UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

Murilo de Oliveira Guimarães

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

MURILO DE OLIVEIRA GUIMARÃES

Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): MURILO DE OLIVEIRA GUIMARÃES

Título do trabalho: Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Murilo De Oliveira Guimaraes**, **Discente**, em 12/01/2025, às 14:08, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 15/01/2025, às 16:25, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089799** e o código CRC **013E0D14**.

Referência: Processo nº 23070.001596/2025-01 SEI nº 5089799

MURILO DE OLIVEIRA GUIMARÃES

Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.
Orientador: Prof. Dr. Fernando Marques Federson

Goiânia
2025

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

GUIMARÃES, MURILO DE OLIVEIRA
Inteligência Artificial Generativa para Cenários Tridimensionais [manuscrito] : Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas / MURILO DE OLIVEIRA GUIMARÃES. - 2025.
136 f.

Orientador: Prof. Dr. Fernando Marques Federson.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. geração de cenários. 3. realidades imersivas. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

MURILO DE OLIVEIRA GUIMARÃES

Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 17 de dezembro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Prof. Dr. Arlindo Rodrigues Galvão Filho
(INF-UFG)

MURILO DE OLIVEIRA GUIMARÃES

Inteligência Artificial Generativa para Cenários Tridimensionais

Estudo sobre a Inteligência Artificial para Geração de Cenários Virtuais para Realidades Imersivas

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Geração de Cenários Tridimensionais (Realidades Imersivas)**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Generation of Three-Dimensional Scenarios (Immersive Realities)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada

Murilo de Oliveira Guimarães

Especialista em: Geração de Cenários Tridimensionais (Realidades Imersivas)

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: **Murilo de Oliveira Guimarães**

Especialidade: **Geração de Cenários Tridimensionais (Realidades Imersivas)**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha Jornada começou na **Semana 1** com o foco em avaliar diversas áreas de aplicação de Inteligência Artificial (IA) dentro de Realidades Imersivas. A partir da chamada dos Congressos, em especial do Congresso IEEE VR 2024, pude perceber a abrangência dos grandes temas que podem ser visualizados no **Apêndice 1**. A IA ainda não é muito explorada dentro das Realidades Imersivas, diversos dos artigos publicados não possuem uma alta integração com IA, apenas algumas aplicações muito simples. Porém, devido a formação obtida durante as disciplinas de Visão Computacional e alguns projetos influenciaram a minha escolha para **Geração de Cenários Tridimensionais (Realidades Imersivas).**

Depois de algumas leituras, pude me aprofundar, na **Semana 2**, sobre como está o “estado da arte” na geração de cenários e um pouco sobre como é feito também em alguns jogos atualmente que geram cenários proceduralmente. Algumas observações que considero importantes podem ser obtidas em detalhes no material disponibilizado no

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.

Apêndice 2. Em relação ao estado da arte, que atualmente se resume na utilização de Neural Radiance Fields e de 3D Gaussian Splatting para a parte de Reconstrução e Geração 3D são Mildenhall[2] e Kopanas, Kerbl [3], onde os autores criaram essas técnicas que estão sendo exploradas até o presente momento.

Com essas técnicas em mente, foi possível, na **Semana 3 e 4**, realizar diversos experimentos com ferramentas open source e closed source também, esses experimentos envolvem a utilização das técnicas citadas anteriormente para reconstrução de um cenário 3D a partir de um conjunto de imagens do mesmo. Alguns desses resultados podem ser verificados no **Apêndice 3**, onde alguns ambientes foram fotografados, sejam com câmera de celular, ou até mesmo drones e reconstruídos tridimensionalmente. Além disso, é possível também gerar esses cenários através apenas de texto, com a utilização de modelos de Difusão, que são capazes de gerar imagens com o uso de IA. Dessa forma, não são mais necessários a utilização de um conjunto de imagens tiradas por um celular, apesar do resultado ainda ser bastante inferior. Posteriormente, meu foco serão esses modelos que utilizam de difusão, a possibilidade de criar cenários a partir de um prompt de texto acabou me animando mais.

Depois de conseguir gerar e reconstruir alguns cenários, as **Semanas 5 e 6**, foram utilizadas para conseguir pesquisar sobre alguns frameworks que poderiam ser utilizados para interagir e manusear esses cenários 3D. Os frameworks estudados foram a Unity, Unreal Engine que são motores gráficos e também alguns visualizadores como o WebGL viewer, SuperSplat e Spline. Os motores gráficos estudados conseguem interagir com esses cenários através de alguns plugins, podem ser visualizados no **Apêndice 4**, falando sobre o funcionamento de cada um e suas limitações. Devido a algumas dessas limitações e dificuldades, o framework que será utilizado até o final das semanas foi a Unity, aproveitando sua vasta comunidade e suporte. Eu aproveitei para estudar um pouco mais sobre Unity e seu funcionamento, principalmente o plugin para conseguir importar os 3D Gaussian Splatting para dentro do motor gráfico.

A **Semana 7** serviu para organizar os modelos e datasets encontrados que estão no **Apêndice 5** e ter uma visão mais clara das possibilidades. As alternativas que pensei seria

de realizar um fine-tuning em algum dos modelos ou de integrar algum desses modelos na Unity. Decidi seguir a ideia de integrar algum dos modelos na Unity, poderia ser feito também um fine-tuning dos modelos que utilizam de difusão, porém aproveitei que estava estudando um pouco mais de Unity e segui com a integração.

Com isso, o restante das **Semana 8, 9 e 10** foram para fazer essa implementação, foi possível criar um workflow completo onde o usuário de dentro do VR consegue pedir por um cenário que ele desejar através da voz ou por escrito. A partir dessa requisição, a Unity se conecta em um servidor remoto que tem um modelo já configurado para gerar o cenário 3D a partir dessa requisição do usuário. Com esse cenário 3D gerado, o usuário consegue importar para a Unity esse cenário, que é um arquivo, e explorar e interagir com o novo ambiente. As implementações e resultados podem ser encontrados no **Apêndice 6**.

Em função de tudo que vivi nesta Jornada, gostaria de deixar registrado que foi uma experiência bem divertida e esclarecedora. Eu tinha um conhecimento breve sobre o assunto, mas com essa Jornada foi possível explorar com calma e foco, além também de criar uma ideia clara de como explorar qualquer outro assunto da mesma forma. Consegui aprender bem o processo, e foi de extrema importância essa Residência para minha formação, deixando bem claro para o caminho que vou seguir adiante.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O foco dessa semana foi procurar por diversas áreas que a IA pode interagir com Realidades Imersivas. Por meio de blogs e artigos de revisão da literatura, consegui encontrar diferentes áreas de aplicação de Realidade Virtual e também de IA na Realidade Virtual. Organizei um mapa mental para melhor visualização dessas áreas.

- Mapa mental sobre áreas de interesse

A área de interesse que resolvi escolher foi IA Generativa para modelagem 3D de objetos, avatares ou cenários. A área de geração de objetos é a mais desenvolvida, seguida de avatares e por fim cenários, então resolvi explorar a parte de cenários.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

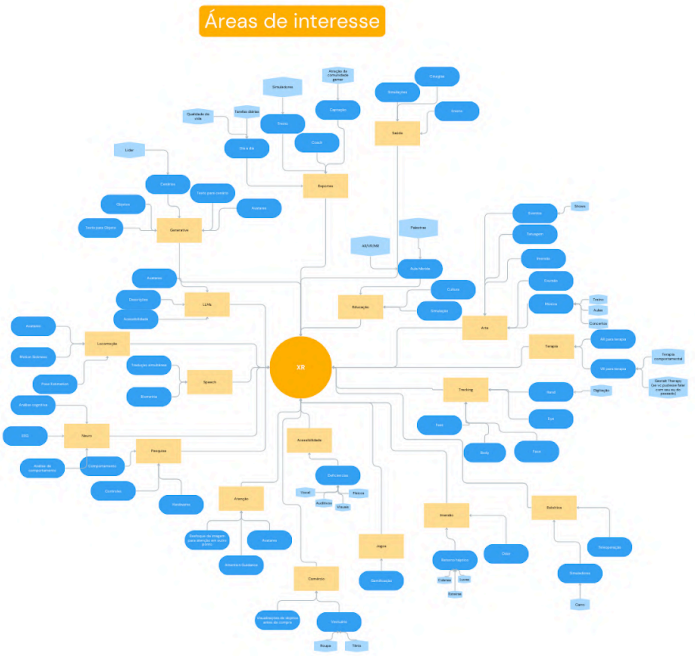
Pesquisar sobre geração procedural e a diferença para IA Generativa.
https://medium.com/@furisticstudio/ai-and-procedural-generation-37563ba149bc

Buscar o estado da arte para IA generativa para cenários.
https://ar5iv.labs.arxiv.org/html/2401.17807v1

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!



Onde lê-se:

Áreas de Interesse XR (Realidade Estendida) Generative

- Cenários
  - Texto para cenário
  - Texto para Objeto
- Objetos
- Avatares

Acessibilidade

- Deficiências
    - Auditivas
    - Físicas
    - Visuais
    - Vocais

Speech

- Tradução simultânea
- Biometria

Tracking

- Hand
- Eye
- Face
- Body
- Feet

Imersão

- Retorno háptico
    - Coletes
    - Luvas
    - Esteiras
- Odor

Locomoção

- Motion Sickness
- Pose Estimation

Atenção

- Desfoque da imagem para atenção em outro ponto
- Attention Guidance

Neuro

- EEG
- Análise de comportamento
- Análise cognitiva

Educação

- AR/VR/MR
- Palestras
- Aula híbrida
- Cultura

Terapia

- AR para terapia
- VR para terapia
- Terapia comportamental
- Gestalt Therapy (se você pudesse falar com seu eu do passado)

Saúde

- Simulações
- Ensino
- Cirurgias

Esportes

- Treino
    - Coach

    - Simuladores
- Captação
    - Atração da comunidade gamer
- Dia a dia
    - Tarefas diárias
    - Qualidade de vida

Jogos

- Gamificação

Artes

- **Arte**
    - Imersão
    - Excursão
    - Eventos
    - Música
    - Aulas

Robótica

- Teleoperação
- Simuladores
    - Carro autônomo

Comércio

- Visualizações de objetos antes da compra
    - Vestuário
        - Roupa
        - Tênis

Pesquisa

- Pesquisa de aplicação

- Hardwares

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 26 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Foi feita uma pesquisa sobre o histórico da geração de cenários digitais e também sua evolução para geração procedural até os dias de hoje, onde é utilizado IA.

Geração de Cenários

Busquei também sobre o estado da arte em IA e técnicas para geração de cenários. Evitei um pouco a área de geração de cenários em jogos digitais e foquei mais em técnicas utilizadas para geração de cenários no geral utilizando IA. Encontrei diversas técnicas que foram muito utilizadas e que estão sendo utilizadas hoje em dia. Que envolvem Fotogrametria, Neural Radiance Fields e 3D Gaussian Splatting.

Photogrammetry / NeRF / Gaussian Splatting comparison

Nesse vídeo é possível ver as diferenças entre cada uma das técnicas. Sendo que cada uma foi uma evolução para a outra sendo que na atualidade 3DGS e NeRF competem na geração de melhores resultados.

Procurei a existência de ferramentas que utilizam essas técnicas e encontrei algumas como Luma AI e Polycam, fiz um teste com cada uma delas.
Fotogrametria: Fotogrametria
NeRF: NeRF
Gaussian Splatting: 3D Gaussian Splatting

Estudei um pouco mais sobre cada uma das técnicas
Geração de Cenários

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Estudo sobre as técnicas descobertas e testar em cenários mais abertos.

Estudo sobre outras formas de geração de cenários como por exemplo texto para cenário.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Geração de Cenários

Histórico da Geração de Cenários

A geração de cenários digitais tem suas raízes na evolução dos gráficos computacionais e na necessidade de criar ambientes detalhados e interativos para jogos e simulações. No início, o processo era puramente manual, com artistas gráficos desenhando cada detalhe do cenário. Em jogos clássicos como **Super Mario Bros.** (1985), cada plataforma, obstáculo e fundo era meticulosamente desenhado e posicionado. Isso não apenas demandava muito tempo e esforço, mas também limitava a capacidade de criar ambientes expansivos e dinâmicos.

Conforme a demanda por mundos mais ricos e detalhados cresceu, surgiram técnicas para automatizar partes do processo. Algoritmos de geração de terreno, baseados em fractais e ruído Perlin, foram algumas das primeiras ferramentas usadas para criar paisagens vastas e realistas sem a necessidade de modelagem manual de cada colina e vale. Esses algoritmos permitiam que designers gerassem mundos complexos a partir de poucas variáveis, facilitando a criação de ambientes como montanhas, florestas e oceanos.

Evolução para a Geração Procedural

A geração procedural evoluiu como uma resposta à limitação de recursos e à necessidade de variedade em jogos e simulações. Diferente da abordagem manual, onde cada elemento é criado individualmente, a geração procedural utiliza algoritmos para criar ambientes inteiros com base em conjuntos de regras predefinidas. Esses algoritmos são capazes de gerar desde terrenos e estruturas até padrões complexos de comportamento e interação.

Princípios Básicos

O princípio fundamental da geração procedural é que pequenos conjuntos de regras podem produzir uma grande variedade de resultados. Por exemplo, um algoritmo pode gerar diferentes tipos de árvores variando apenas parâmetros como altura, número de galhos e densidade de folhas. Da mesma forma, um algoritmo de geração de mapas pode criar terrenos variando parâmetros como altitude média, presença de rios e densidade de vegetação.

Apesar dos benefícios, a geração procedural tradicional tem suas limitações. Como os algoritmos seguem regras fixas, o resultado pode se tornar previsível ou carecer de uma "humanidade" ou toque artístico que a criação manual traz. Por exemplo, em um mundo gerado proceduralmente, pode faltar o tipo de narrativa visual ou a composição intencional de cenas que um designer experiente criaria.

Geração Procedural com IA

Com o avanço da IA e do aprendizado de máquina, a geração procedural deu um salto significativo. A IA trouxe a capacidade de não apenas seguir regras, mas de aprender e se adaptar. Em vez de depender exclusivamente de parâmetros fixos, a geração procedural com IA pode utilizar grandes quantidades de dados para aprender padrões e estilos, criando conteúdo que é não apenas complexo, mas também contextualmente relevante.

A IA permite a criação de cenários que respondem dinamicamente às ações do usuário. Por exemplo, em um jogo de mundo aberto, a IA pode gerar eventos e ambientes com base no comportamento do jogador, criando uma experiência personalizada. Isso é possível porque modelos de IA podem ser treinados para entender a intenção e o contexto, ajustando a geração de conteúdo para manter o interesse e o desafio.

Uma das abordagens mais promissoras é o uso de redes neurais generativas, como GANs (Generative Adversarial Networks) e modelos baseados em Transformers. Esses modelos podem ser treinados com grandes bases de dados de cenários reais ou estilizados, aprendendo a gerar ambientes que não apenas parecem realistas, mas também seguem estilos artísticos ou temáticos específicos.

Por exemplo, um modelo treinado com imagens de cidades históricas pode gerar uma cidade fictícia que segue os mesmos padrões arquitetônicos e de layout urbano. Da mesma forma, um modelo treinado com paisagens alienígenas de filmes de ficção científica pode criar mundos que parecem tirados diretamente de uma obra de arte conceitual.

Fotogrametria: Funcionamento, História e Estado da Arte

A fotogrametria é a ciência e a arte de extrair informações tridimensionais de imagens bidimensionais, permitindo a reconstrução precisa de objetos, cenários e paisagens. Amplamente utilizada em áreas como engenharia, arquitetura, arqueologia, geologia e realidade virtual, a fotogrametria evoluiu ao longo dos séculos, acompanhando avanços na tecnologia de captura e processamento de imagens. Este texto explora o funcionamento técnico da fotogrametria, sua história e o estado da arte na aplicação desta técnica.

Funcionamento

A fotogrametria funciona com base em princípios de visão estereoscópica e geometria projetiva. O processo envolve a captura de múltiplas imagens de um objeto ou cenário sob diferentes ângulos. A partir dessas imagens, algoritmos computacionais detectam pontos de interesse em comum entre as fotografias e calculam a posição espacial desses pontos utilizando conceitos de triangulação.

Os passos básicos do processo de fotogrametria incluem:

1. **Captura de Imagens**: Fotografias de alta qualidade são tiradas sob diferentes perspectivas, garantindo que cada parte do objeto ou cena seja coberta por pelo menos duas imagens sobrepostas. Drones e câmeras manuais são frequentemente utilizados.
2. **Extração de Características**: Algoritmos como SIFT (Scale-Invariant Feature Transform) ou ORB (Oriented FAST and Rotated BRIEF) identificam pontos de interesse nas imagens, como arestas ou texturas.
3. **Matching**: Os pontos de interesse comuns entre pares de imagens são correspondidos para determinar a relação espacial entre as fotografias.
4. **Reconstrução 3D**: Com base no posicionamento relativo das imagens e nos pontos de correspondência, os algoritmos de reconstrução criam uma nuvem de pontos 3D representando o objeto ou a cena.
5. **Texturização e Malha**: A nuvem de pontos é convertida em uma malha tridimensional, e as texturas das imagens originais são projetadas sobre ela, gerando um modelo 3D fotorealista.
6. **Pós-Processamento**: Ferramentas de modelagem refinam o modelo, corrigindo imperfeições e otimizando sua aplicação para diferentes finalidades.

A precisão do processo depende da qualidade das imagens, do grau de sobreposição entre elas e do controle de parâmetros como iluminação e calibração da câmera.

Estado da Arte

Atualmente, a fotogrametria é uma ferramenta essencial em diversas áreas, com avanços significativos impulsionados por técnicas de inteligência artificial, aprendizado de máquina e computação gráfica. Alguns dos aspectos mais recentes e promissores incluem:

1. **Softwares Automatizados**: Ferramentas como Agisoft Metashape, RealityCapture e Pix4D utilizam algoritmos avançados para processar milhares de imagens de forma automatizada, permitindo reconstruções rápidas e detalhadas.
2. **Fotogrametria em Tempo Real**: Com a crescente capacidade de processamento de GPUs, é possível realizar reconstruções 3D quase instantâneas, integrando a fotogrametria em sistemas de realidade aumentada e virtual.
3. **Integração com IA**: Modelos baseados em redes neurais são utilizados para melhorar a correspondência de pontos e reduzir erros de reconstrução, aumentando a qualidade dos modelos 3D.
4. **Aplicações em Realidades Imersivas**: A fotogrametria tem sido fundamental na criação de cenários e objetos hiper-realistas para realidade virtual e aumentada, ampliando sua relevância em entretenimento, educação e treinamentos industriais.
5. **Fotogrametria Multiespectral**: Sensores que capturam diferentes comprimentos de onda, como infravermelho e ultravioleta, permitem a análise detalhada de superfícies e materiais, com aplicações em agricultura de precisão e conservação de patrimônio histórico.
6. **Escalabilidade e Colaboração**: Plataformas em nuvem permitem a colaboração de múltiplos usuários no processamento e análise de modelos fotogramétricos, viabilizando projetos de larga escala, como mapeamento urbano ou monitoramento ambiental.

Desafios e Limitações

Um dos principais desafios da fotogrametria está na sua dificuldade em lidar com superfícies brilhantes e reflexivas. Esse problema ocorre porque a técnica depende da detecção e correspondência de pontos de interesse em múltiplas imagens. Em superfícies reflexivas, a luz é refletida de maneira especular, causando distorções nos padrões visíveis e dificultando a identificação consistente de características. Além disso, o reflexo pode mudar significativamente dependendo do ângulo de visão, resultando em discrepâncias entre as imagens capturadas. Como consequência, os algoritmos de reconstrução frequentemente falham em triangulações precisas ou geram artefatos, comprometendo a qualidade do modelo 3D. Esse desafio é particularmente problemático em aplicações como digitalização

de metais polidos, superfícies de vidro e corpos d'água, onde técnicas adicionais, como sprays matificantes ou sensores complementares como Lidar, são frequentemente necessárias para superar essas limitações.

Neural Radiance Fields (NeRFs)

Neural Radiance Fields (NeRFs) são uma abordagem inovadora para representar e sintetizar cenários tridimensionais a partir de imagens 2D. Baseados em redes neurais, os NeRFs superam muitas limitações das técnicas tradicionais de reconstrução 3D, como a fotogrametria, especialmente em cenários complexos que envolvem superfícies reflexivas, transparências ou oclusões. Este texto detalha o funcionamento dos NeRFs, sua evolução histórica, o estado da arte na área e suas limitações atuais.

Funcionamento

Os NeRFs utilizam redes neurais profundas para modelar a função volumétrica que descreve como a luz interage com o espaço tridimensional. O modelo representa um espaço 3D como um campo contínuo, calculando densidade e cor (emitância) para cada ponto de uma cena. O processo de funcionamento pode ser dividido em etapas principais:

1. **Entrada de Dados**: Uma série de imagens 2D capturadas de diferentes ângulos é fornecida como entrada, juntamente com os parâmetros da câmera, como posição e orientação.
2. **Treinamento do Modelo**: Uma rede neural recebe como entrada coordenadas 3D e a direção do raio óptico correspondente. A saída da rede é composta por:
    - **Densidade**: Indica o nível de opacidade do ponto no espaço.
    - **Cor RGB**: Representa a emitância do ponto na direção do raio fornecido.
3. **Renderização**: A partir da integração volumétrica, é gerada uma imagem sintética, calculando como a luz atravessa e interage com o campo de densidade e emitância. Este processo utiliza o princípio de renderização diferencial, permitindo a comparação direta com as imagens reais e o ajuste dos pesos da rede.
4. **Reconstrução**: Após o treinamento, o modelo é capaz de renderizar novas vistas do cenário com alta fidelidade visual, incluindo superfícies reflexivas, texturas complexas e oclusões.

Os NeRFs superam a fotogrametria em aspectos como a representação contínua do espaço e a capacidade de lidar com propriedades ópticas avançadas, como transparências e reflexos, que são problemáticas para abordagens baseadas apenas em correspondências de pontos visíveis.

Estado da Arte

Os NeRFs continuam a evoluir, impulsionados por avanços em hardware, algoritmos de otimização e integração com outras técnicas. O estado da arte atual abrange:

1. **Renderização em Tempo Real**: Avanços no uso de GPUs e técnicas de amostragem eficientes, como Instant-NGP, permitem renderizar NeRFs em tempo real, tornando-os viáveis para aplicações interativas.
2. **Aprimoramento da Qualidade Visual**: Modelos como Mip-NeRF utilizam representações hierárquicas para lidar com desfoque de movimento e escalas complexas, melhorando a qualidade da reconstrução.
3. **Adaptação a Cenários Dinâmicos**: DinNeRFs e outras variantes foram desenvolvidas para lidar com mudanças temporais em cenas, permitindo representar objetos em movimento ou ambientes mutáveis.
4. **Aplicações Multimodais**: NeRFs estão sendo combinados com técnicas de difusão e redes GAN para gerar cenários que incorporam propriedades físicas, como iluminação dinâmica e texturas realistas.
5. **Ferramentas Open-Source**: Frameworks como NVIDIA Instant-NGP e PyNeRF democratizaram o acesso à tecnologia, permitindo a pesquisadores e desenvolvedores utilizarem NeRFs de forma mais acessível.

Limitações

Apesar dos avanços, os NeRFs enfrentam desafios significativos:

1. **Dependência de Dados**: Os NeRFs exigem conjuntos de dados de alta qualidade e densidade, com múltiplas imagens de diferentes ângulos e iluminação controlada. Dados incompletos ou ruidosos podem resultar em reconstruções deficientes.
2. **Alto Custo Computacional**: O treinamento de NeRFs requer poder de processamento elevado, especialmente para cenas complexas. Apesar de otimizações recentes, o custo ainda é proibitivo para algumas aplicações.
3. **Escalabilidade**: Representar grandes ambientes ou cenários urbanos completos com NeRFs é um desafio devido à necessidade de armazenamento eficiente e à manutenção da fidelidade.
4. **Dificuldade com Superfícies Extremas**: Embora superem a fotogrametria em reflexões e transparências, os NeRFs ainda apresentam dificuldades com superfícies puramente especulares, como espelhos, devido à ambiguidade na correspondência entre reflexos e geometrias subjacentes.

5. **Generalização Limitada**: Modelos NeRFs treinados em um cenário específico não são facilmente reutilizáveis para outros cenários, necessitando de novos treinamentos para cada caso.

3D Gaussian Splatting

Gaussian Splatting é uma técnica inovadora para a representação e renderização de cenas tridimensionais, que utiliza distribuições gaussianas para modelar o espaço 3D de forma contínua e eficiente. Essa abordagem vem ganhando destaque como alternativa para métodos como Neural Radiance Fields (NeRFs), oferecendo vantagens significativas em termos de velocidade de renderização e eficiência computacional. Este texto explora o funcionamento técnico, a evolução histórica, o estado da arte e as limitações do Gaussian Splatting.

Funcionamento

O Gaussian Splatting representa cenas 3D como uma coleção de distribuições gaussianas definidas no espaço tridimensional. Cada gaussiana é parametrizada por uma posição, uma matriz de covariância (que define a forma e a orientação da distribuição), e um valor de cor associado. O funcionamento pode ser dividido em etapas principais:

1. **Aquisição de Dados**: Imagens de uma cena são capturadas de múltiplos ângulos, juntamente com os parâmetros intrínsecos e extrínsecos das câmeras.
2. **Reconstrução da Cena**:
    - Pontos 3D são estimados a partir das imagens, utilizando técnicas de visão computacional, como triangulação multivista.
    - Cada ponto é convertido em uma distribuição gaussiana que captura não apenas sua posição, mas também sua incerteza e densidade no espaço.
3. **Renderização Baseada em Gaussiana**:
    - A renderização é feita projetando as distribuições gaussianas no espaço 2D da câmera de visualização.
    - A integração volumétrica é utilizada para calcular a cor e a opacidade de cada pixel na imagem resultante.
4. **Otimização Iterativa**: Um processo iterativo ajusta os parâmetros das gaussianas para alinhar a renderização com as imagens de entrada, garantindo alta fidelidade visual.

O Gaussian Splatting supera métodos como NeRFs em cenários que exigem renderização rápida, pois elimina a necessidade de inferência neural contínua. Em vez disso, utiliza

cálculos baseados em distribuições gaussianas, que são mais eficientes para representar superfícies complexas e transparentes.

Estado da Arte

O estado da arte em Gaussian Splatting reflete avanços significativos em eficiência, qualidade visual e aplicações práticas:

1. **Renderização em Tempo Real**: Graças à simplicidade computacional das distribuições gaussianas, esta técnica permite renderizar cenas em tempo real, sendo especialmente vantajosa para aplicações interativas como jogos e realidade virtual.
2. **Suporte para Superfícies Complexas**: Gaussian Splatting se destaca em representar superfícies complexas e desafiadoras, como reflexões especulares e transparências, que são problemáticas para métodos como NeRFs.
3. **Modelos Híbridos**: Abordagens híbridas que combinam Gaussian Splatting com aprendizado profundo permitem lidar com cenas mais dinâmicas ou resolver problemas de otimização não-lineares em reconstruções complexas.
4. **Uso em Ambientes Abertos e Urbanos**: Aplicações como mapeamento urbano e reconstrução de paisagens naturais têm explorado o Gaussian Splatting para capturar grandes ambientes de forma eficiente.
5. **Implementações Open Source**: Frameworks como o *3D Gaussian Splatting Framework* estão tornando a técnica acessível à comunidade de pesquisa e desenvolvimento.

Limitações

Embora o Gaussian Splatting apresente vantagens notáveis, ele também possui limitações que desafiam sua adoção mais ampla:

1. **Dependência de Dados Precisos**: A técnica requer dados de entrada altamente confiáveis, incluindo pontos 3D bem alinhados e informações precisas das câmeras. Erros na entrada podem levar a distribuições gaussianas mal ajustadas, comprometendo a qualidade do modelo.
2. **Ambiguidade em Regiões de Oclusão**: Em áreas onde múltiplas superfícies se sobrepõem ou onde há oclusão significativa, a técnica pode gerar artefatos devido à dificuldade em distinguir entre superfícies concorrentes.
3. **Escalabilidade para Grandes Cenários**: Embora eficiente, o Gaussian Splatting pode apresentar desafios de escalabilidade ao lidar com cenas muito grandes, devido ao aumento exponencial do número de gaussianas necessárias para representar todos os detalhes.

4. **Representação Limitada de Dinâmica Temporal**: A técnica é naturalmente estática, o que limita sua aplicabilidade em cenários dinâmicos ou onde mudanças temporais precisam ser modeladas com alta precisão.
5. **Dependência de Recursos Computacionais**: Apesar de ser mais eficiente que métodos baseados em redes neurais, o Gaussian Splatting ainda exige recursos computacionais significativos para otimização e renderização de cenas altamente detalhadas.

Comparação com NeRFs

Enquanto os NeRFs (Neural Radiance Fields) são focados em representar cenas como um campo contínuo de densidade volumétrica e cor, o Gaussian Splatting representa cenas como um conjunto de pontos distribuídos de acordo com distribuições gaussianas. Isso leva a algumas diferenças significativas:

- **Renderização**: NeRFs geralmente requerem integração volumétrica complexa, o que pode ser computacionalmente intensivo. O Gaussian Splatting, por outro lado, permite uma projeção mais direta e eficiente dos pontos gaussianos, tornando-o mais adequado para aplicações em tempo real.
- **Flexibilidade**: O Gaussian Splatting é mais flexível na representação de superfícies complexas e dinâmicas, enquanto NeRFs são mais eficientes para cenas estáticas e fotorrealistas.
- **Uso em Dispositivos Móveis**: Devido à sua eficiência, o Gaussian Splatting pode ser mais facilmente implementado em dispositivos com menor poder computacional, como headsets de VR/AR e dispositivos móveis, em comparação com NeRFs.

APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 2 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Estudei um pouco sobre o estado da Arte atualmente em relação às técnicas, e documentei os achados
Geração de Cenários

Gravei alguns videos e testei outros exemplos que encontrei na plataforma LUMA AI em um óculos de realidade virtual, gravei alguns vídeos da experiência.
Sala de estar Sala de Estar.mp4
Castelo Castelo.mp4
Bitdog Bitdog.mp4

Fui atrás de outras formas de geração de cenários como por exemplo texto para cenário.

Descobri que é o maior desafio atualmente da área e está utilizando modelos de Difusão e CLIP.
Conversei com o colega Arthur Jung sobre os estudos dele sobre a área e li um pouco sobre os principais artigos da área. Documentei os achados

Geração de Cenários

Encontrei também um vídeo bem interessante sobre o funcionamento do Stable Diffusion
What are Diffusion Models?
E para o CLIP
CLIP: Connecting Text and Images

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Testar alguns modelos encontrados de texto para cenário e como é a integração com os óculos de realidade virtual

Começar a pesquisar os frameworks para a área de geração de cenários, modelagem 3D e Realidade Virtual ou Aumentada.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Geração de Cenários

Text to Scene

Os métodos iniciais de geração de cenas frequentemente exigem dados específicos de cenas para o treinamento, a fim de obter geradores de cenas específicos para determinadas categorias, ou implementam uma reconstrução de cena única com base na imagem de entrada. No entanto, esses métodos são limitados pela qualidade da geração ou pela extensibilidade da cena. Com o aumento dos modelos de difusão em tarefas de inpainting de imagem, vários métodos começaram a utilizar as capacidades de completude de cena desses modelos para implementar tarefas de geração de cena. Recentemente, algumas abordagens propõem estratégias de inpainting progressivo e atualização para gerar cenas 3D realistas usando modelos de difusão pré-treinados. Essas abordagens utilizam malhas poligonais explícitas como representação 3D durante o procedimento de geração. No entanto, essa escolha de representação impõe limitações na geração de cenas ao ar livre, resultando em geometria distorcida e artefatos desfocados nas regiões de fusão das faces da malha. Em contraste, outras abordagens adotam representações implícitas, que oferecem a capacidade de modelar geometria e texturas detalhadas sem requisitos específicos de cena. Consequentemente, essas abordagens conseguem gerar tanto cenas internas quanto externas com alta fidelidade.

Modelos de difusão

Proposta

O artigo apresenta um avanço significativo no campo de **modelos de difusão probabilística** (Diffusion Probabilistic Models), uma classe de modelos de variáveis latentes inspirados por conceitos de termodinâmica fora do equilíbrio. O objetivo é demonstrar que esses modelos, quando treinados corretamente, são capazes de gerar amostras de alta qualidade, superando outros modelos generativos, como GANs e VAE. O trabalho propõe uma nova conexão entre os modelos de difusão e a técnica de **denoising score matching** com a dinâmica de Langevin, permitindo um processo progressivo de descompressão com perda. Os autores também destacam que os modelos de difusão probabilística podem ser interpretados como uma generalização de modelos autoregressivos, o que amplia a aplicabilidade do método para geração de imagens de alta qualidade.

Metodologia

O treinamento do modelo baseia-se em um processo de **difusão reversa** que reverte uma cadeia de Markov usada para adicionar ruído às amostras de dados. A abordagem é estruturada da seguinte maneira:

1. **Processo de Difusão**: O modelo utiliza um processo de difusão que aplica incrementalmente pequenas quantidades de ruído gaussiano aos dados, resultando na destruição progressiva do sinal até que os dados fiquem indistinguíveis do ruído. O modelo é então treinado para inverter esse processo, removendo o ruído passo a passo para gerar amostras realistas.
2. **Treinamento Variacional**: Para o processo reverso, os autores treinam uma cadeia de Markov usando inferência variacional. A transição dessa cadeia é modelada como uma distribuição gaussiana condicional, parametrizada por uma rede neural, que aprende a prever o ruído em diferentes níveis. O treinamento é baseado em um bound variacional que é otimizado durante o processo de treinamento.
3. **Parâmetro de Previsão**: A abordagem proposta por este trabalho envolve a previsão do termo "ε" (ruído gaussiano) durante o processo reverso, em vez da média da distribuição posterior. Essa parametrização simplifica a otimização e resulta em melhores amostras quando comparada a outros métodos.
4. **Experimentos**: Os autores treinam seus modelos em vários conjuntos de dados (CIFAR10, LSUN, CelebA-HQ) e obtêm resultados de qualidade superior, como um score FID de 3.17 no CIFAR10, que é competitivo com os melhores modelos de GANs e superiores aos modelos autoregressivos e baseados em flows.

Conclusão

Os resultados obtidos mostram que os **Modelos de Difusão Probabilística** são uma alternativa poderosa para a geração de imagens de alta qualidade, rivalizando com as abordagens de ponta baseadas em GANs. Eles apresentam um processo de amostragem estável, além de uma conexão teórica sólida com **denoising score matching** e a dinâmica de Langevin, o que reforça a eficiência e a qualidade do método. Além disso, o modelo permite um processo progressivo de compressão com perda, o que o torna potencialmente útil em outras modalidades de dados, além da imagem, e para sistemas de compressão. Embora os autores reconheçam que os modelos ainda não atingem uma performance de log-likelihood competitiva com outros modelos baseados em likelihood, eles mostram que a maioria dos bits usados na codelength corresponde a detalhes imperceptíveis nas imagens.

Stable Diffusion

Proposta

O artigo propõe os **Modelos de Difusão Latente (LDMs)** como uma alternativa eficiente aos modelos de difusão tradicionais para síntese de imagens de alta resolução. Os modelos de difusão, conhecidos por sua capacidade de gerar imagens de qualidade elevada, tradicionalmente operam diretamente no espaço de pixels, o que exige enormes recursos computacionais e longos tempos de treinamento. A principal inovação dos LDMs é a ideia de mover o processo de difusão para o **espaço latente** aprendido por autoencoders poderosos, em vez de atuar no espaço de pixels. Isso reduz significativamente a dimensionalidade dos dados, preservando os detalhes perceptualmente relevantes. A proposta é alcançar um equilíbrio ótimo entre a redução de complexidade e a preservação de detalhes, tornando possível a síntese de imagens de alta qualidade com menor custo computacional. Além disso, os autores introduzem camadas de **cross-attention** na arquitetura do modelo, o que permite que os LDMs sejam condicionados por diversas entradas, como texto, mapas semânticos ou caixas delimitadoras, ampliando sua flexibilidade para várias aplicações de síntese.

Metodologia

A abordagem é dividida em duas fases principais:

1. **Compressão Perceptual**:
    - Nesta fase, um **autoencoder** é treinado para aprender um espaço latente que captura as informações mais importantes da imagem, eliminando detalhes de

alta frequência que são menos perceptíveis ao olho humano. Isso é feito utilizando uma combinação de uma **perceptual loss** e uma **objetiva adversarial baseada em patches**, que garantem que as reconstruções geradas pelo autoencoder sejam localmente realistas e preservem a estrutura da imagem.

- O autoencoder comprime a imagem em uma representação latente de baixa dimensionalidade. A redução de dimensionalidade (com diferentes fatores de downsampling, como f = 4 ou f = 8) garante que o modelo de difusão subsequente tenha menos dados para processar, tornando o processo mais eficiente sem comprometer significativamente a qualidade da reconstrução. Dois tipos de regularização foram utilizados: a penalidade KL e a quantização vetorial (VQ-reg).

2. **Modelos de Difusão Latente (LDMs)**:
   - Após a fase de compressão, os LDMs são treinados diretamente no espaço latente gerado pelo autoencoder. Esse espaço reduzido permite que os **modelos de difusão**, que são compostos por uma cadeia de autoencoders de denoising, se concentrem nos aspectos semânticos da imagem, em vez de gastar recursos computacionais processando detalhes irrelevantes no espaço de pixels.
   - O backbone do modelo de difusão é um **UNet condicional** que pode ser aplicado de forma convolucional, permitindo o treinamento em imagens de alta resolução (até 1024x1024 pixels). Além disso, a arquitetura foi aprimorada com mecanismos de **cross-attention**, o que permite que os LDMs utilizem vários tipos de entradas condicionais, como texto, mapas semânticos e imagens de baixa resolução, tornando-os aplicáveis a uma ampla gama de tarefas de síntese.
   - Os LDMs foram testados em várias tarefas de síntese, incluindo síntese de imagens incondicionais, geração de imagens condicionadas por texto, super-resolução e inpainting. Em todas essas tarefas, os LDMs mostraram desempenho competitivo com as técnicas mais avançadas da área, como **GANs** e **modelos autoregressivos**.

Conclusão

Os **Latent Diffusion Models** (LDMs) apresentados no artigo oferecem uma solução eficiente e eficaz para a síntese de imagens de alta resolução, resolvendo os desafios de custo computacional elevado e longos tempos de treinamento associados aos modelos de difusão convencionais. Ao mover o processo de difusão para o espaço latente, os LDMs

reduzem significativamente a complexidade computacional, sem sacrificar a qualidade da imagem gerada. Os autores demonstram que essa abordagem pode ser usada para uma variedade de tarefas, como super-resolução, inpainting e síntese de imagens condicionadas por texto, superando ou competindo com os métodos de última geração, como GANs e modelos baseados em transformadores. Além disso, o uso de camadas de cross-attention amplia a flexibilidade dos LDMs, permitindo sua aplicação em diferentes tipos de entradas condicionais.

Os LDMs reduzem os requisitos de recursos, facilitando o acesso a esse tipo de tecnologia para pesquisadores e desenvolvedores com menor infraestrutura computacional. Embora os LDMs ofereçam um grande avanço em eficiência, os autores reconhecem algumas limitações, como a necessidade de múltiplas etapas de amostragem, que pode ser mais lenta do que a amostragem direta de GANs. O impacto societal dessa tecnologia também é destacado, com os autores alertando para o potencial uso inadequado de imagens geradas para manipulação e desinformação, mas ao mesmo tempo reconhecendo o potencial criativo e democratizador da tecnologia.

CLIP

Proposta

O artigo propõe o **CLIP** (Contrastive Language-Image Pretraining), um novo método de aprendizado que utiliza supervisão de linguagem natural para treinar modelos visuais transferíveis, sem a necessidade de conjuntos de dados rotulados especificamente para cada tarefa visual. A ideia central é alavancar a enorme quantidade de pares de imagem-texto disponíveis na internet para ensinar os modelos a reconhecer e associar conceitos visuais com descrições em linguagem natural. O modelo CLIP é treinado para prever qual descrição textual corresponde a qual imagem, aprendendo representações visuais robustas que permitem a transferência para tarefas visuais sem necessidade de re-treinamento. O artigo demonstra que esse tipo de pretreinamento pode alcançar performance competitiva em várias tarefas de visão computacional, sem dados específicos de treinamento para essas tarefas, facilitando assim a **generalização zero-shot**.

Metodologia

A metodologia consiste em:

1. **Criação do dataset**: Os autores criaram um novo dataset contendo 400 milhões de pares de imagem-texto extraídos de diversas fontes públicas na internet, com o objetivo de capturar uma ampla gama de conceitos visuais.

2. **Treinamento contrastivo**: CLIP utiliza um treinamento contrastivo onde um codificador de imagem e um codificador de texto são treinados para maximizar a similaridade de pares corretos (imagem e descrição correspondente) e minimizar a similaridade entre pares incorretos. O modelo aprende a projetar as imagens e descrições em um **espaço de embeddings** multimodal onde a similaridade pode ser medida.
3. **Arquitetura**: O modelo de imagem utiliza tanto **ResNet-50** quanto **Vision Transformers (ViT)** como codificadores de imagem. Para o texto, um transformer é utilizado como codificador. Durante o treinamento, o modelo é otimizado com base em uma **perda cross-entropy** entre as previsões e as correspondências reais.
4. **Zero-shot learning**: Após o treinamento, o CLIP pode realizar tarefas de classificação de imagens com base em descrições textuais, sem necessidade de ajustar o modelo para um novo conjunto de dados. Isso é feito simplesmente criando descrições textuais das classes do novo conjunto de dados e selecionando a classe mais similar à imagem no espaço de embeddings.

Conclusão

Os resultados mostram que o **CLIP** é altamente eficiente na transferência para tarefas de visão computacional, alcançando resultados comparáveis a modelos treinados de forma supervisionada em benchmarks populares, como ImageNet, sem usar dados rotulados. O modelo também se mostrou eficaz em uma ampla gama de tarefas, como reconhecimento de ações em vídeos, OCR, e geo-localização. A abordagem oferece uma maneira promissora de aprender representações visuais transferíveis de maneira escalável, aproveitando a supervisão implícita nos textos da internet. Além disso, os autores apontam que a robustez e a eficiência do CLIP sugerem que ele pode servir como base para futuros avanços na combinação de linguagem natural e visão computacional.

Sala de Estar

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Castelo

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Bitdog

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta entrega eu consegui testar alguns modelos pagos e open source para conseguir criar cenários 3D a partir de texto e testar também a integração com os óculos de realidade virtual.
Os registros dessa integração estão nos vídeos:
Lucid Dreamer
- Postapocalyptic city in desert.mp4
- A vibrant, colorful floating community city, clouds above a beautiful, enchanted landscape filled with …
- Cozy livingroom in christmas.mp4
- anime style, animation, best quality, a boat on lake, trees and rocks near the lake. a house and port i…

Skybox
- interior of a large old warehouse, big windows along one wall, lots of shelves for storing boxes and o…
- ruins of a lost civilization on an alien landscape, rusting advanced futuristic technology, abandoned w…
- swirling puffy clouds, dark night sky, dusk.mp4

A integração não é difícil mas também não é muito prática, essencialmente essa integração é feita por motores gráficos que facilitam a visualização e a edição, além também de permitir a utilização dos modelos 3D gerados para diversos propósitos. O motor gráfico que eu encontrei foi a Unreal Engine. Porém, essa integração é feita através de plugins que podem chegar a custar mais de 500 reais. Eu testei uma integração básica utilizando um plugin grátis mas a qualidade não ficou muito boa.

Tentei testar mais alguns modelos open source mas alguns estão desatualizados:

Text2Room: https://arxiv.org/pdf/2303.11989
RoomDreamer: https://arxiv.org/pdf/2305.11337

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Testar mais alguns modelos que acabaram dando bastante erro:

PanFusion: https://arxiv.org/pdf/2404.07949
MVDiffusion: https://arxiv.org/pdf/2307.01097

Testar mais alguns plugins da Unreal Engine como o https://github.com/xverse-engine/XV3DGS-UEPlugin

Pesquisar sobre a integração dessas cenas em outro motor gráfico como a Unity.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Post Apocalyptic city in desert

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

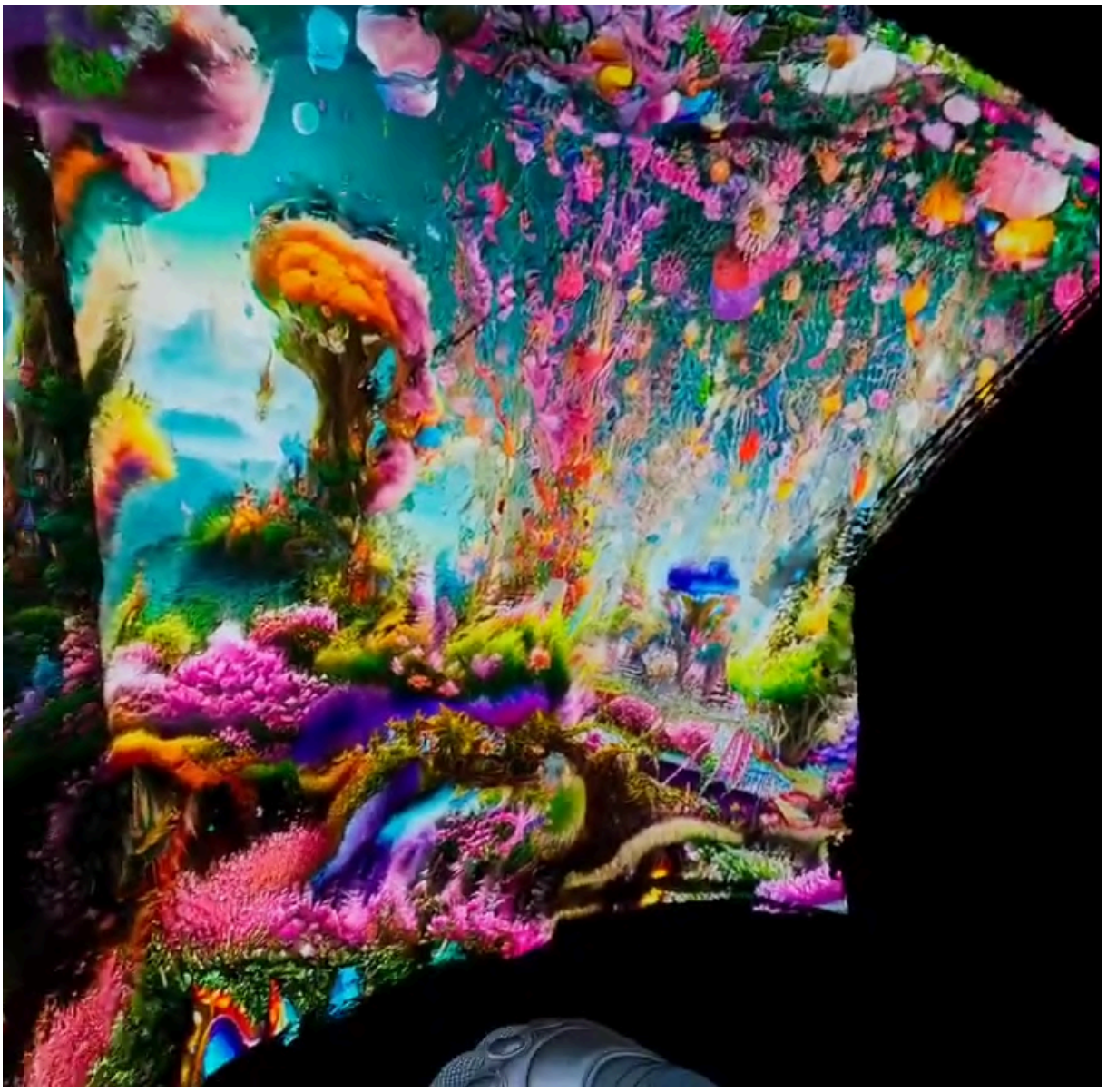
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

A vibrant, colorful floating community city … and color vibrancy

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Cozy living room in christmas

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

anime style … front of a house

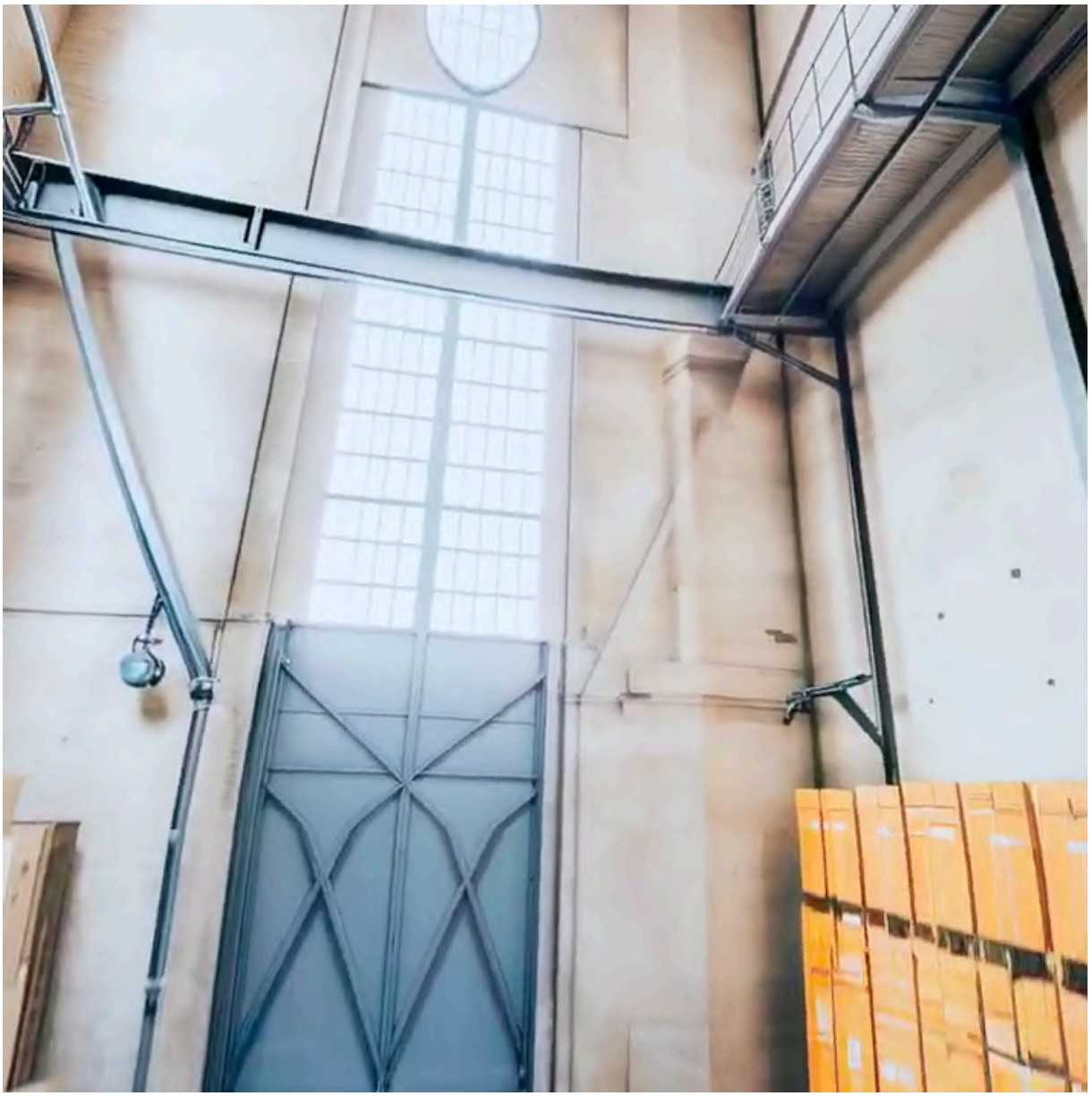
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

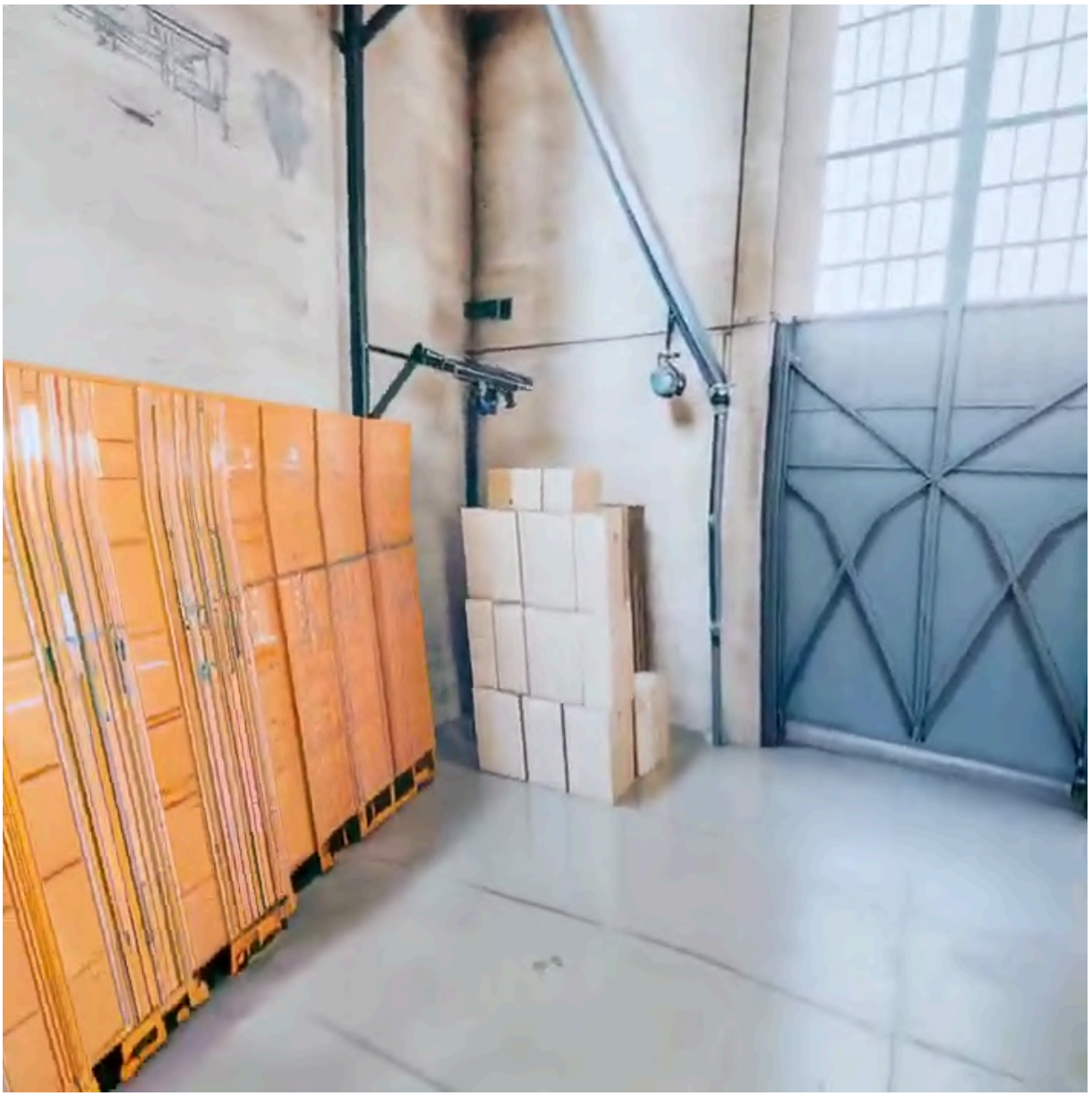
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

interior of a large … graffiti, symmetrical

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

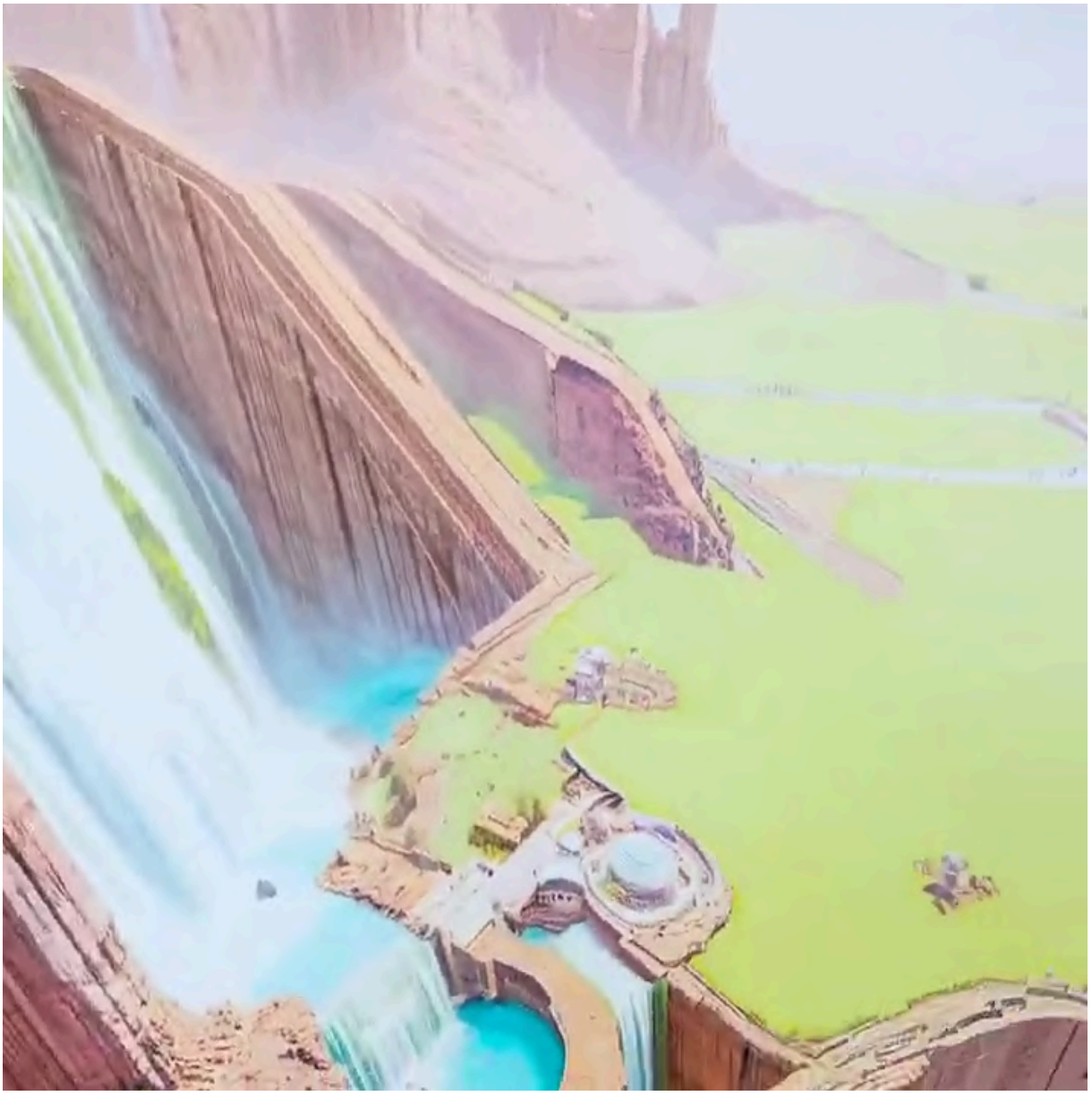


Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

ruins of a lost civilization … micro detailed, masterpiece



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

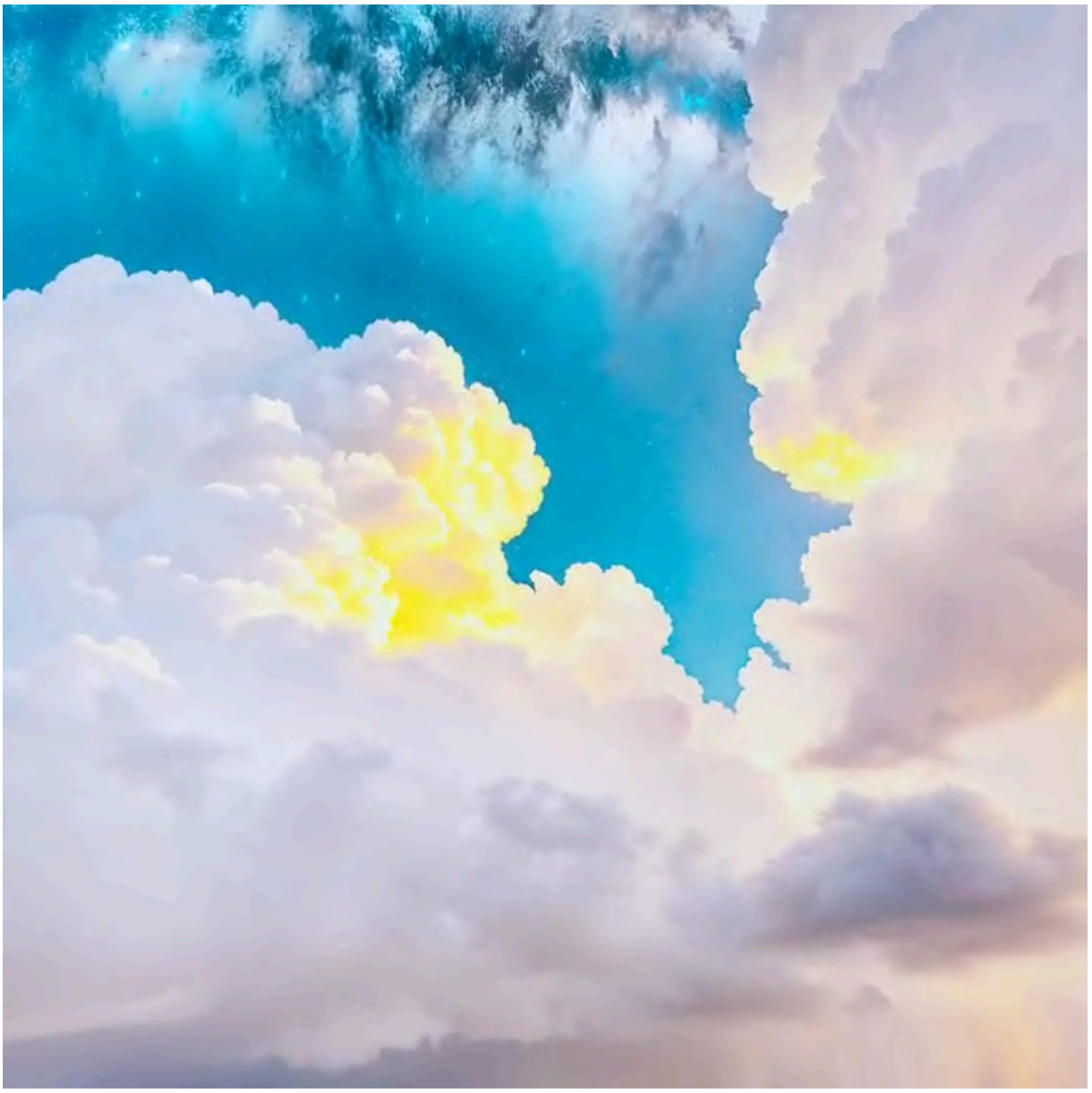
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

swirling puffy clouds, dark night sky, dusk

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 17 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Nesta entrega eu consegui resolver os erros que estava encontrando e testei dois modelos que criam panoramas a partir de texto. Apesar de não ser inteiramente 3D, já consegue proporcionar uma experiência imersiva bacana.

Pan Fusion
- a hallway in a house.mp4

MVDiffusion
- This kitchen is a charming blend of rustic and modern, featuring a large reclaimed wood island with …

Testei também outro plugin para a Unreal Engine e percebi que realmente os plugins gratuitos conseguem fazer o trabalho de passar a representação 3D para a Unreal, sendo possível interagir e editar os cenários 3D.

- XV3DGS-UEPlugin.mp4

Instalei também a Unity, outro motor gráfico bem famoso para a produção de jogos e experiências em Realidade Virtual. A Unity é um opção bem mais preparada para a integração de NeRFs e 3D Gaussian Splattings, ela também faz a integração por meio de plugins, mas a comunidade disponibiliza diversos plugins gratuitamente, capazes até mesmo de refinar a representação 3D do Gaussian Splatting através de uma limpeza manual das nuvens de pontos.

- Gaussian Splatting playground in Unity.mp4

O vídeo para fazer a limpeza dessa cena que eu encontrei foi esse.

Let's Clean the 3D Gaussian Splatting models and remove floating artifacts!

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Planejo testar um pouco mais a Unity e ver as interações possíveis.

Além disso, planejo testar outros frameworks para visualização, os chamados visualizadores, alguns famosos são o MeshLab, Super-Splat e o WebGL viewer.

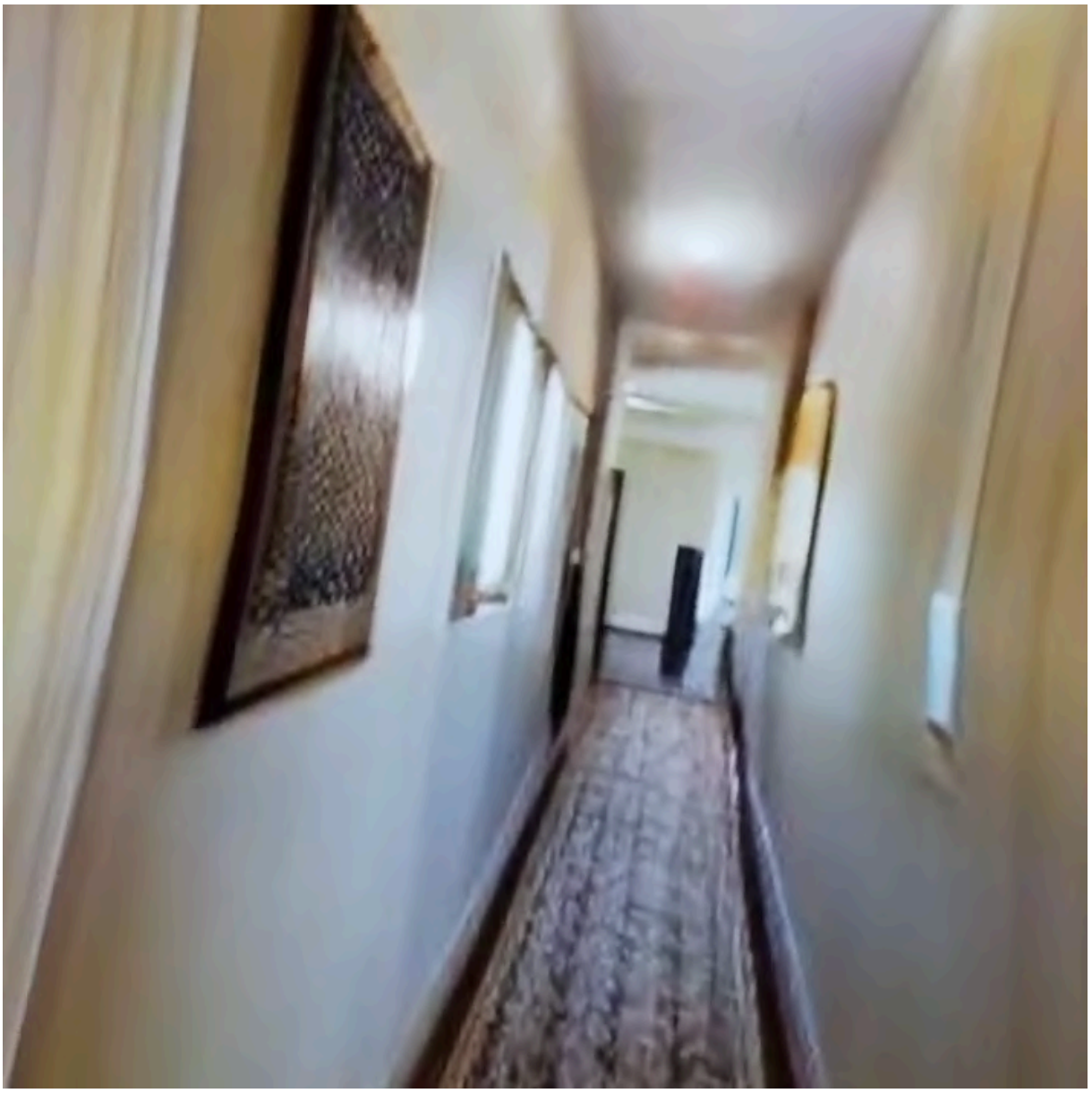
Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

a hallway in a house

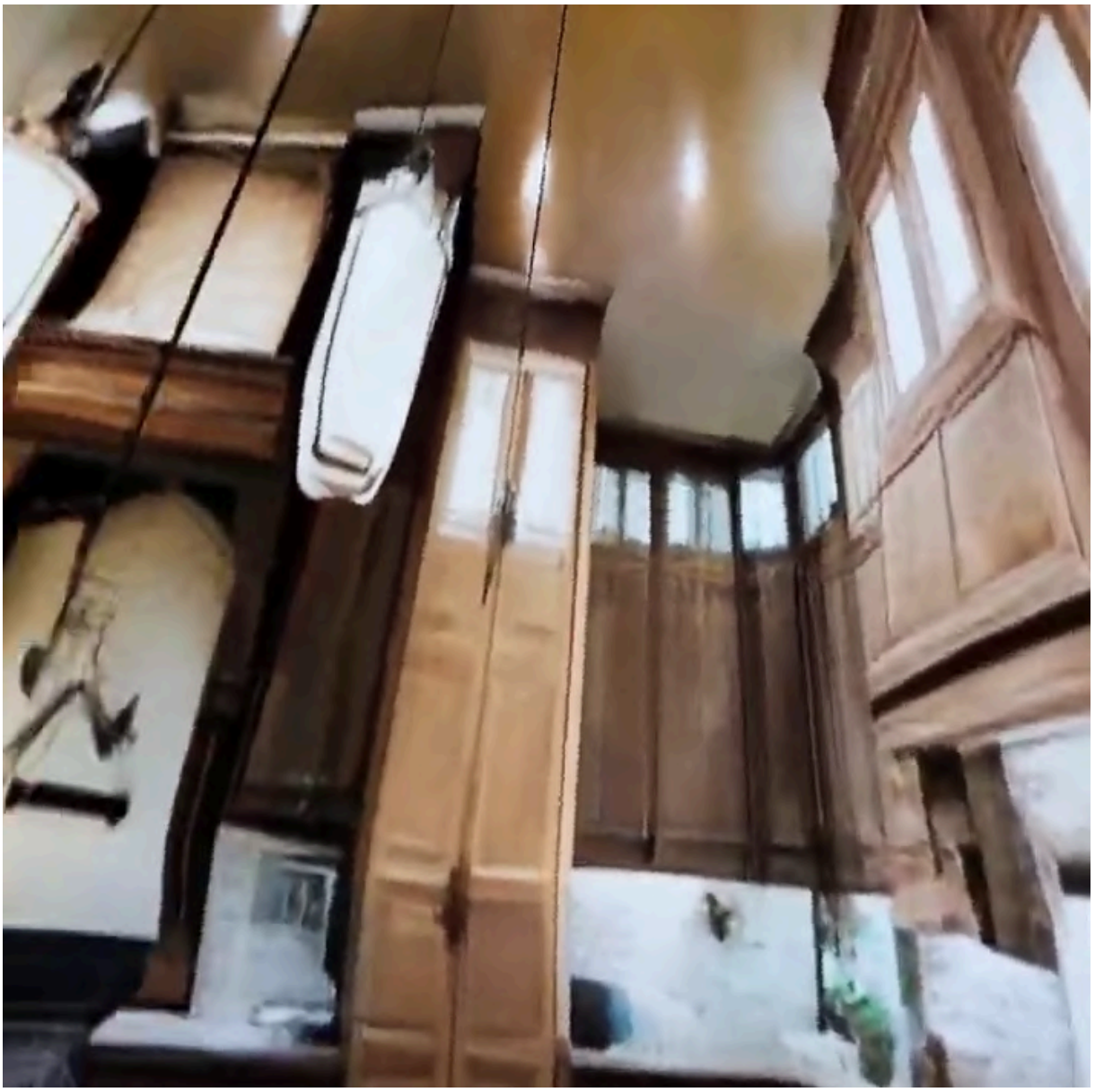
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

This kitchen is … countertop



Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Gaussian Splatting Playground in Unity

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 30 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Tentei me familiarizar com a Unity e continuei um pouco os testes para visualização dos Gaussian Splatting e NeRFs.

Unity

Testei um novo modelo chamado Text2Room, que tem a proposta de utilizar estimadores de profundidade, além da utilização de geração de imagens. Um método que eu não estava encontrando antes.

Text2Room - Meshlab.mkv

Além disso, consegui testar também os visualizadores, apesar de eles não serem para programação e interação direta, eles ajudam bastante com o processo de visualização, além de serem bem mais leves pois a maioria deles são na web, facilitando também a visualização sem a necessidade de uma máquina mais potente.

Os visualizadores testados foram o Meshlab no vídeo acima, além do:

WebGL viewer
- Kitchen.mp4
- AnimeLake.mp4
- Fantasy.mp4

SuperSplat
- Ruins.mp4
- Christmas.mp4

Spline
- Fantasy.mp4
- Christmas.mp4
- AnimeLake.mp4

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Separar alguns modelos que mais se destacaram. Quero começar a procurar um dataset pensando para eventualmente conseguir treinar um modelo.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

LEONARDO ALVES: Go!

Unity

Tutorial: Configurando o UnityGaussianSplatting

Pré-requisitos

1. **Unity Instalado**:
   - Use uma versão recente (Unity 2022 ou superior é recomendada).
2. **Clonar o Repositório**:
   - Tenha o Git configurado no seu sistema.
3. **Compatibilidade com GPU**:
   - Este plugin aproveita o GPU para desempenho ideal.

Passo 1: Clonar o Repositório

1. Abra um terminal ou use um cliente Git para clonar o repositório:

   ```
   git clone https://github.com/aras-p/UnityGaussianSplatting.git
   ```

2. Navegue até o diretório do repositório:

   ```
   cd UnityGaussianSplatting
   ```

Passo 2: Importar o Plugin no Unity

1. **Abra o Unity**:
   - Crie um novo projeto Unity ou abra um existente.
2. **Adicionar o Plugin**:
   - Copie a pasta do repositório clonado para dentro da pasta `Assets` do seu projeto Unity.
     - Exemplo: Coloque o conteúdo do repositório em `Assets/UnityGaussianSplatting`.
3. O Unity irá compilar automaticamente os scripts e recursos adicionados.

Passo 3: Configurar a Cena

1. **Criar uma Cena**:
   - Crie uma nova cena ou use uma existente.
2. **Adicionar um GameObject Gaussian Splatting**:
   - Vá até o diretório onde você importou o plugin.
   - Localize os exemplos ou scripts principais relacionados ao Gaussian Splatting.
   - Arraste o prefab ou script principal para a sua cena.

Passo 4: Carregar os Dados Gaussian

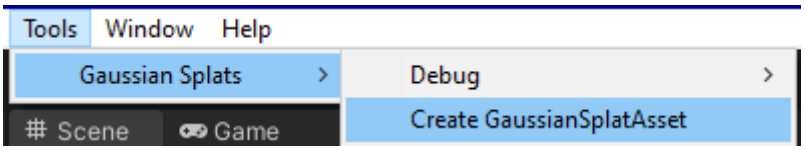
1. **Formatos de Arquivo**:
   - O plugin aceita arquivos preparados para Gaussian Splatting. Certifique-se de que os dados estejam no formato correto.
   - Exemplos de formatos:
     - Arquivos `.bin` ou similares, dependendo do suporte do repositório.
2. **Carregar Arquivo**:
   - Use as ferramentas do plugin para importar os dados Gaussian. Isso pode ser feito através de:
     - Scripts fornecidos no repositório.
     - Prefabs configurados com scripts para carregar os dados.

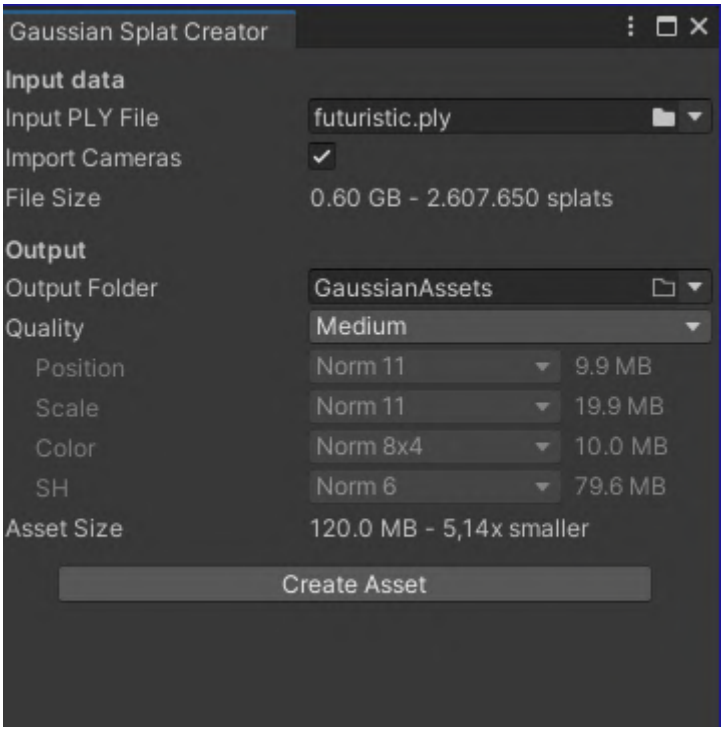
Passo 5: Ajustar os Parâmetros de Visualização

1. **Shader Gaussian Splatting**:
   - O repositório inclui shaders especializados para visualização de Gaussian Splats.

- Certifique-se de que o material associado ao GameObject está usando o shader correto.

2. **Configurações do Renderizador**:
   - Configure as propriedades do renderizador, como tamanho e opacidade dos splats.

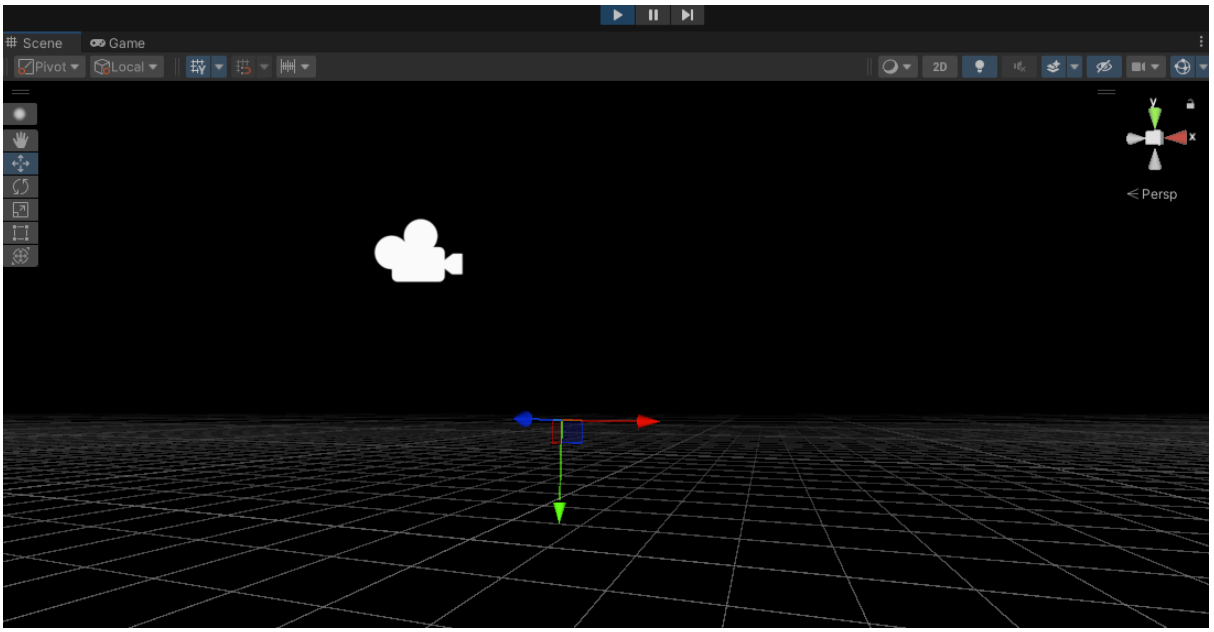




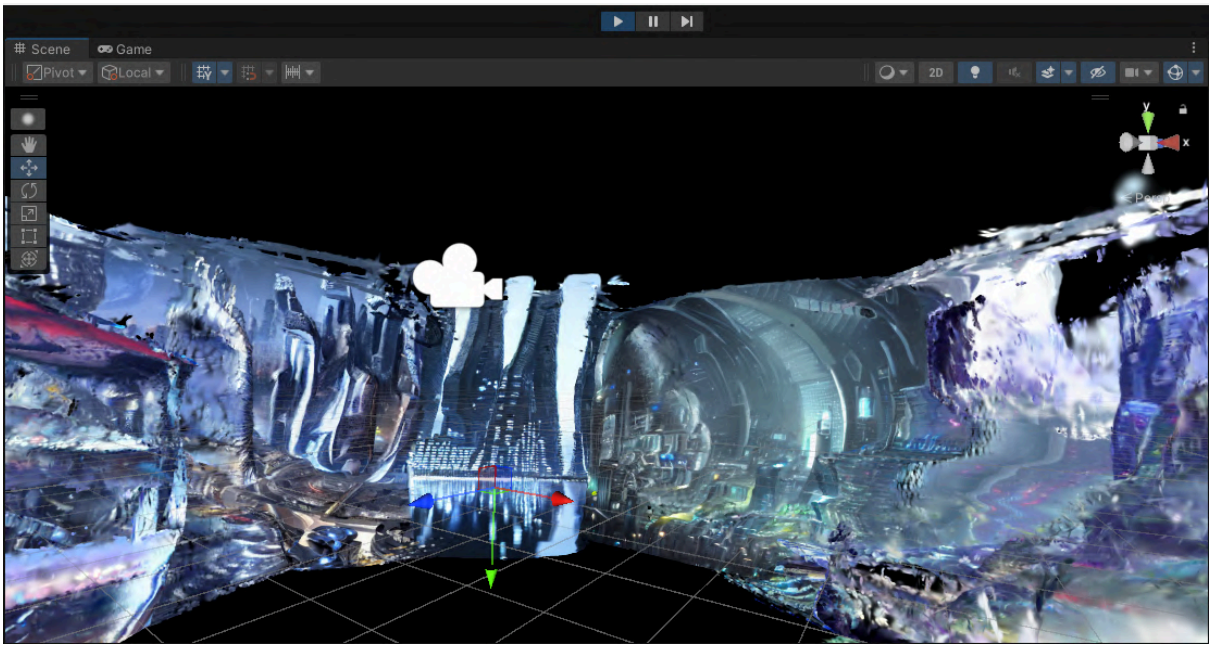
Dessa forma os seguintes arquivos são gerados:

Dessa forma é possível arrastar esse arquivo para o GameObject.

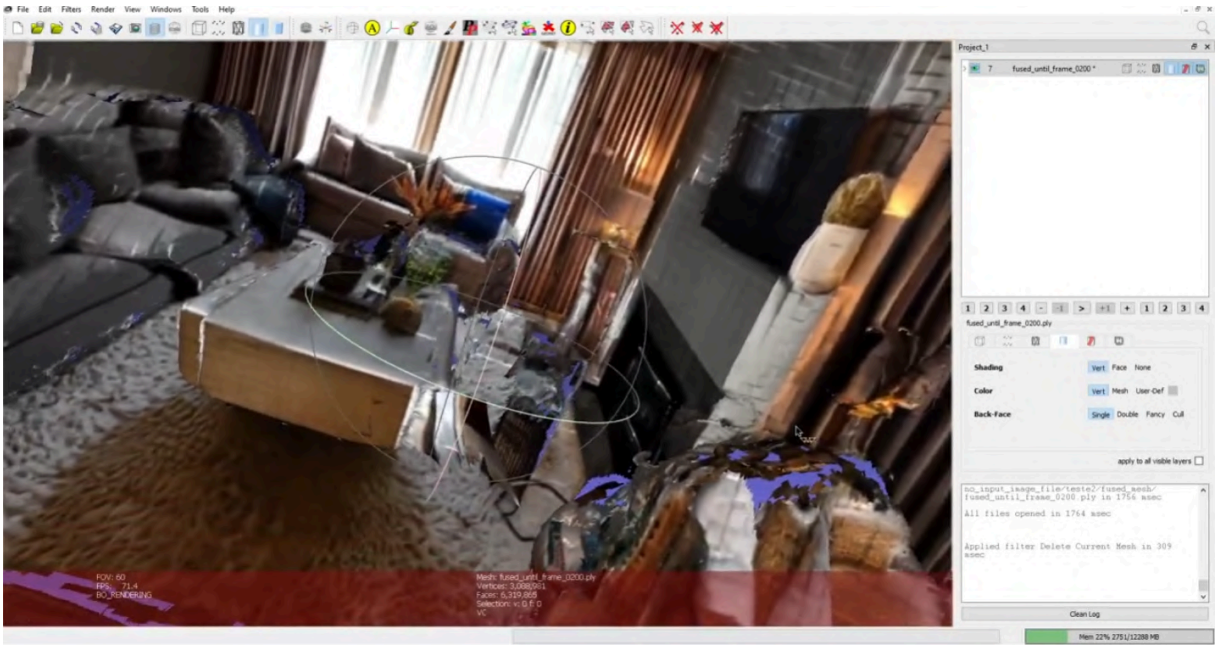
Antes:

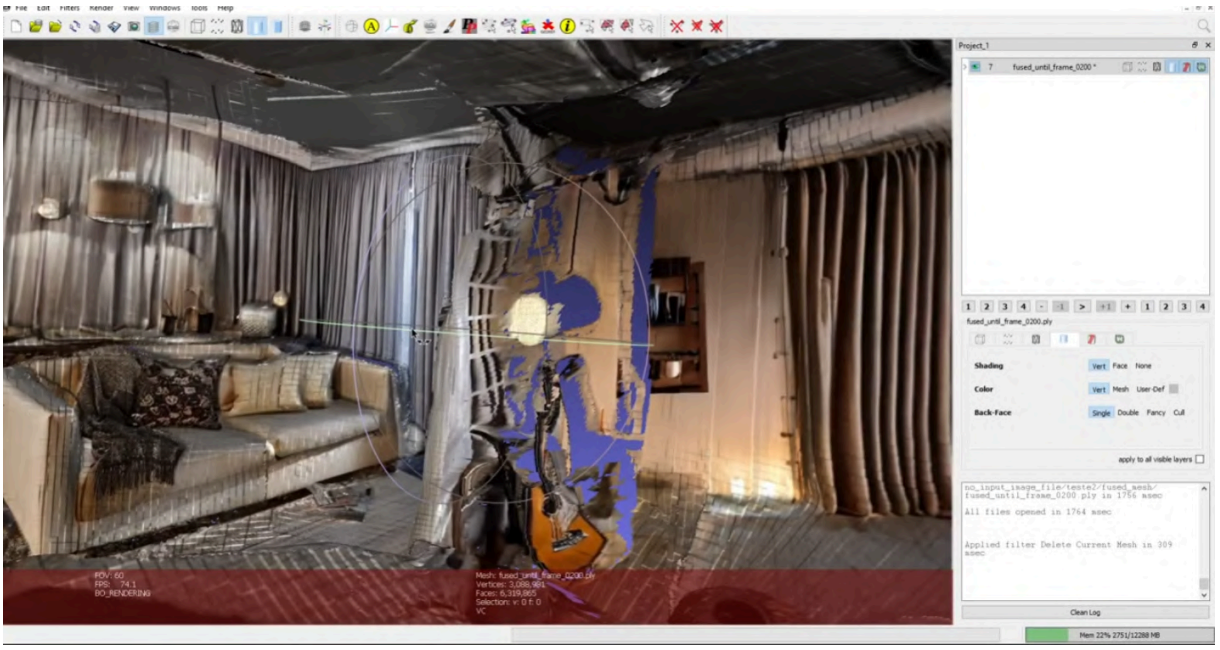


Depois:



Text2Room

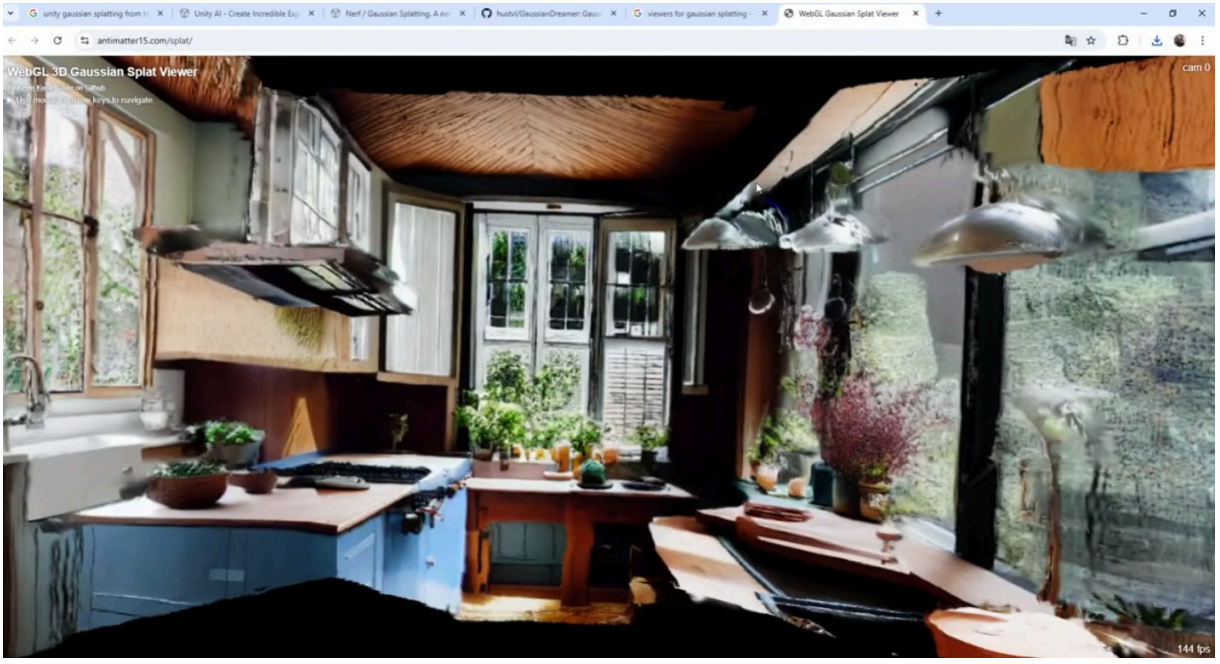


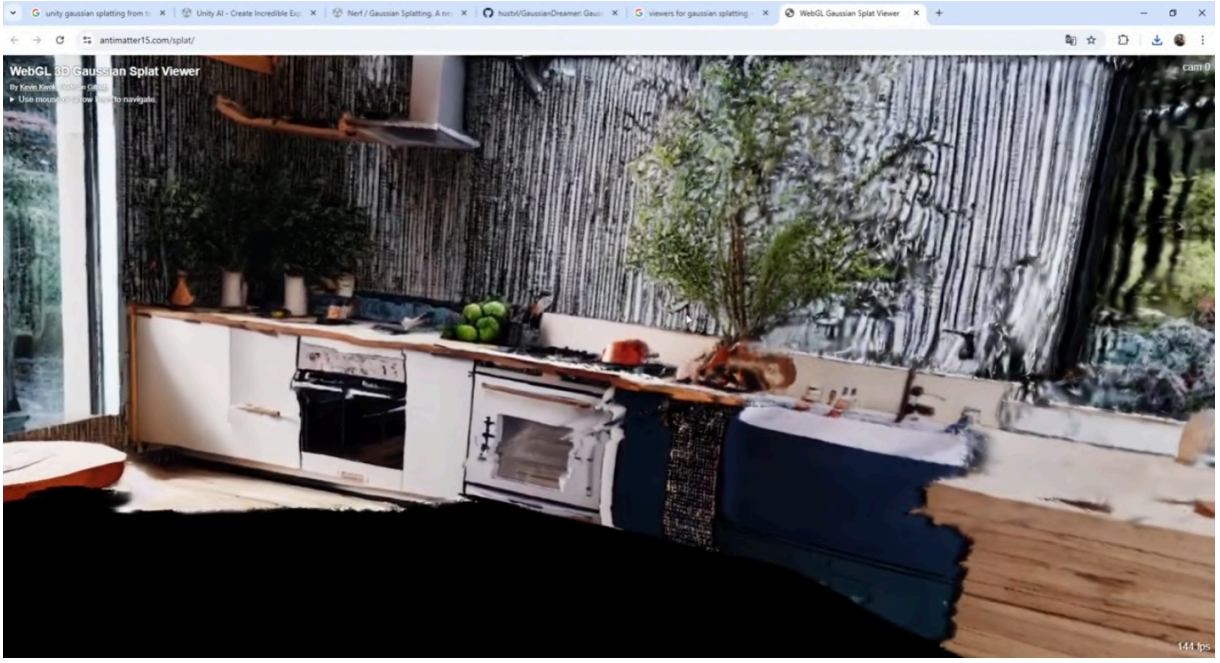


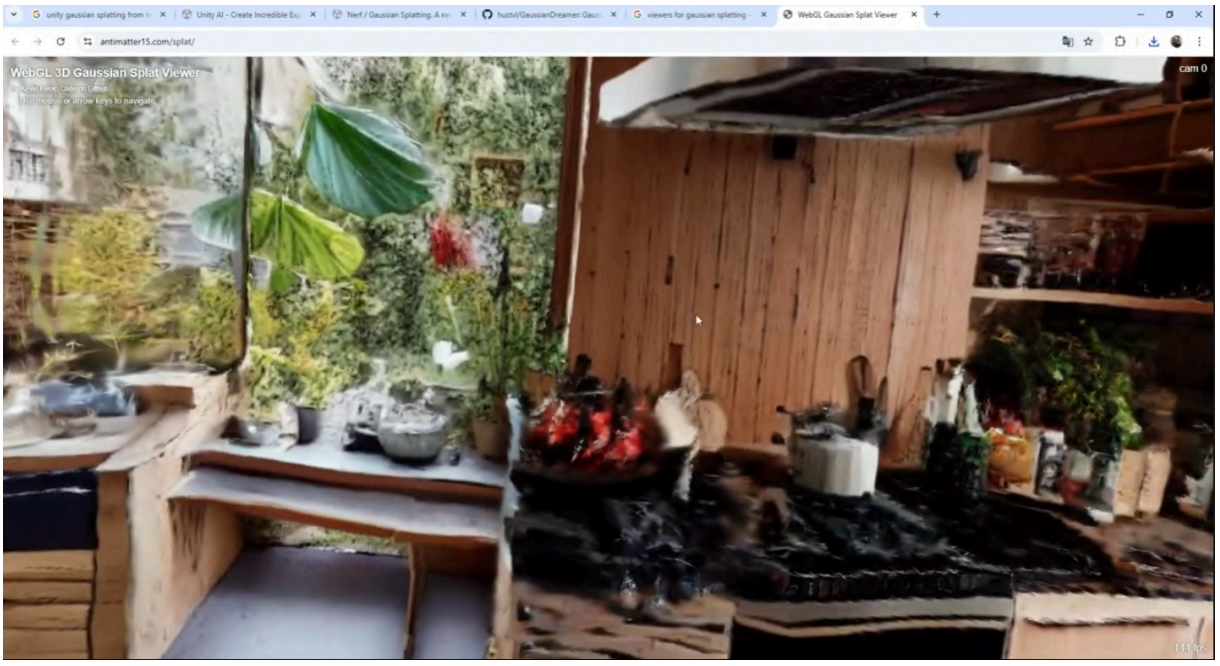
WebGL viewer

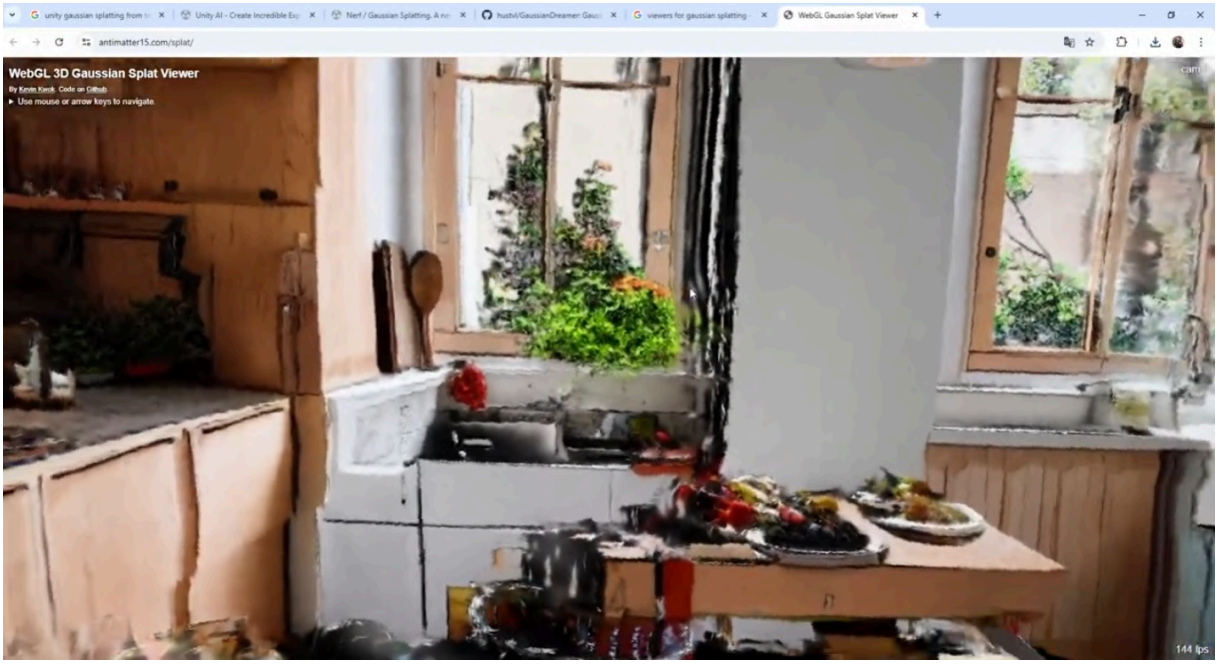
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Kitchen

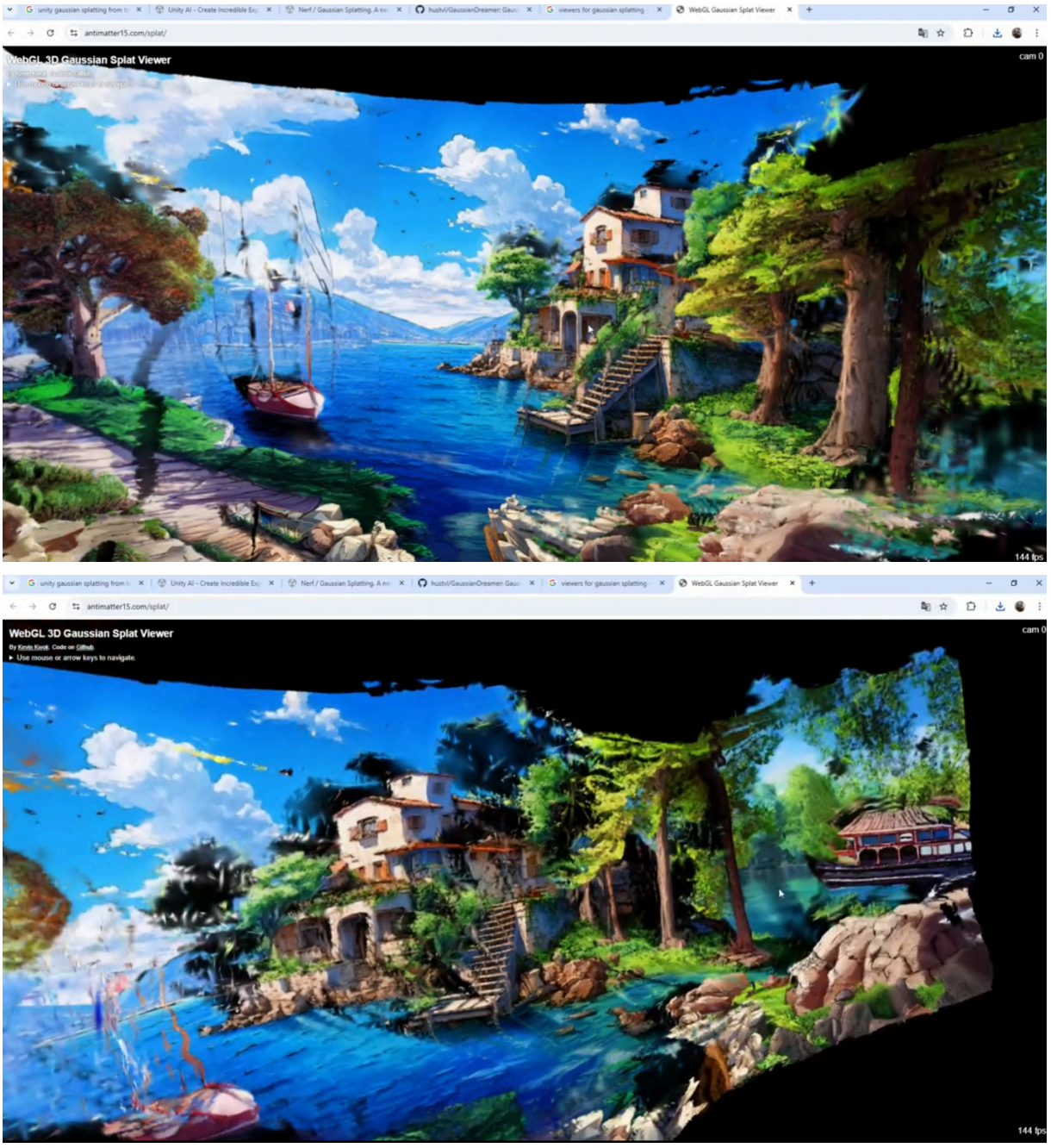

WebGL 3D Gaussian Splat Viewer
cam 0
144 fps


WebGL 3D Gaussian Splat Viewer
cam 0
144 fps

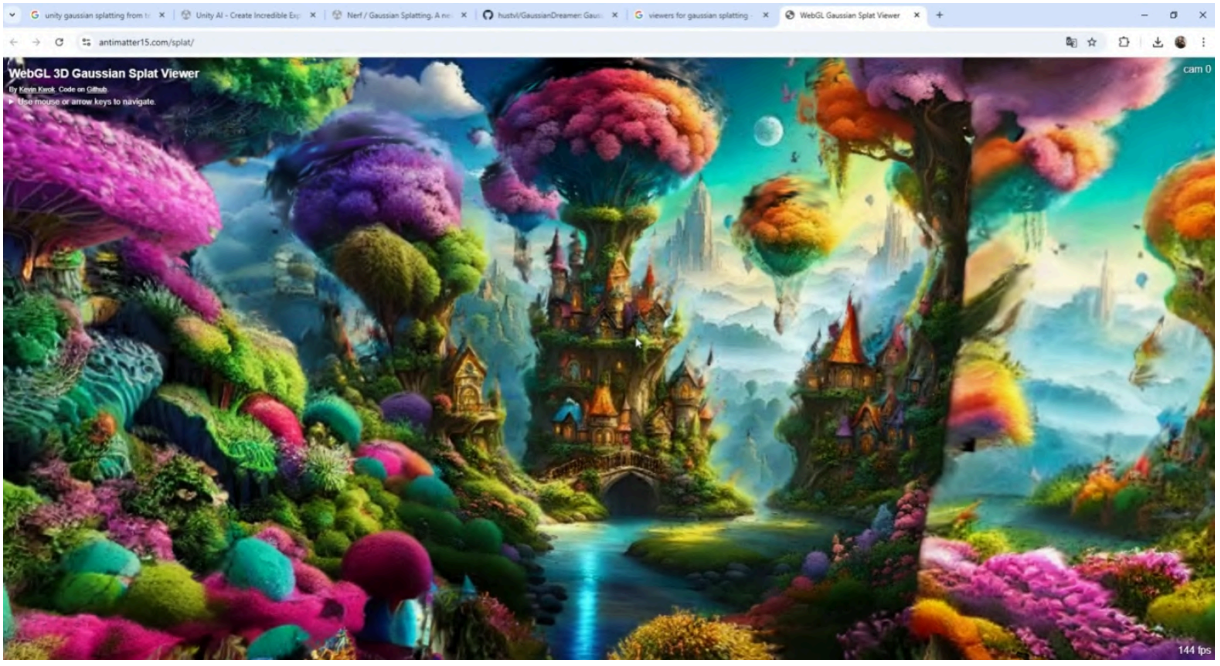

antimatter15.com/splat/
WebGL 3D Gaussian Splat Viewer
cam 0
144 fps

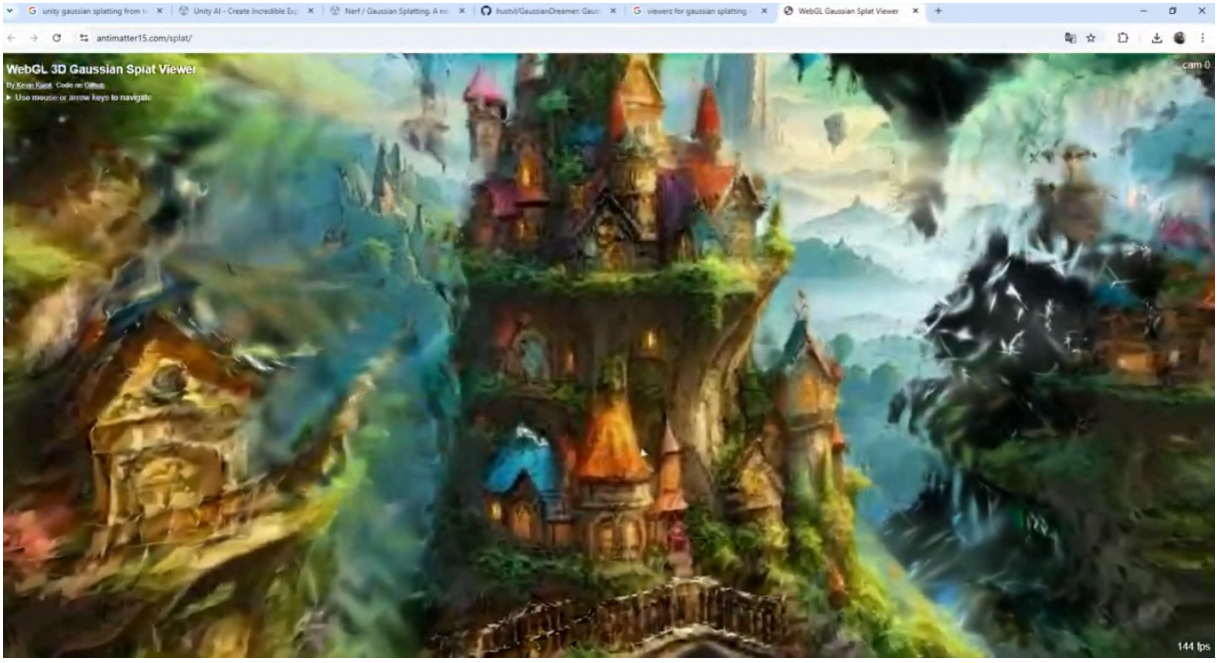

antimatter15.com/splat/
WebGL 3D Gaussian Splat Viewer
Use mouse or arrow keys to navigate.
144 fps

AnimeLake


antimatter15.com/splat/
WebGL 3D Gaussian Splat Viewer
cam 0
144 fps
antimatter15.com/splat/
WebGL 3D Gaussian Splat Viewer
By Kevin Kwok. Code on GitHub.
Use mouse or arrow keys to navigate.
cam 0
144 fps

Fantasy





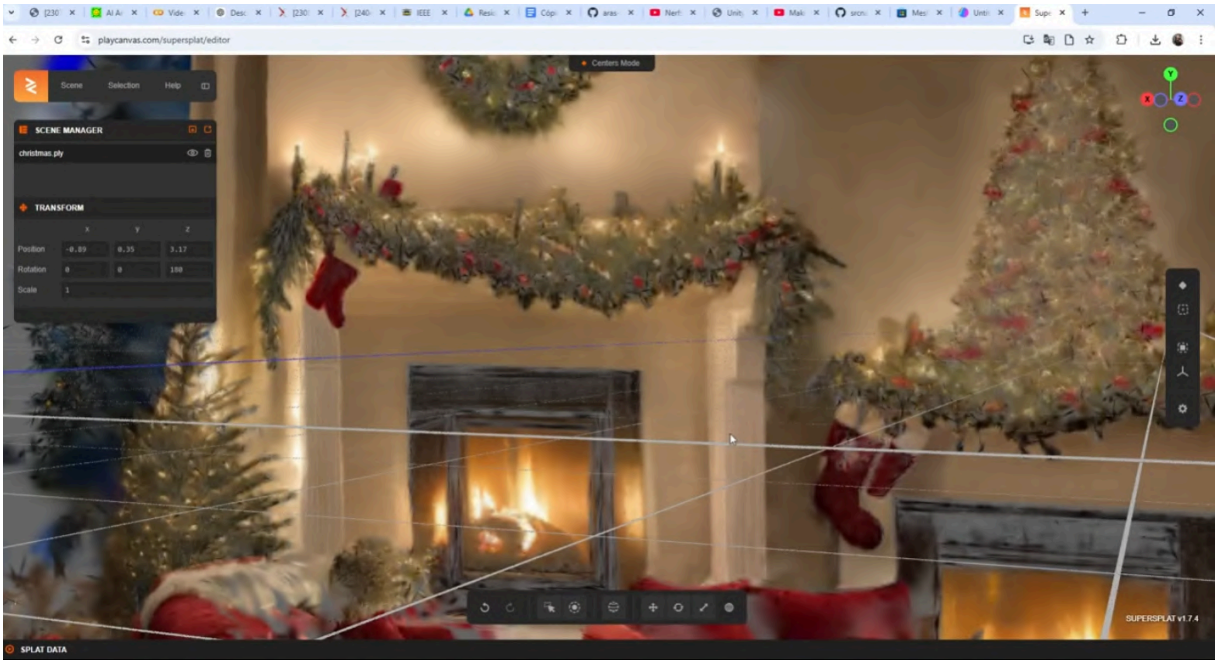
SuperSplat

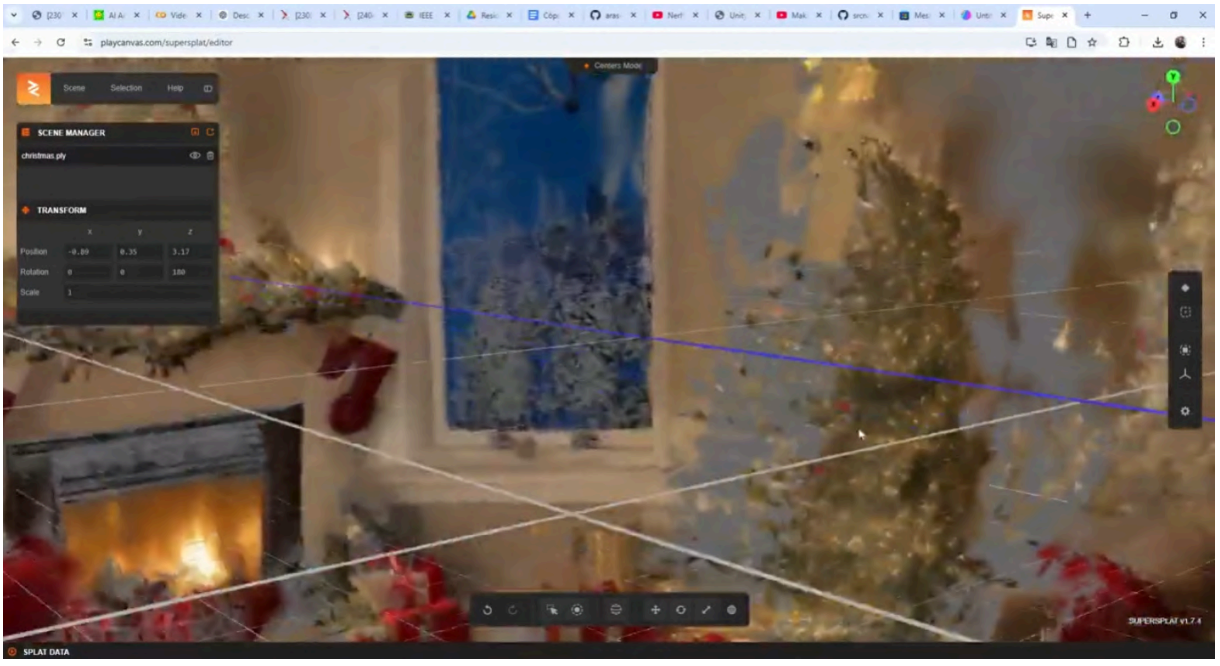
Ruins


playcanvas.com/supersplat/editor
Centers Mode
SCENE MANAGER
christmas.ply
ruin.ply
TRANSFORM
Position
Rotation
Scale
SUPERSPLAT v1.7.4
SPLAT DATA


playcanvas.com/supersplat/editor
Centers Mode
SCENE MANAGER
christmas.ply
ruin.ply
TRANSFORM
Position
Rotation
Scale
SUPERSPLAT v1.7.4
SPLAT DATA

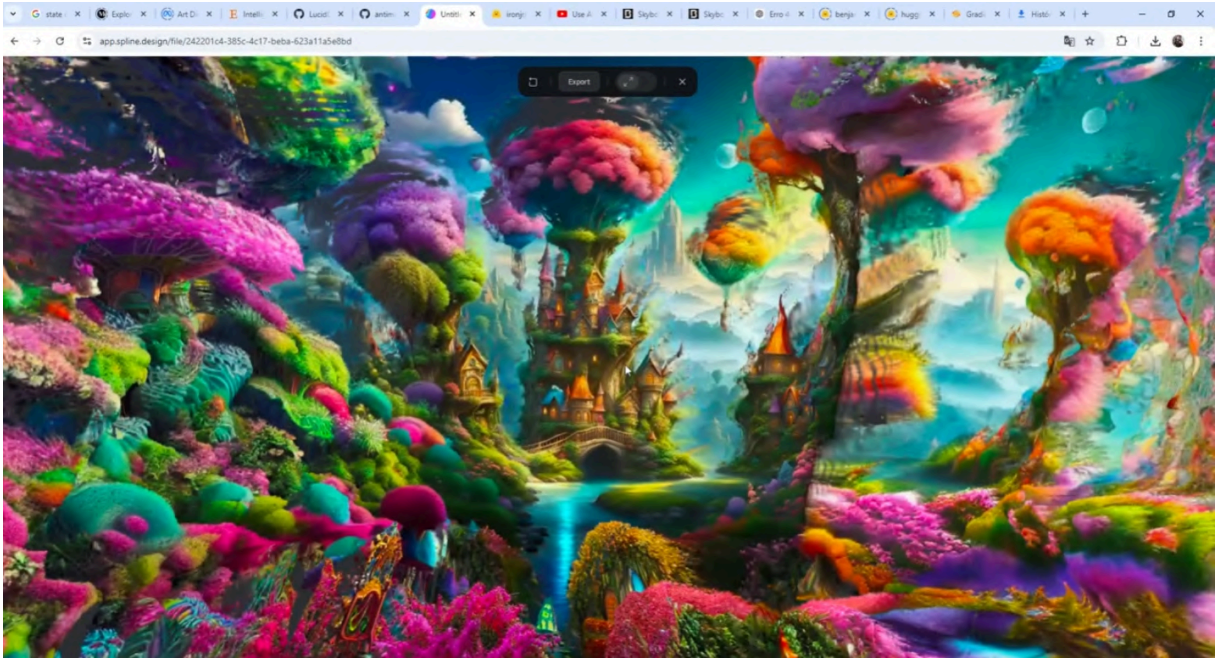
Christmas

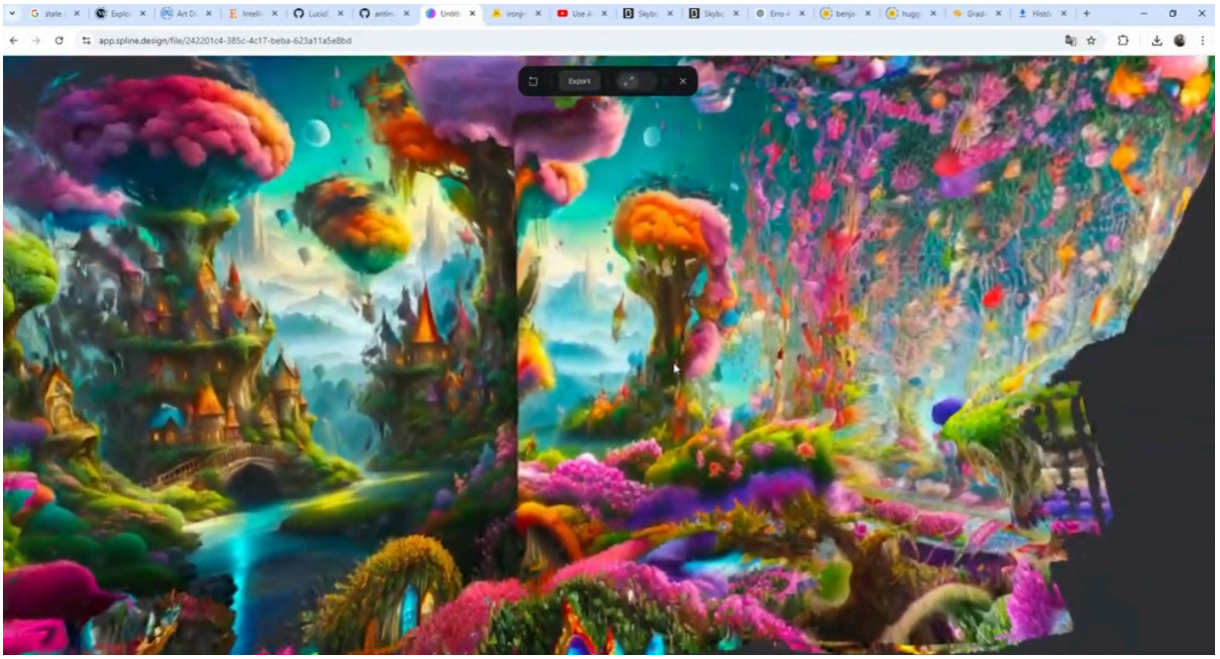




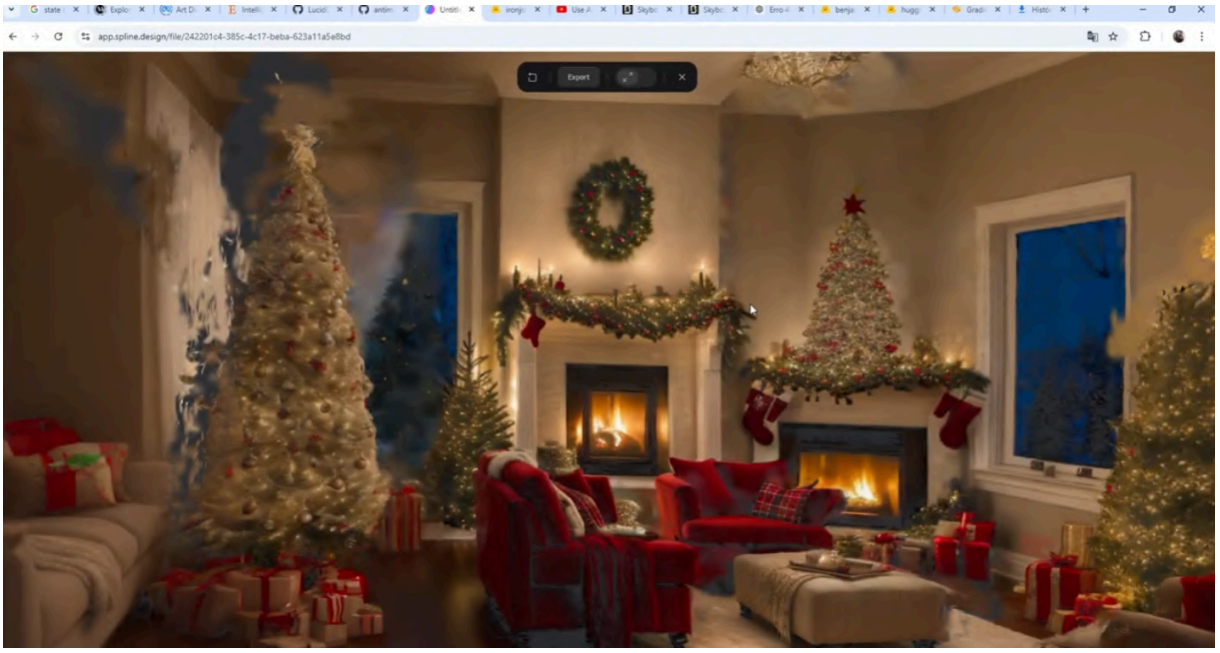
Spline

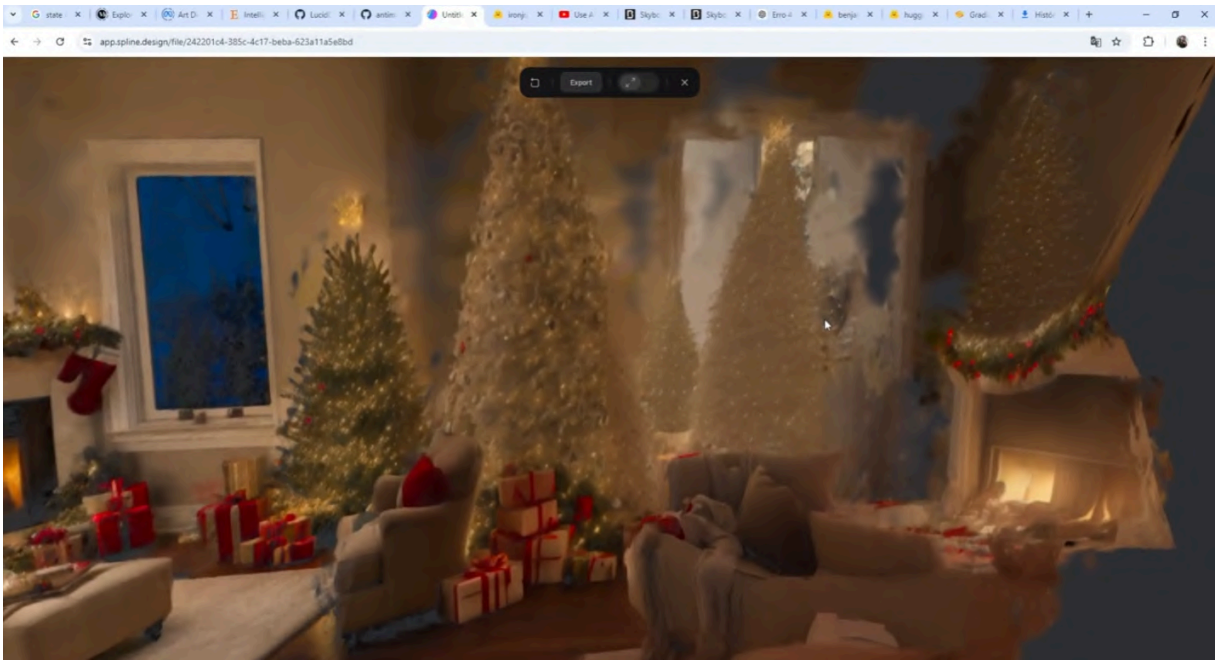
Fantasy

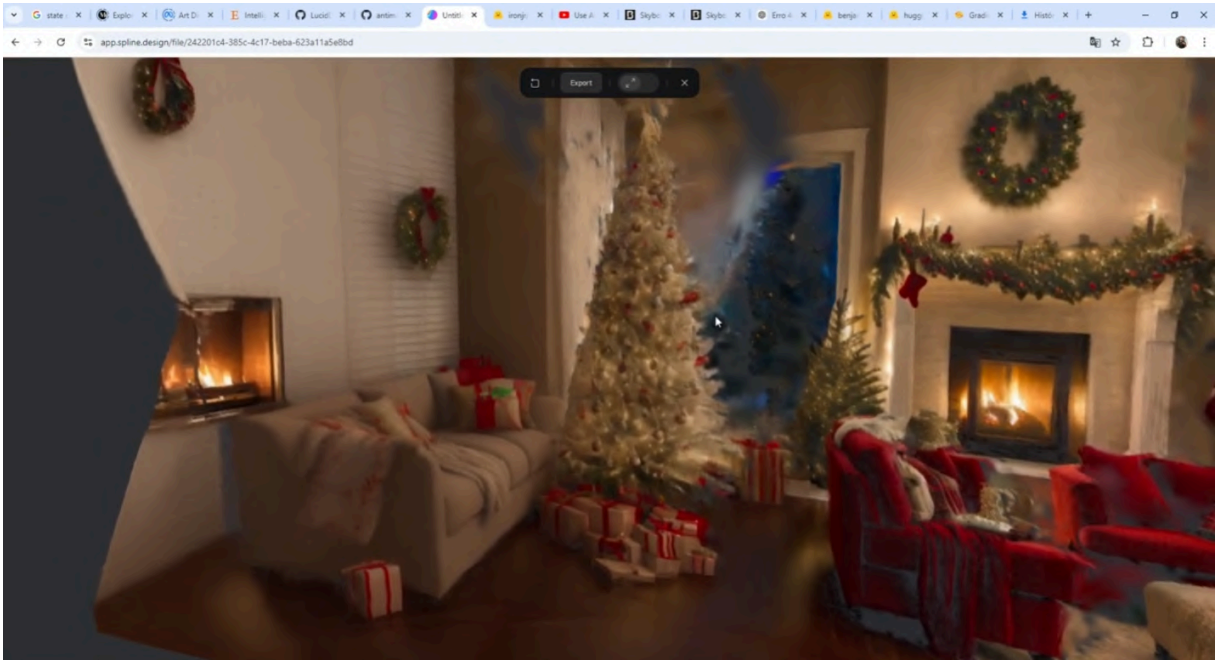

app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export


app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export

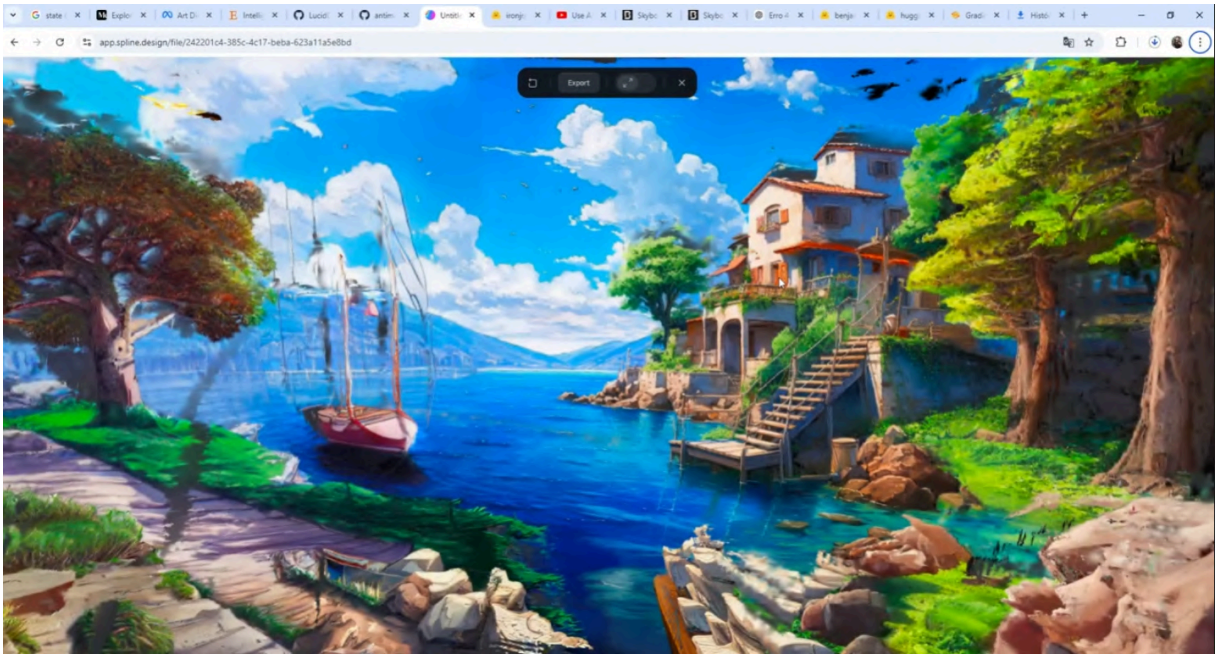
Christmas

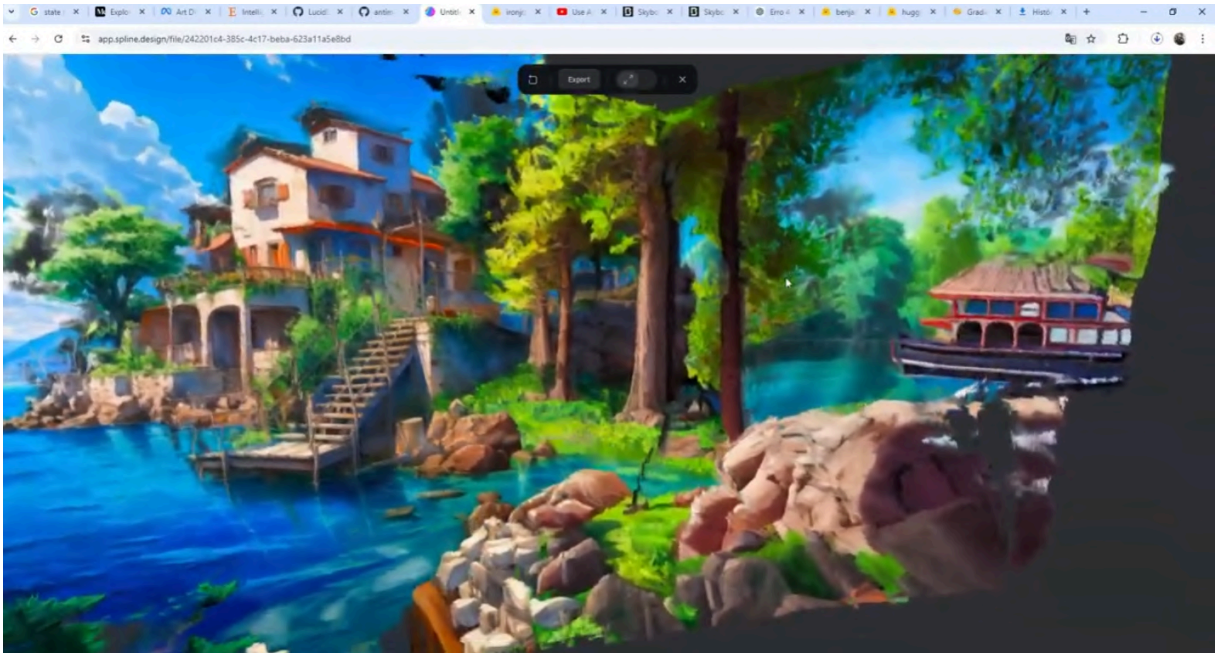

app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export

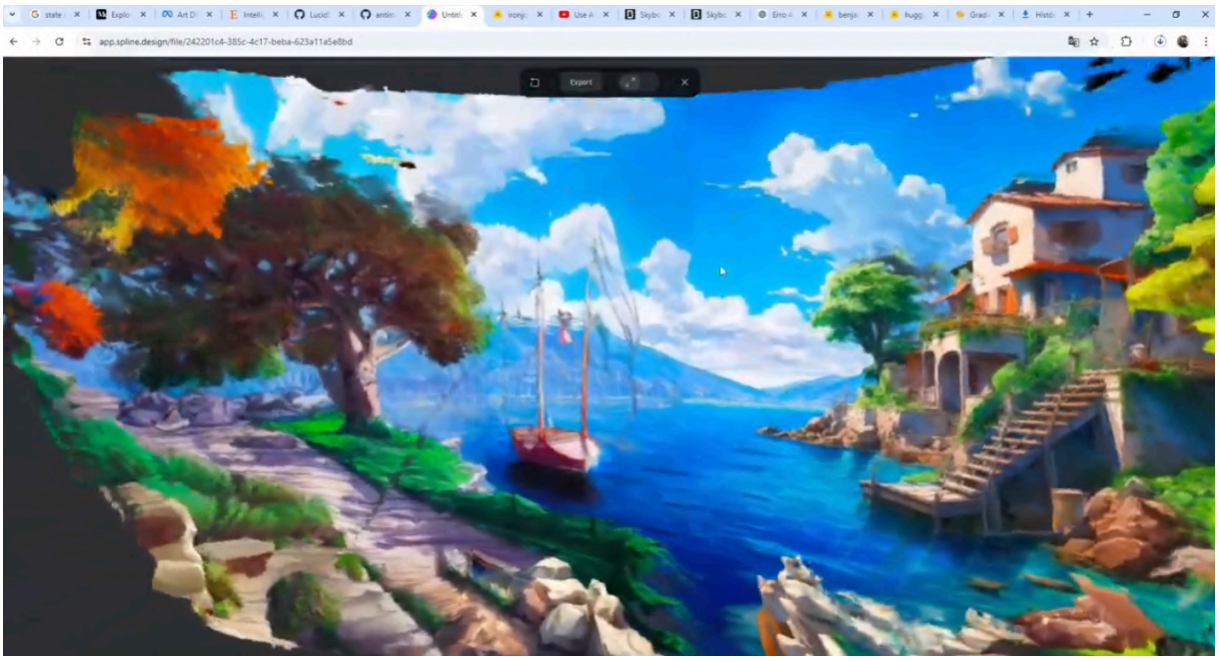

app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export


app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export

AnimeLake


app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export


app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export


app.spline.design/file/242201c4-385c-4c17-beba-623a11a5e8bd
Export

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 6 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Descrição da Pipeline
Modelos

Essa semana, organizei os modelos e datasets vistos ao longo das entregas para ter uma visão mais clara das possibilidades de fine-tuning direcionada a algum propósito. O que percebi é que a maioria dos datasets disponíveis são para indoor scenes. Ou seja, para cenários em apartamentos, casas e tudo mais.

Essa direção que estou considerando de realizar o fine-tuning em um modelo existente, como o DreamScene, que pode oferecer resultados promissores ao melhorar funcionalidades específicas ou adaptá-lo a datasets particulares. O fine-tuning pode permitir outputs mais personalizados, especialmente em ambientes que exigem uma geração precisa de cenários 3D a partir de prompts de texto.

Outra possibilidade é a integração de um desses modelos, como o LucidDreamer, diretamente com Unity ou Unreal. Essa integração criaria uma ponte entre as capacidades de geração 3D e os motores de renderização em tempo real, oferecendo novas possibilidades em aplicações interativas.

Pretendo explorar ambas as abordagens com mais profundidade para avaliar a viabilidade e os requisitos técnicos para a implementação, com o objetivo de expandir as aplicações práticas desses modelos em plataformas de renderização em tempo real.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Definir o caminho que será tomado, seja de um fine tuning ou então a implementação para um motor gráfico.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Descrição das Pipelines

Text2Nerf

O paper sobre Text2NeRF descreve uma pipeline para geração de cenários 3D a partir de prompts de texto, combinando Neural Radiance Fields (NeRF) com um modelo de difusão pré-treinado de texto para imagem. Aqui está um resumo direto da pipeline do modelo:

1. **Inicialização do Cenário**: O modelo começa gerando uma cena 2D inicial baseada no prompt de texto usando um modelo de difusão, complementado por um modelo de estimativa de profundidade para fornecer dados geométricos iniciais. Essa combinação forma a entrada base para o NeRF.
2. **Criação de Conjunto de Suporte**: Para evitar overfitting com dados de uma única visão, o modelo gera um conjunto de suporte com múltiplas visões usando renderização baseada em imagem de profundidade (DIBR), o que ajuda a restringir o modelo NeRF com visões adicionais de câmera.
3. **Inpainting Progressivo e Atualização**: As novas visões são sintetizadas progressivamente. Regiões faltantes em cada nova visão são preenchidas com inpainting baseado em texto a partir do modelo de difusão, garantindo consistência multi-visual. As visões preenchidas são então usadas para atualizar iterativamente o modelo NeRF.
4. **Alinhamento de Profundidade**: Uma estratégia de alinhamento de profundidade em duas etapas reduz inconsistências entre mapas de profundidade gerados em regiões sobrepostas entre visões. O alinhamento global corrige escala e offset, enquanto o alinhamento local, via rede neural, ajusta a precisão no nível de pixel.
5. **Objetivo de Treinamento**: O treinamento do NeRF utiliza perdas de RGB, profundidade e transmitância. As perdas de RGB e profundidade asseguram fidelidade nas cores e profundidade, enquanto a perda de transmitância promove coerência geométrica ao reduzir inconsistências de densidade.

O treinamento usa o modelo Stable Diffusion 2.0 para imagens iniciais e o LeReS para estimativa de profundidade, otimizando o NeRF para gerar cenários 3D fotorealistas e coerentes em múltiplas visões.

Text2Room

O modelo Text2Room cria malhas 3D de ambientes internos com texturas a partir de descrições em texto, usando modelos de texto-para-imagem em 2D e estimativa de profundidade em uma pipeline estruturada:

1. **Geração da Cena**: O processo começa com uma imagem 2D inicial gerada a partir do prompt de texto, que é projetada em 3D usando um estimador de profundidade monocular. Visões adicionais são sintetizadas iterativamente para expandir a cena, garantindo uma malha coesa.
2. **Seleção de Pontos de Vista em Duas Etapas**: Na primeira etapa, o modelo gera a estrutura principal do ambiente e os objetos selecionando pontos de vista predefinidos. Na segunda etapa, ele escolhe pontos de vista adicionais para preencher lacunas restantes, criando uma cena completa e sem buracos.
3. **Fusão de Malha**: As informações de profundidade e textura de cada visão são alinhadas e filtradas para evitar distorções. O modelo usa filtros para remover faces esticadas e suavizar bordas, garantindo consistência na geometria.
4. **Conclusão e Finalização**: Regiões não observadas são preenchidas usando inpainting baseado em texto, e a reconstrução de superfície de Poisson é aplicada para fechar pequenos espaços, resultando em uma malha à prova de vazamento pronta para renderização.

O modelo utiliza o Stable Diffusion para a geração de imagem a partir de texto e o IronDepth para estimativa de profundidade, com todo o processo levando cerca de 50 minutos por cena em uma GPU RTX 3090.

DreamScene

O modelo DreamScene utiliza uma abordagem inovadora para gerar cenas 3D a partir de descrições textuais, estruturada em uma pipeline focada em consistência, qualidade e flexibilidade de edição. Aqui está um resumo direto da pipeline:

1. **Amostragem de Padrão de Formação (FPS)**: Esta etapa utiliza a amostragem multi-timestep com filtros Gaussianos em 3D, otimizando a estrutura e textura de objetos. Esse processo permite criar representações 3D ricas em detalhes e semântica.
2. **Estratégia de Câmera em Três Etapas**:
    - **Etapa 1**: Gera uma representação inicial do ambiente (paredes internas ou ambientes externos distantes).
    - **Etapa 2**: Otimiza o chão e o solo, garantindo a integração entre objetos e o ambiente.
    - **Etapa 3**: Consolida a cena, garantindo consistência 3D em toda a visualização, com câmeras cobrindo a cena de forma uniforme.
3. **Integração de Objetos e Ambientes**: Objetos são colocados na cena com transformações específicas para evitar duplicação ou erros físicos, permitindo

flexibilidade na edição e ajustes individuais.
4. **Reconstrução de Texturas**: Técnicas de reconstrução rápida geram texturas plausíveis e detalhadas para superfícies, otimizando a criação de cenas de forma rápida e coesa.

O DreamScene usa modelos como o GPT-4 para decompor prompts textuais e o Point-E para criar nuvens de pontos iniciais, com um tempo de geração de cerca de 1 hora em uma GPU NVIDIA 3090.

LucidDreamer

O modelo LucidDreamer é uma pipeline para geração de cenas 3D de alta qualidade a partir de entradas variadas, como texto, imagens RGB e RGBD, utilizando um processo estruturado em duas etapas principais:

1. **Construção de Nuvem de Pontos**:
   - A partir de uma imagem inicial (ou de uma imagem gerada a partir de texto usando Stable Diffusion), é estimado um mapa de profundidade monocular. Os pixels da imagem e profundidade são "elevados" ao espaço 3D, formando uma nuvem inicial de pontos.
   - Em um processo iterativo de "sonhar" e "alinhamento", o modelo gera imagens adicionais em novos ângulos, usando inpainting para completar áreas visíveis e criar pontos adicionais, que são alinhados suavemente com a nuvem original para garantir consistência multi-angular.
2. **Renderização com Gaussian Splatting**:
   - Após a construção da nuvem de pontos, o modelo otimiza uma representação em Gaussian Splatting para preencher lacunas e gerar uma cena 3D mais realista e sem buracos.
   - Imagens re-projetadas são usadas para supervisionar o ajuste dos pontos Gaussianos, garantindo uma convergência rápida e um alto nível de detalhe.

Para a geração, o LucidDreamer usa Stable Diffusion para inpainting e ZoeDepth para profundidade, criando uma cena de alta qualidade e consistência em várias vistas, aplicável a estilos variados.

MVDiffusion

O modelo MVDiffusion introduz uma pipeline para gerar imagens consistentes em múltiplas vistas a partir de prompts textuais, com foco na geração de panoramas e imagens condicionadas por profundidade. Aqui está um resumo direto da pipeline:

1. **Arquitetura Multi-Branch com Atenção CAA**: O modelo usa várias cópias de um modelo de difusão (Stable Diffusion) em paralelo, com um mecanismo de atenção "correspondência-aware" (CAA) inserido em cada bloco do UNet para promover a consistência entre vistas.

2. **Geração de Panorama**: Para criar panoramas, o modelo gera múltiplas imagens com campo de visão de 90° que se sobrepõem. Cada uma dessas imagens é criada simultaneamente com o modelo de difusão e unidas por meio do CAA para manter a consistência entre bordas e evitar erros de acumulação.
3. **Geração de Imagens a partir de Profundidade**: Na tarefa de depth-to-image, o modelo cria imagens RGB a partir de entradas de profundidade, ajustando o conteúdo visual de acordo com os dados geométricos para manter consistência multi-visual.
4. **Módulo de Interpolação**: Um módulo adicional de interpolação é utilizado para gerar frames intermediários entre pares de imagens chave, facilitando a criação de sequências mais detalhadas.

O treinamento é feito em duas etapas: inicialmente, o modelo é ajustado em dados de uma única vista, e na segunda etapa, o bloco CAA é ativado e refinado para garantir a consistência em múltiplas vistas.

PanFusion

O modelo PanFusion utiliza uma arquitetura de difusão de dois ramos para gerar panoramas 360° de alta qualidade a partir de prompts textuais, abordando problemas de consistência e distorção em imagens panorâmicas. Aqui está um resumo da pipeline:

1. **Modelo de Difusão de Dois Ramos**: O PanFusion possui dois ramos – um ramo panorâmico, que fornece orientação de layout global e cria o panorama final, e um ramo de perspectiva, que explora a geração detalhada de imagens em perspectiva. Ambos os ramos compartilham uma arquitetura UNet com camadas LoRA para adaptação de resolução.
2. **Mecanismo de Atenção de Projeção Equiretangular-Perspectiva (EPPA)**: Para manter a integridade geométrica e consistência entre os ramos, o EPPA projeta informações entre os ramos, usando um codificador posicional esférico e uma máscara de atenção para alinhar detalhes.
3. **Condição de Layout**: O PanFusion permite geração condicionada pelo layout ao aplicar um mapa de distância do layout como entrada de um ControlNet, garantindo que o panorama siga as restrições espaciais do layout.

Durante o treinamento, o modelo usa o dataset Matterport3D e adota uma estratégia de inicialização conjunta de mapa latente para sincronizar a geração entre os ramos. Essa abordagem assegura a consistência no panorama, minimizando distorções e duplicação de elementos.

Modelos

MODELOS		TIPOS	INTERMEDIÁRIO	DATASET UTILIZADO
Text2Nerf	https://github.com/eckertzhang/Text2NeRF	Text-to-3D	Diffusion - Depth Estimator	
Text2Room	https://lukashoel.github.io/text-to-room/	Text-to-3D	Diffusion - Depth Estimator	ScanNet
DreamScene	https://github.com/DreamScene-Project/DreamScene	Text-to-3D Gaussian	Diffusion - Point Cloud	Utiliza Stable Diffusion
LucidDreamer	https://github.com/luciddreamer-cvlab/LucidDreamer	Text-to-3D Gaussian	Diffusion - Point Cloud	Utiliza Stable Diffusion
MVDiffusion	https://github.com/Tangshitao/MVDiffusion	Text-to-Panorama	Diffusion - Inpainting	
PanFusion	https://github.com/chengzhag/PanFusion	Text-to-Panorama	Diffusion - Inpainting	
DATASETS				
3D-FRONT	https://tianchi.aliyun.com/specials/promotion/alibaba-3d-scene-dataset			
ARIA	https://www.projectaria.com/datasets/ase/			
Replica	https://github.com/facebookresearch/Replica-Dataset			
ARKitScenes	https://github.com/apple/ARKitScenes			
ScanNet	http://www.scan-net.org/			
Matterport3D	https://niessner.github.io/Matterport/			
Outros	https://github.com/alelopes/awesome-rgbd-datasets?tab=readme-ov-file			

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 13 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Eu deveria durante essa Semana escolher um caminho, que estava entre um fine tuning ou uma implementação para um motor gráfico. Depois de uma conversa com o Professor Leonardo, decidi focar na implementação com o motor gráfico, mas acabei enfrentando uma lista enorme de erros. Algumas abordagens que testei foi passar o código para C++/C#, que são os aceitos pela Unity, mas logo percebi que se eu fosse utilizar o modelo generativo de dentro da Unity, eu enfrentaria diversos problemas de recursos de hardware. Já que os modelos de Gaussian Splatting são pesados.

Uma segunda abordagem que testei foi utilizando requisições HTTPS, eu coloquei o modelo para rodar em um servidor remoto, e coloquei a interface gráfica que o LucidDreamer disponibiliza para conseguir fazer essas requisições, mas acabei enfrentando mais dificuldades, uma vez que a Unity estava tendo problemas para conseguir fazer essas requisições.

Uma terceira abordagem foi testar a mesma ideia de requisições HTTPS mas dessa vez na Unreal Engine, mas enfrentei problemas dessa vez foi com os plugins para conseguir mostrar os modelos 3D. Os plugins não são facilmente modeláveis igual na Unity.

Uma decisão que tomei foi tentar simplificar ainda mais essa abordagem, o que resultaria em um workflow fora da Unity, mas sendo ela o destino final para a visualização do cenário 3D. Eu consegui mudar o workflow para que ele conseguisse baixar o .ply resultante e adicionasse ao Unity.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Prosseguir nos testes e aproveitar essa Semana para concretizar o workflow.
Pensar em possíveis aprimoramentos para o workflow.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Uma correção que eu queria fazer também, foi que eu havia dito que talvez seria possível fazer uma

aplicação em tempo real, mas os modelos demoram um tempo considerável para conseguir gerar o cenário 3D (cerca de 5 minutos).

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 28 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Relembrando:

Integração do modelo Lucid Dreamer em um workflow dentro da Unity.
Essa integração poderia ser feita de diversas formas, como o modelo é muito pesado, ou seja, exige mais de 12 GB de VRAM, não seria possível rodar localmente na minha máquina. Algumas abordagens para contornar isso:
Abordagens Unity

Progressos da Semana:

Atualmente o workflow é esse: Workflow_Atual.png

As interações estão todas de acordo com o esperado, podem ser visualizadas no vídeo:
LucidDreamer_Unity.mp4

Depois de integrar uma etapa de Stable Diffusion o usuário já consegue gerar o cenário 3D a partir de um único prompt, a ideia agora é tentar otimizar ainda mais, colocando uma etapa de STT para que o Workflow final seja alcançado:

Workflow_Completo.png

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Terminar o workflow, integrando STT
Speech-to-text
Speech to Text with OpenAI Whisper in Unity!

ou por uma entrada de texto digitada dentro da aplicação.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Abordagens Unity

A primeira ideia de utilizar a Unity seria passando o modelo para rodar nativamente na Unity ou então rodando localmente, mas o modelo escolhido Lucid Dreamer (https://github.com/luciddreamer-cvlab/LucidDreamer) precisa de 12 GB de VRAM, então rodar localmente ou na Unity fica muito custoso. Apesar disso, um tutorial que eu encontrei seria assim:

Integrando um Modelo Keras na Unity

A integração de um modelo de IA Keras na Unity pode adicionar funcionalidades avançadas de IA aos seus jogos e experiências imersivas. O **Keras.NET** é uma biblioteca .NET que permite que você carregue e use modelos Keras diretamente na Unity, oferecendo uma forma eficaz de aproveitar modelos de aprendizado de máquina em ambientes de jogos.

Visão Geral do Keras.NET

O **Keras.NET** é uma ponte que permite aos desenvolvedores Unity usar modelos de aprendizado de máquina treinados em Keras (que utiliza TensorFlow como backend) diretamente dentro de seus projetos. Com o Keras.NET, você pode:

- Carregar modelos pré-treinados salvos em arquivos `.h5`.
- Executar inferências diretamente no Unity sem precisar de servidores externos ou APIs REST.
- Integrar modelos de IA com outras funcionalidades Unity para decisões em tempo real, controle de personagens, geração de conteúdo e mais.

Criando uma Implementação com Keras.NET: Procedimento Geral

Para integrar um modelo de IA Keras na Unity usando Keras.NET, siga estas etapas:

1. Preparar o Modelo de IA Keras

Certifique-se de que seu modelo Keras esteja treinado e salvo em um arquivo `.h5`.
python
Copiar código

```
from keras.models import Sequential

# Exemplo de criação de um modelo simples
model = Sequential()
# (Adicione camadas aqui)

# Salvar o modelo
model.save('caminho/para/o/modelo.h5')
```

-
- Transfira o arquivo `.h5` para um diretório acessível pela Unity.

2. Configurar o Ambiente Unity para Keras.NET

- **Adicionar Keras.NET ao Projeto Unity:**
  - Baixe o **Keras.NET** do NuGet e adicione os binários ao projeto Unity.
  - Alternativamente, você pode usar o Visual Studio para instalar a biblioteca diretamente no projeto, facilitando a gestão de dependências.

3. Integrar o Modelo Keras no Unity

- Crie um script C# para carregar e executar o modelo na Unity:

Importe as bibliotecas necessárias:
csharp
Copiar código

```
using Keras.Models;
using Numpy;
using UnityEngine;
```

1.

Carregue o modelo Keras em um script Unity:
csharp
Copiar código

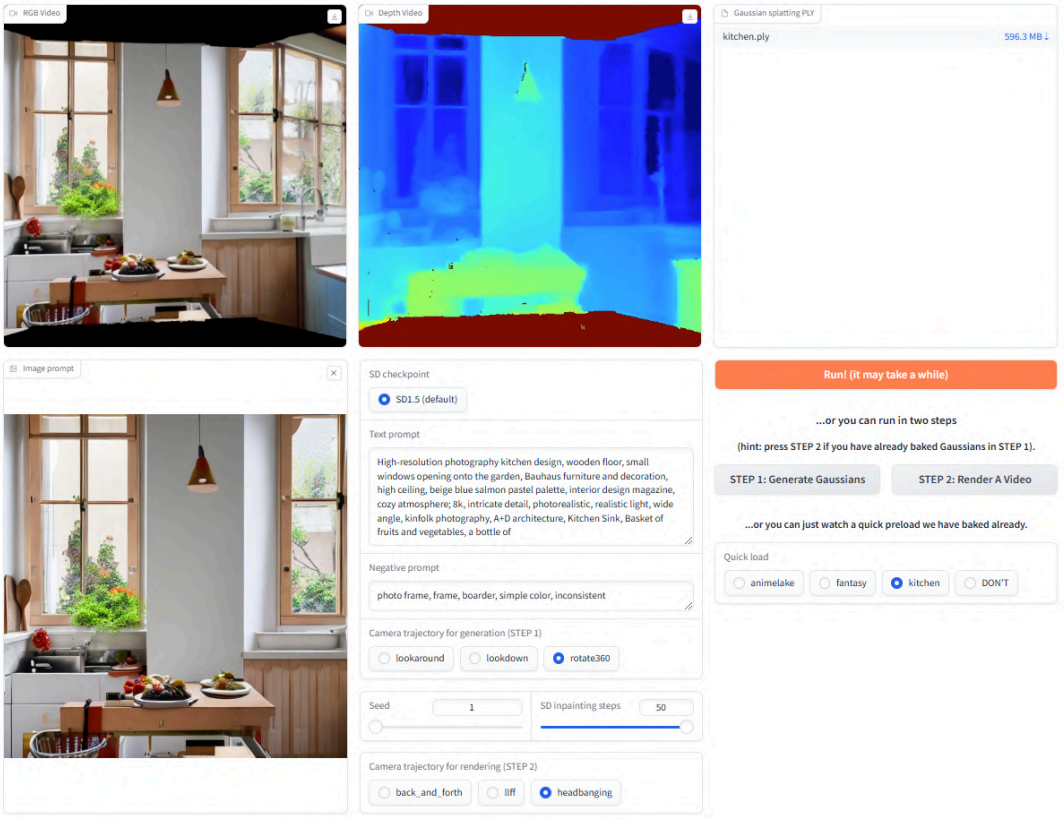
```
public class KerasIntegration : MonoBehaviour
```

```
{
    private BaseModel model;

    void Start()
    {
        // Carregar o modelo Keras salvo
        model = Sequential.LoadModel("caminho/para/o/modelo.h5");
        Debug.Log("Modelo Keras carregado com sucesso!");
    }

    public void Predict(float[] inputData)
    {
        // Converter os dados de entrada em um tensor numpy
        var npArray = np.array(inputData);

        // Realizar a previsão
        var prediction = model.Predict(npArray);

        // Exibir o resultado
        Debug.Log("Predição: " + prediction);
    }
```

2. }

VIA HTTPS

O modelo atual gera uma página do Gradio, onde é possível interagir, gerando os cenários a partir de uma imagem como entrada e um prompt para auxiliar no restante da geração



O resultado são alguns videos para mostrar o cenário 3D gerado além do arquivo .ply

A primeira mudança realizada foi a necessidade de uma imagem de entrada, mudando para que apenas um texto seja necessário através de uma adição de uma etapa do Stable Diffusion para gerar essa imagem inicial.

Após isso, diversas tentativas foram feitas de tentar fazer a Unity acessar esse Gradio por meio de requisições HTTPS, porém nenhuma delas teve sucesso. Encontrando o seguinte erro:

“Failed to download file: HTTP/1.1 403 Forbidden UnityEngine.Debug:LogError (object) Download3DModel/<DownloadFileCoroutine>d__2:MoveNext () (at Assets/Download3DModel.cs:31) UnityEngine.SetupCoroutine:InvokeMoveNext (System.Collections.IEnumerator,intptr)”

Mostrando que algumas permissões estavam sendo negadas através do Gradio, mesmo fazendo uma integração com flask.

Tutorial: Integração do Pacote SSH na Unity

Outra abordagem para conseguir acessar o modelo a partir da Unity é por SSH, um simples tutorial é descrito a seguir:

Pré-requisitos

1. **Unity instalado:** Tenha uma versão do Unity configurada no seu ambiente.
2. **Conhecimento básico em C#:** Você trabalhará com scripts em C#.
3. **Bibliotecas externas:** Vamos usar uma biblioteca SSH, como `SSH.NET`.

Passo 1: Instalar o pacote SSH.NET

O pacote `SSH.NET` é uma biblioteca popular e de código aberto para integração de SSH em projetos C#.

1.1. Adicionar o pacote via NuGet (Recomendado)

1. Abra seu projeto Unity.
2. Instale o pacote `SSH.NET` no Unity:
    - Se estiver usando o Visual Studio como IDE, abra o **Gerenciador de Pacotes NuGet**:
        - Clique com o botão direito no projeto no Visual Studio.
        - Selecione **Gerenciar Pacotes NuGet**.
        - Pesquise por `SSH.NET` e instale a biblioteca.
    - Alternativamente, baixe o pacote diretamente da página do SSH.NET no NuGet.

1.2. Adicionar manualmente (se necessário)

1. Baixe o arquivo `.dll` da biblioteca SSH.NET no GitHub: SSH.NET no GitHub.
2. Copie o arquivo `.dll` para a pasta **Assets/Plugins** no seu projeto Unity.

Passo 2: Criar o Script de SSH

Agora, vamos configurar um script em C# para conectar-se a um servidor via SSH.

2.1. Criar o script básico

1. Na Unity, vá até a pasta **Assets** e crie um novo script:

    - Clique com o botão direito > **Create** > **C# Script**.
    - Nomeie o script como `SSHManager`.

2. Adicione o código abaixo no script:

```
using System;
using Renci.SshNet; // Biblioteca SSH.NET

public class SSHManager
{
    private SshClient _sshClient;

    public SSHManager(string host, int port, string username, string password)
    {
        _sshClient = new SshClient(host, port, username, password);
    }

    public void Connect()
    {
        try
        {
            _sshClient.Connect();
            if (_sshClient.IsConnected)
            {
                Console.WriteLine("Connected to SSH server.");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error connecting to SSH server: {ex.Message}");
        }
    }

    public string ExecuteCommand(string command)
    {
        if (!_sshClient.IsConnected)
        {
            throw new InvalidOperationException("SSH client is not connected.");
        }

        var commandResult = _sshClient.RunCommand(command);
        return commandResult.Result;
    }
```

```
    public void Disconnect()
    {
        if (_sshClient.IsConnected)
        {
            _sshClient.Disconnect();
            Console.WriteLine("Disconnected from SSH server.");
        }
    }
}
```

Passo 3: Usar o SSHManager em um Componente Unity

1. Crie um novo GameObject na cena, como **SSHController**.
2. Adicione um script ao GameObject com o seguinte código para conectar-se ao servidor e executar comandos:

```
using UnityEngine;

public class SSHController : MonoBehaviour
{
    private SSHManager sshManager;

    void Start()
    {
        // Configure os detalhes do servidor SSH
        string host = "example.com";
        int port = 22;
        string username = "user";
        string password = "password";

        // Inicialize e conecte
        sshManager = new SSHManager(host, port, username, password);
        sshManager.Connect();

        // Execute um comando no servidor
        string output = sshManager.ExecuteCommand("ls");
        Debug.Log("Command Output: " + output);

        // Desconecte
```

```
        sshManager.Disconnect();
    }
}
```

Passo 4: Testar a Conexão

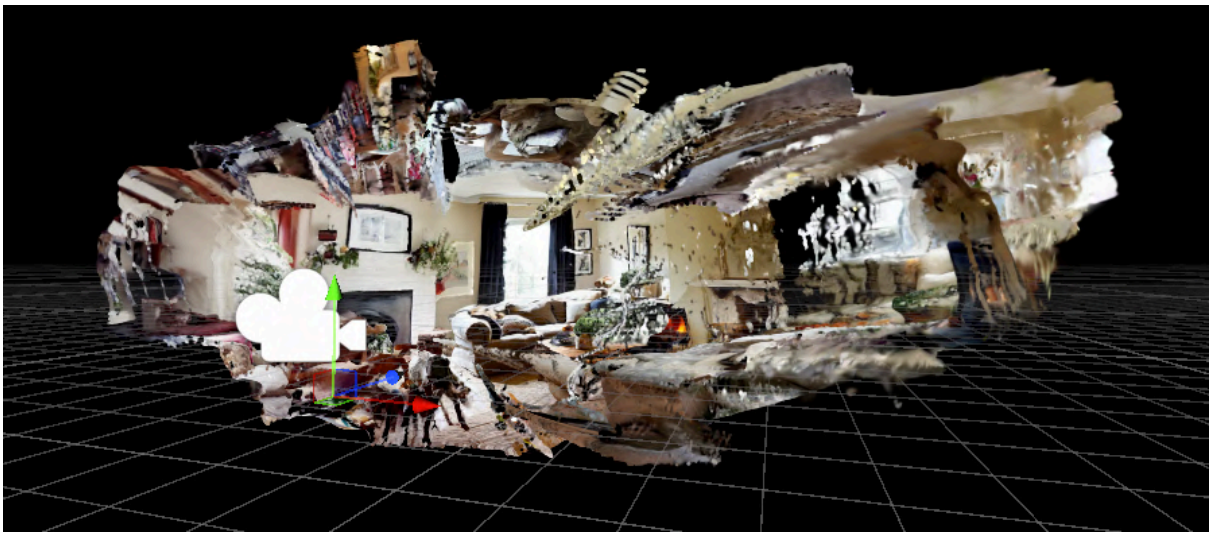
1. Configure as credenciais SSH corretas no código acima.
2. Pressione **Play** no Unity para testar a conexão.
    - O console do Unity exibirá mensagens relacionadas à conexão e os resultados dos comandos.

Abordagem de sucesso

A última alternativa que testei foi a de tentar fazer essa integração via SSH. Aproveitando que o modelo estava em um servidor remoto que ja estava sendo configurado via SSH. Com isso, após instalar algumas dependências, foi possível acessar o servidor via SSH e rodar os comando docker necessários. Mudando o código para que não gerasse mais o gradio e rodasse o código diretamente com o prompt requisitado.

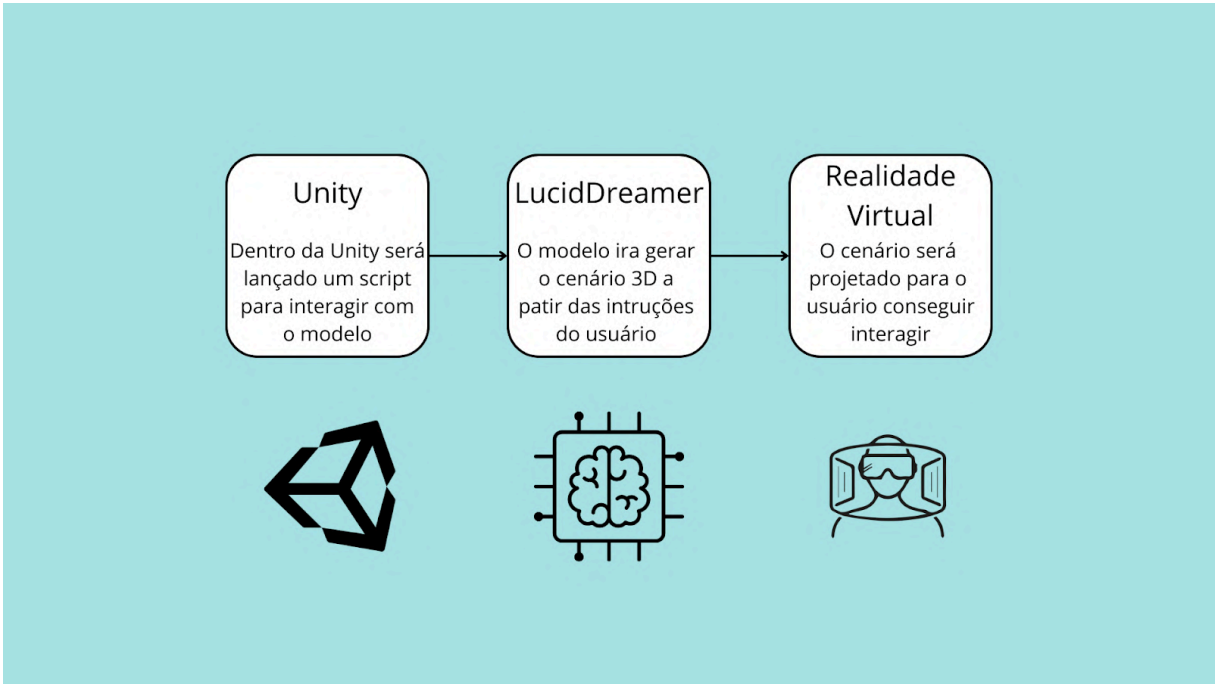
“docker exec -it {containerName} bash -c 'source /opt/conda/etc/profile.d/conda.sh && conda activate lucid && cd /workspace/LucidDreamer && python3 teste4.py --prompt \"{prompt}\" --negative_prompt \"{negativePrompt}\"'”;

Dessa forma, a Unity conecta no servidor remoto, roda os comandos docker, e depois copia os arquivos para a máquina local. Esse arquivo resultante é carregado pelo plugin e mostrado na tela.



Essa interação ainda está sendo feita por comandos ao iniciar a aplicação, a ideia é fazer tudo por dentro da aplicação, através de STT ou por uma caixa de texto.

Workflow Atual



LucidDreamer_Unity

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 5 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

MURILO DE OLIVEIRA GUIMARAES

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Produto dessa Semana: Abordagens Unity

Relembrando:

Ao longo das entregas eu trouxe diversos resultados de ambientes 3D, algumas ferramentas do estado da parte possibilitam a criação de uma ambiente 3D com muita qualidade, desde que muitas imagens sejam utilizadas.

Um exemplo é o vídeo da minha sala de estar: Sala de Estar - Exploração.mp4
É possível ver e explorar esse ambiente 3D com muita liberdade.

Apesar disso, alguns modelos lidam com o desafio de interagir com apenas uma imagem, o que é possível imaginar que não terá o melhor dos resultados, porém é uma área a ser explorada.O modelo utilizado ao longo da implementação da Unity foi o LucidDreamer, que trabalha com uma imagem como entrada e um texto. As alterações feitas para aceitar um audio ou um texto como entrada. Possibilitando o seguinte workflow:

Workflow_Completo.png

Progresso da Semana:

É possível agora trabalhar com áudio como entrada, então é possível agora falar um cenário que você deseja e o workflow irá gerar o cenário 3D. O workflow completo é o mais automatizado possível, porém uma pequena parte ainda está manual, acredito que com um conhecimento maior de Unity é possível automatizar tudo, mas tive dificuldades.

A Unity trabalha com o modo Editor e o modo Runtime.

- No modo Editor diversas alterações podem ser feitas no motor gráfico como a importação da mesh.ply para o diretório.
- No modo Runtime é durante o "play" da Unity, durante esse modo a Unity começa executar os

scripts e impede de utilizar algumas funcionalidades do modo Editor.

Foi possível utilizando duas Unity para fazer o workflow completo, mas dessa forma ficou bem completo, e de acordo com os planos que eu tive no começo.

Alguns cenários resultantes do modelo LucidDreamer:

A cozy living room:
A cozy livingroom.mp4

A huge library:
A huge library.mp4

A futuristic city:
A futuristic city.mp4

Workflow completo:
WorkflowCompleto.mp4

Planejamento: **[descrever o que pretende fazer para realizar a próxima ENTREGA]**

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Abordagens Unity

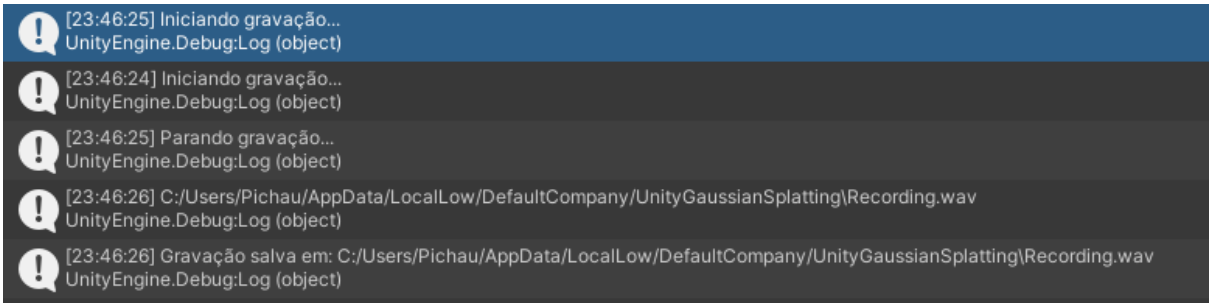
Outras dificuldades

Outras dificuldades encontradas foi na maneira de interagir com a Unity entre o modo Runtime e o modo Editor:

- No modo Editor diversas alterações podem ser feitas no motor gráfico como a importação da mesh.ply para o diretório.

- No modo Runtime é durante o “play” da Unity, durante esse modo a Unity começa executar os scripts e impede de utilizar algumas funcionalidades do modo Editor.

O código atual na Unity verifica o áudio feito pelo usuário, que é capturado através da utilização do microfone e ao pressionar um botão do teclado.

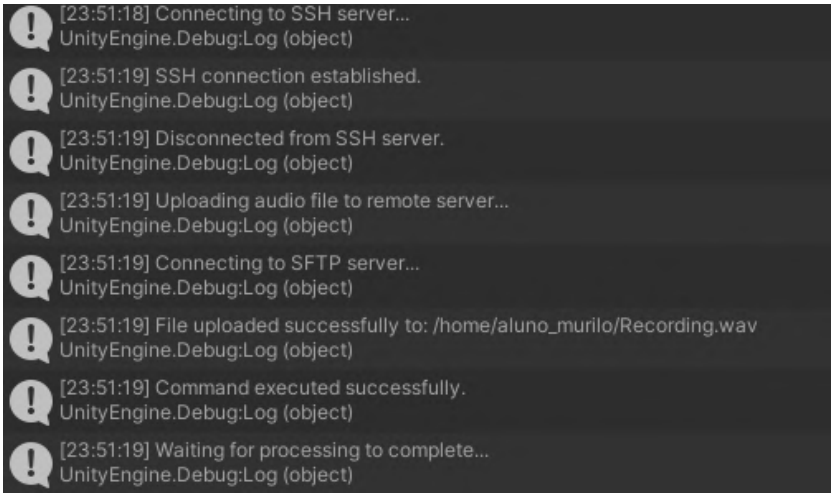


Essa gravação é então mandada por outro script para o servidor remoto onde o modelo está configurado e rodando sem problemas.

Durante a execução do script, o áudio é mandado via scp para o servidor, e depois mandado para o container, o modelo então roda o comando

"docker exec -it {containerName} bash -c 'source /opt/conda/etc/profile.d/conda.sh && conda activate lucid && cd /workspace/LucidDreamer && python3 run.py --audio \"{containerAudioPath}\" --save_dir \"{containerOutputPath}\"'"

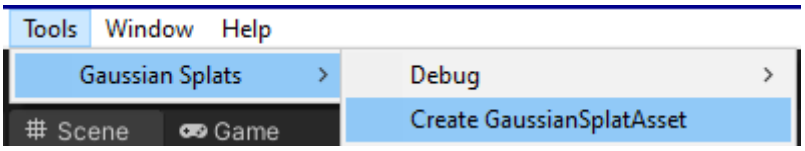
Esse comando executa o container e o script que recebe o caminho do audio ou texto e começa a gerar o ambiente 3D, o output gerado é salvo em uma pasta que depois é retirado do container e baixado do servidor para o computador local.

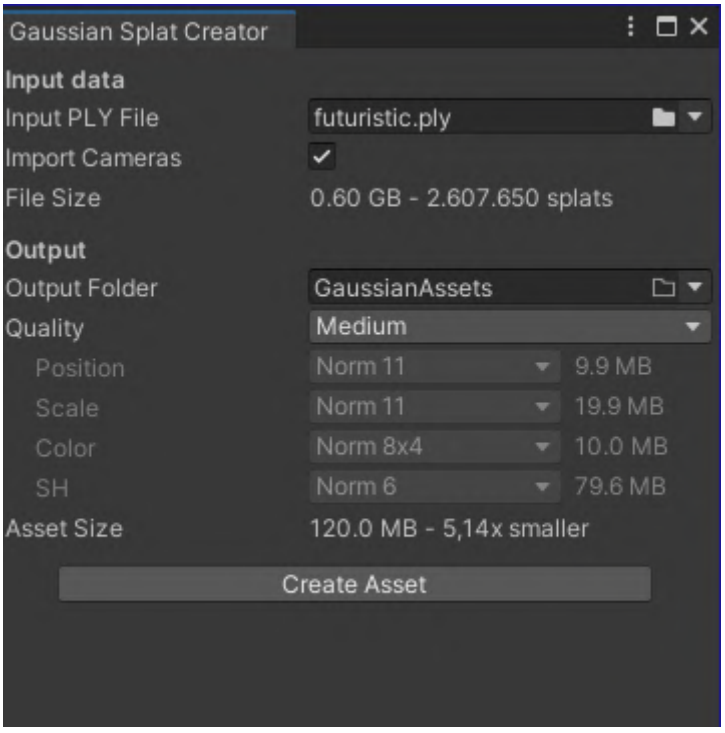


Depois disso, foi o momento dos erros. Pois o plugin da Unity que recebe um arquivo .ply e

lê a nuvem de pontos para gerar o asset só funciona no modo Editor, provavelmente é possível fazer msm em Runtime e automatizar até mesmo essa parte, mas não fui capaz de fazer com as minhas capacidades atuais de Unity.

O método que fiz foi de ter que utilizar a importação do plugin manualmente e arrastar o asset resultante para a tela.

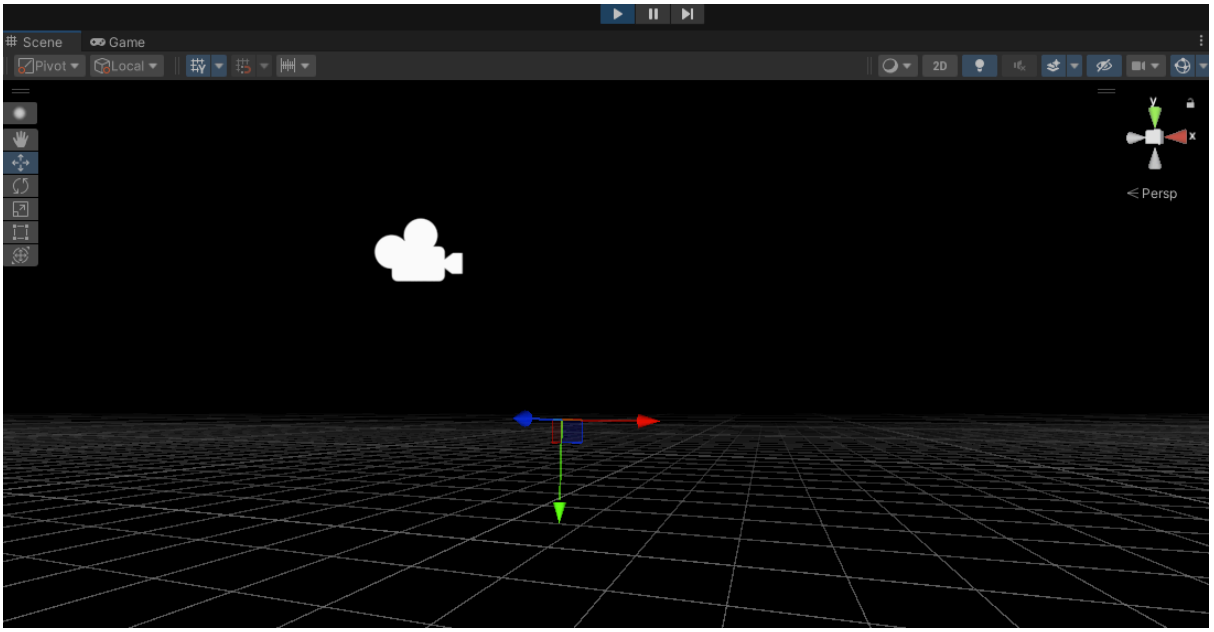




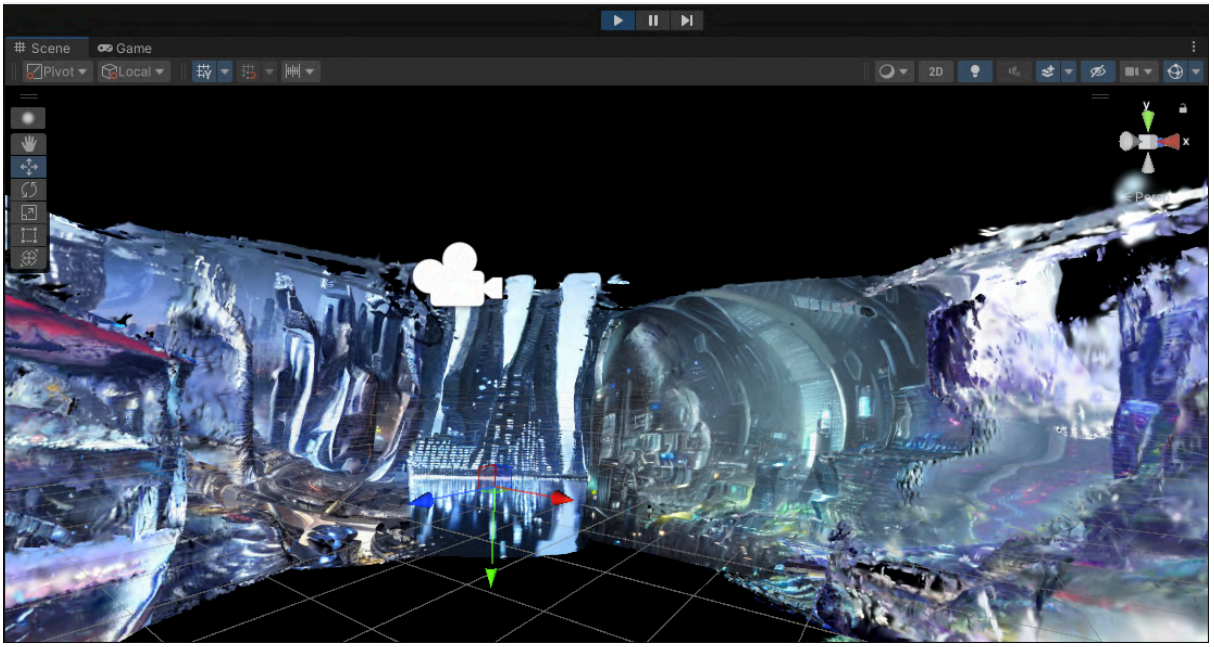
Dessa forma os seguintes arquivos são gerados:

Dessa forma é possível arrastar esse arquivo para o GameObject.

Antes:

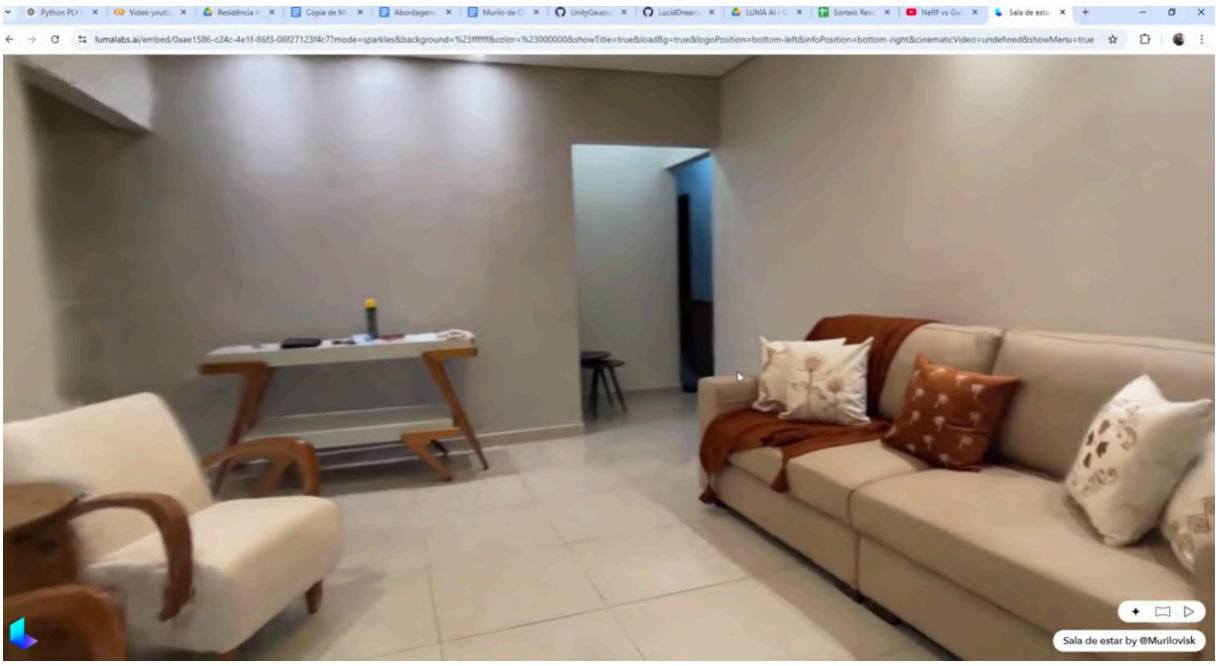


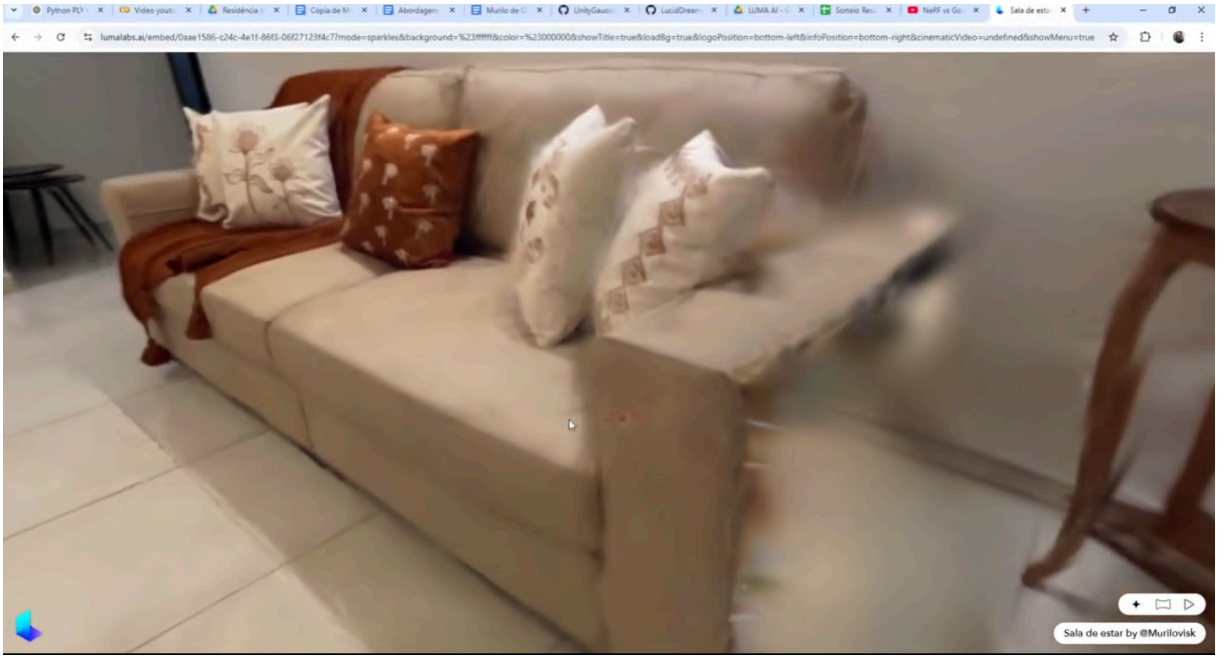
Depois:

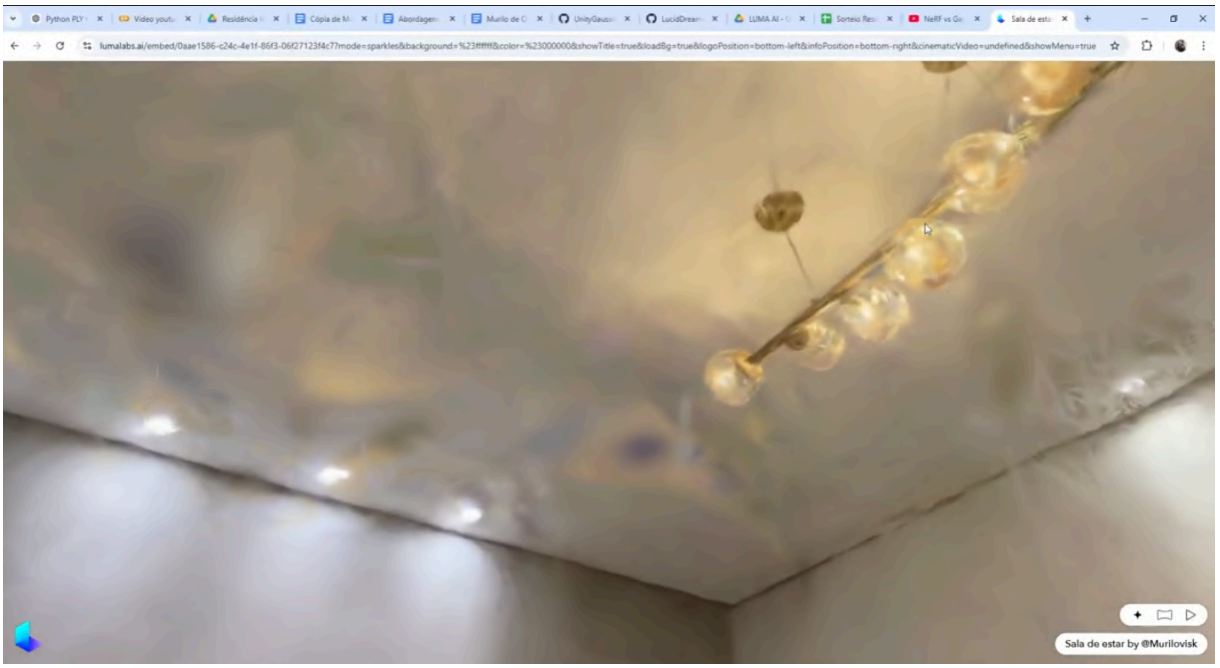


E dessa forma é possível explorar o ambiente.
WorkflowCompleto.mp4

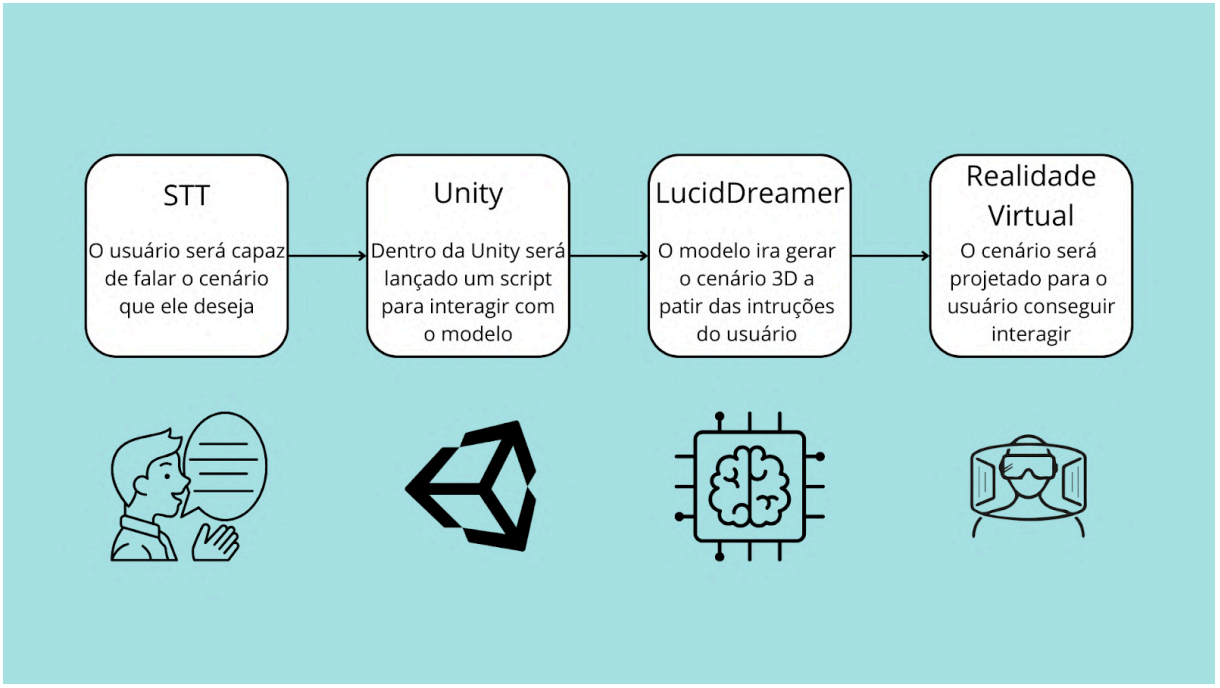
Sala de Estar







Workflow Completo



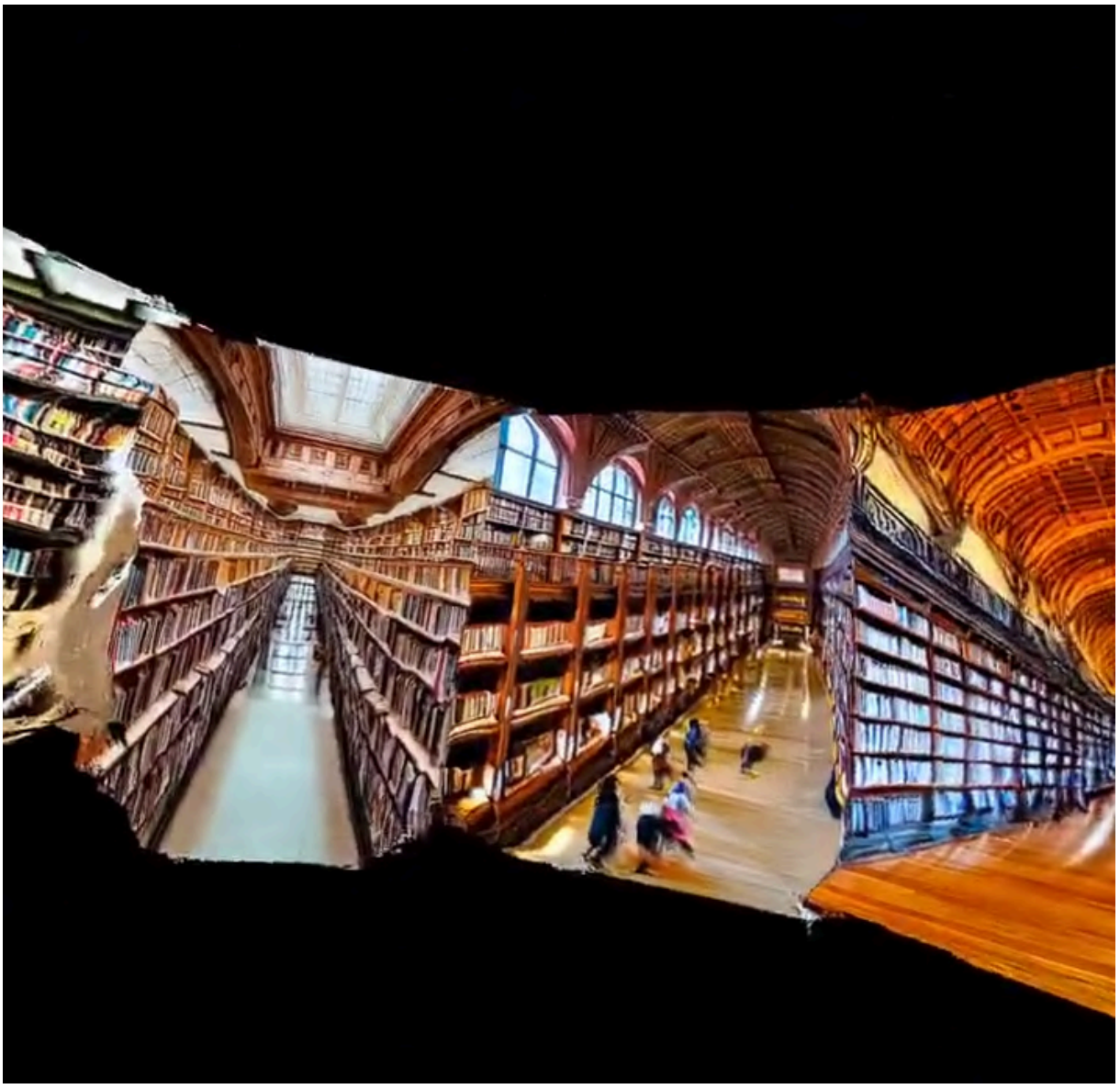
A cozy living room

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

A huge library



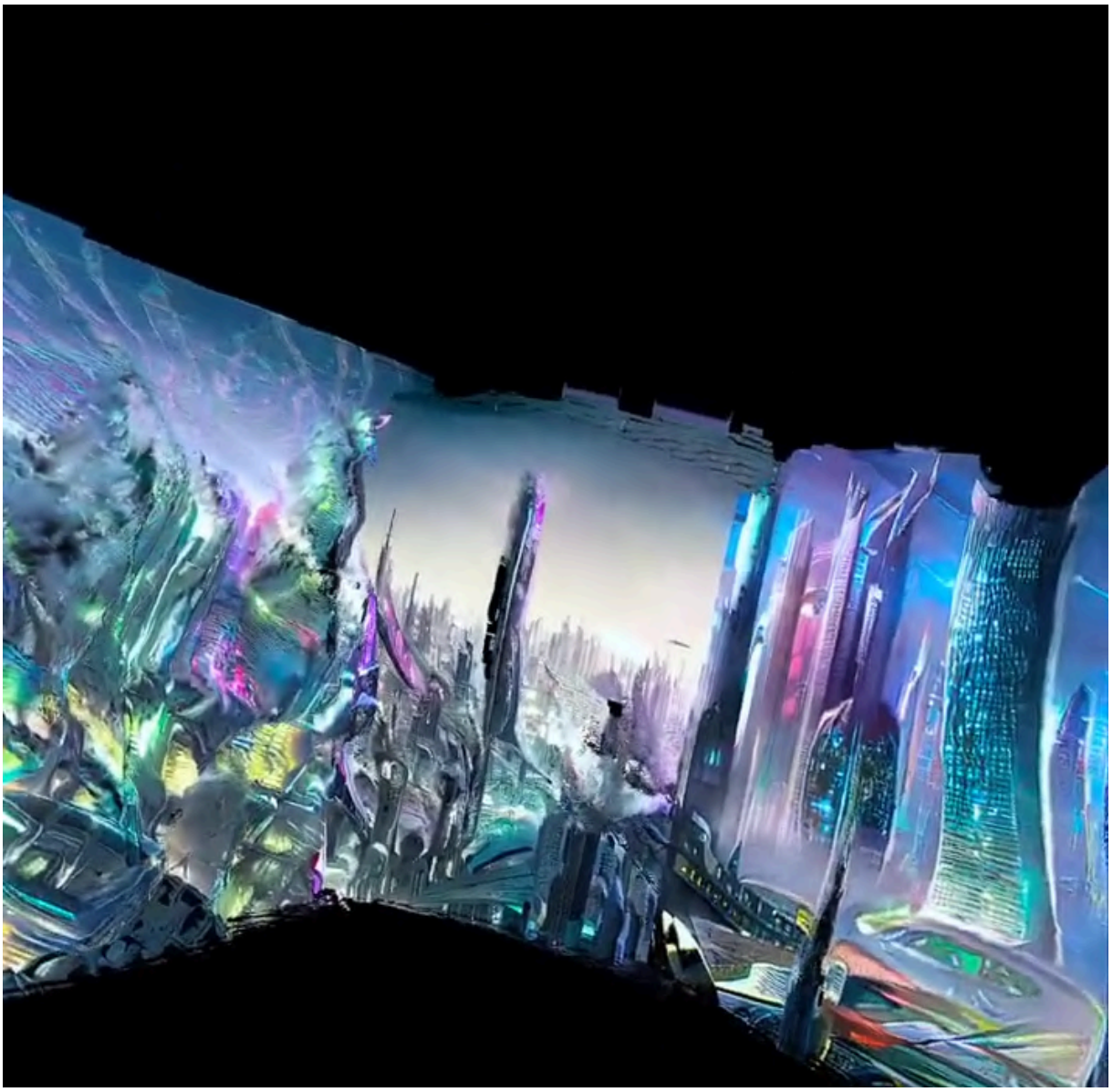
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

A futuristic city

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

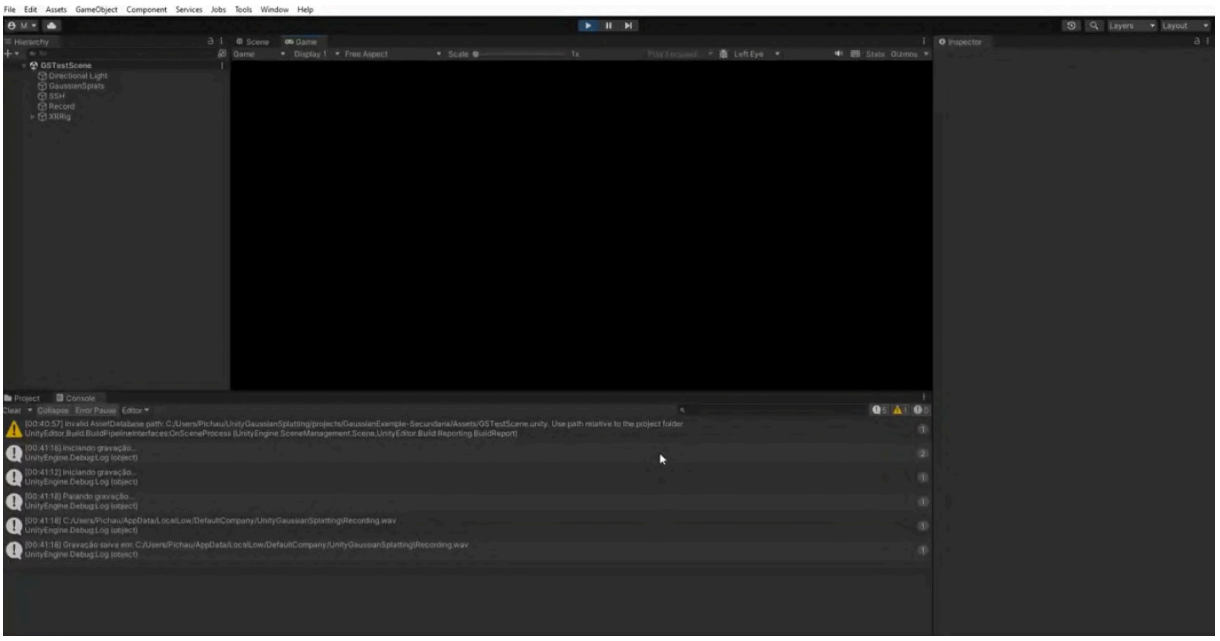


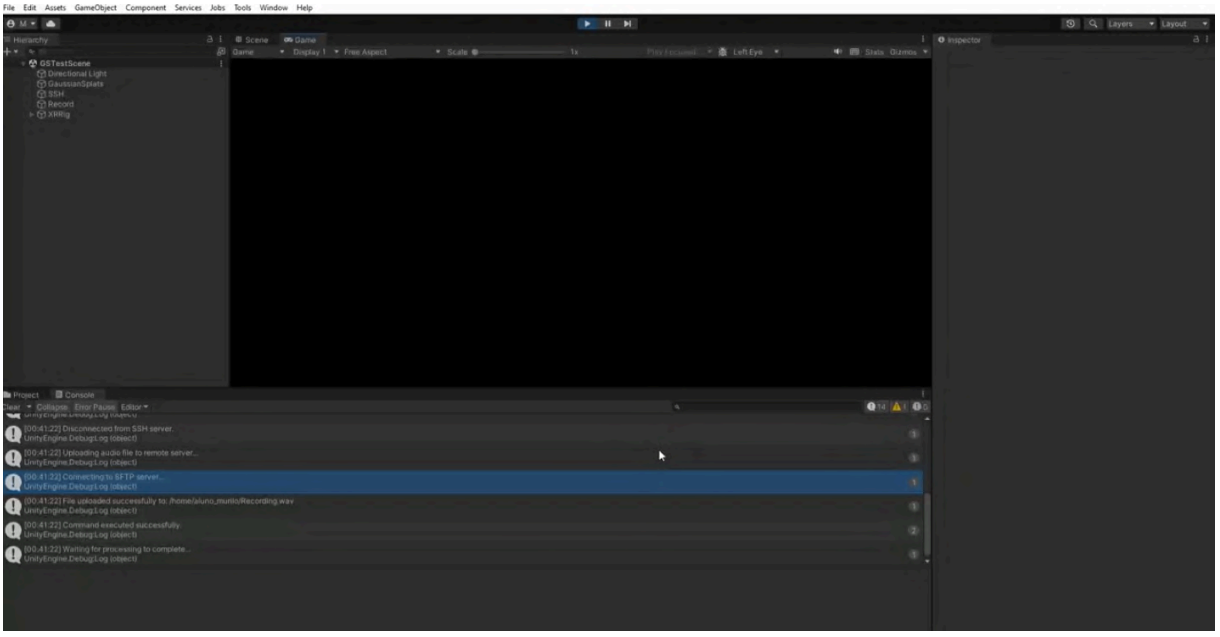
Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

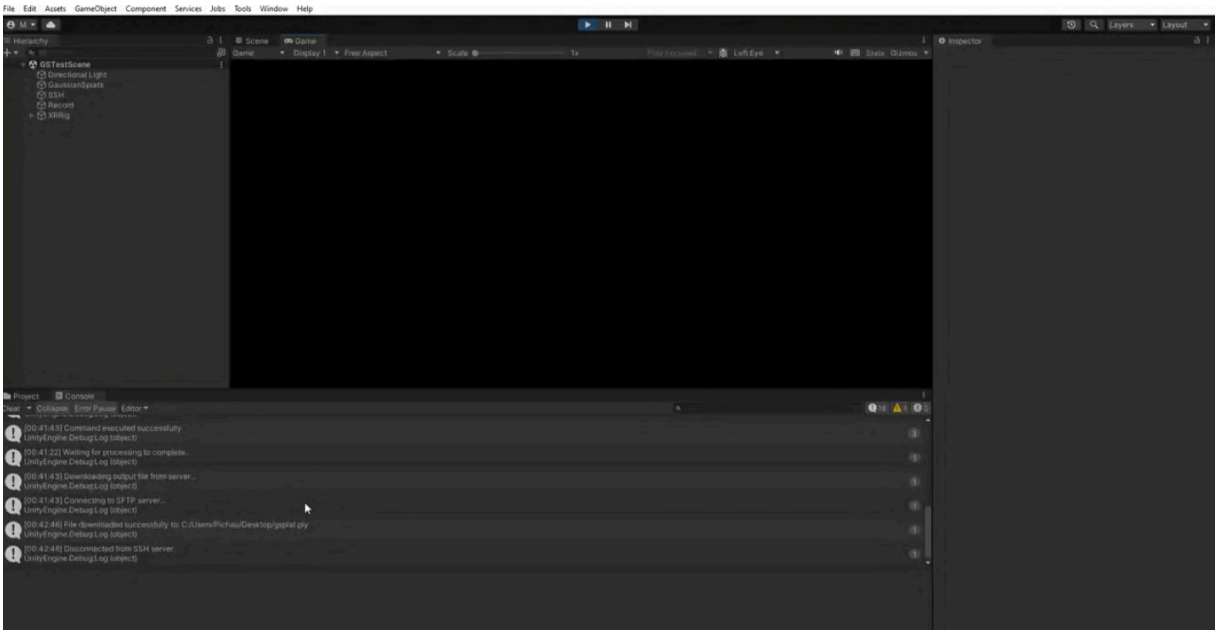
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Workflow Completo

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182






File Edit Assets GameObject Component Services Jobs Tools Window Help
Hierarchy
GSTestScene
Directional Light
GaussianSplats
SSH
Record
XRRig
Scene
Game
Inspector
Project
Console
[00:41:43] Command executed successfully.
[00:41:22] Waiting for processing to complete...
[00:41:43] Downloading output file from server...
[00:41:43] Connecting to SFTP server...
[00:42:46] File downloaded successfully to: C:/Users/Pichau/Desktop/gsplat.ply
[00:42:46] Disconnected from SSH server.