



**UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE  
CAMPUS AVANÇADO DE NATAL  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**MANOEL FELIPE ALVES DE SOUSA**

**ANÁLISE COMPARATIVA ENTRE OPENMVG E INSTANTNGP PARA GERAÇÃO  
DE MODELOS 3D A PARTIR DE IMAGENS 2D**

**NATAL / RN**

**2024**

**MANOEL FELIPE ALVES DE SOUSA**

**ANÁLISE COMPARATIVA ENTRE OPENMVG E INSTANTNGP PARA GERAÇÃO  
DE MODELOS 3D A PARTIR DE IMAGENS 2D**

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação da Universidade do Estado do Rio Grande do Norte – UERN, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

**Orientador (a): Prof.<sup>a</sup> Dr.<sup>a</sup> Adriana  
Takahashi**

NATAL / RN

2024.

© Todos os direitos estão reservados a Universidade do Estado do Rio Grande do Norte. O conteúdo desta obra é de inteira responsabilidade do(a) autor(a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu(a) respectivo(a) autor(a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

**Catálogo da Publicação na Fonte.**  
**Universidade do Estado do Rio Grande do Norte.**

A474a Alves de Sousa, Manoel Felipe  
Análise comparativa entre OpenMVG e InstantNGP para geração de modelos 3D a partir de imagens 2D. / Manoel Felipe Alves de Sousa. - Natal/RN, 2024.  
73p.

Orientador(a): Profa. Dra. Adriana Takahashi.  
Monografia (Graduação em Ciência da Computação).  
Universidade do Estado do Rio Grande do Norte.

1. Modelagem Tridimensional. 2. Estudo de tecnologia.  
3. Visão Computacional. 4. Percepção Tridimensional. 5.  
análise de caso. I. Takahashi, Adriana. II. Universidade do  
Estado do Rio Grande do Norte. III. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pela Diretoria de Informatização (DINF), sob orientação dos bibliotecários do SIB-UERN, para ser adaptado às necessidades da comunidade acadêmica UERN.

**MANOEL FELIPE ALVES DE SOUSA**

**ANÁLISE COMPARATIVA ENTRE OPENMVG E INSTANTNGP PARA GERAÇÃO  
DE MODELOS 3D A PARTIR DE IMAGENS 2D**

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação da Universidade do Estado do Rio Grande do Norte – UERN, como requisito parcial para obtenção do título de bacharel em Ciência da Computação.

Aprovada em: 04/03/2024

**BANCA EXAMINADORA**

*Adriana Takahashi*

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Adriana Takahashi (Orientadora)  
Universidade do Estado do Rio Grande do Norte - UERN

*André Gustavo P. da Silva*

---

Prof. Me. André Gustavo Pereira da Silva  
Universidade do Estado do Rio Grande do Norte - UERN

*Bruno Cruz de Oliveira*

---

Prof. Me. Bruno Cruz de Oliveira  
Universidade do Estado do Rio Grande do Norte - UERN

## **AGRADECIMENTOS**

Agradecimento aos meus familiares, sobretudo àqueles próximos a mim, pela paciência.

Aos meus amigos e conhecidos pelo incentivo a continuar.

Aos meus professores e colegas, ainda mais à minha orientadora, professora Adriana Takahashi, pelo empurrão nas costas para conseguir finalizar o trabalho.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo auxílio financeiro no formato de bolsa de Apoio para Estudante Elaborando TCC em IA, a oportunidade e consideração.

## RESUMO

As tecnologias na área de geração e reconstrução de ambientes virtuais vêm crescendo e ganhando espaço no mercado, como programas para criação de planta baixa e visualização de ambientes virtuais, que auxiliam o usuário na avaliação do seu ambiente. Porém, essas mesmas tecnologias se encontram muitas vezes restritas a quem tem o poder computacional de utilizar as mesmas, sendo desenvolvidas somente por profissionais da área ou oriundos da área de computação gráfica, o que por si só limita a entrada de novos desenvolvedores a essa crescente área de programação. Assim sendo, este trabalho vem com a premissa de fazer uma análise comparativa de duas tecnologias presentes no meio científico de computação. As tecnologias, InstantNGP, que usa como bases os conceitos da tecnologia NeRF, e o OpenMVG são tecnologias que apresentam contínua atualização até a data desse trabalho. Tendo a tarefa de construir ambientes virtuais a partir de fotos, a tecnologia InstantNGP, assim como o NeRF, utiliza redes neurais para reconstrução de modelos 3D em tempo real, já o OpenMVG analisa pontos de correspondência em várias imagens relacionadas, estima a geometria da cena e reconstrói a estrutura 3D do cenário das imagens. Sendo assim, o trabalho tem como inspiração principal comparar os resultados das duas tecnologias ao miniaturizar um local desejado pelo usuário em um ambiente virtual, possibilitando a visualização. Por fim, espera-se que este trabalho sirva de incentivo de referência para outros interessados a desenvolverem trabalhos na área no campo tridimensional por meio da descrição de áreas de pesquisas, assim como análise bibliográfica de autores da área de reconstrução tridimensional e visão computacional.

**Palavras-chave:** Modelagem Tridimensional, estudo de tecnologias, análise de caso, Visão Computacional, Percepção Tridimensional, miniaturização de ambientes.

## ABSTRACT

Technologies in the area of generating and transforming virtual environments have been growing and gaining ground in the market, such as programs for creating floor plans and visualizing virtual environments, which help the user in evaluating their environment. However, these same technologies are often restricted to those who have the computing power to use them, and are only intended for professionals in the field or those from the computer graphics area, which in itself limits the entry of new developers into this growing programming area. Therefore, this work comes with the premise of making a comparative analysis of two technologies present in the scientific computing environment. The technologies, InstantNGP, which uses the concepts of NeRF technology as a basis, and OpenMVG are technologies that continually update the data in this work. Having the task of building virtual environments from photos, InstantNGP technology, like NeRF, uses neural networks to visualize 3D models in real time, while OpenMVG analyzes correspondence points in several related images, estimating the geometry of the scene and reconstructed the 3D structure of the image scenario. Therefore, the work's main inspiration is to compare the results of the two technologies by miniaturizing a location desired by the user in a virtual environment, enabling visualization. Finally, it is hoped that this work will serve as a reference incentive for other interested parties to develop work in the area in the three-dimensional field through the description of research areas, as well as bibliographic analysis of authors in the area of three-dimensional drilling and computer vision.

**Keywords:** Three-dimensional Modeling, technology study, case analysis, Computer Vision, Three-Dimensional Perception, miniaturization of environments.

## LISTA DE ILUSTRAÇÕES

Figura 01 – Representação de imagem digital.....	17
Figura 02 – Aplicação de sensores 3D de alta precisão em objeto históricos. ....	18
Figura 03 –Reconstrução 3D com ultrassom (a) e com ressonância magnética (b).....	20
Figura 04 – Demonstração de Visão Estéreo, e a detecção da disparidade em pixels. ....	21
Figura 05 – Ilustração da reconstrução virtual baseada em raios-X.....	22
Figura 06 – Demonstração de cada abordagem ao decorrer do tempo de processamento. ....	25
Figura 07 – Exemplo de reconstrução 3D a partir de imagens 2D utilizando NeRF. ....	26
Figura 08 – Demonstração de resultados para cada estratégia. ....	28
Figura 09 – Reconstruções feitas por não profissionais utilizando o OpenMVG. ....	30
Figura 10 – Pipeline simplificado do OpenMVG.....	31
Figura 11 – Demonstração de exemplo básico com 10 imagens utilizando o OpenMVG.....	32
Figura 12 – MagicPlan .....	33
Figura 13 – ARPlan 3D .....	34
Figura 14 – Primeiro ambiente de teste. ....	38
Figura 15 – Segundo ambiente de teste, iluminado natural e artificialmente pela câmera. ....	38
Figura 16 – Segundo ambiente de teste. ....	39
Figura 17 – Diversos ângulos da primeira maquete. ....	40
Figura 18 – Segunda maquete simplificada e sem textura, com um objeto laranja. ....	41
Figura 19 – Demonstração da aplicação e treinamento com o gráfico em estágio já estável. .	51
Figura 20 – Exemplo de ponto focal em uma câmera. ....	53
Figura 21 – Resultados de reconstrução da primeira maquete utilizando o InstantNGP. ....	56
Figura 22 – Resultados de reconstrução da primeira maquete utilizando o InstantNGP. ....	57
Figura 23 – Resultados de reconstrução da segunda maquete utilizando o InstantNGP.....	58
Figura 24 – Resultado de reconstrução da primeira maquete com o OpenMVG.....	59
Figura 25 – Resultado de reconstrução da primeira maquete com o OpenMVG.....	60
Figura 26 – Resultado de reconstrução da primeira maquete com o OpenMVG.....	60
Figura 27 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01. ....	61
Figura 28 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01. ....	61
Figura 29 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01. ....	62
Figura 30 – Resultados de reconstrução da segunda maquete.....	62
Figura 31 – Resultado de reconstrução da segunda maquete. ....	63



## LISTA DE TABELAS

Tabela 01 – Informações de outras tecnologias em 2017 em comparação ao OpenMVG.....	33
Tabela 02 – Informações e especificações da primeira máquina utilizada.....	42
Tabela 03 – Informações e especificações da segunda máquina utilizada. ....	43
Tabela 04 – Informações do primeiro dispositivo móvel utilizado e sua câmera embutida. ...	44
Tabela 05 – Informações do segundo dispositivo móvel utilizado e sua câmera embutida. ....	44
Tabela 06 – Tabela para comparação dos resultados lado a lado.....	64
Tabela 07 – Tabela comparativa de informações e resultados alcançados pelo InstantNGP e OpenMVG. ....	65

## SUMÁRIO

<b>1. Introdução .....</b>	<b>11</b>
1.1. Objetivos.....	13
1.1.1. Geral .....	13
1.1.2. Específicos.....	13
1.2. Metodologia.....	14
<b>2. Fundamentação Teórica.....</b>	<b>16</b>
2.1. Computação Gráfica .....	16
2.1.1. Captura de Imagens 3D .....	17
2.1.2. Reconstrução Tridimensional de Imagens.....	19
2.2. Redes Neurais .....	23
2.3. NeRF.....	25
2.4. InstantNGP .....	27
2.5. OpenMVG .....	29
2.6. Trabalhos relacionados .....	33
<b>3. Estudo de caso – Especificações de cenário, hardware e sistemas. ....</b>	<b>36</b>
3.1. Descrição de Cenário.....	36
3.2. Descrição de Hardware e Sistemas.....	42
3.2.1. Dos computadores .....	42
3.2.2. Das câmeras e dispositivos portáteis. ....	43
<b>4. Estudo de Caso – Reconstrução tridimensional de interiores com testes em maquete</b>	<b>45</b>
4.1. Instalação .....	45
4.1.1. Instalação: InstantNGP .....	45
4.1.2. Instalação: OpenMVG.....	47
4.2. Aplicação .....	48
4.2.1. InstantNGP .....	49
4.2.2. OpenMVG .....	51
4.3. Resultados.....	55
4.3.1. InstantNGP .....	56
4.3.2. OpenMVG .....	59
4.3.3. Tabelas Comparativas.....	63
<b>5. Considerações Finais .....</b>	<b>68</b>
5.1. Conclusões.....	68

5.2. Trabalhos Futuros .....	70
<b>Referências Bibliográficas .....</b>	<b>71</b>

## 1. Introdução

As empresas de tecnologia buscam cada vez mais a inserção de ambientes virtuais em nosso meio de vida. Prova disso são os grandes investimentos e produtos que estão surgindo diariamente nas áreas de realidade virtual e realidade aumentada. Não somente no mercado ou futuramente em nossas residências, existe a possibilidade de se ter, parcialmente ou totalmente, alguma forma de visualizar o nosso ambiente virtualmente, seja na área de design, construção ou na área médica. A visão computacional por meio da virtualidade só tende a aumentar. No entanto, ao mesmo tempo em que se encontra uma barreira de custos e pouco conhecimento, a tecnologia ainda é muito cara e complicada, resultando em uma parcela reduzida de mão de obra qualificada para desenvolvimento de tecnologias.

A percepção tridimensional virtual é uma área em constante desenvolvimento. O termo em si se refere à capacidade que alguns animais, inclusive humanos, de criar a estereopsia (RIBAS, RIBAS, RODRIGUES, 2006), um fenômeno de percepção em profundidade tridimensional por meio de olhos virados para frente, cada um com seu ponto de vista próprio, em que a fusão das imagens resulta na forte percepção tridimensional ou relativa (GONZALEZ e WOODS, 2010). Na computação, a percepção tridimensional está presente nas áreas de Inteligência Artificial e Visão Computacional, que possibilita a máquina “enxergar” o meio em que se encontra e interagir com o mesmo, como exemplo, os carros autônomos (GUIZZO, 2011) ou software de reconhecimento facial (BIESSECK, ZACARKIM, 2021). Demonstrando assim, o potencial que a área possui, mesmo que ainda seja somente seu início.

Uma das dificuldades que os desenvolvedores encontram é demonstrar à máquina como captar e interpretar algum ambiente fechado ou aberto de forma automática. O motivo para essa dificuldade seria a barreira entre o bidimensional, o qual o computador consegue interpretar, e o tridimensional, o mais complicado, visto a necessidade de um número maior de variáveis para ser representado a profundidade do ambiente de estudo (CYGANNEK e SIEBERT, 2009).

Nesse contexto, esse trabalho visa analisar duas soluções para o problema de modelagem geométrica tridimensional. De forma que poderiam ser utilizados em qualquer ambiente fechado, levando em conta certo nível de luminosidade, e gerar um ambiente virtual de qualidade e fácil visualização em qualquer sistema, tendo as limitações do programa em mente. Importante ressaltar que as estratégias testadas são focadas em construção de objetos

sólidos, portanto, foram necessárias adaptações na fase de teste para que a visão fosse feita pelo interior desses sólidos.

A primeira analisada foi a implementação do Instant-NGP (*Instant Neural Graphics Primitives*) (MÜLLER et al., 2022), que se trata de uma série de diretivas baseadas na NeRF (*Neural Radiance Fields*) (MILDENHALL et al, 2020), ambas se utilizando das linguagens Cuda, Python e C++. O NeRF, no entanto, tem um algoritmo que necessita um maior processamento e maior necessidade de um hardware com especificações mais potente. As duas tecnologias, NeRF e InstantNGP, utilizam de imagens e volumes neurais, empregando aprendizado de máquina, a MLP (*Multilayer Perceptron*). O treinamento tem o propósito de fazer a rede executar a renderização de pontos em comum entre várias imagens bidimensionais distintas de um mesmo objeto/ambiente. Dessa forma, a rede neural ao mapear esses pontos, os armazena em uma matriz, resultando em uma malha virtual de pixels, para posterior modelagem do objeto/ambiente em uma cena virtual.

A segunda técnica utilizada foi o OpenMVG (*Open Multiple View Geometry*) (MOULON et al, 2016), a qual ao utilizar principalmente de linguagem C++, utiliza da mesma técnica de comparação e localização de pontos em comum em imagens, no entanto, ao se aproveitar de pontos de vista distintos, o OpenMVG se vale de técnicas de visão computacional para encontrar seu objetivo. A utilização do OpenMVG se mostra bem mais leve em relação ao processamento e exigência da máquina em relação ao InstantNGP, se tornando um método mais acessível de construção virtual, já que tem como objetivo a extensão do reconhecimento 3D ao somente utilizar imagens e fotos no seu desenvolvimento.

Na fase de testes, serão apresentados os resultados alcançados tanto em ambientes reais com diferentes graus de iluminação para verificação e estabelecimento das medidas do ambiente a serem utilizadas para miniaturização. O fator de miniaturizar diz respeito tanto ao aspecto do modelo virtual em menor escala, quanto para a necessidade que deve ser demonstrada de se utilizar ambientes construídos de forma artificial (maquetes) durante a fase de testes, conforme a necessidade.

Para demonstrar e comparar duas soluções para o problema de virtualização de interiores, este trabalho visa demonstrar etapas e métodos para implantação das mesmas, bem como descrevê-las e demonstrar seus pontos altos e baixos, e através das métricas o quão confiáveis são. Dessa forma, que este trabalho sirva de ponto de partida para novos desenvolvedores alcancem resultados melhores, seja em qualidade ou mesmo economia de tempo.

## 1.1. Objetivos

### 1.1.1. Geral

Construção de um modelo 3D utilizando as técnicas InstantNGP e OpenMVG de um ambiente controlado a partir de maquetes e análise comparativa entre os resultados e as duas técnicas.

### 1.1.2. Específicos

São objetivos específicos deste projeto de pesquisa:

- Identificar alguns dos principais métodos para geração de ambientes virtuais atuais.
- Demonstrar a instalação e dificuldades inerentes ao uso das técnicas escolhidas.
- Geração de modelo 3D com foco nas características de interior de ambientes.
- Fazer uma comparação concisa entre as técnicas apresentadas e seus pontos fortes e fracos para reconstrução virtual.
- Analisar o contexto atual de tecnologias e técnicas de gerenciamento nas áreas de visão computacional, assim como suas limitações e dificuldades em aplicar métodos de modelagem virtual ao converter ambientes reais para virtuais.

## 1.2. Metodologia

Uma revisão bibliográfica na área de visão computacional e computação gráfica foram necessárias com o intuito de servir de base para construção do modelo tridimensional. Como exemplo, vale mencionar o trabalho de JUNIOR (2013), com a atenção que o autor teve para reconstruir os ambientes que foram sendo adquiridos, frame a frame, demonstrando como um ambiente pode ser virtualizado de maneira mais detalhada. Posteriormente, foi feito um refinamento e um levantamento bibliográfico mais refinado sobre o tema do trabalho, a fim de servir como efeito comparativo com a solução a ser encontrada com este projeto. Assim sendo, se espera que a pesquisa bibliográfica seja retomada quando novas tecnologias forem desenvolvidas, visto que novos conhecimentos poderão vir a surgir, sendo necessária constante atualização a respeito do assunto, sendo que novos métodos para resolver o problema possam vir a surgir.

Dessa forma, o trabalho assume caráter de pesquisa exploratória tendo em vista a necessidade de se familiarizar com o objetivo da pesquisa, isto é, o ambiente para reconstrução virtual a partir de locais físicos reais, para então se iniciar o desenvolvimento do protótipo. Para se alcançar um resultado satisfatório, serão feitas comparações com os aplicativos já presentes no mercado, dessa forma cumprindo a proposta de ser feita uma comparação com técnicas já presente no mercado. Para efeito de comparação, os aplicativos mencionados anteriormente visam a obtenção das medidas do ambiente do usuário para uma construção bidimensional, que apesar de ser efetiva para uma parcela de usuários, deixa a desejar aos mesmos que gostariam de, por exemplo, um modelo mais visual para uma possível melhor manipulação.

No processo de desenvolvimento do trabalho será documentado o levantamento de requisitos de software e hardware feito, de modo que foi executado o processo de compreensão e identificação das necessidades que as técnicas têm, bem como, chegar a conclusão se de fato o problema do trabalho foi solucionado. No entanto, inicialmente, já se pode estabelecer que, como requisitos mínimos de hardware, o trabalho inicialmente que deveria compreender, somente um computador e um celular smartphone para aquisição de imagens e vídeos, foi usado um novo computador mais capaz e outro smartphone, totalizando assim duas máquinas de testes e duas câmeras.

O método de calibração da câmera deverá ser feito para que sejam obtidas as medidas aproximadas dos locais testados, ao menos para que sejam demonstrados confiáveis e possam

ser de fato condizentes com o ambiente. Essas medidas deverão ser armazenadas e usadas para construção do ambiente miniaturizado em proporção ao ambiente que representa, por isso, essas dimensões necessitam serem as mais fidedignas possíveis. Como adendo, visto a necessidade de melhorar a precisão da representação virtual, técnicas para medidas e escala serão utilizadas para melhor estimar o projeto.

Para se alcançar uma maior qualidade quanto à funcionalidade da máquina utilizada durante a execução das técnicas analisadas, foram feitas várias rodadas de teste com as tecnologias instaladas, de modo que o ambiente testado se encaixava nos padrões do problema. Com relação ao ambiente, levando em conta o uso de maquetes, a especificação dos materiais, medidas e escala serão bem definidas e explicadas, de modo a cobrir o quanto possível de variáveis ao exemplificar ambientes reais. Durante os testes, seguindo os objetivos do projeto e utilização de maquetes que, visto a disponibilidade, possam vir a serem testados, para demonstrar os diferentes tamanhos, formato dos ambientes, grau de iluminação, imperfeições com o máximo de precisão possível.

Com o objetivo de demonstração, o processo de instalação e suas dificuldades deverão ser apresentados, levando em conta, principalmente, os processos principais e suas dependências. Espera-se, dessa forma, que o processo de obtenção das tecnologias avaliadas seja facilitado a outras pessoas que tiverem contato com esse trabalho, possibilitando às mesmas a reprodução dos resultados, ou ao menos que sirva de caráter avaliativo.

Nos próximos capítulos é documentado as pesquisas executadas das áreas que serviram de base para avaliação das técnicas trabalhadas, bem como a demonstração de tecnologias já presentes no mercado.



## 2. Fundamentação Teórica

Este capítulo aborda conceitos fundamentais da área de computação gráfica, reconstrução tridimensional, bem como apresenta os resultados da pesquisa exploratória tanto das tecnologias escolhidas quanto de outras desenvolvidas dentro da mesma área.

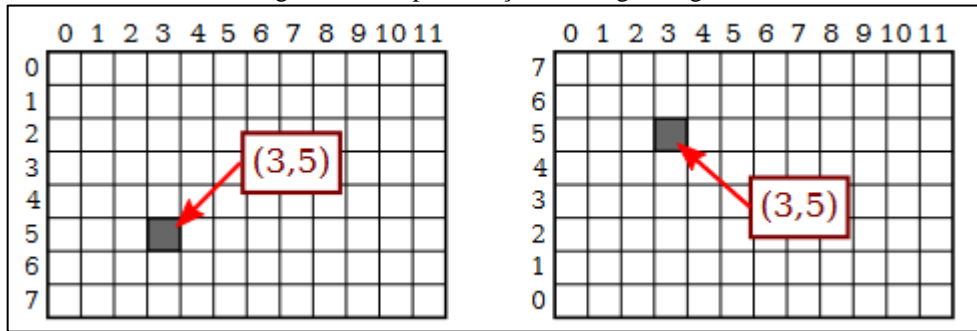
### 2.1. Computação Gráfica

Com a surpreendente velocidade que as máquinas vêm evoluindo nas últimas décadas, os computadores têm se tornado um rápido e eficiente modo de interpretar o mundo, com quase nenhuma área comercial ou residencial no qual uma tela digital não forneça algum benefício, seja para demonstração ou interpretação (HEARN, 1997). Embora os maiores avanços na área tendem a ser alcançados em computadores de última geração, os avanços em tecnologias computacionais possibilitaram que a maioria das pessoas conseguisse vivenciar esses avanços no conforto de suas casas ou qualquer lugar que quisessem.

A computação gráfica atualmente é indispensável nas mais diversas áreas, nas quais as mais notáveis são ciências, engenharia, medicina, indústria, entretenimento, educação, notícias, educação, treinamento, entre outras. O termo computação gráfica se refere a qualquer coisa que envolva a criação e manipulação de imagens em computador, sendo elas estáticas ou móveis (ECK, 2023). Um dos campos da computação gráfica é a reconstrução virtual, que através de métodos e técnicas computacionais buscam transformar o mundo físico real em virtual, de forma que o usuário possa manipulá-lo, adaptá-lo e modificá-lo para resolver quaisquer problemas.

Uma imagem, quando representada em uma tela digital, é feita a partir de pequenas partes de formato retangular e determinada cor num plano bidimensional (2D) chamada de pixels (Figura 01). Uma imagem digital é essencialmente uma matriz retangular composta por pixels organizados em linhas e colunas (ECK, 2023). As técnicas abordadas neste trabalho se baseiam em manipular os pixels em diferentes imagens sobrepostas, localizando, comparando e por fim encontrando os mesmos pontos em comum em outras imagens e reconstruindo em uma nova representação, desta vez tridimensional em ambiente virtual controlado.

Figura 01 – Representação de imagem digital.



Fonte: (ECK, 2023).

As duas tecnologias, InstantNGP e OpenMVG, detalhadas no capítulo adiante, utilizam desse mesmo princípio, diferenciando, no entanto, a forma que será trabalhada e a técnica empregada, bem como o quanto dependente do hardware cada uma requer, sendo, portanto, um limitador a possíveis usuários. O InstantNGP utiliza de redes neurais pré-treinadas que através de análise pesada por meio do aprendizado de máquina consegue estabelecer uma malha de pontos diretamente num plano tridimensional virtual (3D) (MÜLLER et al, 2022). O OpenMVG, por sua vez, se utiliza principalmente de técnicas de visão computacional (localização, triangulação, reconstrução), aproveitando significativamente melhor o tempo e reduzindo a necessidade de hardware mais robusto do usuário (MOULON et al, 2017).

### 2.1.1. Captura de Imagens 3D

O processo de captura de imagens do meio físico para o meio digital vem evoluindo continuamente. A captura de imagens tridimensionais (3D) representa uma área crucial no campo da computação gráfica e visão computacional, proporcionando a capacidade de adquirir informações espaciais detalhadas de objetos e ambientes. Suas aplicações variam de capturas geométricas de objetos em indústrias, até controle de qualidade e visualização de ambientes remotos para uso nas ciências. Por causa disso, a variedade de áreas que se utilizam das técnicas de captura de imagens ultrapassou até mesmo os setores de manufatura, áreas humanas, como medicina, análise criminal e de padrões, ou mesmo áreas de precisão, como análises de possibilidades e ocorrências (SANSONI, TREBESCHI, DOCCHIO, 2009).

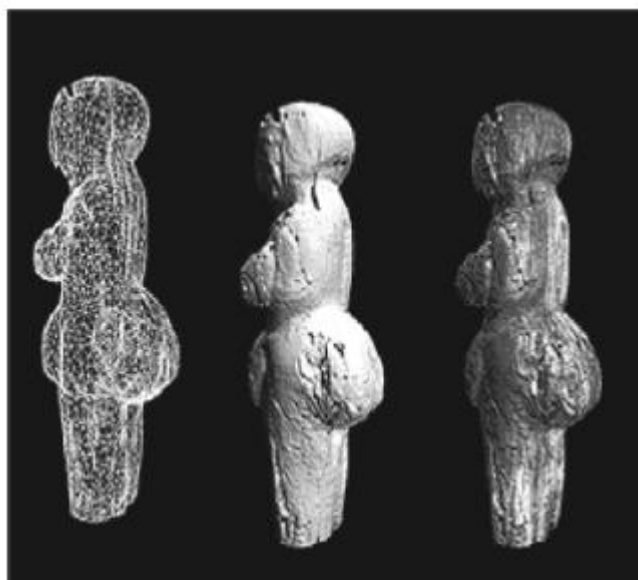
A flexibilidade na captura de superfícies em alta qualidade e precisão, que são empregadas em áreas como a médica, por exemplo, quando comparada às utilizadas para indústrias de produção, se demonstra mais evidente ao se colocar ao lado dos avanços em

captação de imagens. A demanda por tecnologias ópticas só tende a aumentar devido à sua capacidade de resolução e precisão cada vez mais confiáveis e flexíveis (KUS, 2009).

As imagens digitais, em sua definição tradicional (bidimensionais 2D), são imagens que são vistas em televisores, e diversas outras mídias, sendo representadas por um plano (largura e altura). Os sensores de capturas dessas imagens estão no dia a dia de todas as pessoas, em seus celulares (câmeras digitais). Já as imagens 3D visualizam não só a largura e altura de uma cena capturada, mas também a profundidade (Figura 02). Os sensores para captura de imagens 3D receberam avanços consideráveis, contudo, ainda necessitam de aparatos tecnológicos, como lasers, adaptações de câmeras ou o uso de mais uma câmera para serem capazes de recriar imagens 3D. Alguns exemplos de sensores para captura 3D são: câmera 3D, scanner 3D, *side-by-side*, *beam splitter*, entre outros.

As técnicas empregadas de captura de imagens tridimensionais geralmente operam projetando ou capturando alguma forma de energia a partir do objeto em análise, sendo então essa energia transmitida ou refletida (SANSONI, 2009). Um dos maiores exemplos disso seria, na área médica, com a tomografia computadorizada, a qual usa de medidas de radiação de raios-X que passam através dos objetos.

Figura 02 – Aplicação de sensores 3D de alta precisão em objeto históricos.



Fonte: (BLAIS, 2004)

As técnicas de captura tridimensional se baseiam em triangulação óptica ou fotogrametria (BERNARDINI, RUSHMEIER, 2002). O processo de triangulação utiliza o atraso de captura do sinal emitido e o uso de imagens de ponto focal único (monocular). Os métodos passivos se baseiam no reflexo do objeto e na iluminação do ambiente como

informação para determinar o formato da imagem. Os métodos ativos se utilizam de fontes de luz presentes no ambiente para medições diretas e indiretas, provendo informações essenciais, como orientação e formato, se baseando nas distâncias entre o sensor e o objeto em análise (SANSONI, 2009).

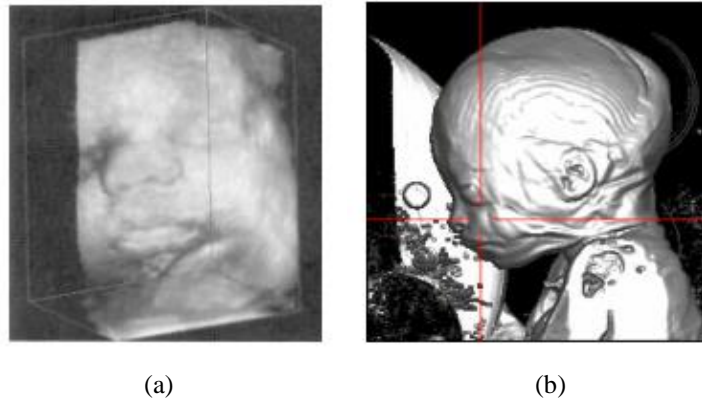
Os algoritmos em visão computacional, no entanto, se chocam com a dificuldade que as mesmas possuem em determinar pontos exatos, se restringindo a estimativas, o que em alguns casos acabam produzindo resultados indesejados, como a perda de qualidade das imagens, que por sua vez ocasiona uma análise deficiente (JANG et al, 2022). Os algoritmos de estimativa baseadas em aprendizagem de máquina vêm como alternativa para suplantam as lacunas perdidas em determinar, por exemplo, a profundidade precisa de cenas e assim providenciar performances de melhor qualidade quando se trabalhar com imagens em planos 2D para 3D. Por causa da necessidade de aproximação, ao se trabalhar com graus de profundidade diferentes, se torna fator crítico analisar o quão preciso a visão estéreo se torna ao utilizar múltiplos pontos de vista.

Entre os diferentes métodos que surgiram, o método que serviu de base para ambas as tecnologias analisadas e que será analisada a seguir é o que mais se assemelha a visão humana, a visão estéreo. Esse método segue a sequência padrão: aquisição de imagem, extração de informações, análise de correspondências e triangulação, servindo como critério comparativo entre diferentes pontos de vista do mesmo ambiente, utilizando no mínimo dois pontos de análise. Esse processo fornece à reconstrução tridimensional a profundidade tão desejada para montagem visual.

### 2.1.2. Reconstrução Tridimensional de Imagens

Tendo como principal objetivo a recuperação das informações resgatadas de ambientes reais, a reconstrução 3D de imagens exerce uma vital necessidade em áreas como medicina, robótica e indústria. Dados geométricos de como objetos ou ambientes são realmente, se corretamente transmitidos ao computador, se tornam um diferencial na hora de saber como se aplicar a modificação necessária, seja de que área for. Alguns exemplos típicos seriam de ultrassonografia 3D (Figura 03a) e ressonância magnética (Figura 03b), que possibilita a intervenção médica caso necessária (JOSÉ, 2008).

Figura 03 –Reconstrução 3D com ultrassom (a) e com ressonância magnética (b).



Fonte: (JOSÉ, 2008)

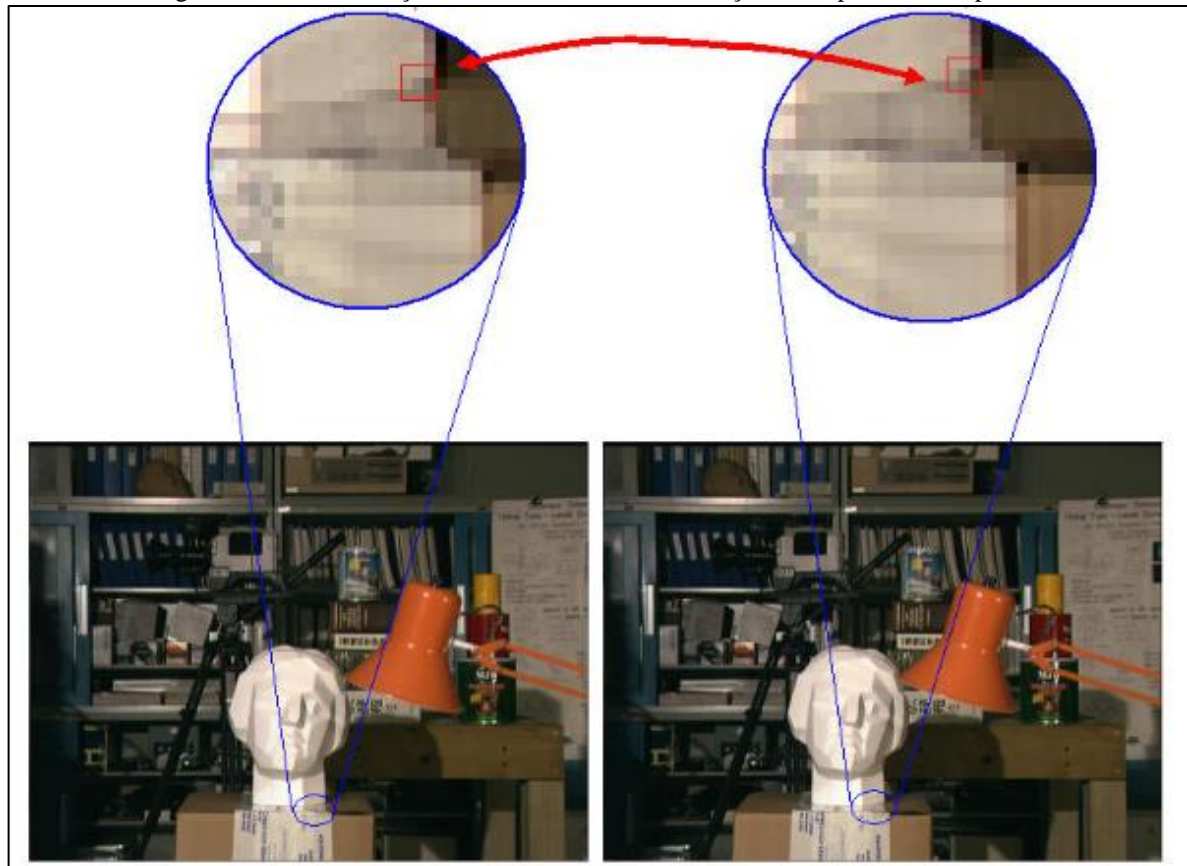
Dentre os sistemas que utilizam fotografias como base, o posicionamento dos equipamentos, exemplo câmeras digitais, devem ser planejadas, visto que a facilidade da identificação de pontos de convergência em diferentes figuras contribui na qualidade da reconstrução (JOSÉ, 2008), podendo até mesmo, ser necessária uma intervenção manual nessa identificação, o que não seria ideal.

As técnicas para visualização tridimensional precisam se preocupar com um eixo a mais: a profundidade. Para encontrar e adicionar essa dimensão adicional à modelagem, o dilema pode se resolver ao tentar imitar o funcionamento do sistema visual humano, que envolve utilizar duas imagens, ou pontos de vista distintos de um mesmo local, conhecida como visão binocular (estereoscópica) (JOSÉ, 2008). Esse efeito é alcançado ao encontrar a disparidade entre as duas imagens, conhecida como deslocamento lateral visual, que por sua vez possibilita a visão estéreo (SCHARSTEIN, 1999). O sistema visual humano evoluiu a tal ponto que executa todo esse processo sem esforço algum, ao extrair os dados 3D fornecidos pela diferença na captura visual de cada olho (Figura 04).

Um dos algoritmos mais populares para simular a visão humana é: o *Multi-view Stereo* (MVS). Entre as variações deste algoritmo, o ponto principal gira em torno de se empregar o fenômeno físico real, ou seja, capturar numerosos pontos de vista de um objeto ou ambiente através de informações do sensor utilizado, geralmente uma câmera (DRISSYA, 2022). Uma das versões do processo envolve a extração de quanto mais imagens forem possíveis como entrada ao programa, gerando assim modelos tridimensionais com precisão elevada, as chamadas nuvens de pontos densos capturados (*dense points clouds*) (LIN, KONG, LUCEY, 2018). O fundamento da técnica é a reconstrução dos pontos de confluência armazenados

nessas nuvens para recriar o objeto ou sólidos a partir de diferentes pontos de cada superfície do objeto.

Figura 04 – Demonstração de Visão Estéreo, e a detecção da disparidade em pixels.



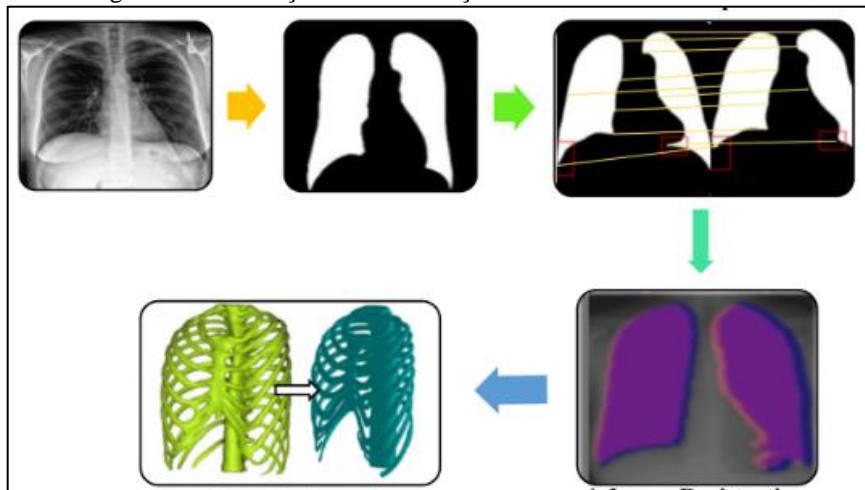
Fonte: (SCHARSTEIN, 1999)

Desde o final do século XIX, a humanidade foi capaz de usar raios-X na área médica, e com o avanço da computação, se tornou possível utilizar essas radiações em alta frequência no processo de reconstrução virtual do corpo humano. Existem vários tipos de soluções ao se reconstruir imagens de corpo inteiro a partir de raios-X, entre esses, os que utilizam formas modificáveis a partir de deformações de modelo se destaca (AHMAD et al., 2010). O procedimento se baseia em: inicialização um modelo básico pré-construído genérico, detectar diferenças entre esse modelo e as imagens raios-X, para então modificar o modelo virtual base para condizer ao corpo do paciente; tudo isso ao mesmo tempo em que se lida com os ruídos das imagens utilizadas.

A Figura 05 demonstra esse funcionamento, o processo envolve: aquisição/calibração da imagem 2D, extração de contornos (segmentação), correspondência de pontos, o registro das imagens, e a deformação do formato 3D virtual até corresponder ao corpo do paciente. A qualidade da reconstrução 3D do paciente é proporcional à quantidade de informação captada

pelos raios-X nas imagens fornecidas, levando em conta as imperfeições e a parte do corpo analisada (MAKEN E GUPTA, 2022).

Figura 05 – Ilustração da reconstrução virtual baseada em raios-X.



Fonte: (MAKEN E GUPTA, 2022).

Entre as principais dificuldades ao se lidar com a área de reconstrução virtual se deve pela dificuldade de ser transmitir fielmente a cena/objeto para o computador. Imagens manchadas, excesso de iluminação, foco, reflexo, entre outras, acabam por causar dois problemas impossíveis de se ignorar, exclusivamente relacionados à reconstrução. Primeiro, a flutuação de interferência (*noise*), ou ruído, na imagem acaba por, ao se fazer associações entre imagens distorcidas, confundir o algoritmo de qual sinal de imagem é o correto para a cena, o que ocasiona na interpretação exagerada do meio e formação durante a reconstrução de sólidos inexistentes à cena (PUETTER, GOSNELL, YAHIL, 2005).

Em segundo lugar, com posteriores acréscimos de imagens ao algoritmo e com o mesmo se tornando cada vez mais robusto, o erro se não tratado ou filtrado, pode se tornar impossível ao programa discernir o que é interferência e o que é de valor real a cena. Ainda mais, é possível ainda que dados de um conjunto de imagens sejam descartados por causa de pontos de ruído em comum, ocasionando em descartes de ponto de inferência importantes à cena (PUETTER, GOSNELL, YAHIL, 2005). Algoritmos de reconstrução tridimensional virtual exercem sua função ao menos tempo que combate esses dois problemas. A chave para uma reconstrução estável, assim como empregados nos mais modernos algoritmos atuais, é manter uma balança entre restrição de imagens, análise comparativa e escolher quando restringir os modelos. Quanto mais restritivo maior a estabilidade, mas também mais resultados positivos descartados, o desafio é conseguir manter uma razão saudável.

Um dos aliados na área de reconstrução de imagens é o aprendizado de máquina, sobretudo com o uso de rede neurais, pelo motivo de complementar as etapas que na maioria dos casos estão mais sujeitos a ruídos, e na correção de imagens onde essas interferências ocorrem com mais frequência (AHISHAKIYE et al, 2021). As técnicas de aprendizagem de máquina surgiram para servir de auxílio a outras técnicas, fornecendo resultados mais confiáveis e de classe superior a partes críticas de análise, detecção e compreensão de imagens.

A reconstrução tridimensional envolve a formação aproximativa de medidas durante a análise de imagens bidimensionais, no passo que as técnicas de aprendizado de máquina fornecem melhorias de velocidade, precisão e qualidade ao se remontar a imagem. No ambiente médico, por exemplo, as imagens de exames são usadas como diagnóstico e terapia, as técnicas de processamento melhoram a qualidade dessas imagens, métodos automáticos auxiliam por sua vez na aplicação dessas técnicas melhorando precisão e confiança. Ao se utilizar de aprendizado de máquina para fornecer melhoria nas pequenas etapas, toda a área é incrementada por fornecer precisão, confiabilidade e correção de erros tanto na captura, tratamento e reconstrução da imagem virtual (AHISHAKIYE et al, 2021).

As técnicas, InstantNGP e OpenMVG, analisadas neste trabalho partem da ideia de semelhança de imagens distintas de um mesmo local para encontrar a profundidade necessária para a posterior reconstrução, diferenciando entre si nas formas de captura de imagens, nos modos de comparação e como fazer a reconstrução sem exigir exageradamente do hardware.

## 2.2. Redes Neurais

Para auxiliar na eterna busca de respostas, desde o período do pós Segunda Guerra Mundial, os cientistas sempre buscaram pela capacidade de tornar as máquinas mais inteligentes e capazes de encontrar os resultados de questões. Inicialmente, somente com o objetivo de resolução de problemas matemáticos, com o avanço da tecnologia e hardware cada vez mais capazes, o termo de Inteligência Artificial ganhou cada vez espaço e atenção tanto na área científica quanto no meio popular (RUSSEL e NORVIG, 2010).

Como uma subárea da inteligência artificial, o aprendizado de máquina (*Machine Learning*) consiste em um conjunto de algoritmos capazes de aprender por si mesmos. Isso é possível através da análise de dados e do aprimoramento contínuo do próprio pacote de regras e de métodos de se atualizarem a cada interação. O resultado disso são os chamados



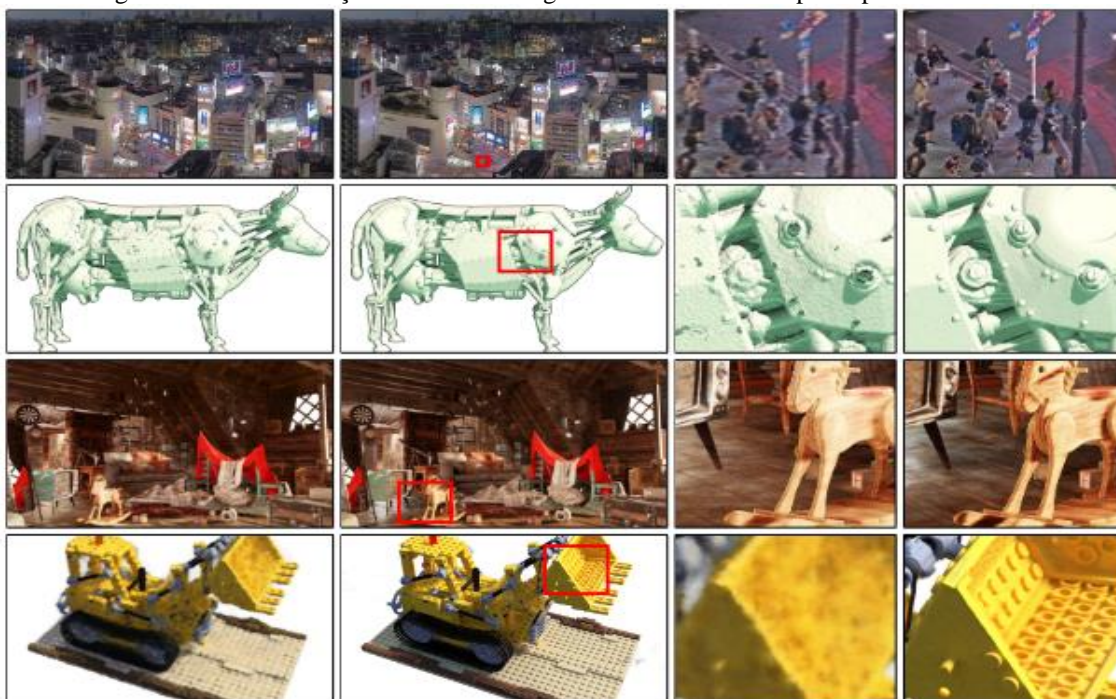
algoritmos de aprendizado, os quais, a partir de um determinado ciclo de estratégias e inferências, são capazes de chegar à resolução de problemas antes visto como sem soluções. Técnicas de aprendizado são empregados para reconstrução tanto pelo NeRF quanto pelo InstantNGP.

Os métodos tradicionais assim como os primeiros anos da área de aprendizado de máquina são focados no aprendizado em pequena escala, com as fases de testes compreendendo entre dezenas até milhares de exemplos. Em contraste, o aprendizado de máquina nos anos recentes geralmente alcança a capacidade de milhões de exemplos. Isso se deve por causa pela generalização do erro, que sempre está presente, ocupar um espaço no campo da hipótese (RUSSEL e NORVIG, 2010). A estimativa do erro, como uma forma de aumentar a precisão da função de resolução do algoritmo, nunca consegue chegar ao seu verdadeiro valor, tornando o erro sempre presente. Quanto maior a quantidade de exemplos fornecidos ao algoritmo mais preciso ele se torna, no entanto, os limites da computação moderna também entram em foco, tendo em conta a quantidade e qualidade de dados suficiente disponível à máquina (RUSSEL e NORVIG, 2010).

Com avanços nas áreas de hardware, e com equipamentos cada vez mais potentes, o ramo de Aprendizado Profundo (*Deep Learning*) para resolução de problemas ainda maiores da área de ciências da computação vem se destacando pela sua capacidade surpreendente de evolução contínua. As RNAs (Redes Neurais Artificiais), da mesma forma, surgiram para suprir a necessidade de reconhecimento de padrões cada vez mais robustos e o alto fluxo de dados (MOSTAFA, ISMAIL, LIN, 2012), o conceito foi elaborado usando como inspiração os neurônios biológicos do ser humano.

Simulando o funcionamento de um único neurônio humano que cumpre uma função e passa o resultado para o próximo neurônio através de funções chamadas sinapses, e assim por diante criando uma rede, se foi idealizado o Perceptron e o conceito de rede neural, as RNAs. No contexto tecnológico atual, umas das redes neurais que se destacam são os Perceptrons de Multicamadas, que através dos processos de treinamento, teste e validação são capazes de entregar resultados elaborados por se utilizar de vários níveis de perceptrons conectados entre si para resolução de problemas complexos. A Figura 06 demonstra o uso do aprendizado de máquina na melhoria na qualidade de imagens.

Figura 06 – Demonstração de cada abordagem ao decorrer do tempo de processamento.



Fonte: (MULLER, 2022).

O resultado que se espera alcançar ao se treinar uma RNA define que modelo deve ser utilizado. Entre as principais arquiteturas, as MLPs (multi-layer perceptrons) apresentam uma estratégia válida para se lidar com alto fluxo de dados e assim, atingir o objetivo citado acima. Isso se deve por causa da natureza das MLPs terem como objetivo reconhecer padrões sutis nos mais diversos problemas (GARDNER, 1998). A atribuição de valores (peso) entre as camadas de neurônios conectados entre si formam uma rede capaz de chegar a um resultado satisfatório e lidar com uma alta gama de resultados diferentes.

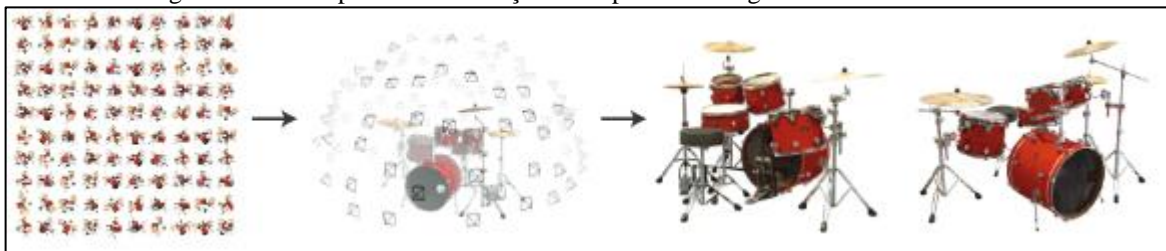
### 2.3. NeRF

Objetivando solucionar o recorrente problema da perda de escalamento que geralmente ocorre ao se processar a síntese de visão, Mildenhall et al. (2010) desenvolveram o algoritmo NeRF (*Neural Radiance Field*), um conjunto de métodos para tratar da geração de cenas complexas utilizando campos de radiância através de um rede neural. A NeRF se resume em uma rede neural completamente conectada e treinada capaz de gerar complexas cenas 3D a partir de imagens 2D. Para isso trabalha recebendo como entrada diversas imagens representando um cenário e interpolando as mesmas para formação e renderização da cena completa (MILDENHALL, 2020). A pesquisa e métodos desenvolvidas no NeRF serviu de

base para desenvolvimento da primeira tecnologia escolhida, o InstantNGP bem como a utilização do aprendizado de máquina para modelagem 3D.

O processo envolve representar uma cena estática junto a uma função contínua, para isso se utiliza de raios de radiância emitidos de pontos em duas direções ao mesmo tempo em que cruzam mais três pontos no espaço determinado, criando um plano 5D (três pontos no plano 3D e mais dois num plano 2D). Para ser capaz de uma criar uma única malha volumétrica de alta densidade de pontos de múltiplas coordenadas e sensibilidade a paleta de cores RGB, o NeRF emprega e otimiza passo-a-passo uma rede neural completamente conectada sem camadas (MILDENHALL, 2020). A rede acaba acumulando uma alta quantidade de representações da cena de cada ângulo do objeto, possibilitando uma renderização de qualquer ponto de vista. A Figura 07 ilustra um exemplo de reconstrução 3D utilizando o NeRF.

Figura 07 – Exemplo de reconstrução 3D a partir de imagens 2D utilizando NeRF.



Fonte: (MILDENHALL, 2020).

O processo de renderização do campo de visão neural envolve uma série de passos que são seguidos à risca durante o processo de compreensão da rede. O primeiro passo é gerar um conjunto de pontos 3D de entrada do cenário através da movimentação da câmera, dessa forma todo o entorno do objeto assim como seu redor fica registrado. O segundo passo é utilizar esses pontos de correspondência espacial para alimentar a rede neural como entrada, produzindo como resultado um conjunto de cores e densidades de pontos. Para só então utilizar as técnicas clássicas de renderização de volume para condensar as cores e densidades em uma imagem 3D virtual.

O processo de renderização volumétrica baseada em técnicas clássicas é um dos diferenciais do algoritmo, junto à otimização das representações de imagens de forma padrão. Importante mencionar também a realocação de recursos da rede neural conforme os pontos de radiância são mapeados, configurando em uma otimização de tempo. Todo o posicionamento espacial de cada entrada na rede neural através da otimização dos campos de radiância possibilita o algoritmo representar todo seu conteúdo de forma foto-realística em altas

resoluções e frequências, assim como o trabalho de treinamento de redes neurais em representações volumétricas (MILDENHALL, 2020).

A rede neural de camada única utilizada no algoritmo NeRF é treinada para mapear diretamente desde a visualização e localização espacial até a opacidade e a cor. No entanto, o NeRF é um algoritmo computacionalmente complexo, como etapas essenciais e um alto fluxo de dados, o que resulta em processamentos mais demorados de forma que até mesmo em máquinas com o hardware recomendado tomam um elevado tempo de processamento. Por usar renderização volumétrica para gerar novas visualizações e com código aberto, a tecnologia NeRF permite que novos algoritmos sejam criados constantemente visando melhorar o desempenho da tecnologia.

O trabalho de Mildenhall et al serviu de base para vários outros autores, no entanto, o NeRF em si é restritivo devido ao seu alto requerimento de hardware. Apesar de originalmente ser a tecnologia escolhida, o custo computacional e o tempo de treinamento foram fatores que contribuíram para a escolha de outra técnica, o InstantNGP, uma variação do NeRF.

#### 2.4. InstantNGP

A fidelidade visual de cenários representados de forma matemática é fortemente atrelada à sua qualidade e desempenho, de forma que qualquer olho, treinado ou não, consegue avaliar se qualquer trabalho na área atinge certo nível. Representações que se mantêm rápidas, fluidas e compactas ao mesmo tempo em que preservam um alto grau de qualidade de imagem e alta frequência na taxa de quadros sempre são desejadas, apesar de raramente serem alcançadas (MULLER, 2022).

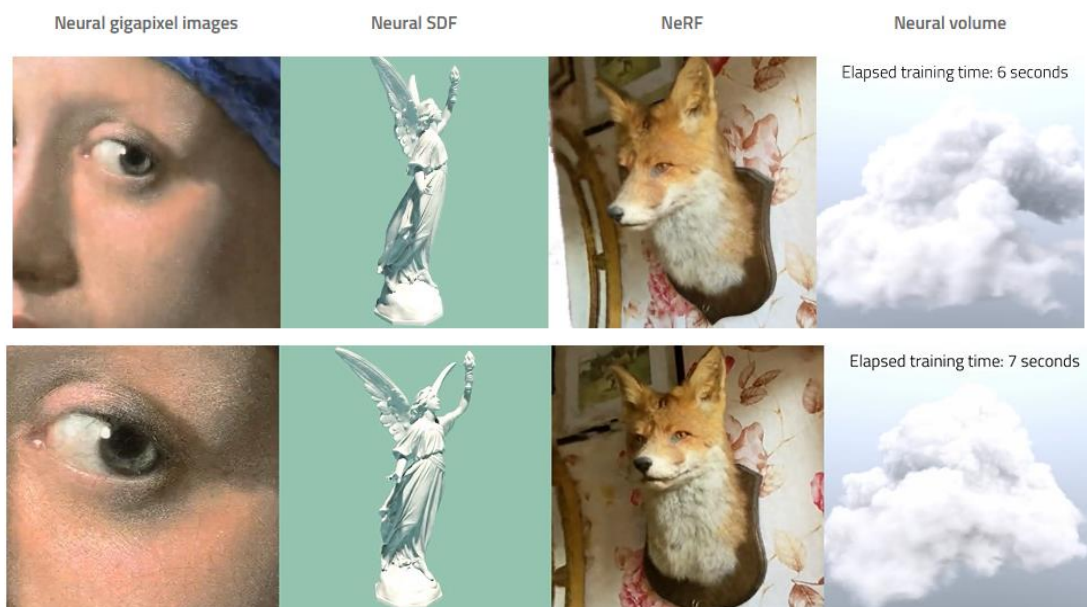
O aprendizado utilizando retropropagação de erro possibilita à tecnologia de MLPs entregar resultados altamente confiáveis ao selecionar quais dados são considerados satisfatórios ou descartáveis. Isso permite o uso de MLPs mais eficientes e de melhor qualidade. No entanto, isso traz uma exigência maior às GPUs (unidade de processamento gráfico) da máquina, limitando o uso (MULLER, 2022). As MLPs que são treinadas com estruturas de dados específicas para uma determinada tarefa apresentam os melhores resultados.

Como primeira técnica escolhida, o InstantNGP (Instant Neural Graphic Primitive) tenta resolver os problemas citados utilizando MLPs menores porém mais eficientes, como as

utilizadas para resolução de campos de radiância (MILDENHALL, 2020). O InstantNGP mapeia a entrada em espaços altamente dimensionais, capazes de extrair aproximações mais confiáveis em relação a conjuntos de modelos mais compactos. No entanto, ao se utilizar de redes neurais que empregam heurísticas e modificações estruturais, o processo de treinamento das mesmas passa a ser mais complicadas, restringindo o método a casos mais específicos, com parâmetros limitados, ao mesmo tempo em que requer um hardware mais robusto, tornando o uso mais custoso de tempo (MULLER, 2022).

O InstantNGP utiliza de primitivas gráficas para treinamento de redes neurais, tendo como objetivo se adaptar e aprender de forma mais rápida, auxiliando o treinamento de alguns seletos problemas de representação de imagens. A técnica originalmente possui codificações para quatro tarefas com imagens: o treinamento de coordenadas bidimensionais em imagens com alta resolução (*Gigapixel image*); mapeamento de pontos 3D para determinar pontos de profundidade em imagens 2D a partir da profundidade da tela (*Neural signed distance functions* – SDF); aprendizado de volume neurais em campos de densidade e radiância direto de pontos marcados em volume; e principalmente o NeRF (MILDENHALL, 2020), que faz o aprendizado de densidade para reconstrução 3D baseado em campos de luz volumétricos retirados de imagens 2D em várias perspectivas distintas. A Figura 08 demonstra os resultados dos diferentes algoritmos que são desenvolvidos pelo InstantNGP. Da esquerda para direita: Imagens em alta resolução; coordenadas 3D de objetos – SDF; campos de luz para treinamento de ambientes 3D – NeRF; treinamento para aprofundamento volumétrico.

Figura 08 – Demonstração de resultados para cada estratégia.



Fonte: (MULLER, 2022).

Com os conhecimentos de redes neurais e aprendizado de máquina, as resoluções de questões mais complexas tornam-se possíveis, assim como se tornar uma porta de entrada à área de inteligência virtual ao público mais leigo. A capacidade de uma rede neural de aprender e aprimorar a própria rede pode soar até assustador. No entanto, algoritmos como o InstantNGP, capazes de utilizar esse processo de aprendizagem complexo para resolução de problemas, criam novas possibilidades de ideias.

No meio desse conceito, o propósito do InstantNGP busca uma implementação eficiente que visa proporcionar rapidez, diminuindo o tempo gasto principalmente com o treinamento; uma qualidade elevada nas imagens, com resolução alta e sem comprometer a performance; e seguindo o lema dos pesquisadores, manter a simplicidade de código, tornando possível o treinamento de redes para imagens de grau elevado de resolução ainda se mantenha o uso de redes neurais de pouca grau de magnitude (MULLER, 2022).

## 2.5. OpenMVG

Avanços na área de Visão Computacional vêm ganhando cada vez mais importância na computação, seja pelo uso em aplicativos em nossos smartphones ou em reconhecimento de imagens espaciais. Isso se deve pela necessidade gradativamente maior de se obter informações confiáveis de significado a partir de imagens digitais, vídeo, ou quaisquer meios de entrada digital, que é o foco principal dessa área. Em suma, a captura, interpretação e compreensão de imagens são abrangidas pela Visão computacional, uma tentativa de treinarmos o computador para agir e “pensar” assim como a visão e raciocínio humano é capaz, levantando em conta tanto experiência, adaptação e compreensão.

A visão computacional, assim como a humana que se espelha, depende de uma fonte de entrada, se valendo de sensores, câmeras, imagens e algoritmos, no lugar de olhos e nervos ópticos, para observar e inspecionar o ambiente e objetos a sua volta. A aplicação dessa tecnologia é praticamente ilimitada, desde a análise de objetos microscópicos, até a análise de sistemas estelares, se restringindo somente ao que o hardware é capaz. Sendo utilizada nas mais diversas ramificações, desde a indústria com análise precisa, quanto à avaliação de sistemas de risco, a visão computacional junto com a inteligência artificial somente tende a crescer e evoluir.

Uma das aplicações que apresenta atualmente ótimos resultados seria a da aplicação de técnicas de análise de imagens com as técnicas de reconstrução 3D, algo que apresenta

resultados tão favoráveis que já há alguns anos, comunidades inteiras na internet são criadas para compartilhamento de algoritmos e técnicas para um treinamento mais preciso e menos sujeitos a erros. A ideia de uma abordagem (framework) mais comum, de propósito mais generalista, com baixo ou nenhum custo, seja compartilhada em uma comunidade entusiasta, capaz de comunicar experiências, fazer testes e divulga-los, com novos protótipos derivados seria um modelo ideal para todos os pesquisadores na área.

Como segunda tecnologia a ser analisada, o OpenMVG (*Open Multiple View Geometry*), uma biblioteca vasta de ferramentas e algoritmos de geometria de múltiplas visões, que tem o objetivo de difusão do uso da Visão Computacional para resolução de problemas com representação e reconstrução visuais digitais. A tecnologia foi criada para ser e se manter simples, com o intuito de manter uma coleção modular de algoritmos, bibliotecas e recursos, a ser utilizada em blocos ou individualmente para construção de sistemas maiores. Por ser modificável e de fácil uso, o OpenMVG possui uma comunidade de considerável tamanho e dos mais variados campos, seja educacional ou de mercado, o que contribui para ser manter ativo e atual (MOULON, 2017). Na Figura 09, exemplos empregando o OpenMVG para modelagem de paisagens.

Figura 09 – Reconstruções feitas por não profissionais utilizando o OpenMVG.



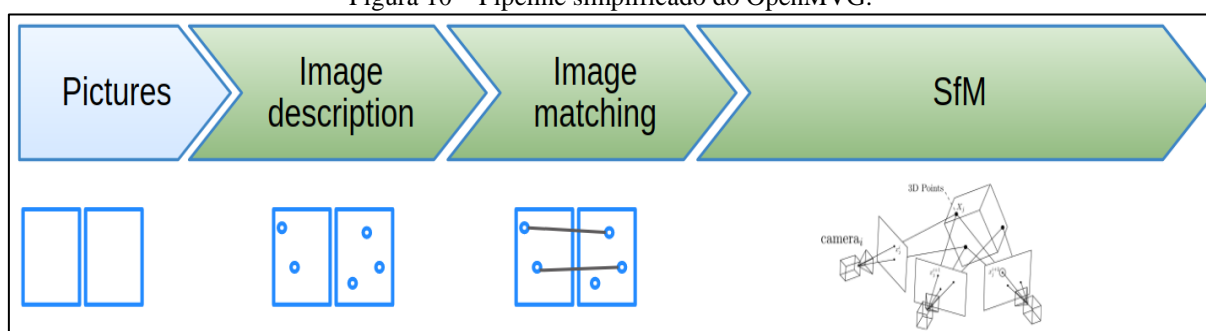
Fonte: (Perrot, 2016)

As contribuições na área de Visão Computacional são várias: um acesso facilitado para implementações precisas de algoritmos de geometria de múltipla visão, uma biblioteca de código de fácil entendimento, um conjunto de ferramentas para construção completa de cenários dependente de pipelines diferentes (MOULON, 2017). Com um considerável número de funcionalidades, o OpenMVG fornece alternativas para processamento e carregamento de imagens, bem como para detecção e comparação de padrões visuais. Aplicações que utilizam a técnica *Multiple View Geometry* tentam recuperar, através de cálculos, a posição, localização e orientação da câmera ao utilizar os dados contidos tanto nas imagens quanto na câmera de foram tiradas, como por exemplo, o ponto focal e o comprimento do foco.

O OpenMVG foi escrito em C++, no padrão C++11, e usando o sistema CMake para montagem do programa. O OpenMVG fornece um conjunto de pequenas bibliotecas que podem ser usadas de forma independente ou como blocos de pipeline - uma cadeia de elementos a serem processados consecutivamente até a conclusão do algoritmo - para reconstrução tridimensional a partir de imagens. Entre os alicerces para o alto desempenho do OpenMVG estão: a biblioteca Eigen (JACOB e GUENNEBAUD, 2010) que permite realizar operações de em álgebra linear com alta performance; a biblioteca Ceres para realizar otimizações não-lineares de grande escala; e a biblioteca OSI-CLP para resolução de programação linear (MOULON, 2017).

Trabalhando com containerização, o processamento em *pipeline* (Figura 10) do OpenMVG é o alicerce que permite a comunicação suave entre as ferramentas durante todo o processo, desde a inserção das imagens, pelos dados de imagem e câmera computados, a associação de ponto em comum, os resultados documentados e processados pelo contêiner. O container armazena as imagens e suas informações relacionadas: visão abstrata (*metadata* da imagem, identificação da câmera e a posição), modelos abstratos de câmera, posição da câmera, marcações de estrutura e as identificações observadas em cada imagem. Graças à interface do programa, o container pode ser salvo em binário, para rápida compactação e leitura/escrita, ou em formato JSON/XML (para fácil exportação) (MOULON, 2017).

Figura 10 – Pipeline simplificado do OpenMVG.

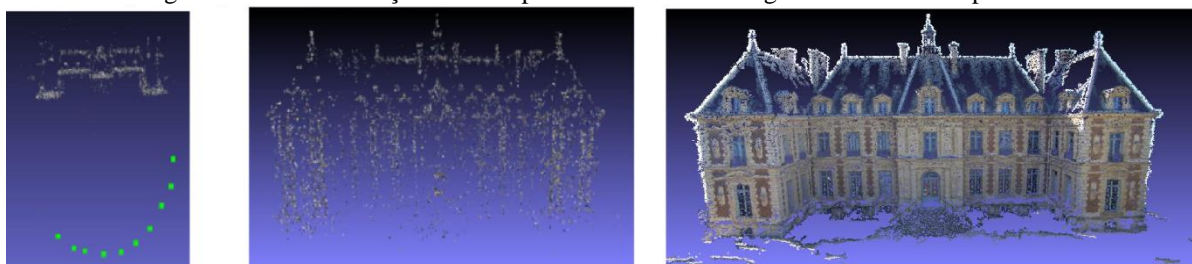


Fonte: (Moulon, 2017).

Para correspondência entre imagens é ainda utilizada o algoritmo do vizinho mais próximo (*nearest neighbor search*) que pode ser utilizado em qualquer dimensão de vetor (MOULON, 2017). Ao utilizar algoritmos de busca é possível computar os pontos 3D próximos ou mesmo encontrar pontos correspondentes em pares de imagens de um determinado cenário (Figura 11).



Figura 11 – Demonstração de exemplo básico com 10 imagens utilizando o OpenMVG.



Fonte: (MOULON, 2017)

Graças a essa customização, o usuário pode controlar a relação de precisão e tempo na tarefa exaustiva de pontos similares. Os pontos similares quando detectados são filtrados como combinações geométricas coerentes dentro dos modelos de geometria de visão múltipla. Por utilizar o formato de dados SVG se torna possível preservar os detalhes das imagens ao expandir a reconstrução virtual graças à natureza do formato. Além desses ainda são utilizados os conceitos de geometria de visão múltipla: posição relativa, posição absoluta, transformação por similaridade, triangulação 3D (MOULON, 2017).

Por utilizar dados de mundo real como entrada, o problema do ruído digital e falha na correspondência nos pares de pontas se torna um impedimento à tecnologia. Portanto, se torna necessário o emprego do método estimação de modelo robusta, no qual o OpenMVG propõe vários métodos para estimação, nos quais fica a cargo do usuário definir o limite que deseja que o programa siga, ou caso não deseje utilizar os definidos por padrão baseados em balanço estatístico pré-avaliado (MOULON, 2017). O programa por padrão utiliza: limiar anterior (*Threshold prior*), quadrados menos aparados (*Least trimmed squares*), limiar livre (*Threshold free*).

Dentre os motivos principais para escolha de análise do OpenMVG, o primeiro ponto a se destacar seria a facilidade que a tecnologia tem de não exigir demais do hardware, caso que foi restrigente na hora de vários testes. Isso ficou evidente no momento em que a máquina em que foram feitos os testes iniciais de outras técnicas não se mostrou capaz de completar os testes, fato que não ocorreu ao utilizar o OpenMVG, por utilizar bem menos memória e uso da CPU em comparação aos demais.

Outro fator importante a acrescentar é o fato do OpenMVG possuir uma comunidade ativa e disposta a ajudar ao se encontrar dificuldades de manipulação e reconstrução utilizando cenários originais próprios. A Tabela 01 demonstra as estatísticas do site Github com relação a outras tecnologias na mesma área de atuação, dando uma ideia do tamanho da comunidade, levando em conta o ano de cada.

Tabela 01 – Informações de outras tecnologias em 2017 em comparação ao OpenMVG.

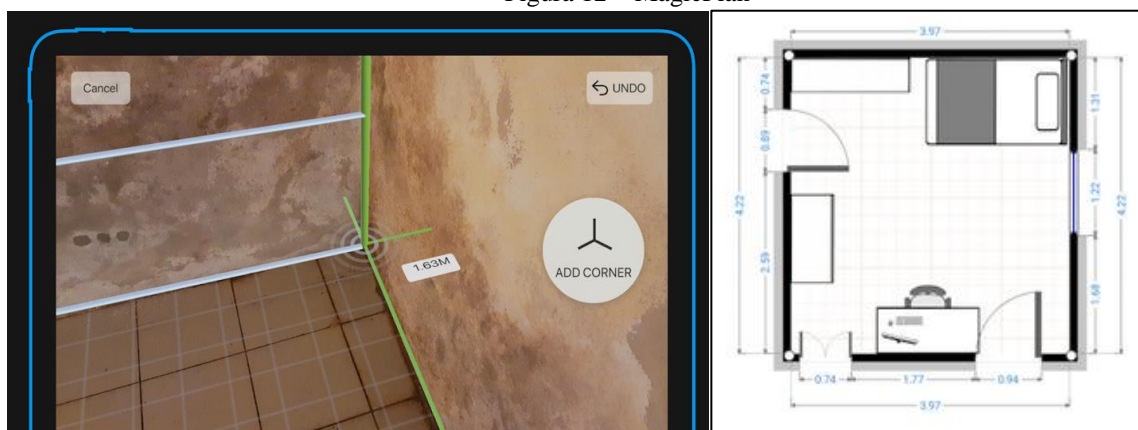
Nomes dos projetos	Ano de criação	Contribuição ao projeto	Acompanhamentos	Favoritos	Vertentes do projeto original
bundler_sfm	2008	8	108	530	245
COLMAP	2016	5	14	82	34
MVE	2012	13	61	188	131
<b>OpenMVG</b>	<b>2013</b>	<b>31</b>	<b>156</b>	<b>802</b>	<b>392</b>
THEIASfm	2015	15	43	165	80

Fonte: (MOULON, 2017)

## 2.6. Trabalhos relacionados

A fim de salientar a necessidade e importância da construção deste trabalho, foi realizada uma pesquisa sobre trabalhos e aplicações que contribuíram com ideias, conceitos ou mesmo comparações. Dessa maneira, foram utilizados: a plataforma Google Scholar, utilizando-se de palavras chaves que sincronizem com os temas abordados no projeto e tenham conceitos ou técnicas que tornem o projeto mais conciso e objetivo; e as principais lojas de aplicativos para celular, Google Play e Apple Store, para encontrar aplicativos que estejam relacionados à área da pesquisa, de forma que forneça uma forma diferente de solucionar o problema alvo. A Figura 12 mostra o funcionamento do MagicPlan.

Figura 12 – MagicPlan



Fonte: [www.magicplan.app](http://www.magicplan.app)

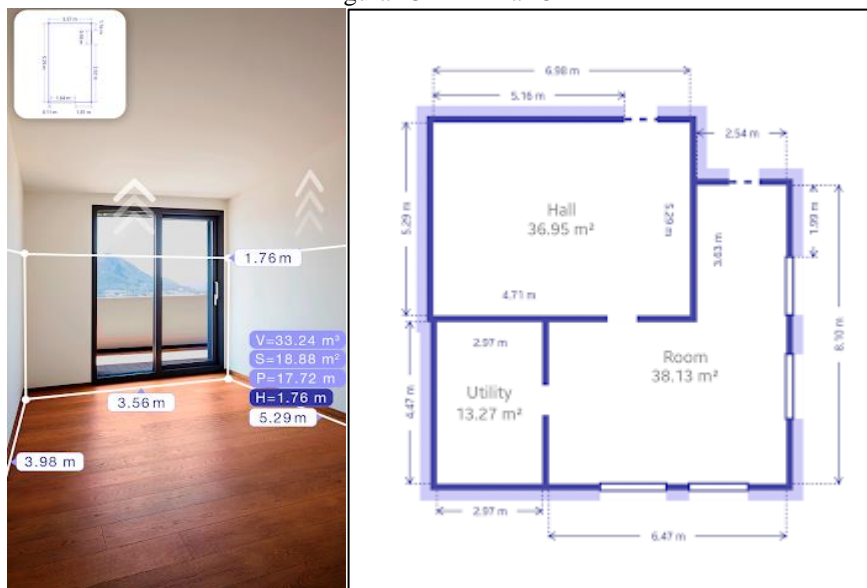
O aplicativo MagicPlan, disponível nas lojas Google Play e Apple Store, trabalha com geração de plantas baixas, possibilitando o usuário usar a câmera do celular para obter as medidas do local que se encontra, essas medidas são armazenadas e utilizadas para criação de uma planta do local, que poderá ser acessada e compartilhada com, por exemplo, o contratante de uma obra, ou um arquiteto.

Apesar de possuir um interface limpa, a versão gratuita do MagicPlan possui algumas limitações, algumas colocadas para incentivar a compra do mesmo, entre elas: o fato que, por

ter como objetivo de trabalhar somente com medidas bidimensionais para geração de suas plantas, o usuário não possuirá visão do local como um todo, mesmo que por fotos; e as restrições que o aplicativo possui em suas funcionalidades na versão gratuita, sendo necessária a compra para completo uso sem limites.

De mesma forma, o aplicativo ARPlan 3D (Figura 13), se encontra disponível nas plataformas de aplicativos para Android e Apple, possuindo a mesma funcionalidade de extração de medidas de ambientes que o MagicPlan e a captura das medidas de altura através do celular do usuário. Apesar de ter como diferença positiva a utilização de uma interface sem tanta poluição visual na tela, o aspecto pago destoa da quantidade de funcionalidades que o MagicPlan disponibiliza. O uso é regulado por tempo de uso com poucas utilidades disponíveis, sendo preciso que o usuário faça uma assinatura para continuar a desfrutar.

Figura 13 – ARPlan 3D



Fonte: [https://play.google.com/store/apps/details?id=com.grymala.arplan&hl=en\\_CA&gl=ES](https://play.google.com/store/apps/details?id=com.grymala.arplan&hl=en_CA&gl=ES)

Apesar de eficiente em sua proposta, pois o usuário consegue extrair um modelo virtual com as medições dos ambientes, os usuários não conseguem visualizar o ambiente virtualmente, nem mesmo por fotos, já que elas não são salvas. Apesar de tudo, o aplicativo possui alguns pontos fortes que servem de referência ao trabalho, como por exemplo, a captura das medidas em três dimensões e a divisões dos cômodos de um recinto, que posteriormente podem ser acessados pelo usuário, subdividindo o local escaneado, o que aumenta a praticidade de uso.

No trabalho de HENRY et. al. (2012), os autores abordaram o problema de uma forma diferente. Utilizando câmeras do tipo RGB-D similares ao Kinect da Microsoft, capazes de

captar as cores dos pixels e de calcular a distância para objetos, eles conseguiram mapear e reconstruir ambientes com os vários pedaços captados. O objetivo do trabalho foi o de melhorar aplicações que necessitavam de uma alta precisão ao seu redor, como sistemas robóticos autônomos ou ainda o mapeamento e manipulação visual.

Com esse objetivo foram utilizados vários algoritmos de otimização combinados com recursos de combinação visual e alinhamento de formas, resultando em uma reconstrução parcial de ambientes em que o sistema seria capaz de armazenar e fazer a visualização em profundidade dos dois grandes ambientes internos que foram utilizados. O trabalho de HENRY et. al. (2012), servirá como inspiração ao decorrer do trabalho, tendo em vista a capacidade de extração pequenos pedaços de frames maiores capturados, bem como a análise dos mesmos, informações de vital importância no decorrer da análise das técnicas.

### **3. Estudo de caso – Especificações de cenário, hardware e sistemas.**

Com o objetivo de analisar os resultados ao utilizar tanto o InstantNGP quanto o OpenMVG, sendo possível demonstrar o conceito de reconstrução virtual a partir de um cenário real, este trabalho utiliza de técnicas atuais e tecnologias de propósitos variados para alcançar o objetivo de conseguir, ou ao menos abrir caminho, a capacidade de representar virtualmente um ambiente real na tela de um dispositivo computacional.

No âmbito da escolha do ambiente foram feitos vários testes aplicando as tecnologias escolhidas em diferentes ambientes, até que se optou por testes em menor escala, por meio de maquetes e miniaturas de objetos. O motivo para isso se deve pela dificuldade em encontrar ambientes reais vazios, que é o foco do trabalho. Nas primeiras rodadas de testes, em que se foram usados cômodos vazios em residências em manutenção e o tempo diminuto para captura de imagens, os resultados alcançados foram de uma qualidade muito abaixo do esperado. Arelado a esse fato, a exigência que era requerida de hardware era bem maior ao se usar fotos com resoluções bem elevadas, visto a necessidade de obter a maior qualidade de imagem possível. Foram esses fatores que levaram à escolha das tecnologias analisadas.

As técnicas escolhidas para comparação (InstantNGP e OpenMVG) foram escolhidas pela facilidade que ambas as tecnologias apresentaram em ser capaz de escanear o ambiente e definir um ponto focal ideal de acordo com o problema inicial para visualização tridimensional virtual. Não sendo as primeiras tecnologias escolhidas, ambas se sobressaíram, principalmente, pela exigência reduzida de hardware que possuem.

Tido como resultado principal deste trabalho, a análise de resultados de ambas as técnicas e seus respectivos desafios durante sua aplicação visa, principalmente, demonstrar as dificuldades tanto de hardware como de software, ou mesmo no desenvolvimento inicial, as inúmeras complicações que se pode encontrar ao lidar com modelagem tridimensional à partir de imagens e suas aplicações em computador, 2D e 3D.

#### **3.1. Descrição de Cenário**

Para realizar os testes das tecnologias escolhidas, foram utilizados alguns cenários, tanto em escala real (cômodos reais) quanto em proporção (maquetes ou miniaturas). Conforme mencionado, as ferramentas utilizadas neste trabalho utilizam imagens 2D como entrada. Nos primeiros testes foram utilizados vídeos para aquisição de imagens do

cenário/ambiente, e a extração das cenas (frames) das gravações foram feitas através da seleção manual das imagens a serem utilizadas.

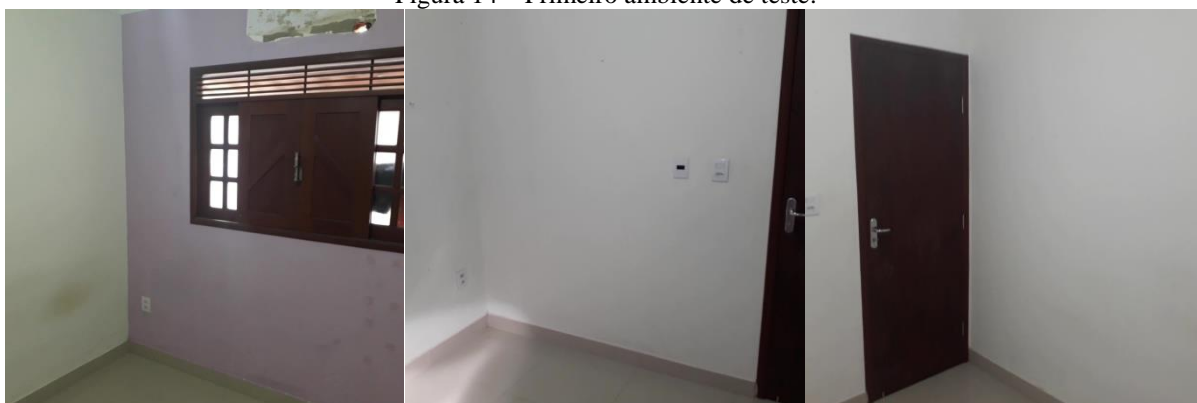
Uma vez que as gravações foram feitas a partir de um aparelho móvel do tipo celular com a necessidade constante de se focalizar, foram necessárias gravações longas para ter certeza que cada ambiente possuía várias imagens com suficiente qualidade, sem borrões, com pontos focais bem localizados e focados, e sempre tendo a certeza que todos os pontos do local foram captados e com múltiplos pontos de vista de qualidade. Esse processo é um requisito necessário para que a reconstrução se mostrasse possível e fiel ao ambiente original.

Primeiramente, deve-se saber que ambas as tecnologias são baseadas nas premissas de reconstrução 3D por meio de imagens, levando em conta o princípio que um objeto 3D em um ambiente virtual, na verdade é a união de pequenas partes. Essas partes se apresentam como uma pequena porção do todo, sendo necessários os processos de localização, captação, extração, e posicionamento. Mesmo com todo o processo sendo, essencialmente, diferente, no final o resultado parte do mesmo objetivo: a realocação das pequenas partes do ambiente de estudo para reconstruí-lo no computador.

As tecnologias escolhidas, apesar de cumprirem eficientemente o objetivo do estudo, não foram as primeiras escolhidas, o motivo para isso foi principalmente a barreira de hardware. O primeiro computador de uso pessoal não foi capaz de suportar a grande exigência de memória e processamento que era necessário para a primeira tecnologia escolhida, o NeRF, sendo necessário escolher outras que se adequasse melhor ao problema e às exigências para execução e com resultados satisfatórios. Por causa desse mesmo problema, os ambientes de testes também tiveram que ser adaptados à máquina à disposição.

Inicialmente, após a procura de local que se estivesse vazio, foi encontrado em uma residência em reformas um cômodo a disposição com proporções suficientes para um primeiro teste, sendo elas: 2,80 metros de comprimento, 2,60 metros de largura e 2,70 metros de altura (Figura 14). A razão para a escolha se deveu pela disponibilidade, tomando em nota a dificuldade em se encontrar um local completamente vazio com o dono ou morador à disposição de relegar o espaço mesmo que momentaneamente, e em segundo lugar o desejo do primeiro teste ser feito em ambiente de pequenas proporções para não sobrecarregar o programa, e por consequência, o computador. O resultado foi bastante abaixo do esperado ao se utilizar o InstantNGP, com a remodelagem gerando uma massa disforme na hora de reconstruir 3D, devido principalmente ao baixo grau de iluminação do ambiente. Posteriormente, o ambiente foi descartado devido à perda da disponibilidade do local.

Figura 14 – Primeiro ambiente de teste.



Fonte: De autoria própria.

Com a necessidade de um novo ambiente de testes, foi encontrado um novo cômodo com maior disponibilidade, novamente em reformas. Como se tratava de um local pertencente a conhecidos, foi possível utilizar novos testes com taxas de iluminação diferentes, tendo o intuito de avaliar qual grau de iluminação ideal para os testes, assim como determinar se sua origem seria melhor natural ou artificial. As medidas do local eram, no tempo de captura das imagens, 2,64m de altura por 2,70m de largura e 2,70m de comprimento. As Figuras 15 e 16 ilustram esse ambiente, com a segunda ilustração apresentando iluminação feita pela única lâmpada do cômodo, sem a iluminação da câmera e a luz externa bloqueada por plástico fosco preto.

Figura 15 – Segundo ambiente de teste, iluminado natural e artificialmente pela câmera.



Fonte: De autoria própria.

Figura 16 – Segundo ambiente de teste.



Fonte: De autoria própria.

A primeira captura de imagens por meio de vídeo se procedeu com uma iluminação natural sem incidência direta do sol pela tarde, capturando todos os pontos do local evitando o ofuscamento da lente da câmera pelos raios de sol. A segunda captura foi executada à noite, com a única janela tampada por fora para evitar qualquer iluminação não planejada, utilizando somente da iluminação artificial da única lâmpada fixa no local. A terceira captura foi executada pela parte da manhã utilizando três pontos de luz: o reflexo dos raios de luz que entravam pela janela, a luz artificial fixa do cômodo, e ainda a luz artificial do próprio aparelho de captura; tudo com o intuito de obter o máximo possível de iluminação sem sobrecarregar a lente da câmera com luz.

Nos três momentos de captura, o mesmo problema surgiu. Durante a reconstrução 3D, a iluminação se provou excessiva, visto que alguns cenários virtuais até mesmo a luz incidindo em alguma face sólida e refletida na lente era reconstruída como um sólido, tornando a imagem tridimensional resultante, novamente, em uma massa disforme irreconhecível e, portanto descartada.

Importante mencionar que até então somente foi utilizado somente o InstantNGP para ambos os testes, uma vez que era esperado resultados mais conclusivos e confiáveis dessa tecnologia por utilizar rede neural. Com o objetivo de ser possível realizar a análise comparativa entre as tecnologias, somente os testes em que ambas as tecnologias conseguiram entregar resultados com uma qualidade razoável foram selecionados, por exemplo, a remodelagem 3D de forma minimamente reconhecível.

Em todo caso, a solução para testes foi encontrada ao mudar novamente o ambiente de testes. De modo a testar de que forma seria possível determinar o grau necessário de qualidade de imagem e luz, a solução foi diminuir o ambiente de testes, estando este proporcional a um ambiente real. Para isso, duas maquetes foram construídas: a primeira de



proporções 1:7,5 (para cada centímetro da maquete equivale a 7,5 no ambiente real) com medidas de 36cm de largura por 36cm de comprimento por 35 cm de altura, aproximadamente; a segunda maquete de proporções 1:13,0 e medidas de 22,5 cm de largura por 30 cm de comprimento por 21 cm de altura. A Figura 17 apresenta a primeira maquete, com visão realizada de cima, feita com materiais simples e utilização de imagens de um ambiente real para referência, com a preocupação de proporções e a tentativa de realismo.

Figura 17 – Diversos ângulos da primeira maquete.

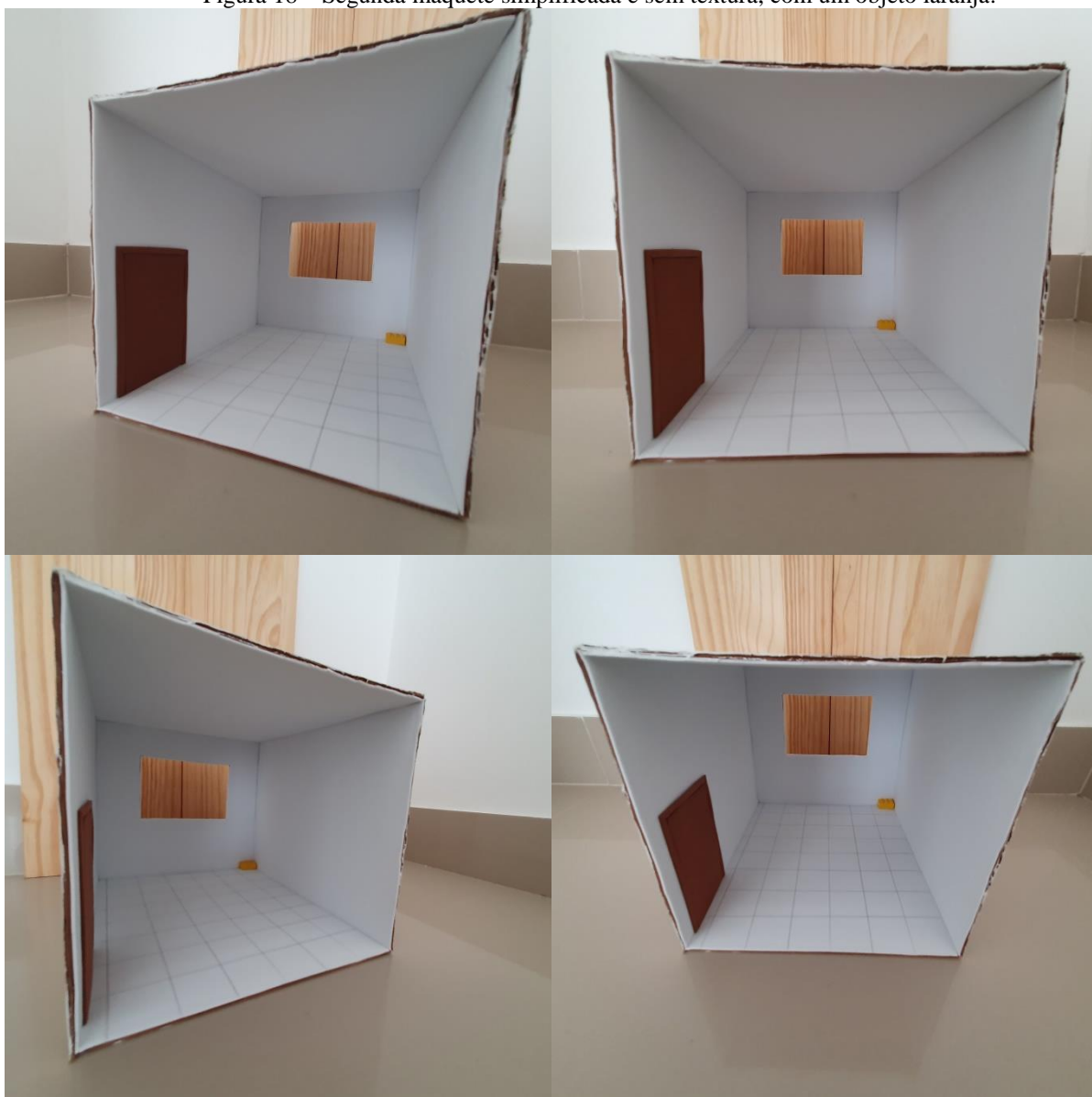


Fonte: De autoria própria.

A primeira maquete tinha o intuito de servir como um ambiente mais flexível menor e portátil baseado no segundo cômodo de teste para testes. Para transmitir a sensação de uma miniatura do cômodo, foram utilizadas as primeiras fotos do cômodo capturadas. As imagens foram impressas em folhas de papel, buscando manter a proporção do local. Essas fotos foram então encaixadas na maquete condizente com as localidades no ambiente real. No entanto, mesmo com a reconstrução virtual ter surtido uma significativa melhora, ainda estava da qualidade desejada.

Assim uma nova maquete foi feita, com visão lateral, a segunda (Figura 18), totalmente original, sem utilização de imagens, na cor branca utilizando também papelão e folha de EVA. Como ponto de referência para o programa foi recortado uma abertura na parte de fundo da maquete de forma a simular uma janela e fornecesse uma noção de profundidade à câmera, bem como o chão com uma malha imitando uma padronização de cerâmica/piso e um pequeno objeto laranja ao canto da cena como ponto de referência ao programa.

Figura 18 – Segunda maquete simplificada e sem textura, com um objeto laranja.



Fonte: De autoria própria.

Com a segunda maquete, a reconstrução e a manipulação do ambiente alcançou um nível de qualidade até então desejado. Ao utilizar as técnicas para encontrar pontos de correspondência, um maior número de pontos para construção de uma malha para o modelo 3D foi alcançado, assim como a qualidade boa para a reconstrução 3D. O qual por sua vez, ao serem utilizados para remapeamento e, por conseguinte reconstrução, o modelo virtual alcançou sua qualidade desejada, bem como sua manipulação virtual.

## 3.2. Descrição de Hardware e Sistemas

### 3.2.1. Dos computadores

Foram utilizados dois computadores para o desenvolvimento deste trabalho, além de câmeras embutidas em celulares pessoais para aquisição das imagens (vídeos e fotos).

A primeira máquina, um computador próprio, de uso pessoal foi selecionada, principalmente pela facilidade ao acesso e pela máquina possuir uma placa de vídeo integrada com memória também dedicada. A escolha, inicialmente, parecia atender às necessidades dos programas que seriam utilizados no decorrer do trabalho. No entanto, na prática, foi percebido que a máquina não conseguia atender os resultados desejados. Os motivos, aparentemente, se centraram no fato que a máquina não possuía uma memória dedicada na placa de vídeo suficiente para comportar o funcionamento do programa, resultando muitas vezes em meios a diversas etapas sendo encerrado previamente pelo sistema devido à memória RAM e dedicada chegando ao limite durante a execução.

Vale destacar que apesar da máquina ser um computador portátil tipo notebook com ao menos nove (09) anos de uso, ele conseguiu executar algumas etapas durante o projeto, como por exemplo, a extração de frames de vídeos captados pela câmera e execução do OpenMVG, que possui uma exigência menor de hardware. Por esses motivos, a primeira máquina se encaixa melhor no papel de teste com os resultados servindo de parâmetros para controle, demonstrando soluções alcançadas em máquinas menos robustas e custo pouco elevado (*mid-end*) com capacidades mais limitadas. Suas especificações estão na Tabela 02.

Tabela 02 – Informações e especificações da primeira máquina utilizada.

Modelo	ASUS DESKTOP-AP1524R
Processador	Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.20 GHz
Memória RAM	6,00 GB
Sistema Operacional	Linux Ubuntu Versão 22.04.1
Tipo de sistema	Sistema operacional de 64 bits, processador baseado em x64
Placa de vídeo	NVIDIA GeForce 930M
Memória Dedicada	2,00 GB

Fonte: De autoria própria.

Com a primeira máquina se tornando impotente visto a quantidade de dados a ser processada, uma nova máquina teve de ser selecionada para computar as novas remodelagens necessárias. O segundo computador foi escolhido, tendo sido emprestado pela orientadora, novamente a máquina é posse própria e uso pessoal, por ser mais atualizado e melhor estado,

seria mais adequado ao processamento bruto de imagens e dimensionamento virtual das mesmas. As especificações e parâmetros de sistemas foram listadas na Tabela 03.

Tabela 03 – Informações e especificações da segunda máquina utilizada.

Processador	Intel core i7-8700
Memória RAM	32 GB
Sistema Operacional	Linux Ubuntu Versão 22.04.1
Tipo de sistema	Linux e MacOS
Placa de vídeo	NVIDIA RTX 2070
Memória Dedicada	8 GB

Fonte: De autoria própria.

No entanto, é importante mencionar que apesar de a segunda máquina ter sido escolhida pelo seu poder de processamento superior, a primeira máquina continuou sendo utilizada principalmente como um critério de comparação, assim como tarefas mais simples, extração de frames e armazenamento, e também para a elaboração do documento do trabalho.

### 3.2.2. Das câmeras e dispositivos portáteis.

Inicialmente, a ideia do projeto que levou ao desenvolvimento desse trabalho girou em torno de ser direcionado ao uso em dispositivos móveis, isso dizendo respeito tanto ao uso quanto às técnicas que poderiam ser aproveitadas. No entanto, mesmo que o foco do trabalho tenha sido desviado da área de aplicativos móveis, o uso de câmeras para captura de imagens se manteve, levando em consideração o fácil acesso a esses dispositivos. Portanto, todas as capturas foram feitas utilizando dispositivos móveis do tipo smartphone com suas respectivas câmeras integradas.

Novamente, foram utilizados mais de um dispositivo, com o propósito de aumentar a diversidade de resultados. A escolha do segundo dispositivo se deu ao mesmo tempo em que o segundo computador foi selecionado, o motivo sendo novamente pela disponibilidade, a familiaridade com o aparelho e a proximidade com a segunda máquina. Sendo do tipo smartphone a gama de aplicativo para auxílio à câmera e à exposição de especificações seria vantajoso ao projeto.

As configurações dos dois dispositivos móveis utilizados, assim como os modelos de suas câmeras digitais embutidas estão, respectivamente, nas Tabelas 04 e 05. Contêm dados como especificações e sistema, além de informações de lente.

Tabela 04 – Informações do primeiro dispositivo móvel utilizado e sua câmera embutida.

Fabricante/Marca	Samsung
Modelo	SM-J600GT
CPU	Exynos7880
Velocidade	546 – 1598 MHz
Proporção	18.5 : 9
Memória RAM	2 GB
Modelo Câmera	s5k3l2
Resolução	12.9 MP (4144x3106)
Comprimento do foco	3.6 mm
Abertura	f/1.9

Fonte: De autoria própria.

Tabela 05 – Informações do segundo dispositivo móvel utilizado e sua câmera embutida.

Fabricante/Marca	Samsung
Modelo	Samsung S10e
CPU	Exynos 9820
Velocidade	ISO-250
Proporção	16:9 / 1:1 (vídeo)
Memória RAM	6 GB
Modelo Câmera	SM-G970F
Resolução	12.2 (4032x3024)
Comprimento do foco	4.3 mm
Abertura	f/2.4

Fonte: De autoria própria.

Como será explicada posteriormente, a especificação dos modelos das câmeras é de suma importância devido aos programas necessitarem de serem alimentados com suas informações, versões e pontos focais.

## 4. Estudo de Caso – Reconstrução tridimensional de interiores com testes em maquete

Este capítulo tem como objetivo explicar passo a passo a reconstrução 3D do cenário escolhido. Sendo detalhado o processo de instalação, destacando as principais dificuldades encontradas, e o uso das técnicas escolhidas, até a apresentação e explicação dos resultados.

### 4.1. Instalação

O sistema operacional Linux Ubuntu foi utilizado para ambas as tecnologias, portanto o processo de instalação apresentado neste trabalho corresponde ao sistema operacional utilizado, de forma que o processo demonstrado é composto basicamente por linhas de comando executadas em um terminal e pelos resultados esperados em cada etapa. No entanto, é importante ressaltar que ambas as tecnologias (InstantNGP e OpenMVG) possuem versões compatíveis para sistemas como Microsoft Windows ou MacOS. A escolha do sistema Linux foi justamente pela versatilidade no processo de instalação das tecnologias.

#### 4.1.1. Instalação: InstantNGP

Tendo sido utilizado o sistema operacional Linux, o processo de instalação a ser mostrado apresentará os comandos utilizados e os resultados esperados para cada um.

Pelo InstantNGP possuir um código amplo com uma rede neural integrada robusta, os requerimentos de sistema também são igualmente exigentes: obrigatoriamente é necessário uma placa de vídeo GPU NVIDIA modelo RTX 2070, ou uma placa de vídeo dedicada equivalente; um compilador de linguagem C++ versão 14 ou acima; o pacote para desenvolvimento CUDA, versão 10.2 ou maior, da NVIDIA para aplicações utilizando GPU; pacote de desenvolvimento CMake versão 3.21 ou maior; e como programas opcionais para diminuição de erros ou construções melhores: linguagem Python versão 3.7 ou maior; pacote OptiX 7.6 ou maior; e pacote Vulkan SDK para suporte DLSS.

Por segurança e conveniência, o primeiro comando será listado para que se assegurar que todos os pacotes básicos esperados em um sistema Linux estejam instalados, baseando-se na distribuição Debian, entre eles os necessários para utilização durante a execução do programa também serão instalados:

```
$ sudo apt-get install build-essential git python3-dev python3-pip libopenexr-dev libxi-dev \
libglfw3-dev libglew-dev libomp-dev libxinerama-dev libxcursor-dev
```

Para obter o repositório é necessário que o pacote Git esteja devidamente instalado (através do comando anterior) e que o comando abaixo seja realizado em um terminal do sistema.

```
$ git clone --recursive https://github.com/nvmlabs/instant-ngp
$ cd instant-ngp
```

É importante que os caminhos dos dois pacotes, CUDA e OptiX, estejam inseridas nas diretivas de variáveis do sistema, sendo altamente recomendado para o caso do método não conseguir localizar prontamente os pacotes mencionadas. Na distribuição Ubuntu do sistema Linux, PATH, é possível se fazer isso com o comando abaixo do Linux, mas variam entre versões e entre sistemas.

```
$ export PATH="/usr/local/cuda-11.4/bin:$PATH"
$ export LD_LIBRARY_PATH="/usr/local/cuda-11.4/lib64:$LD_LIBRARY_PATH"
```

O processo neste ponto se diferencia de certo modo àqueles que utilizam o sistema Windows, pois no caso do Linux é necessário “construir” a instância do processo utilizando o programa CMake, portanto uma instância do projeto será criada e estabelecida na máquina que for instalada, por isso é necessário utilizar os seguintes comandos dentro da pasta que for clonada da página do GitHub do InstantNGP (MULLER, 2022):

```
instant-ngp$ cmake . -B build -DCMAKE_BUILD_TYPE=RelWithDebInfo
instant-ngp$ cmake --build build --config RelWithDebInfo -j
```

Com isso, e levando em conta que não houve erros durante o processo de instalação, o programa está pronto para uso, e aguardando os conjuntos de imagens para o processo de reconstrução. Caso o processo de compilação CMake falhe ou mesmo demore uma quantidade excessiva de tempo para se concluir, é bastante provável que o processo esteja esgotando a memória, e portanto sendo cancelada ou mesmo à espera de mais memória

disponível, permanecendo sempre em espera. Este problema foi encontrado ao utilizar o primeiro computador para testes, sendo posteriormente substituído.

Caso o processo foi bem sucedido e o programa foi instalado, é possível executá-lo com o comando “./instant-ngp” ou encontrar o arquivo run.py dentro da pasta “scripts” dentro da página principal do programa InstantNGP.

#### 4.1.2. Instalação: OpenMVG

O OpenMVG é uma tecnologia mais simples que a primeira escolhida já que não utiliza uma rede neural. Portanto, ela exige ainda menos do processamento do hardware do que o InstantNGP, nem necessita de uma placa de vídeo, é mais leve, e possui menos requerimentos. Por essa razão, apesar de ter sido utilizado o segundo computador mencionado como resultado final, o primeiro computador também foi utilizado visto que era capaz de executar o OpenMVG, apesar dos resultados servirem mais como critérios de ilustração e comparação para máquinas menos avançadas.

Como programas obrigatórios: CMake versão 3.2 ou mais atual; Git para clonagem do projeto; e um compilador de linguagem C++11. Como opcionais para um framework mais suave entre plataformas, caso aja necessidade, o QT versão 5.4 ou maior. Para visualização dos resultados tanto o Meshlab quanto o CloudCompare são os recomendados.

Novamente, os comandos que serão listados são focados no sistema Linux Ubuntu 22.04.1, portanto o uso do terminal do sistema é o local para inserção dos mesmos. Começando com a clonagem do projeto e seus módulos, ao utilizar o Git, há duas opções:

```
$ git clone --recursive https://github.com/openMVG/openMVG.git
```

```
$ git clone https://github.com/openMVG/openMVG.git
```

```
$ cd openMVG
```

```
$ git submodule init
```

```
$ git submodule update
```

Para ter certeza que as bibliotecas básicas do Linux estão instaladas, antes de se iniciar a compilação principal, é ideal utilizar o seguinte comando:



```
$ sudo apt-get install libpng-dev libjpeg-dev libtiff-dev libxxf86vm1 libxxf86vm-dev libxi-dev libxrandr-dev
```

Após a clonagem, criar uma nova pasta, nomeá-la e acessá-la com o comando:

```
$ mkdir openMVG_Build && cd openMVG_Build
```

Com isso, basta somente utilizar os dois comandos de configuração e instalação utilizando o programa CMake para ser finalizada a compilação:

```
$ cmake -DCMAKE_BUILD_TYPE=RELEASE ../openMVG/src/  
$ cmake --build . --target install
```

Caso não tenha sido detectado nenhum erro, e a instalação foi bem sucedida, ainda na página de compilação existem dois comandos simples a serem usados para conferência e teste do programa, no caso de sucesso uma mensagem será mostrada confirmando o processo. Importante mencionar que o OpenMVG, utilizado neste trabalho, não possui interface, portanto o uso se deve somente por linhas de texto no terminal. Os comandos são:

```
$ make test  
$ ctest --output-on-failure -j
```

## 4.2. Aplicação

Com os programas devidamente instalados, sem erros e iniciando apropriadamente, a próxima etapa é a reconstrução 3D do cenário escolhido. Os passos a seguir especificam cada uma das tecnologias e suas particularidades, mas o conceito principal é essencialmente o mesmo: reunir um conjunto de imagens de um determinado ambiente ou objeto a partir de diversos ângulos, fornecê-las ao programa e obter a reconstrução 3D. Cada programa apresenta uma forma diferente de representação.

#### 4.2.1. InstantNGP

No processo de iniciar o treinamento da rede neural do InstantNGP, é importante que se tenha um certo nível de restrição no processo de selecionar as imagens a serem utilizadas. Tendo em vista que a qualidade influencia diretamente no procedimento e velocidade de treinamento da rede, as imagens precisam estar nítidas, sem distorção, focadas e com a melhor qualidade possível, tornando o processo de convergência mais eficiente. Nesse contexto, idealmente, as imagens necessitam ser precisas, cobrir o maior número de pontos cegos do ambiente, e com mínimo (preferencialmente nenhum) desfoque de movimento.

Há duas formas de fornecer as imagens necessárias: as fotos do ambiente a ser capturadas individualmente, levando em conta os pontos citados, ou a extração de frames a partir de um vídeo na melhor qualidade disponível. Durante o trabalho, foi escolhido o segundo modo. Para isso, foi utilizado o programa COLMAP (SCHONBERGER et al, 2016), o qual é capaz de extrair diversos frames de um vídeo com qualidade e na taxa escolhida à demanda pelo usuário. Independente da escolha, os arquivos a serem inseridos no programa se resumem a um conjunto de imagens que precisam ser escolhidas manualmente, visto a necessidade de se selecionar o melhor conjunto de imagens de entradas logo no início no processo, a fim de que não ocorram erros posteriores no processo de remontar os pontos de convergência extraídos.

Para utilização do COLMAP, além do mesmo estar instalado no sistema, é necessário possuir a biblioteca de vídeo FFmpeg (hospedado no site <https://ffmpeg.org/>) e a linguagem Python (hospedado no site <https://www.python.org/>) instalado no sistema Linux, e ainda ambas adicionadas ao PATH do sistema. Além disso, é necessário ter certeza que o script “colmap2nerf.py” também esteja dentro da pasta “/script” localizada na pastas de instalação do InstantNGP. Com isso, após a seleção do script a extração dos frames do vídeo escolhido se torna possível, é preferível que o número de imagens gire em torno de 50 a 150 unidades, com elevada qualidade para um treinamento ideal. Para extração utilizando um vídeo:

```
$python [pasta_de_instalação_do_InstantNGP]/scripts/colmap2nerf.py --video_in  
<nome_do_video> --video_fps 2 --run_colmap --aabb_scale 32
```

Do comando mostrado, o parâmetro mais importante é “—aabb\_scale” que por padrão é selecionado como 1. A função desse parâmetro gira em torno do tamanho na cena que o

usuário deseja, ou seja, quanto maior o número passado ao parâmetro “aabb” mais distante do ponto central a cena se tornará, causando dois principais fatores: a maior quantidade de informações da cena serão passadas à rede, assim como exigirá mais processamento do sistema. Esse foi o fato determinante que resultou na seleção de uma máquina nova para testes, sendo que a primeira máquina nem mesmo passando o valor “1” ao parâmetro “aabb\_scale” foi capaz de fazê-lo funcionar, visto a quantidade de memória necessária para essa distância. O comando “--video\_fps 2” é o que faz com que a extração gere o número ideal necessário para um bom treinamento.

O script, além da extração das imagens, emprega ainda a opção de iniciar o treinamento da rede. Isso é possível visto que o COLMAP tem também a função de encontrar pontos em comum (*matcher*) entre as imagens utilizadas, isso serve como um pré-treinamento da rede. Portanto, após a utilização do script, a pasta resultante (que por padrão fica localizada na pasta de usuário), no caso de executar sem problemas, conterá uma subpasta com os frames extraídos e uma subpasta com a relação de pontos em comum identificados em arquivo, que por padrão é nomeado “transformers.json”. Esse arquivo é a base para o treinamento da rede.

Com o sucesso da aplicação do script, e, por conseguinte extração das imagens e seus pontos combinados, o próximo processo do InstantNGP é repassar como entrada para a rede o arquivo .json das características extraídas das imagens. Isso pode ser feito de duas formas. A primeira é feita ao se acessar no terminal a pasta onde está instalado o programa, após isso executar o comando na janela de comandos do Linux listada anteriormente (“./instant-ngp”). Isso resultará em uma janela sendo aberta, o usuário portanto poderá selecionar o arquivo .json e arrastar para a nova janela do programa, isso iniciará o processo de treinamento e ao mesmo tempo que inicia a remodelagem da cena com os frames e os pontos encontrados. O segundo método é executando as mesmas etapas da primeira forma, mas por uma única linha de código listada a seguir, possível após deslocar a seleção do terminal à pasta do programa e passando a localização do arquivo .json ao processo:

```
$ ./instant-ngp “caminho_ao_arquivo_json”
```

Dependendo da quantidade de detalhes a serem repassados e a capacidade do hardware, o processo não deverá demorar mais de que alguns minutos para a cena ser remontada, que é uma das maiores vantagens em relação ao NeRF. O processo de remodelagem pode ser acompanhado durante o treinamento, por um gráfico à direita do

programa, que ilustra a quantidade de dados a serem combinados. Analisando esta linha, quanto mais baixa ela estiver em relação ao gráfico, menor é a chance de haver novas combinações, portanto se estiver muito baixo, pode-se dar o treinamento como concluído. A Figura 19 ilustra o processo descrito acima, na direita da janela é possível se encontrar a linha “loss graph” destacada, indicando a perda de eficiência do processo, caracterizado pelo poder de generalização da reconstrução 3D do InstantNGP.

Figura 19 – Demonstração da aplicação e treinamento com o gráfico em estágio já estável.



Fonte: Página do InstantNGP no GitHub.

A partir desse ponto, o usuário pode utilizar o programa como bem entender: mexer o cenário, modificar ângulos, montar um vídeo, ou uma imagem animada através de pontos definidos pelo usuário. Em suma, o treinamento estará concluído.

#### 4.2.2. OpenMVG

O OpenMVG, diferente da outra tecnologia e seguindo a intenção do desenvolvedor, não envolve procedimentos muito árduos a fim de executá-lo, no entanto, algumas etapas de pré-inicialização devem ser atendidas. Isso se deve à necessidade do programa ter um ponto focal pré-estabelecido e fornecido para começar a se fazer as combinações entre as imagens. Outro ponto importante é um *dataset* de frames anteriormente a disposição, não necessitando de uma quantidade elevada, mas imagens de boa qualidade e resolução.

Desta forma, no desenvolver desse trabalho em ambas as tecnologias foram utilizados os mesmos conjuntos de imagens das maquetes de teste. Razão para isso foi que levando em conta que em ambos os casos, os mesmos requisitos para seleção de imagens se aplicavam: imagens limpas, sem imperfeições, com mínimo de desfoque de movimento possível e resolução a melhor possível. No caso do OpenMVG, mais um ponto a ser levado em conta foi necessário, o comprimento focal (*focal length*) da câmera utilizado, no caso das imagens adquiridas pelos celulares.

Essa informação deve ser adicionada previamente em um arquivo específico, após a instalação do programa, juntamente com a marca ou modelo da câmera. Isso serve como base de dados de diversos modelos pré-adicionados pelo desenvolver e serve como um repositório. Durante a execução do programa, o mesmo compara as informações de cada imagem com o armazenado neste arquivo de texto, isso para que se mantenha um padrão de comprimento focal durante a execução do método de triangulação utilizando vários pontos de vista entre as várias imagens. Essa informação auxilia no processo de se calcular melhor, aproximadamente, a distância entre os objetos e lentes do sensor da câmera.

Para o programa ter acesso a essa informação existem duas formas. A primeira forma seria passando o dado do comprimento focal transformado em pixel ao iniciar o processo de correspondência do programa principal. Isso é feito ao adicionar o comando à linha de inicialização do pipeline “-f X”, sendo o X representando os pixels máximos em uma imagem, definido pela equação  $X = 1,2 * (\text{resolução máxima de imagem da câmera})$ . Por exemplo, se considerarmos uma imagem com resolução 1920 x 1080, o máximo de largura ou altura em uma imagem seria o valor 1920, portanto utilizando a equação o máximo de pixel em qualquer direção em uma imagem nesta resolução seria 2304 pixels. Pelo comando de inicialização, é esperado que o programa automaticamente detecte o ponto focal correspondente. O valor 1,2 se deve pela razão de escalamento definido pelo campo de visão em câmeras, sendo uma constante. Na Figura 20 há a demonstração do que corresponde o ponto focal em uma câmera.

Figura 20 – Exemplo de ponto focal em uma câmera.



Fonte: <https://redproducao.com/distancia-focal/>.

A segunda forma seria acrescentando manualmente ao arquivo de base que funciona como um banco de dados, contendo um vasto conjunto de informações de fabricantes e fornecedores de câmeras diversas, não sendo universal, mas incrementada ao longo dos anos pela comunidade. Mesmo se tornando obrigatório se ter um prévio conhecimento do equipamento a ser utilizado, em casos em que seriam feitos vários testes, o programa já encontraria e faria todas as conversões anteriormente descritas, por esse motivo foi a alternativa utilizada neste trabalho.

Para se adicionar ao arquivo, basta localizá-lo, abrir o arquivo e adicionar uma linha nova ao corpo do texto com a marca ou modelo da câmera e o comprimento focal, seguindo o modelo: “Marca/Modelo; comprimento focal”. O arquivo é por padrão nomeado como “sensor\_width\_camera\_database.txt”, e é encontrado acessando a pasta do OpenMVG através do caminho “openMVG/src/openMVG/exif/sensor\_width\_database/”. Ao salvar o arquivo e permitindo o programa acessá-lo, o mesmo somente precisaria de nova alteração caso a câmera de teste fosse trocada.

Com essas etapas definidas, basta escolher entre dois pipelines disponíveis: Sequencial e Incremental (*Sequential & Incremental SfM pipeline*) ou o Global (*Global SfM pipeline*). Os dois basicamente entregam o mesmo resultado, diferenciando somente no processo a ser empregado e o foco de testes de cada um. O primeiro, o mais ideal para *dataset* menores, com menores quantidades de imagens, e o segundo focado em maiores quantidades com maior fluxo de dados. O primeiro foi o escolhido por ser mais adequado ao ambiente de teste da maquete.

Apesar da escolha, os comandos para execução são praticamente os mesmos, mudando somente o pipeline escolhido no código de inicialização, sendo seguido pelo local diretório onde as imagens do *dataset* estão salvas, e depois pelo diretório resultante onde o programa

salvará os arquivos de correspondência. O comando inicial deve ser executado referenciando onde estão localizados os arquivos dos pipelines a serem escolhidos. Por padrão (se ignorando os colchetes na segunda linha), os comandos são os que se seguem:

```
$ cd openMVG_Build/software/SfM/
```

```
$ python SfM_SequentialPipeline.py [diretório onde estão as imagens] [diretório final]
```

Como resultado, no diretório final, possuirá somente os arquivos importantes. O primeiro deles é, por padrão, nomeado “cloud\_and\_poses.ply” que possui os pontos resultantes da correspondência (*matching*) dos pontos em comum entre as imagens. O segundo arquivo, nomeado “colorized.ply” corresponde à estrutura tridimensional com pontos de vista coloridos e ambiente reestruturado. Ambos os arquivos podem ser visualizados pelos arquivos Meshlab, ou Cloudcompare, com as trajetórias de câmera mostradas com pontos verdes. O arquivo resultante de todo o processo é salvo com o nome “sfm\_data.bin”. O mesmo pode ser convertido para o formato JSON/XML para visualização.

Os passos do OpenMVG para a reconstrução de um modelo 3D à partir de imagens seguem:

1. Definição de um conjunto de imagens
2. Extração de características
3. Feature matching
4. Estrutura de movimento (SfM)
5. Reconstrução de nuvem de pontos densa
6. Reconstrução da malha
7. Refinamento da malha
8. Texturização da malha

Para o passo 1, é necessário garantir que as imagens escolhidas possuam alta qualidade e se sobreponham, bem como, incluir informações do tamanho (largura e altura) da imagem, câmera, tempo de exposição e distância focal. O passo 2 extrai do conjunto de imagens sobrepostas, características específicas, tais como: linhas, bordas, arestas, cantos, segmentos. O próximo passo, passo 3, é a identificação dos recursos correspondentes nas várias imagens, de forma a determinar a posição relativa das câmeras. O passo 4 é necessário para estimar as poses da câmera e a estrutura 3D de todo o ambiente, informações presentes nas imagens de entrada. A reconstrução de nuvem de pontos densa, passo 5, busca dar

profundidade nos pontos que compõem a nuvem densa, através de uma estimativa de profundidade calculadas por mapas de profundidades. A reconstrução da malha, passo 6, utiliza algoritmos de reconstrução de superfície para suavizar e interpolar pontos da nuvem e gerar superfícies que representam formas dos objetos do cenário. O passo 7 otimiza a estrutura da malha, obtida pela triangularização dos pontos da nuvem densa, com objetivo melhorar a qualidade e precisão da malha 3D. Por fim, o passo 8, aplica texturização da malha no modelo 3D final, adicionando propriedades dos materiais e iluminação.

#### 4.3. Resultados

Os resultados obtidos do estudo de caso de maquetes simulando ambientes reais foram alcançados utilizando os computadores e as câmeras antes descritas, assim como os parâmetros estabelecidos e documentados. Vale ressaltar novamente que os resultados do InstantNGP somente conseguiram ser alcançados na segunda máquina escolhida, principalmente devido a limitação de hardware na máquina 01.

Um aspecto que vale mencionar novamente, é que este estudo de caso visa cumprir o objetivo de conseguir obter um modelo 3D de um ambiente real. Esse objetivo foi alcançado em partes, pois apesar de que, de fato, o ambiente foi com sucesso virtualizado, algumas características do ambiente foram perdidas. Contudo, com uma análise, um planejamento maior e, principalmente, um maquinário mais avançado (envolvendo a etapa de processamento, extração de imagens, virtualização), o objetivo final pode em vias de realidade ser alcançado, que é modelar um ambiente real em um virtual compatível.

Tendo em mente isso, alguns dos resultados, mesmo após vários testes e tentativas de melhorias, foram os mais promissores. Um dos principais motivos para isso se deve por ambas as tecnologias não serem focadas em serem usadas para ambientes internos. O trabalho visa demonstrar que ambas as tecnologias podem ser sim usadas para tal fim, no entanto, como foi explicado, os resultados são limitados, encontrando dificuldades em diferenciar interior do exterior, quanto ao ambiente deve ser adaptado ou condizer com os requisitos de uso durante a etapa de captura de imagens.



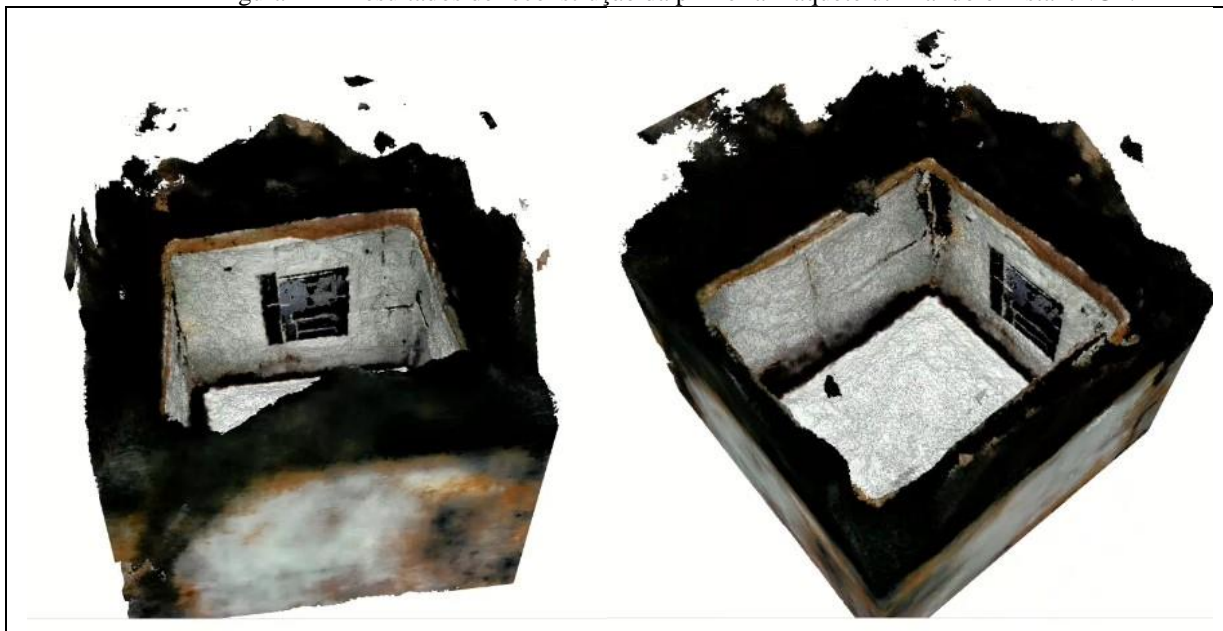
#### 4.3.1. InstantNGP

Assim como já mencionado, os resultados para as duas maquetes utilizando o InstantNGP se resumiram a uma única máquina de teste. A máquina 02 foi utilizada por sua capacidade maior de lidar com o fluxo de dados envolvendo os detalhes do cenário juntamente com os dados vinculados ao sistema de rede neural do InstantNGP.

Os resultados da primeira maquete são ilustrados nas imagens das Figuras 21 e 22. De todas as variações de luminosidade e ângulo de captura que foram feitas para que a primeira maquete se adequasse corretamente às exigências do programa, a reconstrução apresentada foram as que obtiveram os melhores resultados. No entanto, como já enfatizado, aparentemente os resultados do InstantNGP são muito dependentes da qualidade em que se encontra o ambiente a ser testado.

O poder da rede neural do InstantNGP é evidente, ao se perceber que até os arredores da maquete foram captados e reconstruídos. Mesmo que as imagens tenham sido captadas num ambiente escuro, o mesmo foi tanto reconstruído quanto a parte escura foi atribuída como sólido, como apresentando na Figura 21 e 22.

Figura 21 – Resultados de reconstrução da primeira maquete utilizando o InstantNGP.



Fonte: De autoria própria.

Figura 22 – Resultados de reconstrução da primeira maquete utilizando o InstantNGP.

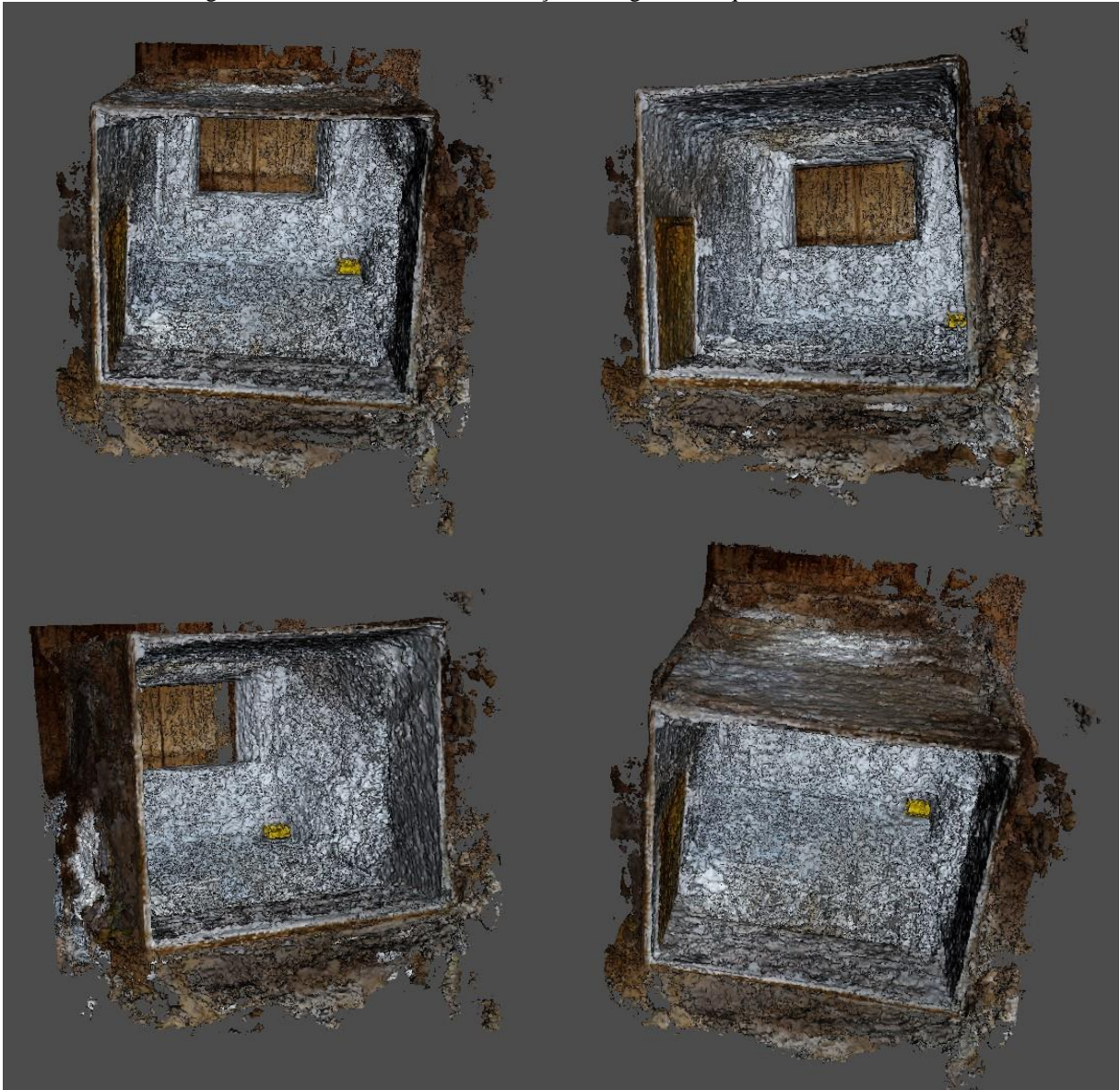


Fonte: De autoria própria.

A maquete 01 simula um quarto com vista isométrica e captada como se tivesse sendo vista pelo teto. O ambiente precisou ser iluminado por luz artificial da própria câmera do smartphone, provavelmente por isso os raios de luminosidade direcionados ao solo e parede em volta possam ter sido captados pela câmera e transmitidos ao InstantNGP, com essa informação o programa interpretou como se o “escuro” em volta fosse um objeto sólido. Esse fato justifica a reconstrução em volta da maquete, marcado com a cor preta e a parede externa da maquete ter ficado como a parede externa da parte sólida, o erro que resultou na construção da segunda maquete.

Na Figura 23, são demonstrados os pontos de importância como a janela, o quadrado amarelo na ponta direita, a abertura ao fundo simulando uma janela, podem ser identificados, no entanto, assim como a primeira maquete, o entorno do sólido não só foi reconstruído como redimensionado como fazendo parte da maquete, além alguns ponto expandidos.

Figura 23 – Resultados de reconstrução da segunda maquete utilizando o InstantNGP.



Fonte: De autoria própria.

A segunda maquete com sua construção pelo InstantNGP pode ser visualizada na Figura 23. Novamente o poder do programa pode ser reconhecido, apesar de várias características do ambiente terem sido perdidas. A maquete 02 pode ser identificada, com seus pontos de identificação em evidência, no entanto, em parte provavelmente pela iluminação, alguns lados do sólido não só foram distorcidos como em alguns pontos ignorados.

Um dos fatos de importância do InstantNGP é que apesar de ser uma tecnologia que exige muito do seu hardware, o resultado consegue demonstrar o poderio que possui em sua reconstrução, isso é, quando seu padrão de excelência é alcançado, atrelado a seus padrões de entrada os melhores possíveis.

O modelo 3D reconstruído, como visto na Figura 23, possui 1.901.302 vértices e é composto por uma malha de 3.716.486 triângulos.

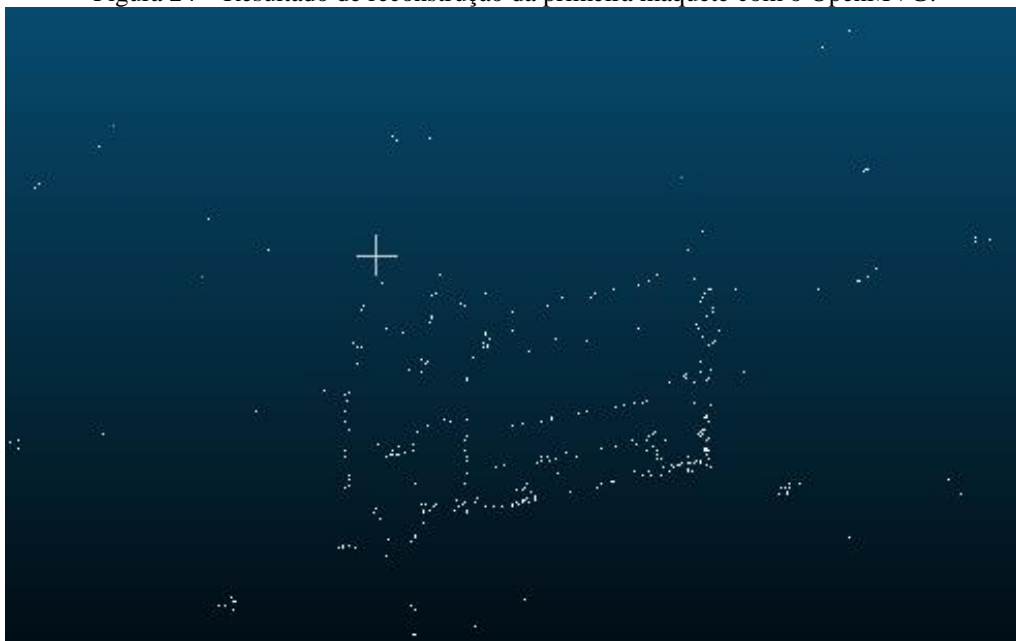
#### 4.3.2. OpenMVG

Os resultados para o OpenMVG puderam ser verificados em ambas as máquinas de teste, isso se deve principalmente ao fato do OpenMVG ser projetado para operar na CPU, otimizando os recursos do processador para a reconstrução 3D, muito embora, o uso da GPU beneficia algumas tarefas específicas as funcionalidade do OpenMVG. Conseqüentemente, pela tecnologia ser bem menos exigente de seu hardware, se torna bem mais acessível.

A máquina de teste 01 pôde ser utilizada para testes, apesar de não conseguir chegar ao mesmo nível de reconstrução, ela foi capaz de extrair os pontos de correspondência das maquetes, que é a base para reconstrução virtual, também conhecida como a malha virtual de pontos. As imagens das figuras 24, 25, 26, 27, 28 e 29 ilustram os resultados alcançados tanto na primeira como na segunda maquete.

Nas Figuras 24, 25 e 26, é apresentado os resultados de reconstrução da primeira maquete utilizando o OpenMVG pela primeira máquina de teste. Apesar de não conseguir reconstruir a maquete, a máquina 01 conseguiu averiguar, determinar e reestruturar os pontos de correspondência da maquete 01 na chamada malha de pontos, que é a base para a reconstrução tridimensional.

Figura 24 – Resultado de reconstrução da primeira maquete com o OpenMVG.



Fonte: De autoria própria.

Figura 25 – Resultado de reconstrução da primeira maquete com o OpenMVG.



Fonte: De autoria própria.

Figura 26 – Resultado de reconstrução da primeira maquete com o OpenMVG.



Fonte: De autoria própria.

As imagens das Figuras 27, 28 e 29 demonstram os resultados encontrados pela máquina 01 para a segunda maquete. Novamente, o processo de reconstrução não foi alcançado, mas os pontos foram detectados e a malha da segunda maquete foi reconstruída de forma simples.

Os resultados da reconstrução da segunda maquete utilizando o OpenMVG pela primeira máquina de teste são apresentados nas Figuras 27, 28 e 29, novamente demonstrando a geração da malha virtual de ponto para posterior reconstrução.

Figura 27 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01.



Fonte: De autoria própria.

Figura 28 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01.



Fonte: De autoria própria.

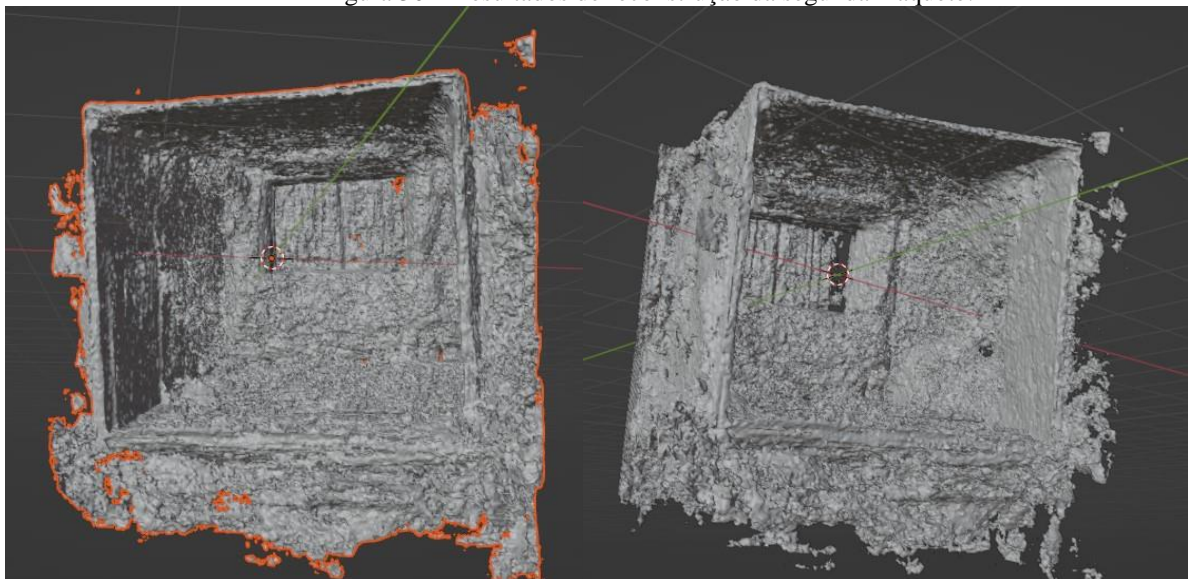
Figura 29 – Reconstrução da maquete 02 com o OpenMVG na máquina de teste 01.



Fonte: De autoria própria.

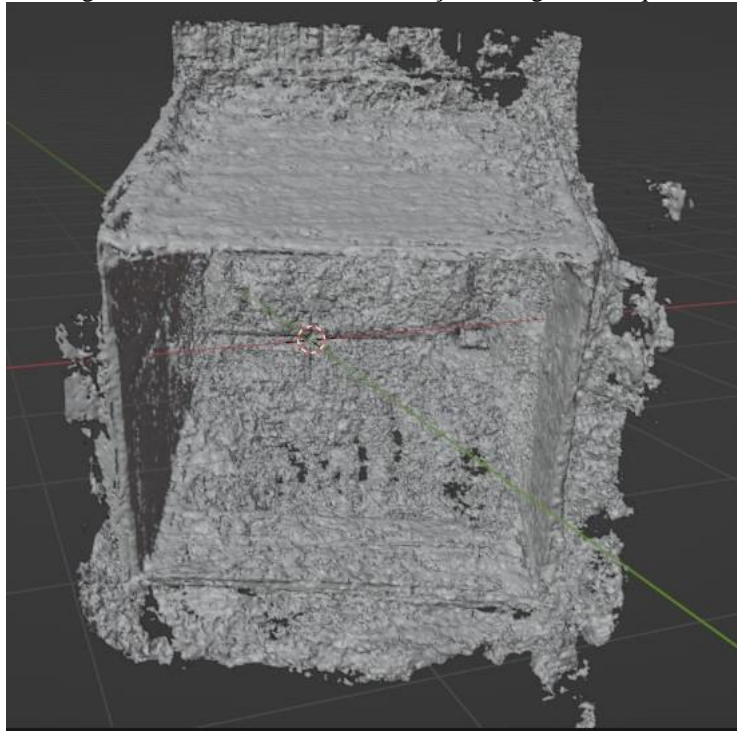
A seguir, as próximas imagens (Figura 30 e 31) ilustram os resultados encontrados utilizando as mesmas imagens que foram utilizadas pela máquina 01, no entanto dessa vez pela segunda máquina (02). Desta vez se observa a reconstrução completa da maquete 02, que apesar de não apresentar os padrões de coloração ainda foi capaz de reconstruir as bases da segunda maquete com suas características básicas.

Figura 30 – Resultados de reconstrução da segunda maquete.



Fonte: De autoria própria.

Figura 31 – Resultado de reconstrução da segunda maquete.



Fonte: De autoria própria.

Como se observa, e levando em conta principalmente os resultados da máquina 02, os resultados obtidos dependem proporcionalmente à capacidade hardware da máquina a ser feita os testes. A máquina 02, por ser mais atual e ser mais poderosa em processamento conseguiu formar o sólido da estrutura externa da maquete.

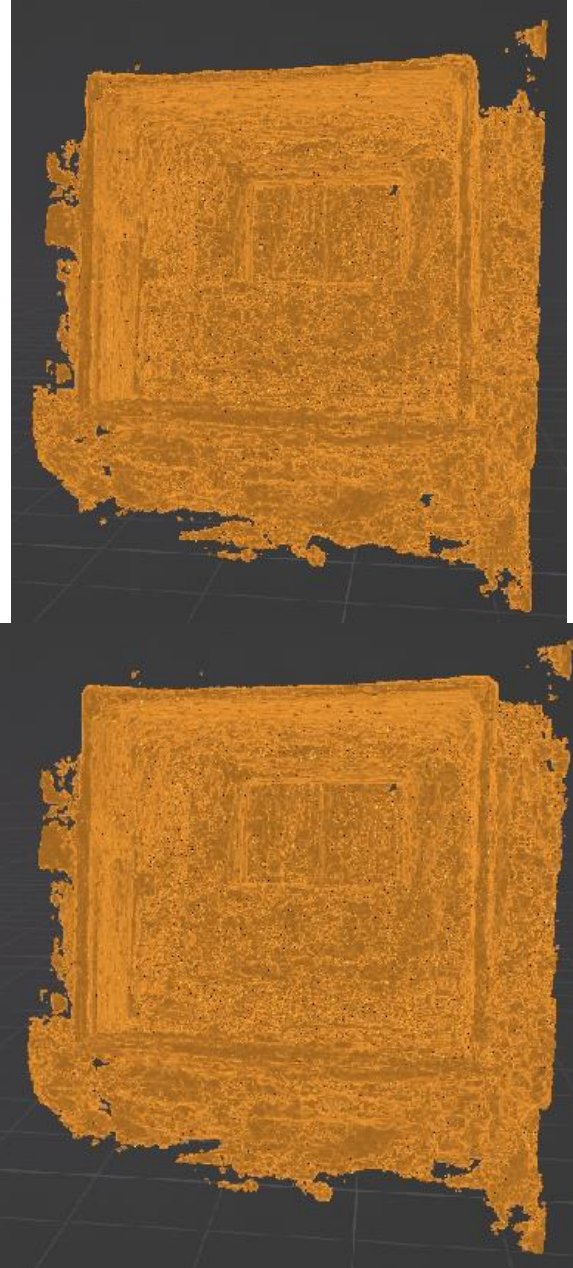
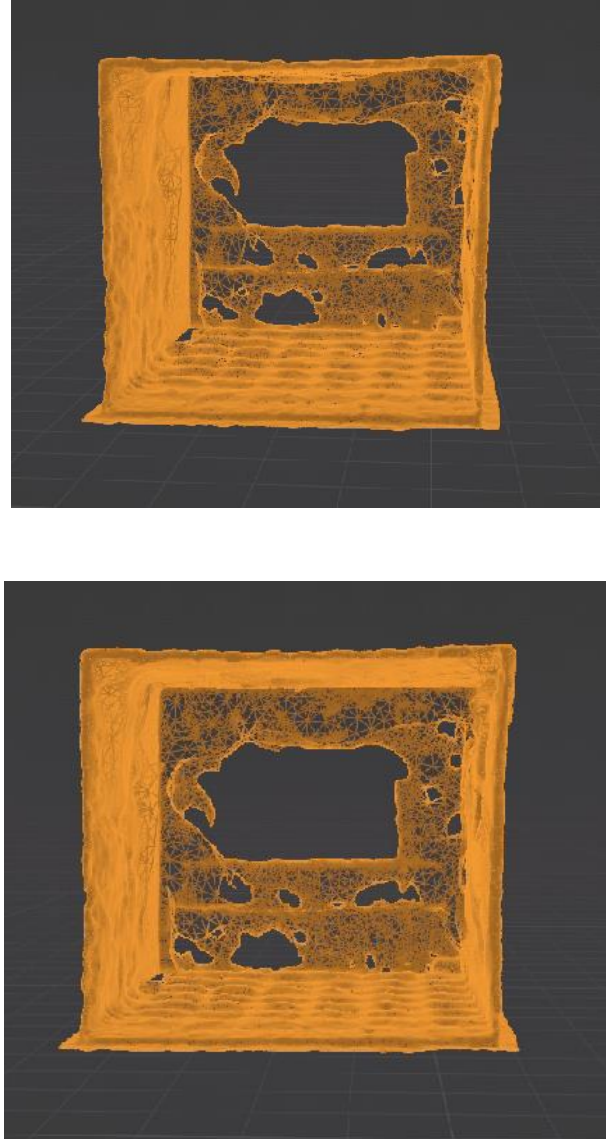
O modelo 3D reconstruído, como visto na Figura 31, possui 781.262 vértices e é composto por uma malha de 1.561.454 triângulos.

#### 4.3.3. Tabelas Comparativas

A seguir, na Tabela 06, será feita a apresentação lado a lado dos resultados obtidos nas modelagens 3D da maquete 02. Os resultados são divididos entre as duas tecnologias, InstantNGP e OpenMVG, para que sirvam de análise comparativa. Os resultados entregues pelo InstantNGP levam em conta a imagem reconstruída em forma completa, sem nenhuma limitação em sua forma. Quanto ao OpenMVG, seus resultados estão apresentados em sua forma de malha reconstruída em imagem bruta e com um filtro de imagem que libera a malha de imperfeições ao seu redor.



Tabela 06 – Tabela para comparação dos resultados lado a lado.

Resultados InstantNGP	Resultados OpenMVG
	

Fonte: De autoria própria.

Tendo como os resultados visuais expostos, serão apresentados os valores e informações numéricas alcançadas. Alguns desses dados foram registrados para que sejam documentados os requisitos mínimos necessários para se replicar o experimento, visto que nem o InstantNGP nem OpenMVG possuem uma avaliação nesse quesito. Posteriormente, os valores de tempo, medição de qualidade, grau de aprendizagem prévio e as medidas do modelo gerado tem o propósito de servir como uma base de comparação para novos

experimentos utilizando as mesmas técnicas, ou mesmo comparação com técnicas diferentes já existentes ou ainda que devam de ser criadas no futuro.

A Tabela 07 apresenta os resultados alcançados para comparação, os dados são referentes à modelagem da maquete 02 para as duas tecnologias e suas respectivas especificações para execução.

Tabela 07 – Tabela comparativa de informações e resultados alcançados pelo InstantNGP e OpenMVG.

	InstantNGP	OpenMVG
CPU	Intel core i7-8700	Intel core i5-5200
GPU	NVIDIA RTX 2070	NVIDIA GeForce 930M
Memória	16GB (8GB dedicada)	4GB (2GB dedicada)
Número de imagens recomendadas	Entre 50 e 150 imagens	Entre 20 e 40 imagens
Quantidade de imagens de entrada	54 imagens	18 imagens
Tempo geração de malha	Mínimo de 15 minutos (depende da quantidade de imagens)	Entre 5 a 10 minutos (podendo aumentar devido ao número de imagens)
Treinamento da rede / Reconstrução da imagem	Mais de 15 minutos	Mínimo de 5 minutos
Grau de conhecimento prévio	Necessita de conhecimento em ambiente de instalação e análise de imagens	Conhecimento de análise de imagens
Dificuldade de instalação	Alta, as bibliotecas devem ser apropriadamente instaladas, necessitando verificação de instalação	Baixa, necessitando de poucos comandos para as poucas bibliotecas necessárias
Necessidade do treinamento de rede	Sim, com tempo variando pela quantidade de imagens	Não
Quantidades de triângulos (para a malha)	3.716.486 triângulos	1.561.454 triângulos
Vértices	1.901.302 vértices	781.262 vértices
Largura e Altura (modelo final)	23.55 x 22.22	24.5 x 24.0

Fonte: De autoria própria.

Quanto ao tempo de modelagem, deve ser levada em conta a diferença nas arquiteturas empregadas nas duas tecnologias avaliadas. No caso do InstantNGP, o tempo deve ser dividido entre, treinar a rede com o registro das informações das imagens, e a remodelagem

das regiões de pontos em comum para montagem 3D. Do mesmo jeito, o tempo utilizado pelo OpenMVG também se divide na análise de imagens para geração de malha e a reconstrução do modelo 3D. Deve ser levado em conta também que, a quantidade de imagens influencia diretamente no processo de associação de pontos, portanto quanto maior o número de imagens maior o tempo total. A quantidade de imagens na Tabela 07 se refere às que foram usadas para a reconstrução da maquete 02 nas duas tecnologias, importante registrar que o número de imagens de entrada foi reduzido levando em conta o tamanho do ambiente analisado e diminuindo sobrecarregar o sistema.

Com o registro dos dados alcançados, é possível afirmar que, apesar de ambas as técnicas terem seu foco na geração de sólidos 3D, o resultado para criação de modelos baseados em ambientes e no mapeamento do interior desses ambientes foi bem sucedido. Os resultados ainda demonstram que as técnicas, ao mesmo tempo em que executam suas funções do modo que foram primariamente desenvolvidas, diferem tanto no resultado do modelo 3D quanto no tempo necessário para reconstrução.

O OpenMVG requer menor poder de processamento e menor número de imagens para gerar seu modelo 3D, no entanto, a malha resultante do modelo 3D falhou em capturar diversos pontos gerando diversos buracos na malha. O InstantNGP, como esperado, tem uma capacidade bem melhor na geração da malha do modelo 3D, mesmo que exija na sua execução mais hardware e uma quantidade superior de imagens de entrada. Sua geração de modelo virtual possui mais detalhes, vértices e triângulos, o tornando mais adequado para se obter uma melhor qualidade na reconstrução do modelo 3D. Essencialmente, o InstantNGP requer o uso de GPU para sintetizar cenas 3D a partir de imagens 2D, bem como um robusto conjunto de dados de imagens. Contudo, apesar da GPU ser indispensável, o conjunto de imagens pode ser redimensionado, desde que se tenha o mínimo (recomendado de 50 imagens) de imagens e estas contenham uma boa cobertura do cenário a ser reconstruído. O OpenMVG é uma opção valiosa em visão computacional para projetos de reconstrução 3D a partir de imagens 2D, e como mencionado anteriormente, foi projetado para operar principalmente na CPU para realizar tarefas como correspondência de pontos-chave, estimativa de geometria fundamental, fusão e otimização de estruturas 3D. Apesar de bem consolidado na área de reconstrução, para os modelos 3D resultados gerados neste trabalho, tanto na qualidade visual dos resultados obtidos, quanto na precisão da geometria, foram inferiores quando comparados ao InstantNGP. Levando em consideração todas essas informações, caso o usuário possua em seu poder uma máquina robusta capaz de suportar o

funcionamento do InstantNGP e um conjunto de dados de imagens que abrange todo o cenário a ser reconstruído, recomenda-se o uso do InstantNGP para sintetizar cenas 3D a partir de imagens 2D, mesmo para reconstrução de modelos 3D caracterizados pelo mapeamento do interior de um objeto, caso contrário, a necessidade de hardware menor e a geração mais simplificada de malha torna o OpenMVG uma escolha mais eficiente em seu objetivo.

## 5. Considerações Finais

### 5.1. Conclusões

A computação gráfica, com suas divisões, se apresenta como uma área em constante evolução, mas que ainda se encontra somente em seu início, levando em conta todo seu potencial ainda inexplorado. Isso se torna verdadeiro ao se levar em consideração que a área está intimamente vinculada aos seus limites tecnológicos de hardware, em que os limites que possuímos hoje podem não existir em um surpreendente curto espaço de tempo.

É possível perceber essa afirmação, ao levar em conta que a maior quantidade dos incentivos de pesquisa na área de computação se localiza temporalmente nas últimas décadas. Tendo em vista que as tecnologias na área de computação existentes possuem uma taxa de evolução assustadoramente elevada se tomar o tempo que a humanidade existe. Colocando em consideração, ainda, as tecnologias de computação gráfica e inteligência artificial, o grau de crescimento é ainda maior.

Ao combinar as áreas da Computação Gráfica, Visão Computacional, Aprendizado de Máquina, Reconstrução Virtual, os resultados tendem a ser imensuráveis. Baseado na pesquisa realizada neste trabalho, as mais recentes e interessantes estratégias à frente do desenvolvimento ao se combinar essas áreas se tornam uma fonte de conhecimento e avanço emocionante de se testemunhar.

A associação e trabalho mútuo entre desenvolvedores e comunidade para obter avanços tecnológicos nas ditas áreas é um fator a se comemorar. Tomando como exemplo o InstantNGP de Müller et al. (2022), a combinação de técnicas de aprendizado de máquina com as subáreas de Computação Gráfica para alcançar resultados de relativamente alta qualidade pode ser considerado um marco. Isso é mais surpreendente tendo em vista a dificuldade que as áreas de tratamento de imagens e reconhecimento de padrões enfrentam constantemente ao tentar equilibrar a equação qualidade/desempenho. Ser capaz de treinar uma máquina para obter modelos virtuais 3D a partir de imagens 2D tendo como base os conhecimentos acumulados pelos campos de radiância do NeRF de Mildenhall et al. (2020) também é um avanço a se comemorar.

Usando como panorama a computação gráfica, a área de visão computacional foi bem representada com suas técnicas de reconstrução utilizadas por Moulon et al. (2016). Desenvolver uma solução ao empregar um algoritmo de correspondência entre imagens 2D, e

utilizá-lo para geração de nuvens de pontos de densidade e reconstrução de malhas não deixa nada a desejar. Isso tudo a torna uma orgulhosa contribuição para a computação gráfica e uma alternativa viável para qualquer pesquisador interessado em ingressar no campo da reconstrução virtual a partir de planos reais.

Acredita-se que os objetivos deste trabalho foram alcançados. Ao utilizar duas tecnologias, InstantNGP e OpenMVG, foi possível ser alcançado a reconstrução de um ambiente real, mesmo que construído de forma artificial, em um ambiente virtual para visualização digital. O estudo das técnicas de reconstrução de cada tecnologia tornou possível a adaptação e utilização das mesmas para captura de objetos em formato oco, a captura se tornou viável somente após vários testes.

Um extenso levantamento bibliográfico foi realizado nas áreas de computação gráfica, aquisição de imagens e reconstrução tridimensional de imagens, assim como a subárea de inteligência artificial, redes neurais e o aprendizado de máquina. Essa pesquisa serviu de aprofundamento pessoal na literatura de forma inestimável, que funcionou não só como guia, mas como incentivo no prosseguimento do trabalho.

Com enriquecimento nas bases da literatura, o levantamento de pesquisa das tecnologias escolhidas foi executado. Um resumo, compreendendo os principais pontos dos algoritmos do NeRF, InstantNGP e OpenMVG, foi utilizado para caráter comparativo, em que foi feito o levantamento de requisitos, suas bases de pesquisas e métodos de captura, análise e modos de execução ao utilizar imagens capturadas 2D para reconstrução 3D.

Foi especificado como o projeto se desenvolveu. Demonstraram-se como os ambientes foram construídos para captura de ambiente real, as dificuldades quanto à iluminação e qualidade de imagens foram listados e testados. As máquinas para teste tiveram suas especificações listadas e explicadas, bem como a documentação das máquinas de captura de imagens utilizando celulares de tipo smartphone para que sirvam como referência futura.

Quanto às tecnologias, foi explicado e registrado para servir como estudo de caso bem como foi demonstrado todo o procedimento instalação do InstantNGP e OpenMVG, e suas dificuldades e os pontos que merecem atenção.

Da mesma forma, o funcionamento do InstantNGP e OpenMVG foram analisadas e documentados, da mesma forma que os resultados todo experimento foram analisados e comparados, demonstrando de acordo com alguns requisitos necessários, qual técnica melhor se adequa ao projeto de reconstrução 3D.

Por fim, os resultados alcançados foram apresentados em formato de imagens, demonstrando como cada desenlace alcançou seu correspondente fim. A tabela comparativa ao final do documento serve para facilitar a correspondência de cada resultado estabelecido a cada tecnologia quando foram submetidos às imagens retiradas da maquete 02. Ao analisar as informações, é possível afirmar que o modelo 3D reconstruído pela técnica empregada pelo InstantNGP supera a qualidade alcançada pelo OpenMVG, levando em conta utilização da tecnologia para reconstruir ambientes em menor escala e modelagem de interiores. Essa afirmação torna-se verdade quando a barreira do hardware é ultrapassada, quando o hardware não suporta o processamento necessário o OpenMVG responde melhor.

## 5.2. Trabalhos Futuros

Sugere-se como trabalhos futuros, levando em conta a importância que a pesquisa bibliográfica teve ao decorrer do mesmo, um novo levantamento de tecnologias mais atuais nas mesmas áreas escolhidas, visto a rápida e constante evolução que a tecnologia digital apresenta. Isso se deve pela constante e garantida evolução em áreas como análise de padrões e aprendizado de máquina estão em evolução crescente. Recomenda-se ainda o emprego de outras formas para testes: utilizando as técnicas escolhidas em maquetes maiores ou ambiente reais em melhor estado; aplicando métodos mais eficiente para miniaturizar os ambientes de teste para melhor qualidade de captura de imagens; ou melhorando a captura de imagens para utilização nas reconstruções em ambiente virtual.

Por fim, é recomendada também a utilização de máquinas mais atuais e com melhor capacidade de processamento, de forma que se obtenha o melhor grau possível de dados e diminuindo o grau de erro que foi detectado e documentado com esse trabalho. Espera-se que em trabalhos futuros, não seja preciso um ambiente miniaturizado para conseguir um nível de detalhamento satisfatório, e, por conseguinte a reconstrução em ambiente virtual seja executada com uma quantidade de maior de detalhes do que a alcançada com esse trabalho. Dessa forma, permanece o desejo que o mundo real seja reduzido a nível virtual algum dia.

## Referências Bibliográficas

AHISHAKIYE, Emmanuel. GIJZEN, Martin Bastiaan Van. TUMWIINE, Julius. WARIO, Ruth. OBUNGOLOCH, Johnes. **A survey on deep learning in medical image reconstruction**. 2021. Published by Elsevier B.V. on behalf of Chinese Medical Association. <<https://doi.org/10.1016/j.imed.2021.03.003>>

AHMAD, Omar. RAMAMURTHI, Krishna. WILSON, Kevin E. ENGELKE, Klaus. PRINCE, Richard L. TAYLOR, Russell H. **Volumetric DXA (VXA): A New Method to Extract 3D Information From Multiple In Vivo DXA Images**. Journal of Bone and Mineral Research, Volume. 25, Issue 12. December, 2010. Pages 2744–2751. Disponível: <<https://doi.org/10.1002/jbmr.140>>

BERNARDINI, Fausto. RUSHMEIER, Holly. **The 3D Model Acquisition Pipeline**. Computer Graphics Forum, v. 21, p. 149-72, 2002. IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA.

BIESSECK, Bernardo Janko Gonçalves, ZACARKIM, Valber Lemes. **Avaliação da CNN FaceNet para reconhecimento facial de estudantes em sala de aula / Evaluation of CNN FaceNet for facial recognition of students in the classroom**. 2021. Brazilian Journal of Development, 7(3), 27558–27563. <https://doi.org/10.34117/bjdv7n3-455>

BLAIS, F. **Review of 20 years of Range Sensor Development**. 2004. Journal of Electronic Imaging, 13(1): 231-240. January 2004. NRC 46531. National Research Council of Canada.

CYGANEK, Boguslaw. SIEBERT, J. Paul. **An Introduction to 3D Computer Vision Techniques and Algorithms**. 2009. John Wiley & Sons.

DA COSTA, Gonçalo Maia Jácome. **Map-It – Mapeamento Tridimensional de Interiores**. 2014. 76 f. Dissertação para obtenção do grau de mestre em Engenharia Eletrotécnica e de Computadores, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

DA SILVA, F. P. DUARTE, L. C. ROLDO, L. KINDLEIN JR, W. **A Digitalização Tridimensional Móvel e sua aplicação no Design de Produto**. 2010. PGDesign. UFRGS.

DRISSYA, K. RAM, Dr. A. Ranjith. **3D Reconstruction from Images : A Review**. Proceedings of the International Conference on Systems, Energy and Environment, 2022. (ICSEE 2022).

DU, Hao. HENRY, Peter. REN. CHENG, Marvin. GOLDMAN, Dan B. SEITZ, STEVEN M. FOX, Dieter. **Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera**. Department of Computer Science and Engineering. University of Washington, Seattle, WA, USA, 2011.

ECK, David J. **Introduction to Computer Graphics**. Versão 1.4. Department of Mathematics and Computer Science. Hobart and William Smith Colleges. New York, USA. 2023. Disponível online no site: <<https://math.hws.edu/graphicsbook/>>.

GARDNER, Howard. **A Multiplicity of Intelligences**. 1998. Scientific American.



GONZALEZ, Rafael C. WOODS, Richards E. 2010. **Processamento Digital de Imagens**. 3ª Edição. São Paulo, São Paulo: Pearson Prentice Hall.

GUIZZO, E. (2011). **How Google's Self-Driving Car Works**. IEEE Spectrum: 18 Oct. 2011. Disponível em: <<http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>>.

HEARN, Donald. BAKER, M. Pauline. **Computer Graphics: C Version**. 2ª Edição. Universidade de Michigan, USA. Prentice Hall, 1997.

HENRY, Peter. KRAININ, Michael. HERBST, Evan. REN, Xiaofeng. FOX, Dieter. **RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments**. 2012. The International Journal of Robotics Research.

JACOB, Benoît. GUENNEBAUD, Gaël. and others. **Eigen v3**. 2010. Disponível em: <<http://eigen.tuxfamily.org>>

JANG, Mingyu. YOON, Hyunse. LEE, Seongmin. KANG, Jiwoo. LEE, Sanghoon. **A Comparison and Evaluation of Stereo Matching on Active Stereo Images**. Sensors. 2022, Switzerland. <<https://doi.org/10.3390/s22093332>>

JOSÉ, Marcelo Archanjo. **Reconstrução Tridimensional de baixo custo a partir de par de imagens estéreo**. Dissertação para a obtenção do título de Mestre em Engenharia. 2008. Escola Politécnica da Universidade de São Paulo. São Paulo, Brasil.

JÚNIOR, Sérgio Carlos Portari. **Um Sistema para Extração Automática de Keyframes a partir de Fluxos de Vídeo direcionada à Reconstrução Tridimensional de Cenários Virtuais**. 2013. 82 f. Exame para conclusão de Mestrado apresentado ao Programa de Pós-Graduação em Televisão Digital, da Faculdade de Arquitetura, Artes e Comunicação, da Universidade Estadual Paulista “Júlio de Mesquita Filho”. Bauru, SP.

KUS, Abdil. **Implementation of 3D Optical Scanning Technology for Automotive Applications**. 2009. Uludağ University, Vocational School of Technical Sciences, Bursa – Turkey.

LIN, C.-H., KONG, C. LUCEY, S. **Learning Efficient Point Cloud Generation for Dense 3D Object Reconstruction**. 2018. Proceedings of the AAAI Conference on Artificial Intelligence, 32. <<https://doi.org/10.1609/aaai.v32i1.12278>>

MAKEN, Payal. GUPTA, Abhishek. **2D-to-3D: A Review for Computational 3D Image Reconstruction from X-ray Images**. Article in Archives of Computational Methods in Engineering. 2022. <<https://doi.org/10.1007/s11831-022-09790-z>>

MILDENHALL, Ben. SRINIVASAN, Pratul P. TANCIK, Matthew. BARRON, Jonathan T. RAMAMOORTHY, Ravi. NG, Ren. **NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis**. 2020. Computer Vision and Pattern Recognition. <[arXiv:2003.08934](https://arxiv.org/abs/2003.08934)>

MOSTAFA, Yaser S. Abu-. ISMAIL, Malik Magdon. LIN, Hsuan-Tien. **Learn From Data: A short course**. 2012. AMLBook. California Institute of Technology.

MOULON, Pierre. MONASSE, Pascal. PERROT, Romuald. MARLET, Renaud. **OpenMVG: Open Multiple View Geometry**. 1st Workshop on Reproducible Research in Pattern Recognition, Dec 2016, Cancun, Mexico. pp.60-74. Hal-01497080.

MÜLLER, Thomas. EVANS, Alex. SCHIED, Christoph. KELLER, Alexander. **Instant Neural Graphics Primitives with a Multiresolution Hash Encoding**. ACM Transactions on Graphics. New York, USA. 2022. Disponível em: <<https://github.com/NVlabs/instant-ngp>>

PUETTER, Richard C. GOSNELL, Timothy R. YAHIL, Amos. **Digital Image Reconstruction: Deblurring and Denoising**. Article in Annual Review of Astronomy and Astrophysics. August, 2005. DOI: 10.1146/annurev.astro.43.112904.104850

RIBAS, G.C., RIBAS, E.C. e RODRIGUES JUNIOR, A.J. 2006. **O cérebro, a visão tridimensional, e as técnicas de obtenção de imagens estereoscópicas**. *Revista de Medicina*. 85, 3 (set. 2006), 78-90. DOI:<https://doi.org/10.11606/issn.1679-9836.v85i3p78-90>.

RUSSEL, Stuart. NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. Third Edition. 2010. Prentice Hall Series in Artificial Intelligence. Pearson Education. New Jersey. United States of America.

SANSONI, Giovanna. TREBESCHI, Marco. DOCCHIO, Franco. **State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation**. 2009. Laboratory of Optoelectronics, University of Brescia. Italy.

SCHARSTEIN, Daniel. **View Synthesis Using Stereo Vision**. Alemanha. Springer, 1999.

SCHÖNBERGER, Johannes Lutz. FRAHM, Jan-Michael. POLLEFEYS, Marc. ZHENG, Enliang. **Structure-from-Motion Revisited. Pixelwise View Selection for Unstructured Multi-View Stereo**. Conference on Computer Vision and Pattern Recognition (CVPR). European Conference on Computer Vision (ECCV). 2016. Disponível em: < <https://github.com/colmap/colmap>>

VIRTANEN, Juho-Pekka. KURKELA, Marti. TURPPA, Tuomas. VAAJA, Matti T. JULIN, Arttu. Et al. **Depth Camera Indoor Mapping for 3D Virtual Radio Play**. 2018. The Photogrammetric Record.