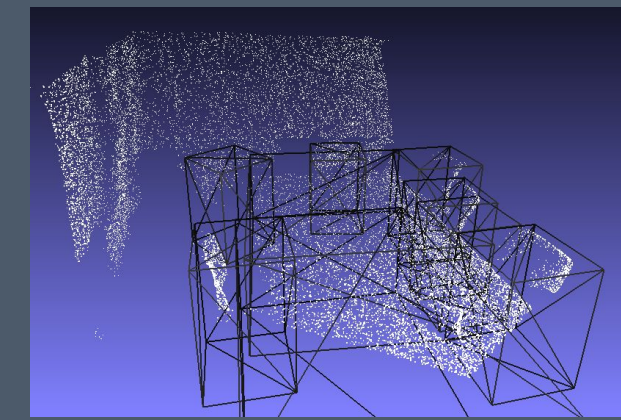




Detecção de objetos em nuvem de pontos

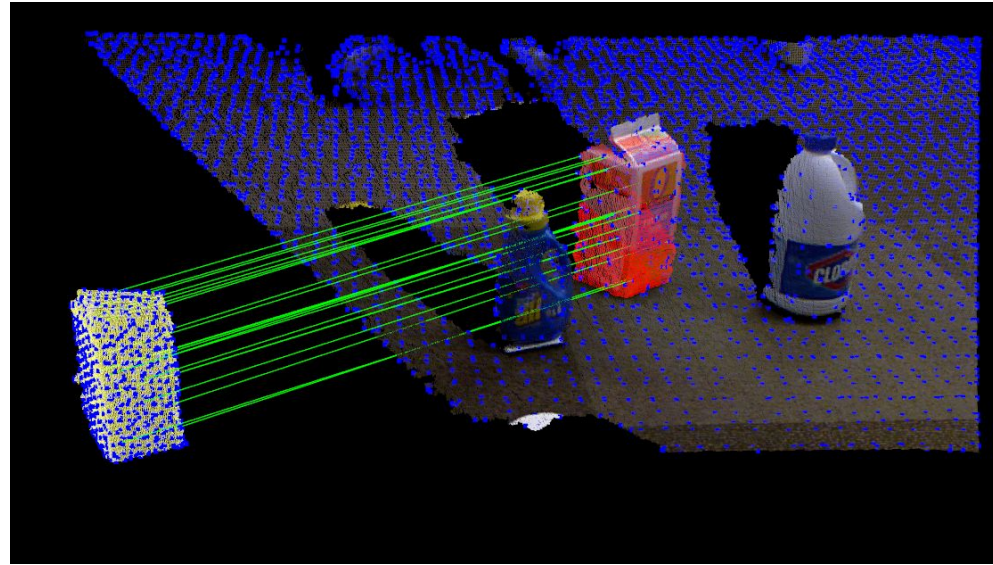
O reconhecimento semântico de objetos é um recurso importante para robôs autônomos que operam em ambientes reais não estruturados. Sensores de profundidade como LiDAR e câmeras e RGB-D são muito usados para evitar obstáculos e mapear, entretanto seu potencial para a compreensão semântica do ambiente ainda é relativamente inexplorado. O objetivo deste trabalho é fazer uso de dados em nuvem de pontos com ricas informações volumétricas para o reconhecimento e localização de objetos.



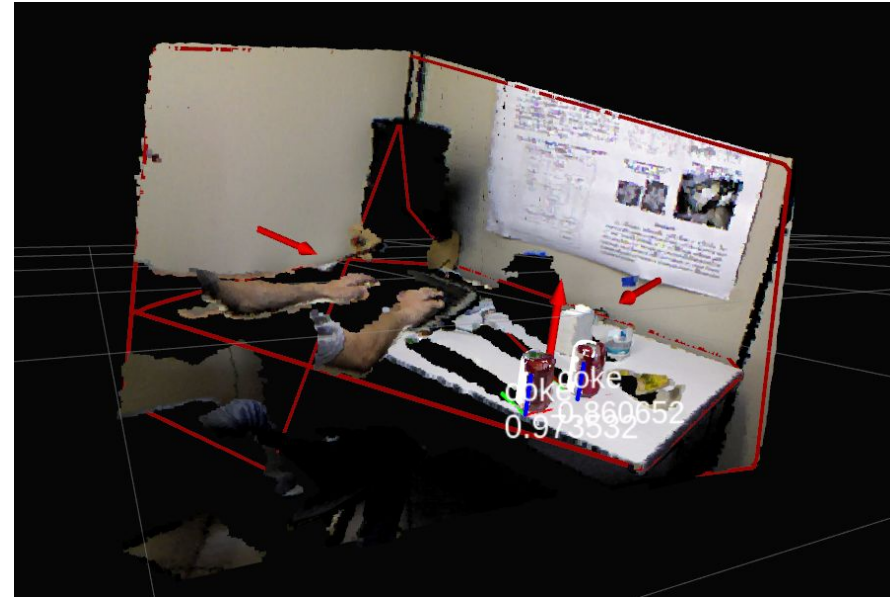
SCAN ME

Visão Clássica

Point Cloud Library (2011) Correspondence Grouping:

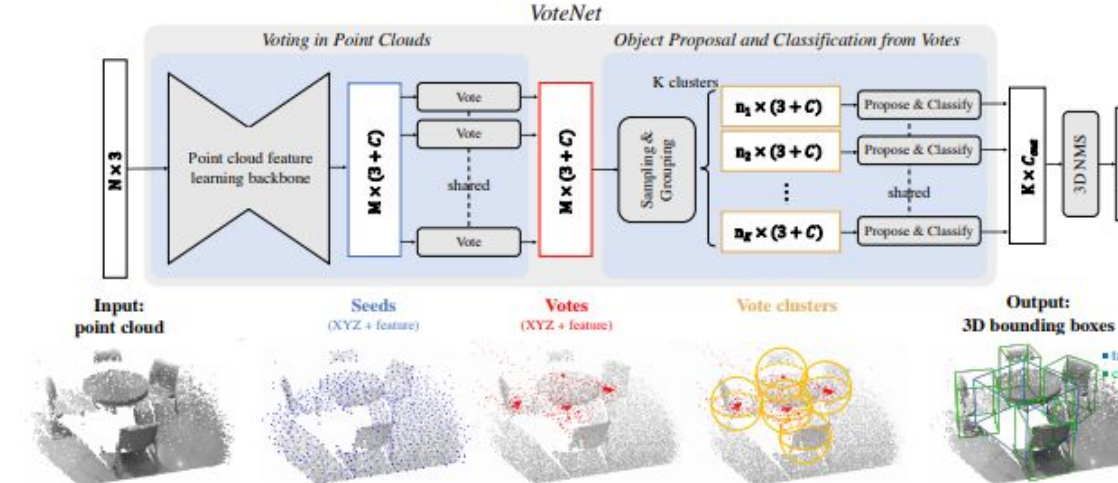


Object Recognition Kitchen (2013):



3D/2.5D CNN's

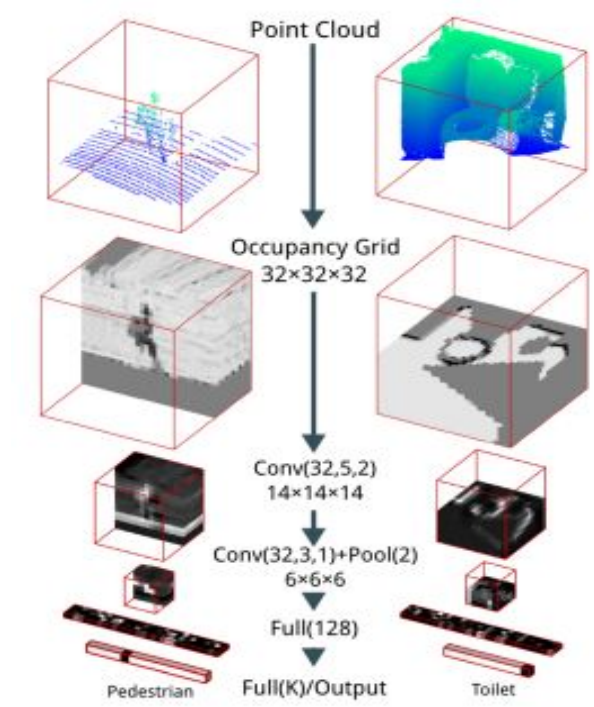
VoteNET (2019):



VoxNET (2015):

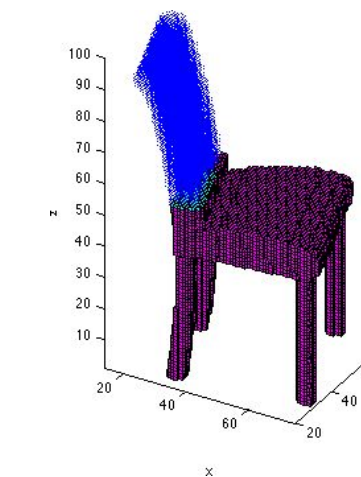
COMPARISONS WITH SHAPENET IN MODELNET (AVG ACC)

Dataset	ShapeNet	VoxNet
ModelNet10	0.84	0.92
ModelNet40	0.77	0.83



Datasets:

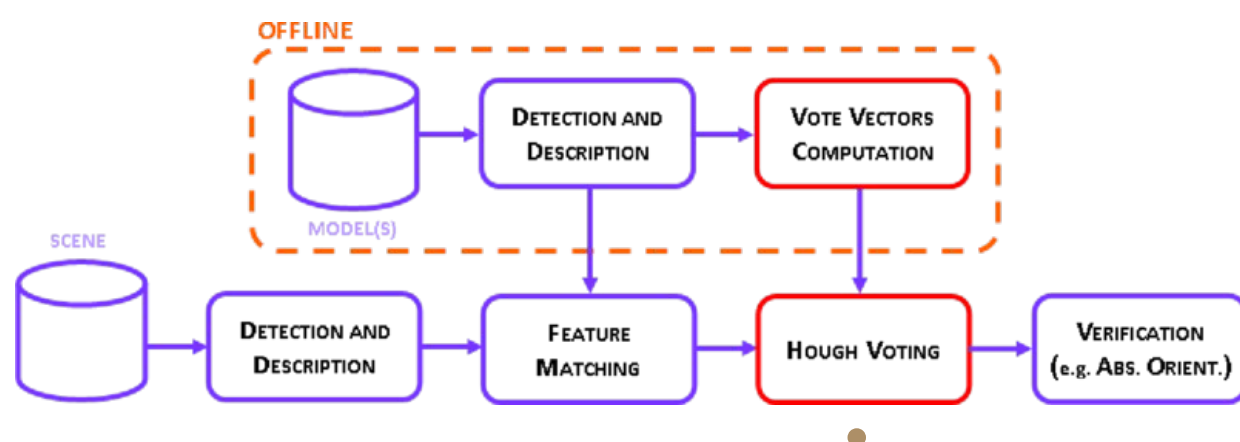
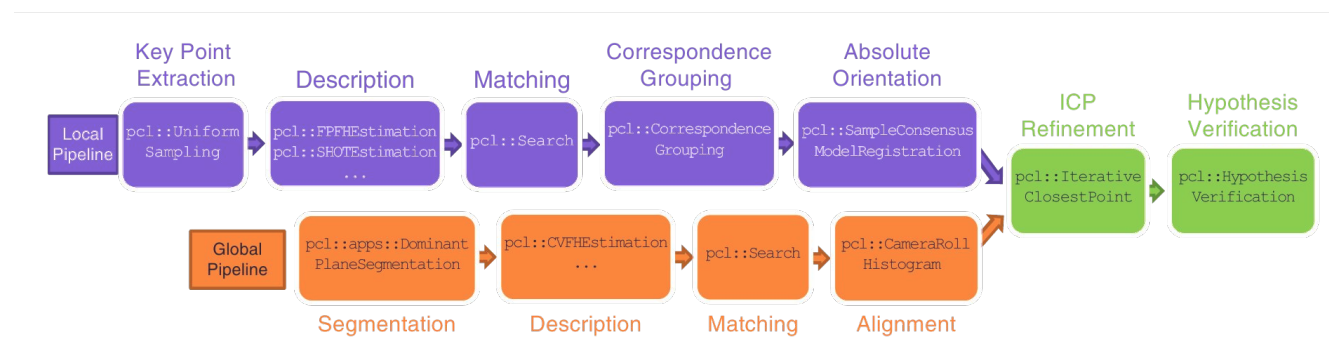
- Sun RGB-D (Câmeras RGB-D)
- ScanNet (LiDAR)
- ModelNet10 (Modelos CAD 3D)
- 3DShapeNet (ModelNet10 voxelizado)



Nossa implementação (Clássica)

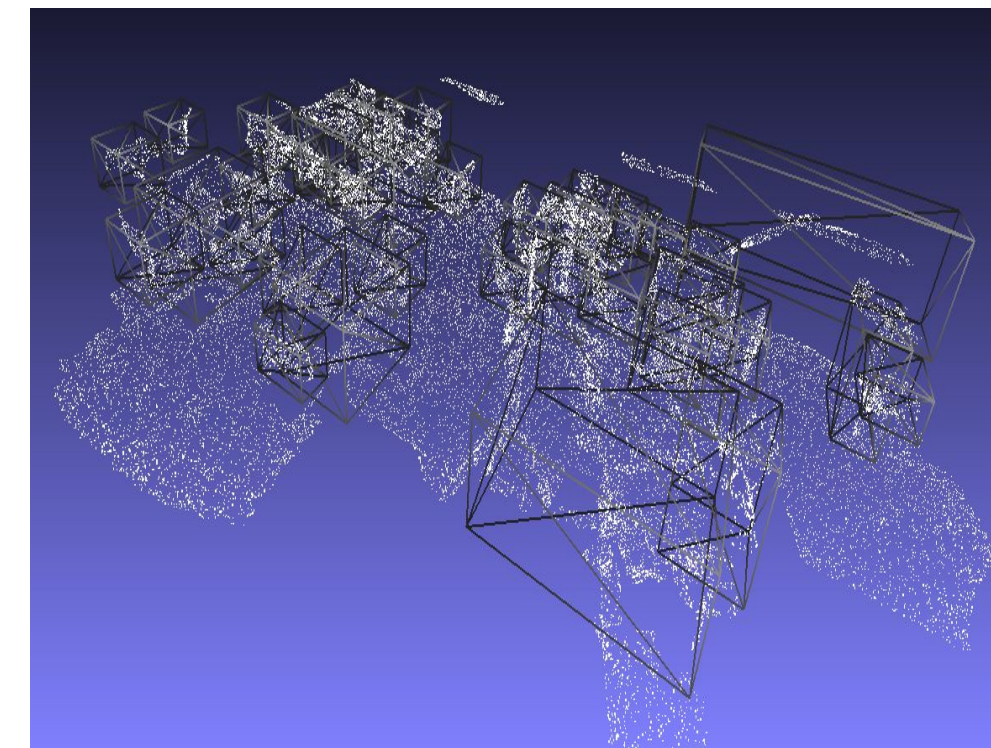
Pacote do ROS (C++) com pipeline local inspirada no *Correspondence Grouping* para detecção em tempo real de um objeto com o sensor Kinect v1.

Descritores locais
PFH (Point Feature Histogram)
FPFH (Fast Point Feature Histogram)
RSD (Radius-Based Surface Descriptor)
3DSC (3D Shape Context)
USC (Unique Shape Context)
SHOT (Signatures of Histograms of Orientations)
Spin image
RIFT (Rotation-Invariant Feature Transform)
NARE (Normal Aligned Radial Feature)
RoPS (Rotational Projection Statistics)
Descritores globais
VFH (Viewpoint Feature Histogram)
CVFH (Clustered Viewpoint Feature Histogram)
OUR-CVFH (Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram)
ESF (Ensemble of Shape Functions)
GFPFH (Global Fast Point Feature Histogram)
GRSD (Global Radius-Based Surface Descriptor)



Desvantagem: Detecta apenas um objeto e se faz necessário o modelo 3D do objeto em nuvem de pontos (.pcd) para extração de características através dos descritores. Existem poucos datasets nesse formato disponíveis, porém existem alguns API's para escanear os objetos através de uma câmera RGBD.

Nossa implementação (3D-CNN)



Implementação de 3D CNN em *google colab notebook* (Python) inspirado na rede VoxNET, utilizando o keras para modelar a rede e TensorFlow como back-end de tensores, enquanto a VoxNET usa Lasagne e Theano respectivamente.

Layer (type)	Output Shape	Param #
conv3d_18 (Conv3D)	(None, 32, 14, 14, 14)	4832
leaky_re_lu_16 (LeakyReLU)	(None, 32, 14, 14, 14)	0
dropout_24 (Dropout)	(None, 32, 14, 14, 14)	0
conv3d_19 (Conv3D)	(None, 32, 12, 12, 12)	27680
leaky_re_lu_17 (LeakyReLU)	(None, 32, 12, 12, 12)	0
max_pooling3d_8 (MaxPooling3D)	(None, 32, 6, 6, 6)	0
dropout_25 (Dropout)	(None, 32, 6, 6, 6)	0
flatten_8 (Flatten)	(None, 6912)	0
dense_16 (Dense)	(None, 128)	884864
dropout_26 (Dropout)	(None, 128)	0
dense_17 (Dense)	(None, 10)	1290
Total params: 917,866		
Trainable params: 917,866		
Non-trainable params: 0		

Data augmentation
shift (-2, 2)
flip (horizontal, vertical)
Optimizer
SGE
Loss Function
Sparse Categorical Crossentropy

Treinamento falhou:
Epoch 319/50000
1430/1430 - 14s - loss: 2.9571 - mean_squared_error: 106.9560 - acc: 0.1150
Epoch 320/50000
1430/1430 - 14s - loss: 3.0182 - mean_squared_error: 106.7032 - acc: 0.1178

Rede de detecção de objetos 3D baseada em redes de pontos e votação Hough. As redes de pontos usam nuvens de pontos cruas, sem a necessidade de voxelização ou outra técnica de pré-processamento. Implementado com PyTorch e TensorFlow em *google colab notebook* (Python).