CSC2457 3D & Geometric Deep Learning

Canonical Capsules: Unsupervised Capsules in Canonical Pose

Weiwei Sun[1,5,*], Andrea Tagliasacchi[3,4,*], Boyang Deng[4], Sara Sabour[3,4], Soroosh Yazdani[4], Geoffrey Hinton[3,4], Kwang Moo Yi[1,5]

[1]University of British Columbia, [3]University of Toronto, [4]Google Research, [5]University of Victoria, [*]equal contributions

Date: March 16th 2021

Presenter: Ioannis Xarchakos

Instructor: Animesh Garg

UNIVERSITY OF TORONTO

Main Problem

Main Problem:

Training 3D deep representations in an unsupervised fashion

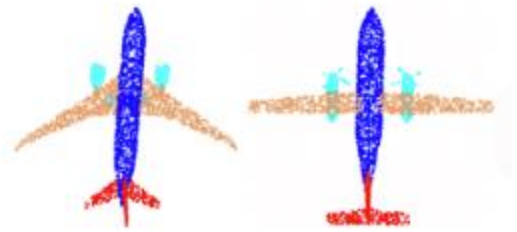
Importance

- This work achieves state of the art perfromance without labeled data
    - Many person-hours are required to extract accurate annotations
- The framework requires no manual object pre-canonicalization

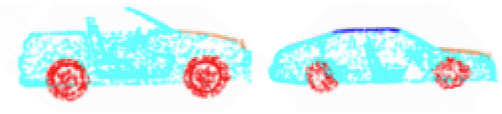
Prior Work

Prior work exploits the inductive bias of the training data sets

- Airplanes cockpit is always along y axis



- Cars always touch z axis



References:

- Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D Point Capsule Networks
- Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning Elementary Structures for 3D Shape Generation and Matching

Contributions

This work proposes :

- Unsupervised learning on 3D point clouds using capsules
- Object-centric unsupervised learning

Past work:

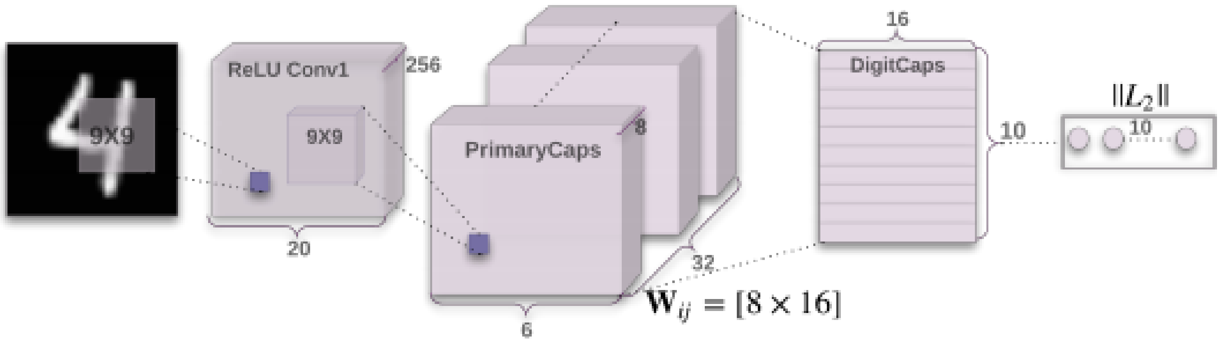
- Requires tons of labeled data to yield state of the art results

This work shows:

- State of the art performance in unsupervised 3D point cloud registration, reconstruction and classification

General Background

Capsule networks



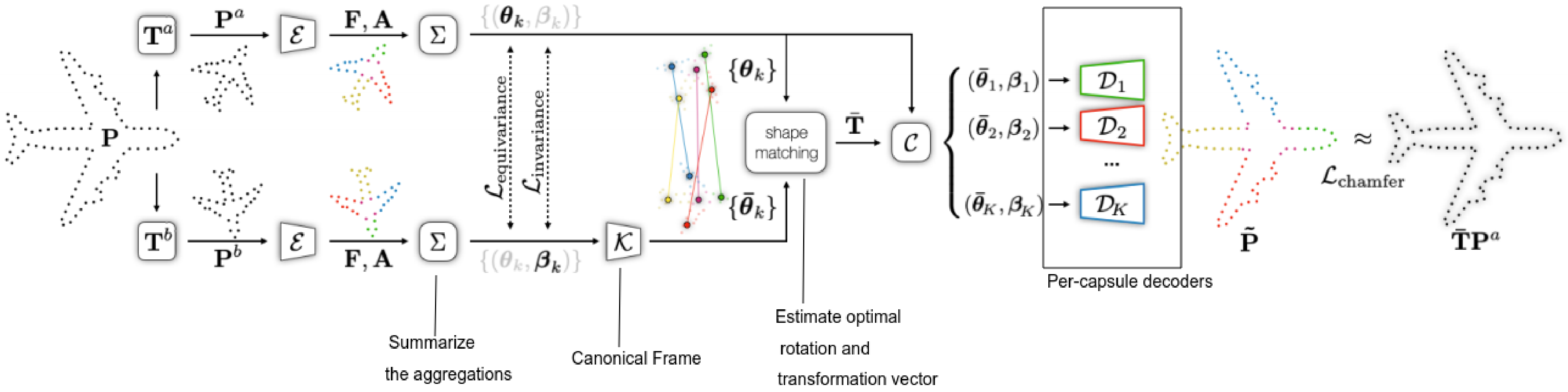
References:

- Dynamic Routing Between Capsules, Sabour et al,. 2017

Notation

- Point cloud $\mathbf{P} \in \mathbb{R}^{P \times D}$
- Random transformations $\mathbf{T}^a, \mathbf{T}^b \in \mathbf{SE}(D)$
- Point clouds after transformation $\mathbf{P}^a, \mathbf{P}^b$
- Capsule Encoder $\mathcal{E}$
- K-fold attention map $\mathbf{A} \in \mathbb{R}^{P \times K}$
- Per-point feature map $\mathbf{F} \in \mathbb{R}^{P \times C}$
- K-th capsule pose $\boldsymbol{\theta}_k \in \mathbb{R}^3$
- Capsule descriptor $\boldsymbol{\beta}_k \in \mathbb{R}^C$

Approach Overview



where $\mathbf{A}, \mathbf{F} = \mathcal{E}(\mathbf{P})$

Method

Decompositions

Pose estimation

$$\boldsymbol{\theta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{P}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Descriptor estimation

$$\boldsymbol{\beta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{F}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Canonicalization

$$\bar{\boldsymbol{\theta}} = \mathcal{K}\left(\boldsymbol{\beta}\right)$$

Autoencoder

$$\tilde{\mathbf{P}} = \cup_k \left\{\mathcal{D}_k(\bar{\mathbf{R}}\boldsymbol{\theta}_k + \bar{\mathbf{t}}, \boldsymbol{\beta}_k)\right\}$$

References:

- Olga Sorkine-Hornung and Michael Rabinovich. LeastSquares Rigid Motion Using SVD

Method

Decompositions

Pose estimation

$$\boldsymbol{\theta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{P}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Descriptor estimation

$$\boldsymbol{\beta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{F}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Canonicalization

$$\bar{\boldsymbol{\theta}} = \boxed{\mathcal{K}}(\boldsymbol{\beta})$$

K is a fully connected network

Autoencoder

$$\tilde{\mathbf{P}} = \cup_k \left\{ \mathcal{D}_k(\bar{\mathbf{R}}\boldsymbol{\theta}_k + \bar{\mathbf{t}}, \boldsymbol{\beta}_k) \right\}$$

References:

- Olga Sorkine-Hornung and Michael Rabinovich. LeastSquares Rigid Motion Using SVD

Method

Decompositions

Pose estimation

$$\boldsymbol{\theta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{P}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Descriptor estimation

$$\boldsymbol{\beta}_k = \frac{\sum_p \mathbf{A}_{p,k} \boxed{\mathbf{F}_p}}{\sum_p \mathbf{A}_{p,k}}$$

Canonicalization

$$\bar{\boldsymbol{\theta}} = \boxed{\mathcal{K}}(\boldsymbol{\beta})$$

K is a fully connected network

Autoencoder

$$\tilde{\mathbf{P}} = \cup_k \left\{ \mathcal{D}_k \boxed{(\bar{\mathbf{R}}\boldsymbol{\theta}_k + \bar{\mathbf{t}}, \boldsymbol{\beta}_k)} \right\}$$

Decoder's input

References:

- Olga Sorkine-Hornung and Michael Rabinovich. LeastSquares Rigid Motion Using SVD

Loss Function

Decomposition Losses

Equivariance

$$\mathcal{L}_{\text{equivariance}} = \frac{1}{K} \sum_k \|\boldsymbol{\theta}_k^a - (\mathbf{T}^a)(\mathbf{T}^b)^{-1} \boldsymbol{\theta}_k^b\|_2^2 \, .$$

Invariance

$$\mathcal{L}_{\text{invariance}} = \frac{1}{K} \sum_k \|\boldsymbol{\beta}_k^a - \boldsymbol{\beta}_k^b\|_2^2 \, .$$

Equilibrium

$$\mathcal{L}_{\text{equilibrium}} = \frac{1}{K} \sum_k \|a_k - \tfrac{1}{K} \Sigma_k a_k\|_2^2$$

Localization

$$\mathcal{L}_{\text{localization}} = \frac{1}{K} \sum_k \frac{1}{a_k} \sum_p \mathbf{A}_{p,k} \|\boldsymbol{\theta}_k - \mathbf{P}_p\|_2^2$$

Loss Function

Canonicalization loss

Canonical $\mathcal{L}_{\text{canonical}} = \frac{1}{K}\sum_k \|(\bar{\mathbf{R}}\boldsymbol{\theta}_k + \bar{\mathbf{t}}) - \bar{\boldsymbol{\theta}}_k\|_2^2 \,.$

Reconstruction loss

Reconstruction $\mathcal{L}_{\text{recon}} = \text{CD}\left(\bar{\mathbf{R}}\mathbf{P} + \bar{\mathbf{t}},\, \tilde{\mathbf{P}}\right) \,.$

The loss functions are employed to train the **encoder**, the **decoder** and a network that represents a learnt **canonical frame** in an unsupervised fashion

Experimental Setup

Datasets

Shapenet (Core)
31747 shapes for training, and 7943 shapes for testing

For single-category experiments, they use:

- the airplane class
- the chair classes

All 13 object classes are used for multi-category experiments

References:

- Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An InformationRich 3D Model Repository

Experimental Setup

Baselines

Auto-encoder evaluation:
- 3D-PointCapsNet[1]
- AtlasNetV2[2]

Registration:
- Deep Closest Points (DCP)[3]
- DeepGMR–RRI[4]
- DeepGMR–XYZ[4]

References:

1. Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3D Point Capsule Networks
2. Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning Elementary Structures for 3D Shape Generation and Matching
3. Yue Wang and Justin M Solomon. Deep Closest Point: Learning Representations for Point Cloud Registration
4. Wentao Yuan, Ben Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. DeepGMR: Learning Latent Gaussian Mixture Models for Registration

Experimental Results

Autoencoder performance

	Aligned			Unaligned		
	Airplane	Chair	Multi	Airplane	Chair	Multi
3D-PointCapsNet [58]	1.94	3.30	2.49	5.58	7.57	4.66
AtlasNetV2 [13]	1.28	2.36	2.14	2.80	3.98	3.08
Our method	**0.96**	**1.99**	**1.76**	**1.08**	**2.65**	**2.25**

Auto-encoding/reconstruction perfromance in terms of Chamfer distance

Experimental Results

Registration performance

	Airplane	Chair	Multi
Deep Closest Points [52]	0.318	0.160	0.131
DeepGMR–XYZ [56]	0.079	0.082	0.077
Our method–XYZ	**0.024**	**0.027**	**0.070**
DeepGMR–RRI [56]	**0.0001**	**0.0001**	**0.0001**
Our method–RRI	0.0006	0.0009	0.0016

Performance in terms of root mean-square error between registered and ground-truth points

Experimental Results

Classification performance

	Aligned		Unaligned	
	SVM	K-Means	SVM	K-Means
AtlasNetV2	94.07	61.66	71.13	14.59
3D-PointCapsNet	93.81	65.87	64.85	17.12
Our method	**94.21**	**69.82**	**87.17**	**43.86**

Unsupervised classification using features extracted from the auto-encoder

Qualitative Results

Ablation Study

Number of points

	1024 pts	2500 pts
3D-PointCapsNet [58]	2.49	1.49
AtlasNetV2 [13]	2.14	1.22
Our method	**1.76**	**0.97**

Loss effect

	Full	$\neg\mathcal{L}_{\text{invar}}$	$\neg\mathcal{L}_{\text{canonical}}$	$\neg\mathcal{L}_{\text{equiv}}$	$\neg\mathcal{L}_{\text{localization}}$	$\neg\mathcal{L}_{\text{equilibrium}}$
CD	**1.08**	1.09	1.09	1.16	1.45	1.61

Discussion of results

- This work achieves state of the art performance in autoencoder, point cloud registration and classification

- On pre-aligned data, this work achieves comparable performance with prior work but in the case of unaligned data, they outperform past work by a large margin

	Aligned			Unaligned		
	Airplane	Chair	Multi	Airplane	Chair	Multi
3D-PointCapsNet [58]	1.94	3.30	2.49	5.58	7.57	4.66
AtlasNetV2 [13]	1.28	2.36	2.14	2.80	3.98	3.08
Our method	**0.96**	**1.99**	**1.76**	**1.08**	**2.65**	**2.25**

	Aligned		**Unaligned**	
	SVM	K-Means	SVM	K-Means
AtlasNetV2	94.07	61.66	71.13	14.59
3D-PointCapsNet	93.81	65.87	64.85	17.12
Our method	**94.21**	**69.82**	**87.17**	**43.86**

Limitations

- Point clouds are the only input allowed
- Experimentally selecting the number of capsules used
- This framework has not tested on scenes with multiple or occluded objects

Contributions (Recap)

This work proposes :

- Unsupervised learning on 3D point clouds using capsules
- Object-centric unsupervised learning

Past work:

- Requires tons of labeled data to yield state of the art results

This work shows:

- State of the art performance in unsupervised 3D point cloud registration, reconstruction and classification