Deep Closest Point: Learning Representations for Point Cloud Registration.--ICCV'19

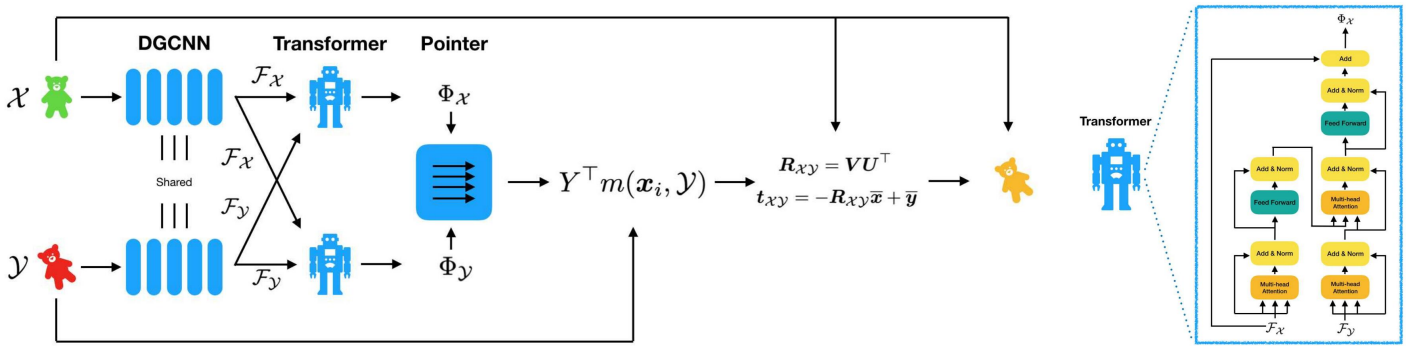
原文

简介

本文以深度学习的方法重构经典的ICP算法，提出了名为DCP的算法，用于对两个点云之间的刚性变换进行基于深度学习的预测。

问题陈述

$\mathcal{X},\mathcal{Y}$代表两个点云，$\mathcal{X}=\{x_1,...,x_i,...,x_N\}\in\mathbb{R}^3,\mathcal{Y}=\{y_1,...,y_i,...,y_M\}\in\mathbb{R}^3$，这里先假设$M=N$，并且本文的方法可以很轻松地被拓展到不相等的情况。假设$\mathcal{Y}$是由$\mathcal{X}$经过刚性变换得到的，刚性变换为$[R_{\mathcal{XY}},t_{\mathcal{XY}}]$，目标是最小化MSE:
$E(R_{\mathcal{XY}},t_{\mathcal{XY}})=\frac{1}{N}\sum_i^N\|R_{\mathcal{XY}}x_i+t_{\mathcal{XY}}-y_i\|$。本文的目标是使用学习的方法来重现一个更好的匹配，之后再使用这个匹配来计算刚性变换。



(a) Network architecture (b) Transformer module

Figure 2. Network architecture for DCP, including the Transformer module for DCP-v2.

Deep Closest Point

初始特征

对于初始特征的提取，本文从PointNet和DGCNN两个模型中选择了DGCNN。首先，PointNet提取的是全局特征，而DGCNN提取的是点特征，而本文需要的是每个点的特征来进行特征点匹配，同时DGCNN还考虑了局部近邻的信息，因此这里选择DGCNN，并用实验证明了选择的正确性。

Attention

为了使模型能够适应不同任务的需要，比如处理有机体形状和尖锐的机械物体需要的特征区别很大。因此本文使用了基于注意力的模型。$\mathcal{F}_{\mathcal{X}}, \mathcal{F}_{\mathcal{Y}}$是DGCNN提取出的特征，这里的注意力模型的任务就是学习一个函数$\phi : \mathbb{R}^{N \times P} \times \mathbb{R}^{N \times P} \rightarrow \mathbb{R}^{N \times P}$。

通过这个函数，就获得了新的特征$\Phi_{\mathcal{X}} = \mathcal{F}_{\mathcal{X}} + \phi(\mathcal{F}_{\mathcal{X}}, \mathcal{F}_{\mathcal{Y}}), \Phi_{\mathcal{Y}} = \mathcal{F}_{\mathcal{Y}} + \phi(\mathcal{F}_{\mathcal{X}}, \mathcal{F}_{\mathcal{Y}})$，对这个特征再做co-attention生成最终的特征

点生成

通过特征的匹配，该模型为每个$\mathcal{X}$内的点生成一个针对于$\mathcal{Y}$内点的概率即$m(x_i, \mathcal{Y}) = softmax(\Phi_{\mathcal{Y}} \Phi_{xi}^T)$

SVD模块

首先根据概率生成一个平均匹配点$\hat{y}_i = Y^T m(x_i, \mathcal{Y})$，在获得了匹配点之后，就可以按照匹配点进行SVD分解并最终得到变换矩阵。

Loss

$$Loss = \|R_{\mathcal{XY}}^T R_{\mathcal{XY}}^g - I\|^2 + \|t_{\mathcal{XY}} - t_{\mathcal{XY}}^g\|^2 + \lambda\|\theta\|^2$$

实验

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP	894.897339	29.914835	23.544817	0.084643	0.290935	0.248755
Go-ICP [55]	140.477325	11.852313	2.588463	0.000659	0.025665	0.007092
FGR [61]	87.661491	9.362772	1.999290	0.000194	0.013939	0.002839
PointNetLK [18]	227.870331	15.095374	4.225304	0.000487	0.022065	0.005404
DCP-v1 (ours)	6.480572	2.545697	1.505548	**0.000003**	**0.001763**	**0.001451**
DCP-v2 (ours)	**1.307329**	**1.143385**	**0.770573**	**0.000003**	0.001786	**0.001195**

Table 1. ModelNet40: Test on unseen point clouds

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP	892.601135	29.876431	23.626110	0.086005	0.293266	0.251916
Go-ICP [55]	192.258636	13.865736	2.914169	0.000491	0.022154	0.006219
FGR [61]	97.002747	9.848997	**1.445460**	0.000182	0.013503	0.002231
PointNetLK [18]	306.323975	17.502113	5.280545	0.000784	0.028007	0.007203
DCP-v1 (ours)	19.201385	4.381938	2.680408	**0.000025**	**0.004950**	**0.003597**
DCP-v2 (ours)	**9.923701**	**3.150191**	2.007210	**0.000025**	0.005039	0.003703

Table 2. ModelNet40: Test on unseen categories

Model	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
ICP	882.564209	29.707983	23.557217	0.084537	0.290752	0.249092
Go-ICP [55]	131.182495	11.453493	2.534873	0.000531	0.023051	0.004192
FGR [61]	607.694885	24.651468	10.055918	0.011876	0.108977	0.027393
PointNetLK [18]	256.155548	16.004860	4.595617	0.000465	0.021558	0.005652
DCP-v1 (ours)	6.926589	2.631841	1.515879	0.000003	0.001801	0.001697
DCP-v2 (ours)	**1.169384**	**1.081380**	**0.737479**	**0.000002**	**0.001500**	**0.001053**

Table 3. ModelNet40: Test on objects with Gaussian noise

# points	ICP	Go-ICP	FGR	PointNetLK	DCP-v1	DCP-v2
512	0.003972	15.012375	0.033297	0.043228	0.003197	0.007932
1024	0.004683	15.405995	0.088199	0.055630	0.003300	0.008295
2048	0.044634	15.766001	0.138076	0.146121	0.040397	0.073697
4096	0.044585	15.984596	0.157124	0.162007	0.039984	0.74263

Table 4. Inference time (in seconds)

idea

可以用于patch之间的配准，在第二次compensation之后另外加一个网络再进行精细化配准？根据数据，ICP匹配MSE高的压缩率小，并且树的差异大。
主要是网络难以处理这么多的点。