e identificação das três sub-regiões definidas.	34
Figura 12 – Representação esquemática do processo de geração de imagens para o P1 e exemplo de imagem gerada. . . . .	35
Figura 13 – Representação esquemática do processo de geração de imagens para o P2 e exemplo de imagem gerada. . . . .	35
Figura 14 – Imagens de satélite da RI. . . . .	36
Figura 15 – Processamento parcial do <i>Meshroom</i> para as imagens do P1. . .	37
Figura 16 – Malha gerada para a SR1. . . . .	38
Figura 17 – Malha gerada para a SR2. . . . .	38
Figura 18 – Malhas das sub-regiões 1 e 2 posicionadas no Blender. . . . .	39
Figura 19 – Malhas 3D das sub-regiões modeladas e posicionadas no Blender.	39
Figura 20 – Malhas 3D das construções da SR1 e SR2 texturizadas. . . . .	40
Figura 21 – Cenário virtual do campus Itabira na <i>UE5</i> . . . . .	41
Figura 22 – Exemplo de imagens de entrada fornecidas para a rede. . . . .	44
Figura 23 – Imagens preditas ao longo das gerações . . . . .	44
Figura 24 – Região do P1 no projeto final da <i>UE5</i> . . . . .	50
Figura 25 – Imagens capturadas em diferentes regiões do DT na <i>UE5</i> . . . .	51

# Lista de abreviaturas e siglas

2D	<i>two-dimensional</i> ou bidimensional
3D	<i>three-dimensional</i> ou tridimensional
AO	<i>Ambient occlusion</i> ou oclusão de ambiente
CAD	<i>Computer-Aided Design</i> ou Design Assistido por Computador
DT	<i>Digital Twin</i> ou Gêmeo Digital
GPU	<i>Graphics processing unit</i> ou Unidade de processamento gráfico
IoT	<i>Internet of things</i> ou internet das coisas
LiDAR	<i>Light Detection and Ranging</i> ou Detecção e Varredura por Luz
PBR	<i>Physically based rendering</i> ou Renderização Baseada em Física
RAM	<i>Random access memory</i> ou memória de acesso aleatório
RGB	<i>Red, Green, Blue</i>
RI	Região de interesse
SR1	sub-região 1
SR2	sub-região 2
SR3	sub-região 3
TD	<i>Texel density</i>
UDIM	<i>U-Dimension</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Considerações Gerais	13
1.2	Objetivos	14
1.3	Estrutura do Trabalho	14
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>16</b>
2.0.1	Trabalhos relacionados a modelagem 3D e Digital Twins	16
2.1	Introdução à Tradução Imagem-a-Imagem	17
2.2	Métodos Base: CycleGAN e Pix2Pix	17
2.3	Trabalhos Relacionados em Tradução Imagem-a-Imagem Não Supervisionada	19
2.4	Trabalhos Relacionados em Tradução Imagem-a-Imagem Específica para Domínio	19
2.5	Trabalho Relacionado em Tradução Imagem-a-Imagem Baseada em Vídeo	20
<b>3</b>	<b>CONCEITOS E ETAPAS BÁSICAS DA CRIAÇÃO DE OBJETOS 3D</b>	<b>21</b>
3.1	Criação de modelos 3D	21
3.2	Modelagem 3D	21
3.3	Mapeamento UV	22
3.4	Texturização	24
3.4.0.1	Materiais PBR	24
3.4.0.2	<i>Texel density</i>	27
3.4.1	Técnicas de texturização	28
3.4.2	Texturas <i>unique</i>	28
3.4.3	Texturas <i>trim</i>	29
<b>4</b>	<b>MATERIAIS E MÉTODOS</b>	<b>31</b>
4.1	Modelagem do ambiente virtual	31

4.1.1	O <i>Software</i> Meshroom . . . . .	32
4.1.2	Captura das imagens . . . . .	32
4.1.3	Processamento das Imagens da SR1 e SR2 . . . . .	36
4.1.4	Modelagem 3D das sub-regiões . . . . .	39
4.1.5	Texturização dos modelos 3D . . . . .	40
4.1.6	Montagem do cenário virtual na <i>Unreal engine 5</i> . . . . .	40
<b>4.2</b>	<b>Treinamento da transferência de estilo . . . . .</b>	<b>41</b>
4.2.1	Captura e tratamento das imagens de treinamento . . . . .	41
4.2.2	Treinamento CycleGAN . . . . .	42
<b>5</b>	<b>RESULTADOS E DISCUSSÃO . . . . .</b>	<b>43</b>
<b>5.1</b>	<b>Resultados da criação do DT . . . . .</b>	<b>43</b>
<b>5.2</b>	<b>Resultados do treinamento da transferência de estilo . . . . .</b>	<b>44</b>
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>46</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>47</b>
	<b>APÊNDICES . . . . .</b>	<b>49</b>
	<b>APÊNDICE A – IMAGENS DO <i>DIGITAL TWIN</i> NA UN- REAL ENGINE 5 . . . . .</b>	<b>50</b>

# 1 Introdução

## 1.1 Considerações Gerais

A modelagem 3D como subarea da computação gráfica teve as principais bases dos seus algoritmos estabelecidas nos anos 60 e 70. Desde então muitos destes algoritmos vem sendo iterados e suas aplicações são aprimoradas juntamente com a evolução dos *hardwares* e poder computacional disponível para a humanidade. Num período de aproximadamente 60 anos o poder da computação gráfica passou de simples linhas e caracteres projetadas em uma tela para simulação completa de ambientes 3D imitando a realidade no que diz respeito a forma de objetos, cálculos de iluminação e aplicação de física sobre estes objetos. Dentre os nomes que possuem grande peso no desenvolvimento dessas bases estão: Ivan Sutherland<sup>1</sup>, Pierre Étienne Bézier<sup>2</sup>, Edwin Catmull<sup>3</sup> e James F. Blinn<sup>4</sup>.

Em 1963 Sutherland[2] criou o *Sketchpad*, que é considerado o primeiro *software* de desenho gráfico interativo, no qual os usuários podiam interagir com polígonos 2D ou objetos gráficos, utilizando um mouse ou uma caneta-ótica, para realizar operações como translação, rotação e escalonamento destes objetos.

Bézier[3], com o seu trabalho acabou popularizando as curvas de Bézier. Curvas essas que são amplamente utilizadas na geração superfícies 3D de maneira precisa, e por consequência, sendo extremamente úteis para a engenharia e computação gráfica.

Edwin Catmull foi aluno de Ivan Sutherland, e dentre as suas publicações forneceu grandes contribuições em algoritmos para subdivisão de superfícies[4]. Processo esse que é muito utilizado na modelagem 3D para suavizar ou adicionar mais detalhes à superfície de um objeto.

---

<sup>1</sup> Ivan Sutherland: [https://amturing.acm.org/award\\_winnners/sutherland\\_3467412.cfm](https://amturing.acm.org/award_winnners/sutherland_3467412.cfm).

<sup>2</sup> Pierre Étienne Bézier: <https://history.siggraph.org/person/pierre-bezier/>.

<sup>3</sup> Edwin Catmull: <https://computerhistory.org/profile/edwin-catmull/>.

<sup>4</sup> James F. Blinn: <https://history.siggraph.org/person/james-jim-f-blinn/>.

James Blinn teve contribuições relacionadas ao mapeamento de texturas de objetos e sua interação com a luz, como em [5]. Campos estes que são fundamentais para atingir realismo na renderização de objetos 3D.

A partir dos fundamentos citados e com a adição de diversas melhorias e aumento do poder computacional ao longo dos anos, hoje é possível recriar ambientes reais em um mundo virtual, de maneira mais visual e com uma carga de cálculos matemáticos extremamente reduzida, por meio do auxílio de *softwares* de modelagem 3D. Assim sendo, o presente trabalho faz uso das facilidades proporcionadas pela evolução da computação gráfica para expor uma metodologia para o desenvolvimento de um *Digital Twin*(DT) da Unifei - campus Itabira e realizar um treinamento de transferência de estilo de imagens a fim de tornar as imagens capturas do DT do campus semelhante às imagens capturadas no campus real.

## 1.2 Objetivos

Um DT é uma ferramenta poderosa e permite a realização de diversas simulações do ambiente que ele representa. Desse modo, esta dissertação tem como objetivo criar e disponibilizar publicamente um DT de parte do campus Itabira a fim de que ele sirva como uma ferramenta para pesquisas futuras relacionadas ao campus. Este DT pode ser utilizado como uma base para pesquisas relacionadas a IoT e campus inteligente, criação de *tours virtuais*, treinamento de veículos autônomos nas vias do campus, dentre outros.

Além disso, com a disponibilização de todos os projetos desenvolvidos busca-se aumentar o interesse em modelagem 3D por parte dos estudantes e propagar o conhecimento em relação às técnicas e ao fluxo de trabalho necessário para a criação de modelos 3D.

## 1.3 Estrutura do Trabalho

Deste ponto em diante a organização deste trabalho ocorre da seguinte maneira: No capítulo II é apresentado o referencial teórico e trabalhos relacionados ao tema. O capítulo III aborda conceitos de modelagem 3D e discorre acerca das

---

etapas e técnicas necessárias para a criação de objetos 3D, desde a elaboração da sua forma até a criação de materiais e texturas. No capítulo IV são explicadas em detalhes as etapas de criação do ambiente virtual do campus Itabira, além disso também é descrito o processo de captura das imagens reais e virtuais para aplicação da transferência de estilo por meio da CycleGAN. O capítulo V expõe os resultados da aplicação da transferência de estilo. E, por fim, o capítulo VI apresenta as conclusões do trabalho.

## 2 Referencial Teórico

Neste capítulo, concentram-se os trabalhos relacionados que exploram a primeira etapa da construção de Digital Twins (DTs), com ênfase na fotogrametria e modelagem 3D a partir de reconstruções tridimensionais.

Posteriormente é apresentado o referencial teórico que fundamenta este trabalho em relação a transferência de imagens. Inicialmente, é fornecida uma visão geral da tradução imagem-a-imagem, destacando sua importância e aplicações, especialmente no contexto da tradução não supervisionada.

### 2.0.1 Trabalhos relacionados a modelagem 3D e Digital Twins

Sommer et al.[6] propõem um método automatizado para gerar DTs, visando reduzir o tempo e o esforço manual envolvidos no processo tradicional. A abordagem é validada em um ambiente industrial, demonstrando a eficácia da identificação de objetos de interesse por meio de redes neurais convolucionais.

Aliaga e Niyogi[7] destaca a importância dos DTs urbanos para simular eventos climáticos extremos. O estudo compara a precisão de modelos 3D gerados por fotogrametria e LiDAR, evidenciando o potencial das técnicas de inteligência artificial generativa para criar modelos detalhados a partir de imagens aéreas e de satélite.

Hellman e Lahti[8] realiza uma revisão abrangente da fotogrametria para a criação de ambientes virtuais 3D. O trabalho apresenta um guia prático para a captura de imagens e a geração de modelos 3D de alta qualidade, além de uma comparação detalhada de diferentes softwares de fotogrametria. O *ContextCapture*<sup>1</sup> é destacado como a ferramenta mais eficiente para a maioria das aplicações.

---

<sup>1</sup> ContextCapture: <https://geo-matching.com/products/contextcapture>.

## 2.1 Introdução à Tradução Imagem-a-Imagem

A tradução imagem-a-imagem é um subcampo da visão computacional que busca aprender mapeamentos entre imagens de entrada e saída. Diferentemente das tarefas tradicionais de processamento de imagem, que frequentemente operam píxel a píxel, a tradução imagem-a-imagem visa capturar e transferir informações semânticas e estruturais de alto nível entre imagens. Essa capacidade tem amplas implicações para aplicações como transferência de estilo, transfiguração de objetos e tradução imagem-a-imagem em domínios específicos, como imagens médicas ou de sensoriamento remoto.

Embora métodos supervisionados de tradução imagem-a-imagem tenham sido explorados como em [9], a escassez de dados de treinamento pareados em muitos cenários do mundo real estimulou um interesse significativo em abordagens não supervisionadas. A tradução imagem-a-imagem não supervisionada busca aprender mapeamentos entre domínios de imagem sem depender de pares de imagens correspondentes explicitamente. Isso representa um desafio considerável, pois o modelo deve estabelecer correspondências significativas com base apenas nas estruturas e conteúdos subjacentes das próprias imagens.

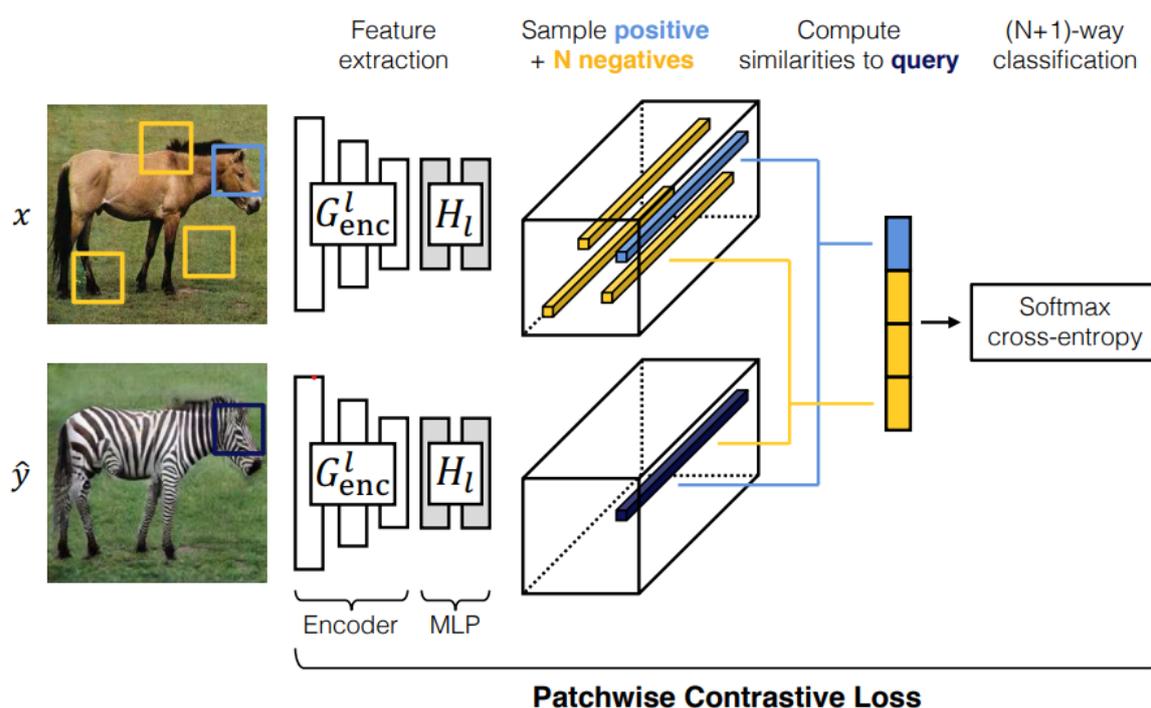
## 2.2 Métodos Base: CycleGAN e Pix2Pix

CycleGAN [1] e Pix2Pix [9] são trabalhos fundamentais no campo da tradução imagem-a-imagem não supervisionada, estabelecendo benchmarks e orientando pesquisas subsequentes.

O CycleGAN introduziu o conceito de consistência cíclica, que impõe um mapeamento entre dois domínios que seja aproximadamente invertível. Ao traduzir iterativamente uma imagem de um domínio para outro e vice-versa, o CycleGAN visa preservar informações semânticas enquanto captura características específicas do domínio. Apesar de seu sucesso, o CycleGAN frequentemente sofre com o colapso de modo, produzindo uma diversidade limitada de imagens geradas. Além disso, a perda de consistência cíclica pode não capturar com precisão o mapeamento desejado entre domínios, especialmente ao lidar com transformações

complexas de imagem. Na Figura 1, pode-se observar um fluxograma simplificado deste método, no qual  $x$  é o domínio da imagem de entrada contendo um cavalo, e  $\hat{y}$  é o domínio da imagem de saída. Em  $\hat{y}$  foi preservado a forma do animal em relação à imagem de entrada, porém as características dos pelos foram alteradas para formar uma zebra.

Figura 1 – Aprendizado contrastivo para tradução imagem-a-imagem não pareadas. Imagem retirada de [1].



O Pix2Pix, por outro lado, utiliza uma estrutura de Rede Adversarial Generativa (GAN) condicional, requerendo dados de treinamento pareados. Ao condicionar o gerador na imagem de entrada, o Pix2Pix pode produzir traduções mais detalhadas e precisas em comparação ao CycleGAN. No entanto, a dependência de dados pareados limita significativamente sua aplicabilidade em cenários do mundo real onde obter tais dados é frequentemente impraticável ou inviável.

## 2.3 Trabalhos Relacionados em Tradução Imagem-a-Imagem Não Supervisionada

Construindo sobre as bases estabelecidas pelo CycleGAN e Pix2Pix, pesquisas subsequentes exploraram diversas técnicas para aprimorar a qualidade, diversidade e controlabilidade da tradução imagem-a-imagem não supervisionada.

Isola et al.[9] introduziu o aprendizado contrastivo ao domínio da tradução imagem-a-imagem. Maximizando a concordância entre visualizações aumentadas da mesma imagem e minimizando a concordância entre imagens diferentes, o aprendizado contrastivo ajuda o modelo a aprender representações mais discriminativas. Essa abordagem demonstrou melhorias na qualidade da imagem e redução do colapso de modo em comparação com métodos tradicionais baseados em GANs.

Pizzati, Cerri e Charette[10] propôs uma abordagem contínua guiada por modelo para gerar traduções imagem-a-imagem diversas e controláveis. Introduzindo um espaço latente contínuo, o CoMoGAN permite interpolação suave entre diferentes estilos ou domínios. Esse método oferece maior flexibilidade e controle sobre as imagens geradas em comparação com abordagens tradicionais baseadas em GANs. No entanto, a complexidade do modelo contínuo e os desafios potenciais no treinamento podem limitar sua aplicabilidade prática.

Xia, Yang e Xue[11] e Huang et al.[12] expandiram a tradução imagem-a-imagem para múltiplos domínios e modalidades. Esses trabalhos exploraram os benefícios de desentrelaçar informações específicas do domínio e de conteúdo, levando a uma generalização e controlabilidade aprimoradas. Embora esses métodos sejam valiosos para cenários multidomínio, sua aplicabilidade direta à tarefa específica de tradução entre ambientes reais e virtuais pode exigir adaptações adicionais.

## 2.4 Trabalhos Relacionados em Tradução Imagem-a-Imagem Específica para Domínio

Embora o treinamento de transferência de estilo realizado neste trabalho seja de tradução imagem-a-imagem de propósito geral, explorar métodos específi-

cos para domínio pode fornecer *insights* valiosos e soluções potenciais.

Wang et al.[13] aborda o desafio de transferir conhecimento de domínios sintéticos para reais na detecção de objetos. Embora não diretamente relacionado à transferência de estilo, este trabalho destaca a importância das técnicas de adaptação de domínio para superar a diferença entre ambientes simulados e do mundo real.

Park e Cho[14] foca na transferência de estilo para cenas 3D, enfatizando o papel da informação semântica na preservação da estrutura e coerência da cena. Este trabalho destaca os benefícios potenciais da incorporação da compreensão semântica em tarefas de tradução imagem-a-imagem.

## 2.5 Trabalho Relacionado em Tradução Imagem-a-Imagem Baseada em Vídeo

Liu et al.[15] aborda os desafios da transferência de estilo de vídeo, enfatizando a importância da consistência temporal e preservação da profundidade. Este trabalho destaca a necessidade de considerar a dinâmica temporal e as relações espaciais em tarefas de tradução imagem-a-imagem envolvendo vídeos.

## 3 Conceitos e etapas básicas da criação de objetos 3D

Para tornar mais compreensível o trabalho de criação do ambiente virtual descrito no capítulo 4, o capítulo atual apresenta alguns conceitos, as principais etapas para a criação de modelos 3D e um dos fluxos de trabalho mais comumente utilizado.

### 3.1 Criação de modelos 3D

A concepção de objetos 3D texturizados possui basicamente as três etapas, as quais estão representadas na Figura 2. Para que estes objetos possam ser representados de forma realista e otimizada, cada uma dessas etapas deve ser realizada atentamente levando em consideração alguns parâmetros fundamentais para a qualidade final dos modelos.

Figura 2 – Processos básicos da criação de *assets* 3D



Cada uma das etapas citadas será explicada em maiores detalhes em sua respectiva seção.

### 3.2 Modelagem 3D

Para a realização de modelagem 3D existem duas abordagens que buscam atender problemas diferentes. São elas a modelagem *Computer-Aided Design* ou Design Assistido por Computador (CAD) e a modelagem poligonal.

A modelagem CAD é aplicada em áreas nas quais especificações técnicas, precisão e controle das formas são de extrema importância; como por exemplo na

engenharia, arquitetura e design automotivo. A precisão do CAD advém do fato que esta abordagem define e manipula as superfícies dos objetos 3D a partir de modelos geométricos, curvas e equações matemáticas. Fato que propicia a realização de simulações precisas. No caso de CAD aplicada ao design automotivo é possível realizar simulações aerodinâmicas muito próximas ao que será obtido na realidade.

Já no caso da modelagem poligonal, que foi o método utilizado na realização deste trabalho, ela se baseia na criação e manipulação de vértices, arestas e faces poligonais para formar uma malha constituída de polígonos que represente um objeto 3D.

A partir deste ponto, a expressão “modelagem 3D poligonal” será referida simplesmente como “modelagem 3D” para fins de simplificação do texto.

As três principais ferramentas da modelagem 3D são as operações de translação, rotação e extrusão. Geralmente a modelagem se inicia a partir de algum sólido geométrico primitivo e o modelador segue modificando os polígonos por meio das operações já citadas e outras mais complexas até atingir a forma desejada.

Geralmente estas operações podem ser executadas de duas maneiras: A primeira é manipulando os polígonos visualmente com o mouse ou algum outro dispositivo como canetas para desenho virtual. A segunda é fornecendo alguns parâmetros como o eixo de execução, graus de rotação e distância de translado nos polígonos desejados.

Vale ressaltar que a complexidade e qualidade dos modelos 3D depende diretamente do conhecimento, prática e habilidade do modelador para com as ferramentas disponíveis nos *softwares*. Isto também é válido para as etapas de mapeamento UV e texturização.

### 3.3 Mapeamento UV

Após a finalização da modelagem 3D é iniciado o processo de mapeamento UV, o qual tem o objetivo de mapear as faces do objeto 3D em um plano bidimensional(2D). Um mapa UV é basicamente um plano 2D contendo coordena-

das determinadas pelos eixos U e V, e este plano é dividido por áreas chamadas de UDIMs. Uma UDIM é a área que representa os limites da textura a ser projetada. Isto é, se estiver utilizando texturas 4k, esta área representa 4096 *pixels* por 4096 *pixels*; já para uma textura 2k ela representará 2048 x 2048px. No mapa UV há uma representação 2D de todas as faces do objeto 3D na UDIM, e dessa maneira ele atua como um mapa para a projeção das texturas no objeto 3D.

À esquerda da Figura 3 está a representação da UDIM feita pelo Blender no *UV editor*. No centro há a projeção da textura do brasão da Unifei na UDIM e destacado em laranja está projetada a representação 2D da face de um plano. À direita desta figura, este plano está renderizado em um ambiente 3D com a sua textura aplicada.

Figura 3 – Exemplo da utilização de UDIM para aplicação de textura



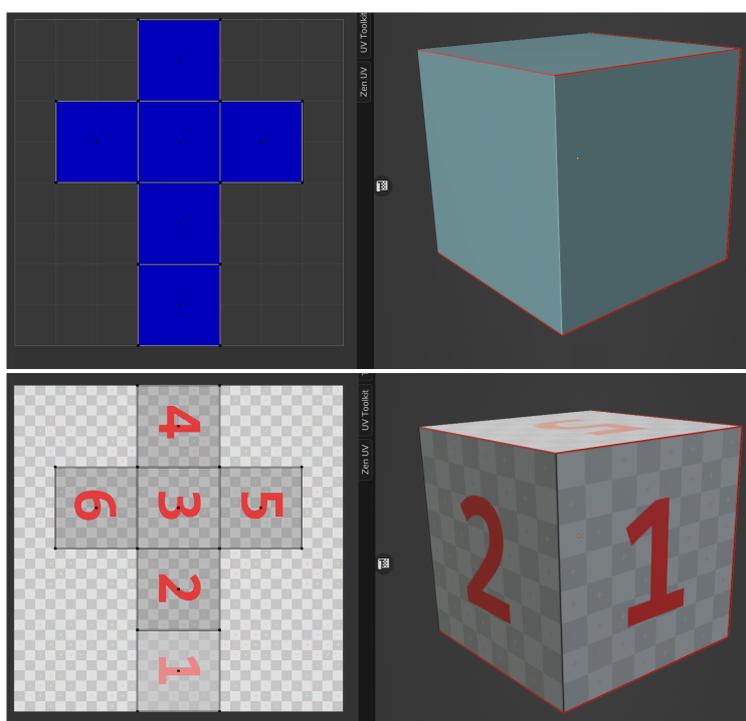
Para criar o mapa UV de algum objeto 3D nos *softwares* de modelagem, o artista realiza a marcação das arestas que podem ser desconectadas para que o objeto 3D possa ser representado em um plano 2D com a menor distorção possível e evitar erros de projeção das texturas. Cada aresta marcada é chamada de *seam* ou costura, e após definir as *seams* basta executar o comando *unwrap* que o *software* irá calcular a projeção 2D das faces em uma UDIM. Como analogia, pode-se dizer que o artista escolhe as arestas que serão *seams* pensando em um processo inverso a criação de um origami, no qual ele parte do origami/objeto 3D montado e vai desfazendo as dobras até retornar a forma original do papel.

O processo de criação de mapas UVs para objetos inicialmente é complicado

e demorado, mas conforme o modelador adquire experiência por meio da prática do raciocínio exigido para esta tarefa, este processo se torna cada vez mais rápido e intuitivo.

Na parte superior Figura 4 à esquerda está o mapa UV gerado para o cubo presente na parte direita com as suas *seams* destacadas em vermelho. Já na parte inferior esquerda na UDIM está a textura do cubo com o seu mapa UV sobreposto à ela. No canto inferior direito é possível observar o cubo no ambiente 3D já com a textura projetada.

Figura 4 – Exemplo de Mapa UV de um cubo e projeção de sua textura.



## 3.4 Texturização

### 3.4.0.1 Materiais PBR

Em um ambiente 3D virtual poligonal, todas as faces e vértices possuem os seus respectivos vetores normais, e estes vetores são utilizados em quase todas as

operações e cálculos realizados em relação aos objetos 3D. Logo eles são essenciais para o funcionamento dos materiais PBR.

Um material é basicamente uma estrutura de dados que contém todas as imagens e configurações necessárias para a renderização do objeto. Os materiais PBR são compostos por um conjunto de mapas de textura que por meio da manipulação dos vetores normais altera a maneira como a luz é refletida nos objetos aos quais eles estão aplicados. Assim, é possível causar ilusões que permitem renderizar os objetos com mais realismo.

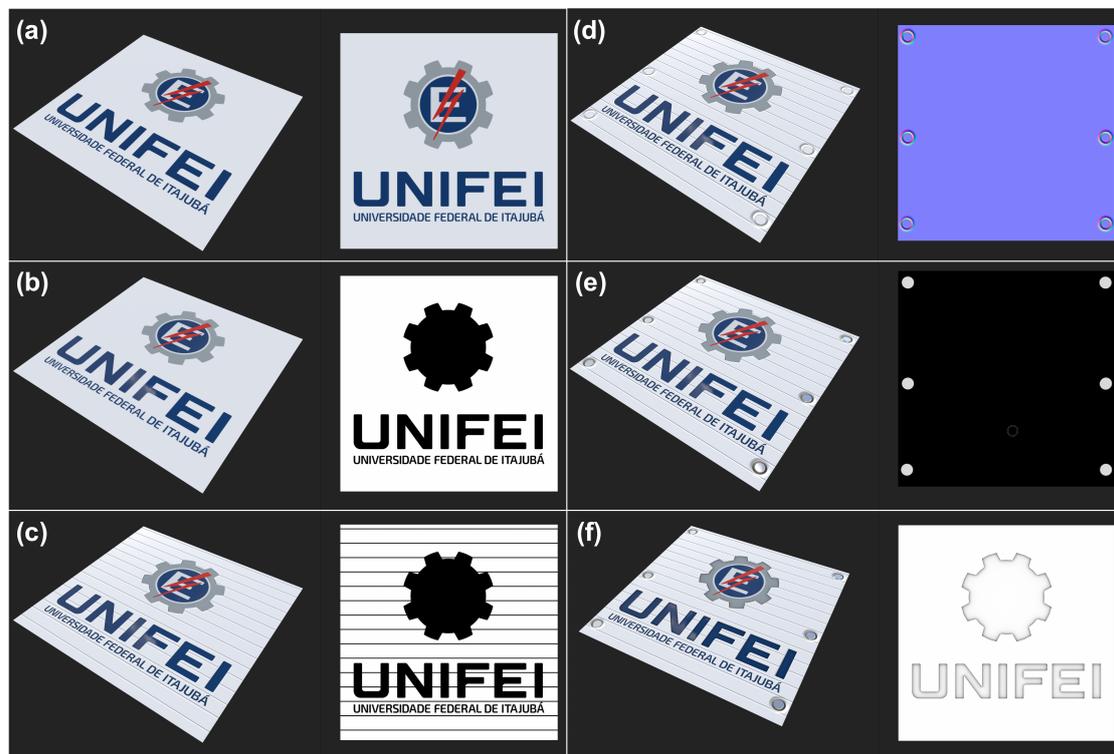
Cada um dos mapas utilizados é responsável por uma propriedade diferente. Os principais mapas de textura utilizados nesta técnica são: *Albedo* ou *diffuse*, *Roughness*, *Height*, *Normal*, *Metallic* e *Ambient occlusion*(AO). A Figura 5 contém exemplos iterativos da aplicação destas texturas num processo que vai de “a” a “f” para a criação de um objeto 3D que representa uma placa com o logo da Unifei fixada por algum tipo de rebite. Em cada imagem presente nesta figura, a esquerda encontra-se um plano com a aplicação das texturas e a direita está a textura do mapa do qual se pretende mostrar o seu efeito. A identificação e função de cada um destes mapas é apresentada posteriormente à Figura 5.

Em “a” está representado o mapa *albedo*. Esta textura é uma imagem que contém os três canais de cor RGB e é responsável por aplicar cores ao objeto.

Em “b”, além do mapa *albedo* está aplicado o *roughness* que é uma textura em escala de cinza responsável por definir o quão reflexiva é a superfície. Neste mapa o branco indica uma superfície sem reflexão e quanto mais próximo do preto mais reflexiva é a superfície. O valor máximo de reflexão é definido pelo artista no *software* de texturização. Neste contexto o *roughness* foi utilizado para deixar o logo e as letras da imagem mais reflexivas.

A aplicação do *Height map* pode ser visualizada na imagem “c” da Figura 5. Este mapa, também caracterizado por ser em escala de cinza, manipula os vetores normais de maneira a criar a ilusão de profundidade. Assim como no *roughness* há um parâmetro que indica a altura máxima, e quanto mais próximo do branco mais esta ilusão se aproxima da altura definida. Neste caso o *Height map* foi utilizado para simular uma superfície corrugada.

Figura 5 – Exemplos dos principais mapas de texturas para materiais PBR



Em “d” o *Normal map* foi utilizado para simular detalhes de deformação causados pelos rebites. Este mapa é uma textura que assim como *height map* é capaz de adicionar detalhes em superfícies, porém ele permite capturar mais detalhes e gerar ilusões mais precisas devido ao fato de ser um imagem RGB. E deste modo, Cada um dos canais de cor atua em uma das componentes dos vetores normais dos objetos. Já no *Height map* há apenas a informação de altura que pode criar ilusões em um único eixo.

Em “e” há a aplicação *Metallic map*. Este mapa é uma textura em escala de cinza que indica o quão metálica é a aparência de uma superfície, quanto mais próximo do branco mais metálica ela se torna. Neste caso este mapa foi utilizado para dar aparência metálica aos rebites.

Na imagem “f” da Figura 5 é mostrado o efeito do AO. Este mapa fornece um pouco mais de realismo ao Objeto 3D, pois ele simula a oclusão ou bloqueio

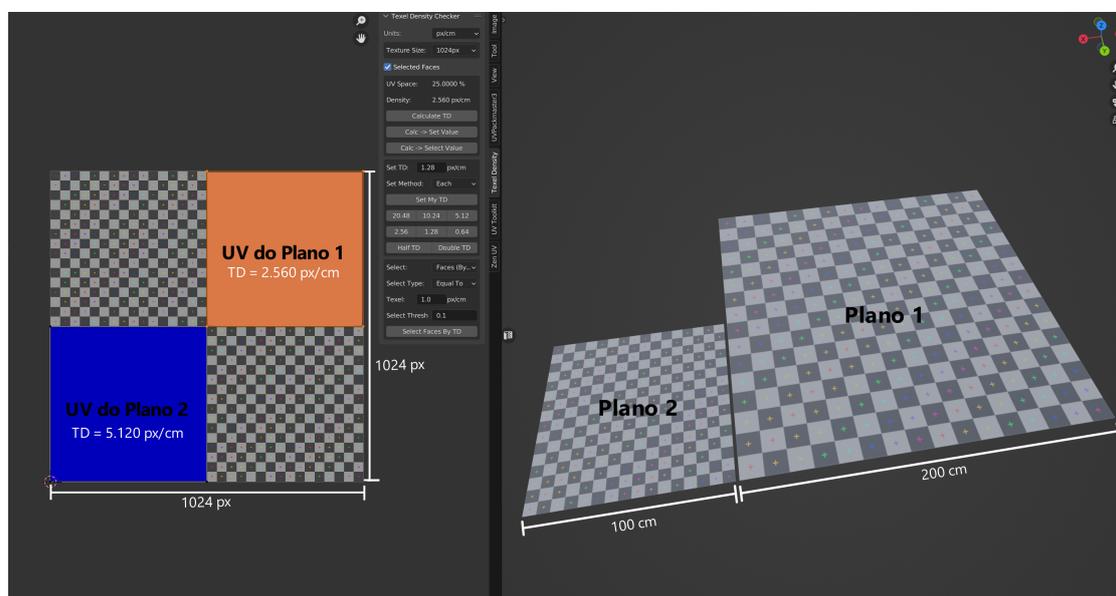
natural da luz ambiente. Neste exemplo, como a logo e as letras da Unifei estão em baixo relevo este mapa adiciona um sombreado nas bordas dessas regiões a fim de simular o que ocorreria com a luz nesta situação em um ambiente real.

### 3.4.0.2 *Texel density*

Um parâmetro essencial para a qualidade e consistência de detalhes de texturas entre diversos objetos é o *Texel density*(TD), Que é a relação entre a resolução das texturas e a área ocupada pelo mapa UV dos objetos na UDIM. Este parâmetro basicamente garante o nível de detalhe que pode ser aplicado ao objeto

Na Figura 6 por exemplo, ambos os planos utilizam a mesma textura, porém Eles apresentam um visual bem diferente um do outro em relação ao tamanho do padrão quadriculado presente na textura ao ser projetada sobre eles. Isso ocorre porque o plano 1, no ambiente tridimensional, um possui uma área de  $4m^2$  e foi representado na UDIM com o mesmo tamanho do plano 2, o qual possui apenas  $1m^2$  no ambiente 3D. Logo o valor de TD do plano 1 é metade do TD do plano 2.

Figura 6 – Representação das UDIMs no *UV editor* do Blender



O *TD* pode ser definido como um padrão de projeto e é importante que o modelador utilize as ferramentas do *software* de modelagem relacionadas ao *TD* para alocar e gerenciar o tamanho de todos os mapas UV dentro da *UDIM* de maneira que estes mapas estejam de acordo com o *TD* desejado.

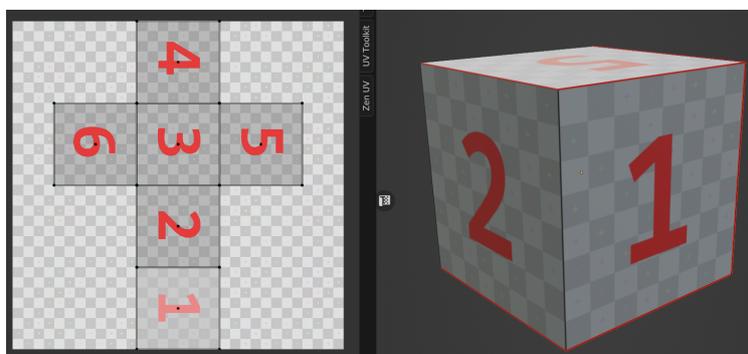
### 3.4.1 Técnicas de texturização

Dentre as principais técnicas de criação de texturas estão as texturas do tipo *unique* e as *trims* ou *trim sheets*. Até este momento, todos os exemplos de objetos texturizados mostrados se tratavam de texturas *unique*.

### 3.4.2 Texturas *unique*

Este tipo de textura se caracteriza por conter todas as faces do objeto dentro da *UDIM* e o seu nome advém da vantagem proporcionada por este tipo de textura. Como todas as faces são representadas em um local específico da *UDIM* é possível pintar detalhes únicos para cada face presente no mapa UV. A Figura 7 contém o exemplo de um cubo texturizado como *unique*, na qual é possível observar o seu mapa UV e como cada face possui detalhes únicos.

Figura 7 – Exemplo de Mapa UV de um cubo e projeção de sua textura



A grande desvantagem deste tipo de textura é a limitação em relação a quantidade e/ou tamanho dos objetos representados no mapa UV, pois com o aumento destas características, a tendência é o *TD* de cada face diminuir. Uma solução para contornar este problema seria dividir os objetos em diferentes materiais que por consequência aumenta a quantidade de arquivos de texturas. Porém,

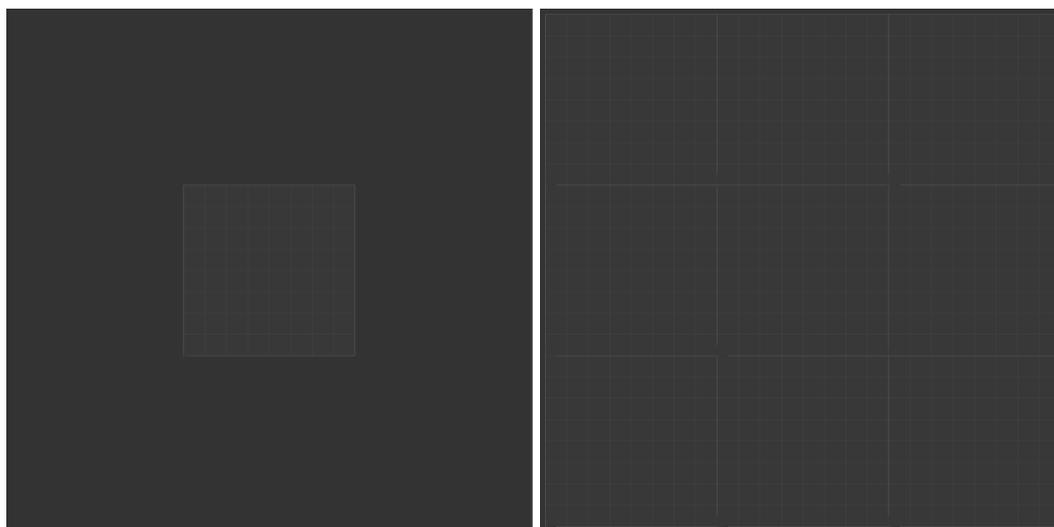
dependendo da quantidade requerida, o aumento da quantidade de texturas pode ter um impacto significativo nos recursos computacionais necessários para renderizar estes objetos. A exemplo de jogos e aplicações 3D em tempo real, um alto número de texturas carregadas simultaneamente pode causar gargalos e reduzir drasticamente o *frame rate* da aplicação.

### 3.4.3 Texturas *trim*

As texturas do tipo *trim* se aproveitam da maneira como as UDIMs funcionam para texturizar objetos grandes mantendo um valor satisfatório de TD. Com esta técnica de texturização é possível obter um número reduzido de texturas ao final de um projeto. Porém, a sua desvantagem é não poder dar detalhes únicos para cada face do objeto texturizado.

O Blender por padrão mostra apenas a UDIM principal. Entretanto existem diversas UDIMs que se estendem em ambos eixos u e v. Por padrão as outras UDIMs são configuradas em modo *tile* ou seja, elas replicam os dados da UDIM principal. Na Figura 8 à esquerda está a representação da UDIM principal, enquanto à direita está a representação de várias UDIMs em torno da principal, com a UDIM principal localizada no centro da imagem.

Figura 8 – Representação das UDIMs no *UV editor* do Blender.



A Figura 9 contém o exemplo de uma textura *trim* utilizada no desenvolvimento do DT deste trabalho. Neste caso a técnica foi necessária porque os prédios possuem paredes muito altas e o comprimento das mesmas é na casa das dezenas de metros. Pode-se observar que os polígonos das paredes extrapolam a UDIM principal na direção horizontal e sobre eles são projetadas as partes da textura da UDIM principal.

Figura 9 – Exemplo de texturas *trim sheet*



## 4 Materiais e métodos

Nesta seção, será descrito o processo de modelagem e composição do ambiente virtual para a realização do trabalho. Além disso também são descritos os métodos de aquisição das imagens reais e virtuais e sua aplicação na rede Cyclegan para transferir o estilo das imagens virtuais para o estilo das imagens reais.

### 4.1 Modelagem do ambiente virtual

Foi definida uma região de interesse (RI) do Campus Itabira para ser replicada em ambiente virtual. Contidos nesta região de interesse estão os Prédios I e II, Anexos I e II, o estacionamento e a via de acesso principal do Campus. A Figura 10 é uma imagem de satélite que contém parte do Campus Itabira. Nesta Figura a área contornada representa a região de interesse definida.

Figura 10 – Imagem de satélite com RI do Campus Itabira destacada.



Com a região de interesse definida, iniciou-se o processo de criação do ambiente virtual replicando parte a Unifei - Campus Itabira. Na primeira etapa deste processo foram investigados os dados disponíveis no software *Google Earth Pro* na latitude e longitude da Unifei. O *Google Earth* é um software que possui diversos tipos de dados do planeta terra, como por exemplo: dados topográficos, representações tridimensionais da superfície do planeta Terra e um histórico de imagens de satélite da superfície terrestre ao longo dos anos. Constatou-se que na localidade do campus não haviam dados topográficos precisos e nem representações 3D dos prédios para serem extraídas e facilitar a construção do ambiente virtual. Como havia apenas o histórico das imagens de satélite da região foi necessário buscar outra alternativa.

#### 4.1.1 O *Software* Meshroom

A solução encontrada foi gerar as próprias imagens para serem utilizadas, permitindo gerar uma malha 3D do campus por fotogrametria utilizando o *software Meshroom*. Este *software* foi definido por já haver uma familiaridade do autor com o seu uso em projetos anteriores que geraram boas malhas 3D. Fato que atestou a viabilidade desta abordagem.

O *Meshroom* é um software que permite criar malhas tridimensionais fiéis ao conteúdo presente nas imagens fornecidas ao seu banco de dados. Este *software* basicamente utiliza fotogrametria para encontrar diversos pontos em comum entre as imagens para assim calcular as distâncias e criar uma malha 3D texturizada. Porém, a qualidade da malha obtida depende diretamente da qualidade das imagens fornecidas. Assim sendo, o *Meshroom* requer alguns cuidados nos processos de captura e seleção de imagens em relação à quantidade e qualidade. Cada uma dessas medidas será abordada na subseção referente ao processo em que foram executadas.

#### 4.1.2 Captura das imagens

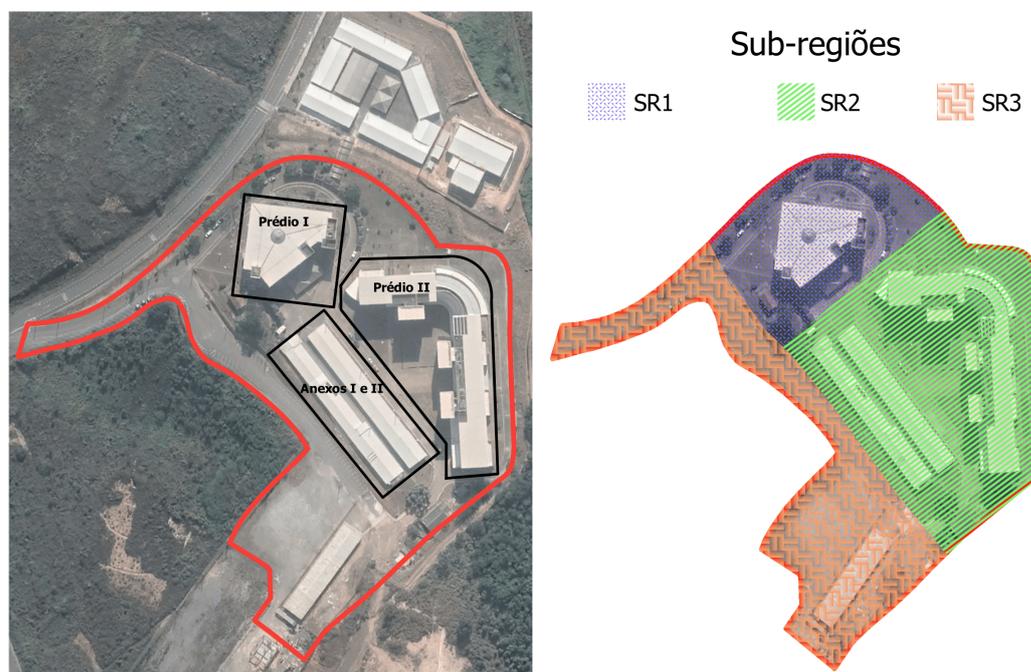
Como as imagens geradas seriam utilizadas como entrada no *Meshroom*, era necessário garantir que todas as imagens utilizadas estivessem sob a mesma

condição de iluminação, ou seja, deveriam ser geradas todas no mesmo horário do dia por se tratar da captura de um ambiente externo. Pois quando o *software* recebe a entrada de imagens do mesmo local com sombras posicionadas de maneira diferente, como por exemplo no caso de misturar imagens obtidas no período da manhã com imagens obtidas no final da tarde, ele pode descartar completamente algumas imagens ou ter dificuldade de encontrar pontos comuns e acabar gerando anomalias na malha de saída como buracos e estruturas imprecisas.

Outro ponto a ser levado em consideração é garantir que haja interseção entre o conteúdo das imagens fornecidas para que o *Meshroom* consiga identificar o caminho percorrido pela câmera, mas mesmo com esse cuidado ainda podem haver falhas no processamento das imagens que inviabilizem a definição caminho percorrido. Uma abordagem para evitar este tipo de problema é capturar fotos enquanto se orbita o objeto com a câmera em altitude superior, focando no objeto com inclinação de aproximadamente  $45^\circ$  em relação ao plano do mesmo, sempre mantendo-o por completo dentro do *frame* da câmera. Foi decidido utilizar essa abordagem, mas como a região de interesse é muito grande, contendo aproximadamente  $0,07Km^2$ , Orbitar todo este terreno capturando as imagens com um *drone* geraria uma malha pouco detalhada para representar as estruturas dos prédios presentes no Campus. Uma vez que devido a distância exigida para manter toda a RI em uma foto seria muito grande e isso acarretaria na perda de detalhe nas fotos.

Posto isso, optou-se por dividir a RI em sub-regiões(SRs), pois assim seria possível obter um conjunto de malhas menores, porém mais detalhadas que poderiam posteriormente ser posicionadas em um software de modelagem 3D de maneira a representar tridimensionalmente toda a RI. Essa representação poderia então ser utilizada como base para modelar manualmente o cenário 3D do Campus Itabira. Foram criadas três SRs. A sub-região 1(SR1) que contém o prédio I(P1) e seus arredores, e a sub-região 2(SR2) contendo o prédio II(P2), Anexos I e II e seus arredores; e a sub-região 3(SR3) abrangendo o estacionamento e a via de acesso ao Campus Itabira. Na porção esquerda da Figura 11 há uma representação da RI com a identificação de alguns prédios do campus. Na parte direita desta figura estão destacadas e identificadas as três sub-regiões definidas.

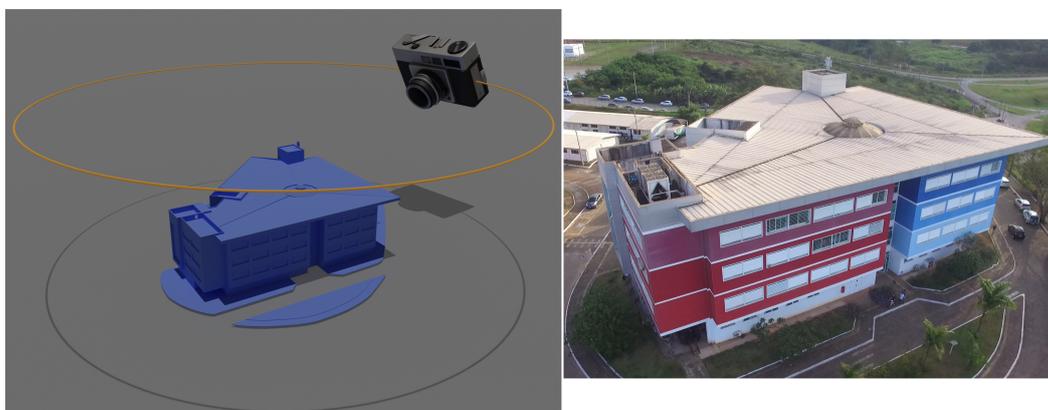
Figura 11 – Representação da RI e identificação das três sub-regiões definidas.



Como a SR1 e SR2 contêm os prédios que devem ser representados fielmente e a maioria das vias internas presentes na região de interesse. Optou-se por capturar as imagens utilizando um *drone*, DJI Phantom 3, operando em torno dessas SRs segundo a abordagem explicada previamente.

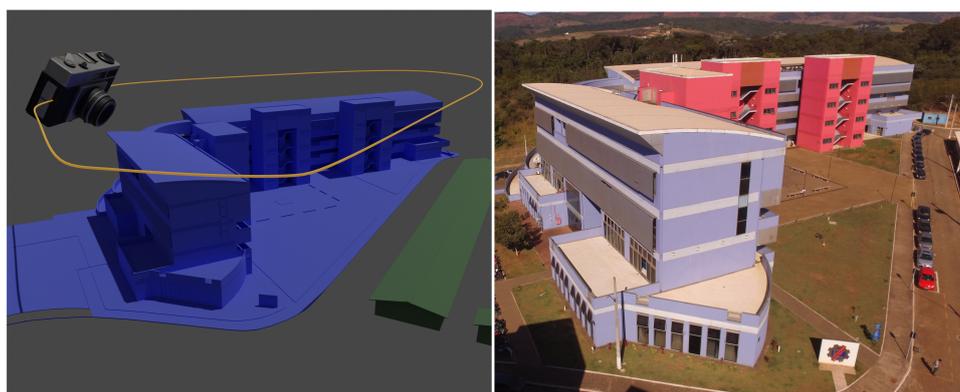
Na Figura 12 à esquerda encontra-se uma representação do esquema utilizado na SR1 para capturar as imagens do Prédio I. Em azul está representado o Prédio I. O toroide amarelo representa o caminho percorrido pelo *drone* em torno do P1, e a inclinação da câmera presente na imagem espelha a inclinação da câmera do *drone* em relação ao prédio I durante todo o processo. A parte direita da Figura 12 contém uma das fotos do Prédio I geradas pelo *drone*.

Figura 12 – Representação esquemática do processo de geração de imagens para o P1 e exemplo de imagem gerada.



Na tomada das imagens da SR2 apenas o prédio II foi orbitado, por ser a estrutura de maior interesse, e desta maneira os Anexos I e II também estariam presentes nas fotos de captura em diversos ângulos possibilitando ao *Meshroom* gerar malhas para essas estruturas. Na parte esquerda da Figura 13 está a representação esquemática do processo de captura de imagens da SR2. Em azul está a representação do P2, em verde está representada parte do Anexo II, Em amarelo está o caminho percorrido pelo *drone*. A câmera existente nesta imagem possui inclinação semelhante à utilizada pela câmera do *drone* no processo. À direita da Figura 13 é exibida uma das fotos do Predio II geradas pelo *drone*

Figura 13 – Representação esquemática do processo de geração de imagens para o P2 e exemplo de imagem gerada.



Já para sub-região 3 por ser muito extensa, conter apenas o estacionamento

e uma construção simples; não foi feita a captura de imagens utilizando o *drone*. Foi utilizada apenas uma foto de satélite do *Google Earth* como referência para o processo de modelagem. Como a imagem utilizada para modelar a sub-região 3 também seria utilizada futuramente para auxiliar no posicionamento das malhas obtidas para a SR1 e SR2 foi necessário percorrer o histórico de imagens recentes do *Google Earth* e encontrar uma foto com a menor inclinação possível para evitar ao máximo erros de posicionamento. À esquerda da Figura 14 está o exemplo de uma imagem ruim, pois nela dos prédios estão muito inclinados e as sombras são muito escuras e impedem a visualização de detalhes. Já na parte direita da Figura 14 está a foto utilizada para a SR3. Ela possui sombras suaves e pouquíssima inclinação nos prédios.

Figura 14 – Imagens de satélite da RI.



### 4.1.3 Processamento das Imagens da SR1 e SR2

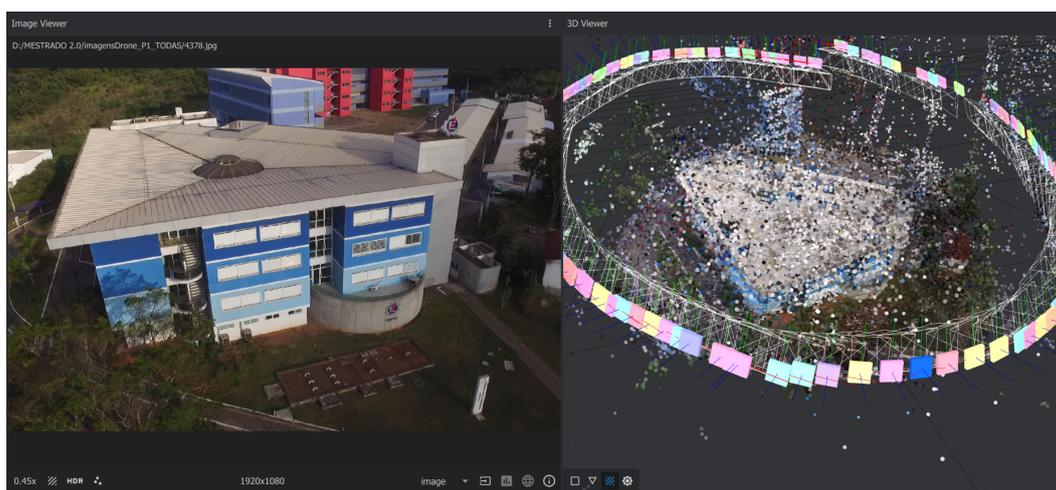
Na etapa de processamento das imagens foi criado um projeto no *Meshroom* para cada sub-região. Ao todo foram capturadas aproximadamente vinte mil imagens advindas dos *frames* da filmagem realizada pelo *drone*. Porém quanto mais imagens são inseridas como entrada maior se torna a complexidade da tarefa para o *Meshroom*, e conseqüentemente mais tempo demora para se obter uma saída. Foi definido que para cada projeto do *Meshroom* uma quantidade de imagens próximas a 100 já seria suficiente para se obter o nível de detalhe desejado com alguns poucos minutos de processamento. As imagens foram processadas em um compu-

tador utilizando o sistema operacional *Windows 11*, com 64Gb de memória RAM, processador Intel Core i7-11700 de 11<sup>a</sup> geração equipado com uma placa de vídeo NVIDIA RTX 3070 com 8GB de VRAM.

Em ambos os projetos tomou-se o cuidado de evitar escolher imagens com muitas superfícies reflexivas, pois reflexões geram imprecisões na malha de saída. Outro ponto levado em consideração na escolha das fotos foi selecionar imagens de diversos ângulos para o *Meshroom* poder criar uma malha com o máximo de detalhes possível. O tempo total de processamento em ambos os casos foi inferior a sete minutos.

Por meio da Figura 15 é possível observar o resultado parcial do processamento das imagens do Prédio I. Na janela *Image Viewer* (Esquerda) está uma das imagens capturadas que pode ser utilizada como referência para compreender a nuvem de pontos presente na janela *3D Viewer* à direita desta figura. Cada ponto presente na nuvem de pontos indica um local em que o *Meshroom* foi capaz de encontrar correspondência entre uma ou mais imagens. O fato de o P1 estar praticamente reconstruído apenas pela nuvem de pontos indica que será gerada uma excelente malha de saída. Cada volume retangular presente no *3D Viewer* indica a posição e orientação da câmera de cada imagem utilizada que foi reconhecida pelo *Software*.

Figura 15 – Processamento parcial do *Meshroom* para as imagens do P1.

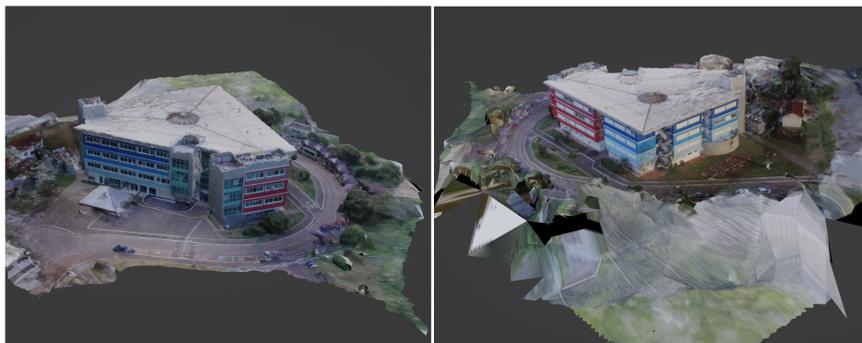


A criação da nuvem de pontos é a antepenúltima etapa de processamento

do *Meshroom*, a penúltima etapa é a geração da malha 3D a partir da nuvem de pontos e na última etapa é criada a textura da malha 3D a partir das fotos fornecidas.

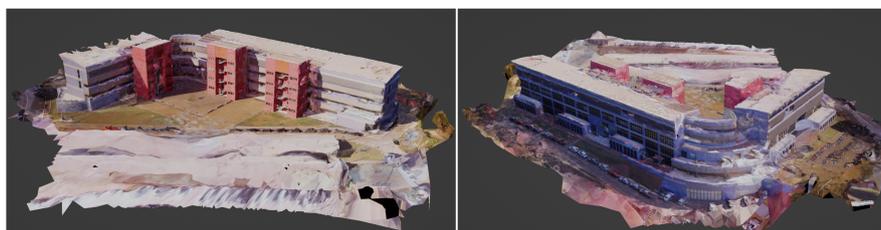
Ambas as malhas geradas para a SR1 e SR2 foram importadas para o Blender e foram tratadas antes de iniciar a etapa de modelagem. Este tratamento envolveu corrigir a orientação das malhas para alinhá-las em relação aos eixos x,y e z do Blender e também deletar as partes desnecessárias da malha. Deixando somente as estruturas de interesse. Na Figura 16 é possível observar a malha de saída obtida para a SR1 de dois pontos de vista diferentes após o tratamento. Por meio desta observação nota-se que a malha gerada representa com detalhes o Prédio I e seus arredores.

Figura 16 – Malha gerada para a SR1.



A Figura 17 contém a malha obtida para a SR2 após o tratamento. Apesar dos Anexos I e II não estarem perfeitamente representados nesta malha, isto foi o suficiente para modelá-los e posicioná-los corretamente devido a simplicidade das construções.

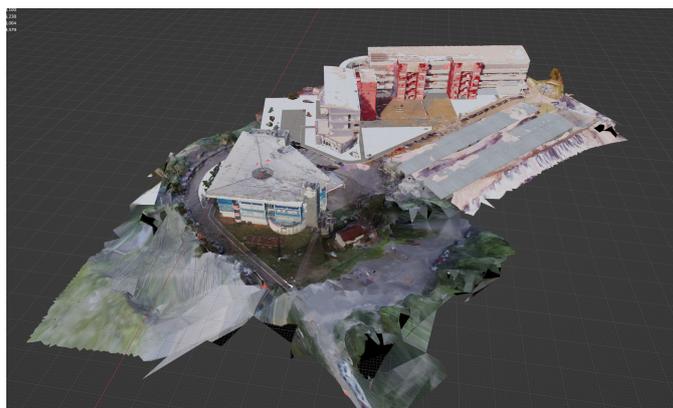
Figura 17 – Malha gerada para a SR2.



#### 4.1.4 Modelagem 3D das sub-regiões

As sub-regiões 1 e 2 foram modeladas no Blender utilizando as malhas geradas pelo *Meshroom* como referência. Estas malhas foram importadas para o Blender e tiveram suas escalas ajustadas de acordo com as medidas reais dos prédios. Depois cada malha foi posicionada de maneira a representar o Campus. A Figura 18 contém todas as malhas posicionadas no Blender.

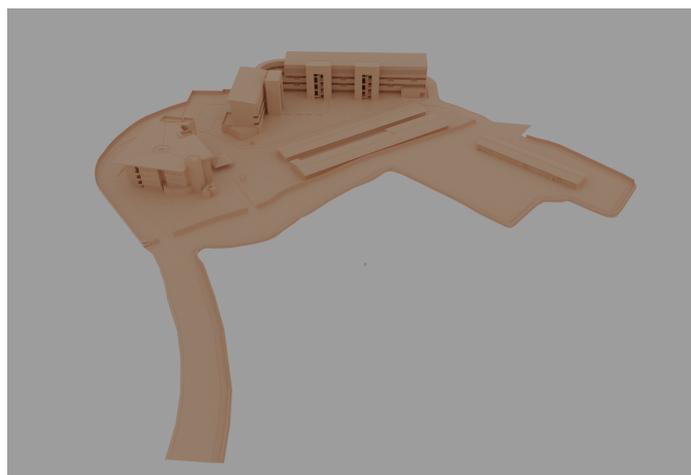
Figura 18 – Malhas das sub-regiões 1 e 2 posicionadas no Blender.



Já a sub-região 3 foi modelada utilizando a imagem de satélite e os modelos 3D da SR1 E SR2 como referência.

Na Figura 19 Estão as malhas 3D modeladas para todas as sub-regiões.

Figura 19 – Malhas 3D das sub-regiões modeladas e posicionadas no Blender.



### 4.1.5 Texturização dos modelos 3D

Os objetos 3D das SR1 e SR2 foram importados para o *software Substance Painter* no qual foram criadas as texturas para estes objetos. Na Figura 20 estão os objetos texturizados.

Figura 20 – Malhas 3D das construções da SR1 e SR2 texturizadas.



A SR3 e as ruas internas do campus foram texturizadas dentro da própria *Unreal Engine 5* após serem importadas.

### 4.1.6 Montagem do cenário virtual na *Unreal engine 5*

Todos os modelos foram então importados para a UE5, utilizando a versão 5.3.2, onde foi montado o cenário definitivo do campus a ser utilizado no processo de captura das imagens virtuais. Para popular o cenário com mais detalhes foram modelados alguns objetos como: cones, placas de trânsito, bancos, etc. Além disso também foram utilizados alguns pacotes de *assets* 3D de árvores e carros. A Figura 21 contém o cenário virtual montado na UE5.

Figura 21 – Cenário virtual do campus Itabira na UE5



## 4.2 Treinamento da transferência de estilo

Após a criação do modelo virtual de parte do campus Itabira, foi iniciado um processo para avaliar a viabilidade de utilizar o DT para melhorar a qualidade das imagens capturadas no ambiente virtual. A proposta é treinar a rede CycleGAN com o objetivo de transformar as imagens do ambiente virtual em imagens que se assemelham ao máximo às imagens reais do campus. Este processo caracteriza-se como primeira etapa de um projeto futuro que utilizará o DT do campus como um ambiente de treinamento para os algoritmos de visão computacional de um veículo autônomo. A CycleGAN será fundamental para diminuir a discrepância entre as imagens virtuais e reais, o que permitirá um treinamento mais preciso da percepção do veículo autônomo, seja em ambientes virtuais, reais ou em uma combinação de ambos.

### 4.2.1 Captura e tratamento das imagens de treinamento

Para a captura das imagens de treinamento foram capturadas imagens tanto no ambiente real quanto no virtual.

A captura das imagens reais foi realizada de forma a percorrer as vias do campus, que estão presentes no DT, realizando filmagens com uma câmera fixada em uma certa altura de maneira a simular a visão do carro autônomo.

Já para a captura das imagens virtuais, as filmagens foram realizadas pela captura de tela do *software OBS*, enquanto o mesmo caminho foi percorrido no DT de maneira a imitar todas as movimentações e ângulos realizados pela câmera real.

A captura de vídeo resultou em um grande conjunto de dados, com dezenas de milhares de imagens. Devido à alta taxa de quadros e ao tempo de gravação, muitas imagens eram praticamente idênticas. Para evitar um treinamento desnecessariamente longo e melhorar a eficiência do processo, foi desenvolvido um código para selecionar as imagens mais informativas. Esse código salta um número específico de quadros, garantindo que apenas as imagens mais distintas sejam utilizadas. Após a seleção, foram obtidas 2055 imagens do ambiente real e 684 do ambiente virtual. Para avaliar o desempenho do modelo as imagens foram divididas em conjuntos de treinamento e teste em seu respectivo tipo de imagem(real ou virtua)l. O conjunto de treinamento contendo 80% das imagens do tipo e o conjunto de teste 20%.

## 4.2.2 Treinamento Cyclegan

O código da rede CycleGAN utilizado é o que está disponível publicamente <sup>1</sup>. Este código foi executado com um parâmetro de duzentas gerações de treinamento, no ambiente virtual *Google Colab* utilizando as opções de configuração para RAM alta e uma GPU NVIDIA T4.

O tempo de treinamento destas 200 gerações foi de aproximadamente 22 horas e o tempo máximo permitido para a sessão do *Colab* era de apenas 20 horas. Assim sendo, foi necessário efetuar uma pequena modificação neste código para que o arquivo de *checkpoint* fosse salvo em um *google drive*, pois originalmente ele é salvo apenas no ambiente da sessão do *Colab* e assim que a sessão é encerrada o arquivo é deletado.

Os resultado e discussões acerca deste treinamento são apresentados na próxima seção.

---

<sup>1</sup> Link do código da cyclegan: <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>.

## 5 Resultados e Discussão

Neste capítulo são expostos os resultados do DT desenvolvido e também os resultados do treinamento da transferência de estilo da *cyclegan*.

### 5.1 Resultados da criação do DT

A região de interesse do campus teve as suas estruturas de interesse representadas fielmente pelo DT dentro da UE5. o Apêndice A contém algumas imagens do DT gerado. A visualização da região de interesse do campus virtual ser realizada acessando o link <sup>1</sup>, o qual contém um vídeo mostrando o DT.

Caso o leitor deseje também é possível abrir o projeto da *Unreal* ou rodar o executável do aplicativo desenvolvido e explorar o DT da maneira que desejar. Estes arquivos estão disponíveis no link <sup>2</sup>. Nesta segunda opção o visual do DT está um pouco melhor em relação ao vídeo, pois até o momento da gravação do mesmo não haviam sido implementados efeitos de pós-processamento.

---

<sup>1</sup> vídeo: <https://youtu.be/e1F1nnGM-bE?si=4X0SM3arV-xd2KdI>.

<sup>2</sup> Arquivos: Link de acesso aos projetos desenvolvidos.

## 5.2 Resultados do treinamento da transferência de estilo

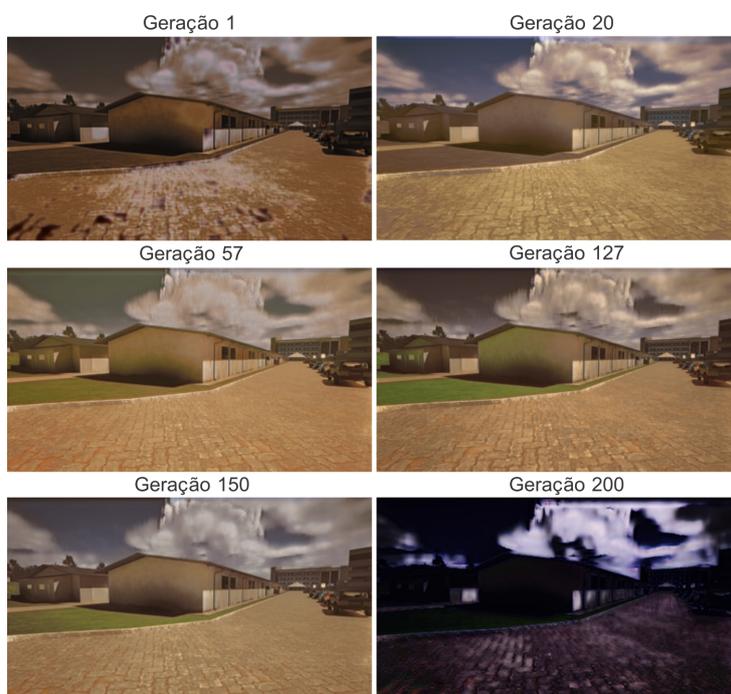
A seguir são mostrados os resultados do treinamento. Na Figura 22 à esquerda está o exemplo de uma imagem de entrada do campus real, a qual contém o estilo desejado que se deseja atingir com este treinamento. Na direita Está uma imagem de entrada do campus virtual que deve ser processada pela rede CycleGAN para transferir seu estilo.

Figura 22 – Exemplo de imagens de entrada fornecidas para a rede.



Já a Figura 23, contém a imagem do ambiente virtual após o processamento da transferência de estilo ao longo das gerações de treinamento.

Figura 23 – Imagens previstas ao longo das gerações



Comparando as imagens a Figura 23 com a imagem de entrada real da Figura 22 nota-se que a rede progrediu em relação transferência de estilo até a geração 150 e aproximou-se bastante do estilo da imagem de entrada. Já na geração 200 há uma queda brusca na qualidade da transferência de estilo o que indica a ocorrência de *overfitting* no processo de treinamento. Sendo assim, o *checkpoint* da rede foi restaurado para a geração 150 e espera-se dar continuidade a aplicação desta em pesquisas futuras.

## 6 Conclusão

Esta dissertação forneceu um panorama geral em relação a técnicas, conceitos e fluxo de trabalho necessário para a criação de modelos 3D, que associados a todos os projetos disponibilizados permite aos leitores deste trabalho, interessados em ingressar nesta área, ter uma boa noção para aumentar o seu conhecimento de maneira mais direta e menos confusa. Buscando assim, contribuir para o crescimento desta área no Brasil, pois os conhecimentos da engenharia associados às capacidades de simulação fornecidas pelo uso de motores gráficos em conjunto com a modelagem 3D abre caminho para pesquisas extremamente interessantes.

O DT desenvolvido por meio da metodologia apresentada foi capaz de cumprir o seu papel em representar a RI do campus. Uma vez que ele está disponibilizado publicamente e a Unifei campus Itabira se encontra em expansão, espera-se que ele continue a ser desenvolvido colaborativamente conforme o campus se desenvolva e sirva como ferramenta de pesquisa para qualquer pessoa que queira utilizá-lo.

Além dos fatos citados, o treinamento realizado para a transferência de estilo apresentou resultados promissores para a realização de uma pesquisa futura relacionada ao treinamento de veículos autônomos em ambiente virtual.

# Referências

- 1 PARK, T.; EFROS, A. A.; ZHANG, R.; ZHU, J.-Y. Contrastive Learning for Unpaired Image-to-Image Translation. 2020. Disponível em: <<https://arxiv.org/abs/2007.15651>>.
- 2 SUTHERLAND, I. E. Sketchpad: a man-machine graphical communication system. In: Proceedings of the May 21-23, 1963, Spring Joint Computer Conference. New York, NY, USA: Association for Computing Machinery, 1963. (AFIPS '63 (Spring)), p. 329–346. ISBN 9781450378802. Disponível em: <<https://doi.org/10.1145/1461551.1461591>>.
- 3 BÉZIER, P. Essai de définition numérique des courbes et des surfaces expérimentales: contribution à l'étude des propriétés des courbes et des surfaces paramétriques polynomiales à coefficients vectoriels. Université Pierre et Marie Curie (Paris VI), 1977. Disponível em: <<https://books.google.com.br/books?id=vK4POAAACAAJ>>.
- 4 CATMULL, E. E. A subdivision algorithm for computer display of curved surfaces. Tese (Doutorado), 1974. AAI7504786.
- 5 BLINN, J. F. Models of light reflection for computer synthesized pictures. SIGGRAPH Comput. Graph., Association for Computing Machinery, New York, NY, USA, v. 11, n. 2, p. 192–198, jul 1977. ISSN 0097-8930. Disponível em: <<https://doi.org/10.1145/965141.563893>>.
- 6 SOMMER, M.; STJEPANDIĆ, J.; STOBRAWA, S.; SODEN, M. von. Automated generation of digital twin for a built environment using scan and object detection as input for production planning. Journal of Industrial Information Integration, v. 33, p. 100462, 2023. ISSN 2452-414X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2452414X23000353>>.
- 7 ALIAGA, D.; NIYOGI, D. Digitizing cities for urban weather: representing realistic cities for weather and climate simulations using computer graphics and artificial intelligence. Computational Urban Science, v. 4, 03 2024.
- 8 HELLMAN, T.; LAHTI, M. PHOTOGRAMMETRIC 3D MODELING FOR VIRTUAL REALITY. 2018.
- 9 ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-Image Translation with Conditional Adversarial Networks. 2018. Disponível em: <<https://arxiv.org/abs/1611.07004>>.

- 10 PIZZATI, F.; CERRI, P.; CHARETTE, R. de. CoMoGAN: continuous model-guided image-to-image translation. 2022. Disponível em: <<https://arxiv.org/abs/2103.06879>>.
- 11 XIA, W.; YANG, Y.; XUE, J.-H. Unsupervised Multi-Domain Multimodal Image-to-Image Translation with Explicit Domain-Constrained Disentanglement. 2019. Disponível em: <<https://arxiv.org/abs/1911.00622>>.
- 12 HUANG, X.; LIU, M.-Y.; BELONGIE, S.; KAUTZ, J. Multimodal Unsupervised Image-to-Image Translation. 2018. Disponível em: <<https://arxiv.org/abs/1804.04732>>.
- 13 WANG, J.; SHEN, T.; TIAN, Y.; YUTONG, W.; GOU, C.; WANG, X.; YAO, F.; SUN, C. A parallel teacher for synthetic-to-real domain adaptation of traffic object detection. IEEE Transactions on Intelligent Vehicles, v. 7, p. 1–15, 09 2022.
- 14 PARK, J.; CHO, K. Neural rendering-based 3d scene style transfer method via semantic understanding using a single style image. Mathematics, v. 11, n. 14, 2023. ISSN 2227-7390. Disponível em: <<https://www.mdpi.com/2227-7390/11/14/3243>>.
- 15 LIU, Y.; JIANG, A.; PAN, J.; LIU, J.; YE, J. Deliberation on object-aware video style transfer network with long–short temporal and depth-consistent constraints. Neural Computing and Applications, v. 33, 07 2021.

# Apêndices

# APÊNDICE A – Imagens do *Digital twin* na Unreal Engine 5

Figura 24 – Região do P1 no projeto final da UE5



Figura 25 – Imagens capturadas em diferentes regiões do DT na UE5

