Simulador de LiDAR para Geração de Nuvens de Pontos Sintéticas

optScan



**Manual do Usuário**

Guilherme Ferreira Gusmão

06 de dezembro de 2021

# **Descrição**:

O simulador de LiDAR apresentado aqui, apelidado de optScan, vem de um histórico de pesquisa submarina, principalmente na área de petróleo e gás. Como as profundezas do oceano são um ambiente muito hostil, somado ao equipamento caro, é impossível adquirir uma quantidade adequada de dados para aplicações como a criação de conjuntos de dados para treinar sistemas de deep learning. Assim, a geração de nuvens de pontos sintéticas ajudaria a criar mais dados para prova de conceito, melhor planejamento de aquisição de dados em campo e aumento de dados na área submarina.

O simulador utiliza um modelo híbrido de sensor (baseado em medições e em erros) para gerar conjuntos de dados sintéticos com anotação semiautomática com fácil adaptação para diferentes campos de pesquisa. Seu sensor virtual calibrado por comparação com um sensor real garante maior robustez neste processo.

# **Composição**:

O sistema foi construído de forma modular, constituído de 3 principais blocos: Um construtor de cenários virtuais baseado em objetos 3D; um modelo matemático do comportamento de um LiDAR real a partir daqui chamado de sensor virtual; e por fim um módulo capaz de processar os dados na forma de nuvens de pontos.

# **Utilização**:

Existem três soluções no github (URL: <https://github.com/Gusmaoguilherme/optScan_Lidar_simulator.git>) deste projeto.

Unity3D-Optix: Este projeto Unity3D versão 2018.3.14f1, encontrada na pasta Scene\_Builder, permite que o usuário construa cenários para escaneamento virtual. Cada objeto colocado em cena pode receber o script Data. Ele é responsável por armazenar o nome do objeto e seu label (um número) de classe, chamado aqui de Id. Depois, objetos do tipo câmera podem ser definidos pela cena para representar a posição do sensor virtual. Definido a cena e as câmeras, com o uso do script Gerenciador, pode-se exportar o arquivo de descrição de cena e caminho do sensor que serão importantes para as próximas etapas. Este projeto também converte os objetos 3D em arquivos .OBJ. Este script é executado clicando na engrenagem do script e escolhendo a opção GerenciadorObjs.

optScan: Este projeto abriga o módulo Sensor Virtual e Gerador de nuvens de pontos. Para construir suas soluções é necessário usar o CMake. Ele vem hardcoded com os parâmetros metrológicos do LiDAR URG-04LX-UG01 da Hokuyo como exemplo. O usuário poderá substituir os parâmetros antes de dar build no projeto com os dados do seu próprio LiDAR.

É necessário instalar OptiX SDK 5.0 (URL:<https://developer.nvidia.com/designworks/optix/downloads/legacy>) e CUDA 10 para construir essas soluções. Como essas são versões legado da NVIDIA, recomenda-se seguir as instruções fornecidas por eles em seu manual na página de download (necessário criar conta NVIDIA). Além disso a placa gráfica deverá ser da NVIDIA.

A tabela a seguir mostra os possíveis comandos para esta etapa:

|  |  |
| --- | --- |
| Comando | Descrição |
| ? | help | --help | Imprime todos os comandos disponíveis |
| -w | --width <int> | Largura da janela para renderização. Junto com height define a quantidade de raios que serão disparados. |
| -h | --height <int> | Altura da janela para renderização. Junto com width define a quantidade de raios que serão disparados. |
| -m | --mesh <mesh\_file> | Especifica o objeto a ser carregado. <mesh\_file> é o caminho do objeto. |
| -c | --camera <camera\_file> | Especifica o caminho com as informações do caminho de câmera. <camera\_file> é o caminho deste arquivo. |
| --position <x,y,z> | Especifica a posição do objeto em relação ao mundo em coordenas cartesianas. |
| --rotation <x,y,z,angle> | Especifica a rotação do objeto na forma de um quartenion. |
| --scale <float> | Especifica a escala do objeto |
| --id <int> | Especifica a identificação de classe do objeto definida pelo usuário |
| -o | --outputPath <path> | Local onde as nuvens de pontos serão armazenadas |
| --optParam <color,id> | Indica se o usuário deseja armazenar a informação de cor e id para cada ponto da nuvem de pontos gerada |
| -c | --configure <config\_file.txt> | Arquivo com linhas de comando |

mergeVertices: Esta solução é responsável por unificar as nuvens de pontos individuais geradas de cada posição de câmera definida pelo usuário. Também é necessário usar o CMake para construir a solução, além de obter as bibliotecas do Geogram (URL:<http://alice.loria.fr/index.php/software/4-library/75-geogram.html>), uma biblioteca para manipulações geométricas.

A tabela a seguir mostra os possíveis comandos para esta etapa:

|  |  |
| --- | --- |
| Comando | Descrição |
| ? | help | --help | Imprime todos os comandos disponíveis |
| -t | --tolerance <float> | Tolerância para distância mínima entre pontos para o processo de decimação espacial |
| -w | --write <output\_file.xyz> | Salva nuvem de pontos em formato xyz |
| -m | --merge <dir\_of\_merge\_files> | Diretório com as nuvens de pontos para merge |
| -n | --normalize | Ajusta a nuvem de pontos no interior de um cubo com aresta de tamanho 2. |
| -c | --configure <config\_file.txt> | Arquivo com linhas de comando |

Cada ponto nessas nuvens tem as seguintes propriedades:

• Coordenadas 3D: Define a posição do ponto em relação ao cenário virtual construído (x, y, z). São as informações básicas de uma nuvem de pontos. A visualização dos pontos é mostrada na Figura 1a.

• Vetor normal: São vetores perpendiculares à superfície dos objetos detectados. Usado para definições de iluminação de materiais e útil em várias rotinas de segmentação de dados. A visualização do vetor em ação é mostrada na Figura 2b.

• Cor: Definido na forma de vetores RGB. Útil para verificar a correspondência da nuvem de pontos com o modelo 3D. Alguns sistemas LiDAR já possuem câmeras no espectro visível para obter informações de cores, pois o sensor sozinho não consegue adquirir esses dados. A visualização da cor é mostrada na Figura 1c.

• Número de Identificação: Este número é um identificador único definido pelo usuário ao criar a cena, conforme mencionado anteriormente. Essas informações facilitam a visualização de diferentes objetos dentro da mesma nuvem, o que também facilita os processos de catalogação e segmentação de dados. A visualização do Id é mostrada na Figura 1d.

Mapa

Descrição gerada automaticamente

Figura 1 - Propriedades diferentes da nuvem de pontos sintéticos: (a) Pontos 3D; (b) Vetores normais; (c) Cor; (d) Id.

Um release do projeto encontra-se na pasta LiDAR\_example com as dll necessárias para rodar sem a instalação prévia do SDK do OptiX e dos pacotes do Geogram. O usuário ainda precisará ter uma GPU da NVIDIA para rodar a aplicação.

# **Requisitos do Sistema:**

|  |  |
| --- | --- |
| Hardware | Requisitos Mínimos |
| Placa Gráfica | GPUs da NVIDIA com Compute Capability 3.0 (Kepler) ou superior |
| CPU | Arquitetura X64 com suporte ao conjunto de instruções SSE2 |
| Memória | 8 GB de RAM |
| Sistema Operacional | Windows 7/8.1/10 64-bit;  Mac OS 10.9 ou superior |