DESENVOLVIMENTO DE AMBIENTE VIRTUAL FOTORREALISTA PARA TREINAMENTO DE ROBÔS EM AMBIENTE RURAL

GUILHERME HIDACA[1]; FABRICIO ARDAIS MEDEIROS[2]

[1]*Universidade Federal de Pelotas, (UFPeL) – ghidaca@hotmail.com*
[2]*Universidade Federal de Pelotas, Campus Capão do Leão – fabricio.medeiros@ufpel.edu.br*

1. INTRODUÇÃO

A automação via mecanismos de *computer vision*, *machine learning* e protocolos mais adaptados a cada objetivo em conjunto com a evolução de sensores, como câmeras RGB de maior resolução e menor tamanho, GPS (*Global Positioning System*) mais precisos, e LIDAR (*Light Detection And Ranging*) carregando todos esses conjuntos de avanços, acarretou-se o desenvolvimento do campo de movimentação autônoma nos últimos anos.

Com objetivo de minimizar erros por falhas humanas, permitir trabalhos em semeaduras mais precisos e rápidos, e em outros campos como gestão de estoques e entregas, um maior controle e confiabilidade, a tecnologia vem ganhando espaço em diversas áreas. Com isso também a necessidade da prototipagem dos equipamentos e de testes para implementação no ambiente desejado, no nosso caso o rural.

Para realização desses testes e treinamento das máquinas existe o meio tradicional de implementação física com experimentos manuais onde é requerido uma réplica ou o próprio ambiente de destino de operação, contudo esta forma requer um orçamento elevado, considerando os gastos com sensores, tempo e locação.

Tendo esses limitadores em mente que foram desenvolvidos os testes em ambientes virtuais, propulsionados nas últimas décadas pelos avanços na capacidade de processamento, tendo como principal limitador o conhecimento das ferramentas de simulação, e a capacidade de renderização gráfica da máquina utilizada para tal.

2. METODOLOGIA

O projeto se dá por duas etapas principais, a primeira sendo o desenvolvimento do ambiente virtual fotorrealista, e para isto a ferramenta utilizada foi o software Unity. Escolhido em detrimentos de outros programas já conceituados na áreas como Gazebo, sendo a Unity selecionada em virtude da sua alta capacidade de gerar cenários realistas, e ampla gama de *assets* disponíveis. Nela diversas variáveis dos materiais podem ser controladas para gerar um ambiente mais fidedigno, através de técnicas como PBR (*Physically Based Rendering*) que realiza cálculos baseados no comportamento da luz no mundo real para simular com maior precisão nossos cenários.

Os denominados *shaders* que irão definir os parâmetros e comportamento dos nossos objetos mediante o cenário, fazem uso de variáveis como iridescência (*Iridescence*), translucidez (*Translucency*), e dispersão (*Subsurface scattering*) com seus efeitos seguindo na Figura 1, além de diversas outras como pode ser visto em (UNITY).

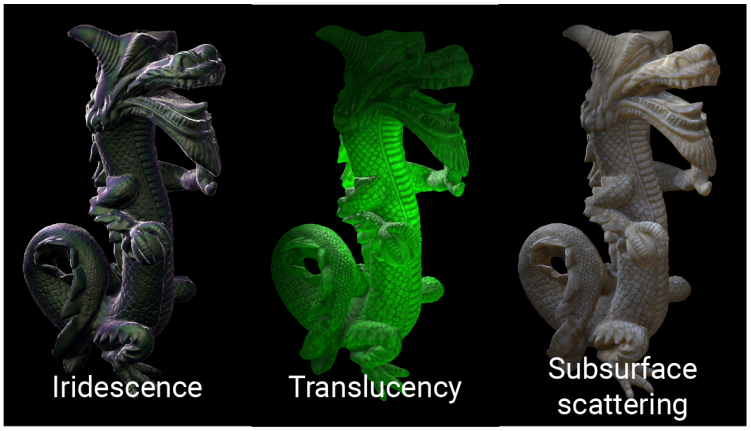

Figura 1: Exemplo das propriedades dos materiais (Fonte: UNITY)

A segunda etapa, seria a comunicação dos sensores implementados com o ROS (*Robot Operating System*), um framework de código aberto que gerencia a comunicação entre os diferentes *scripts* e *softwares.* Ele foi configurado em uma máquina virtual para simular a conexão entre o robô e um computador externo. A transmissão dos dados é realizada através do protocolo TCP/IP, com o *endpoint* sendo a máquina virtual, tendo como base o repositório (Unity-Technologies).

Nesta mesma máquina virtual roda um *script* em Python que realiza a última etapa de tratamento de imagem antes do armazenamento, que seria a adição de um ruído gaussiano. Característica esta vista em diversas câmeras digitais em diferentes níveis, e que também pode ser parametrizado para o determinado sensor escolhido.

3. RESULTADOS E DISCUSSÃO

Os primeiros resultados foram obtidos simulando a webcam C920 da Logitech, de resolução 1920 x 1080 pixels, com os dados sendo recebidos na máquina virtual a uma taxa média de 30 frames por segundo. Na Figura 1 temos um frame sem o pós-processamento de imagem. As imagens são recebidas com um atraso desprezível para a aplicação, podendo serem exibidas em tempo real através do visualizador RVIZ.


Figura 2: Frame sem pós-processamento de imagem (Fonte: autor)

Na Figura 3 podemos ver a diferença quando o pós-processamento de imagem é ativado, neste caso gerando um cenário de pôr do sol. E por fim, na Figura 4 o ruído característico da câmera sendo aplicado sobre a captura nos proporcionando o resultado final, que será utilizado para o treinamento do robô.

Figura 3: Frame com pós-processamento de imagem (Fonte: autor)

Figura 4: Frame com pós-processamento de imagem e ruído (Fonte: autor)

4. CONCLUSÕES

O ambiente simulado permite testes prévios dos sensores implementados no mesmo e a criação de comparativos dos diferentes protocolos de movimentação autônoma, a ênfase escolhida no projeto foi o fotorrealismo, portanto o foco seria a obtenção de dados de imagem através de diferentes tipos de câmeras. Contudo se necessário também podem ser gerados dados como *point clouds* de LIDAR ou gráficos de IMU (*Inertial Measurement Unit*).

O ponto principal é a praticidade da alteração dos parâmetros do projetos, e a velocidade na obtenção de dados, tudo isso sem a necessidade da compra dos sensores que normalmente possuem valores elevados, o que acarreta numa seleção prévia otimizada dos componentes, e por fim uma melhor gestão de orçamento.

5. REFERÊNCIAS BIBLIOGRÁFICAS

KOUBAA, A. **Robot Operating System (Ros): The Complete Reference (Volume 2)**. Poland: Springer Publishing Company, Incorporated, 2017.

UNITY. **Material Type**. Acessado em 23 jun. 2022. Online. Disponível em: https://docs.unity.cn/Packages/com.unity.render-pipelines.high-definition@6.7/manual/Material-Type.html

Unity-Technologies. **Unity-Robotics-Hub**. Acessado em 10 jun. 2022. Online. Disponível em: https://github.com/Unity-Technologies/Unity-Robotics-Hub

Codd-Downey, R., Forooshani, P. M., Speers, A., Wang, H., & Jenkin, M. From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation. **2014 IEEE International Conference on Information and Automation (ICIA)**, Hailar, p.932-936, 2014.