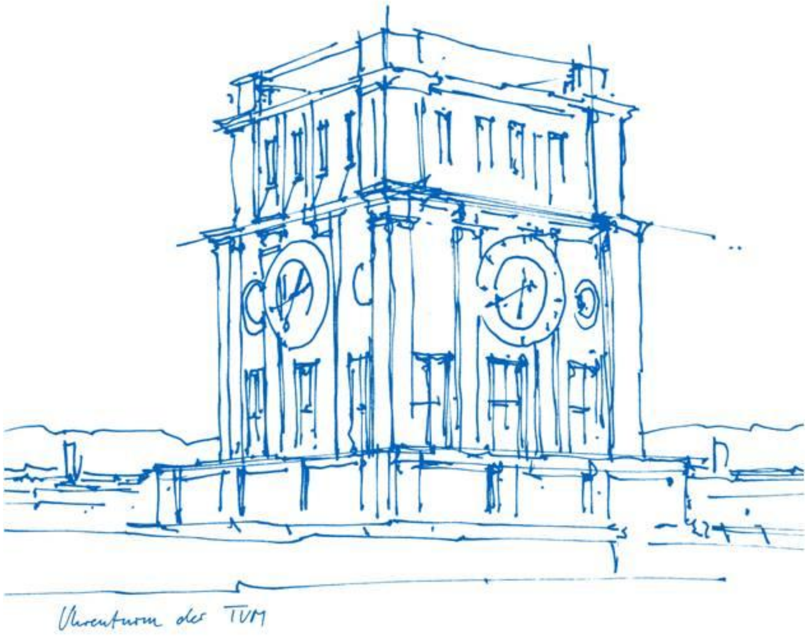
Exploring Deep Closest Point: Learning Representations for Point Cloud Registration

Machine Learning for 3D Geometry



Kerem Yildirir

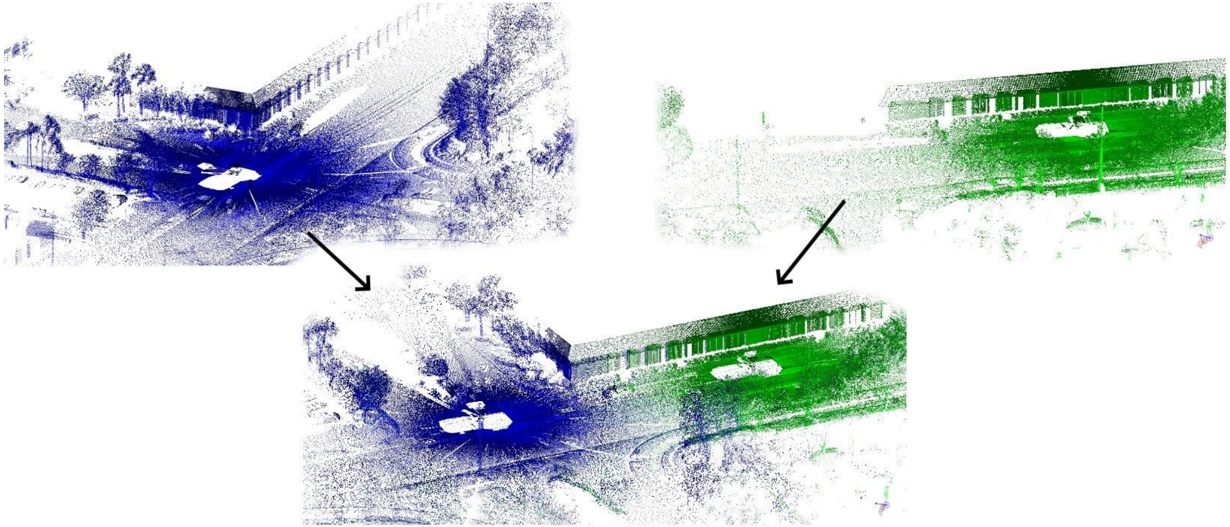
Kaan Oguzhan

Baris Sen

Yigit Aras Tunali

Problem Definition

- **Align** two point clouds
- Find a **rigid transformation** -> **globally** consistent
- Might have **noise**, **occlusions**



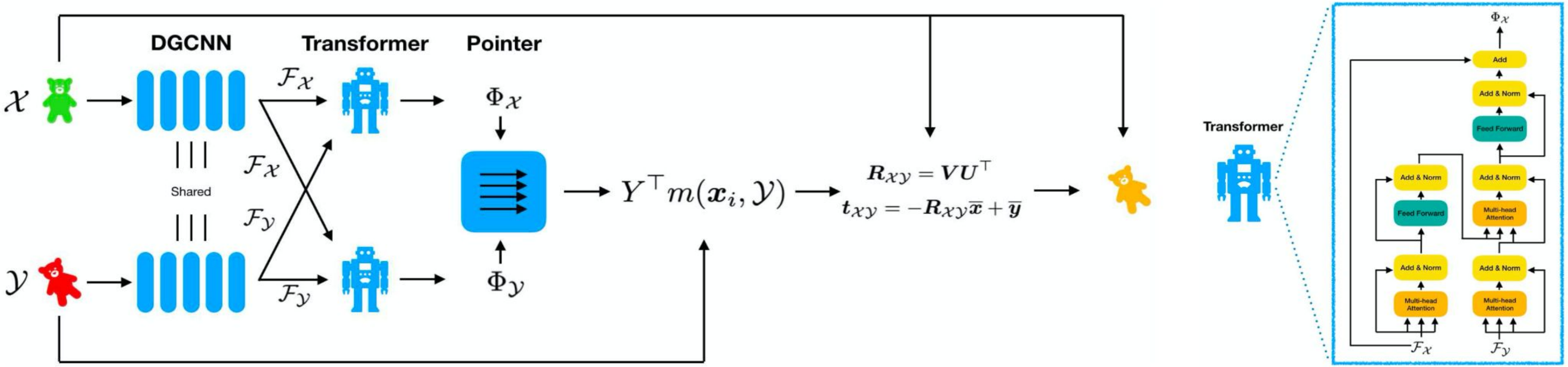
Source: https://new.certainty3d.com/blog/what-is-point-cloud-registration/

Related Work

- Traditional approach:
  - **Iterative Closest Point** (ICP) [Besl and McKay, 1992]
- Deep learning approach
  - Extract **point embeddings**
  - Find **corresponding** points
  - Estimate **transformation** (rotation and translation)
- **Deep Closest Point** [Wang and Solomon, 2019]
- Generate better point embeddings [Kadam et al. 2021]
- Better point matching [Choy et al. 2020, Sarlin et al. 2020]
- Soft assignments, weighted SVD [Yan et al. 2019]

Deep Closest Point (DCP)

- **Feature extraction:** Dynamic Graph Convolutional Neural Networks (DGCNN) [Wang et al. 2019]
- **Point matching:** Transformer [Vaswani et al. 2017]
- **Transformation estimation:** Differentiable SVD

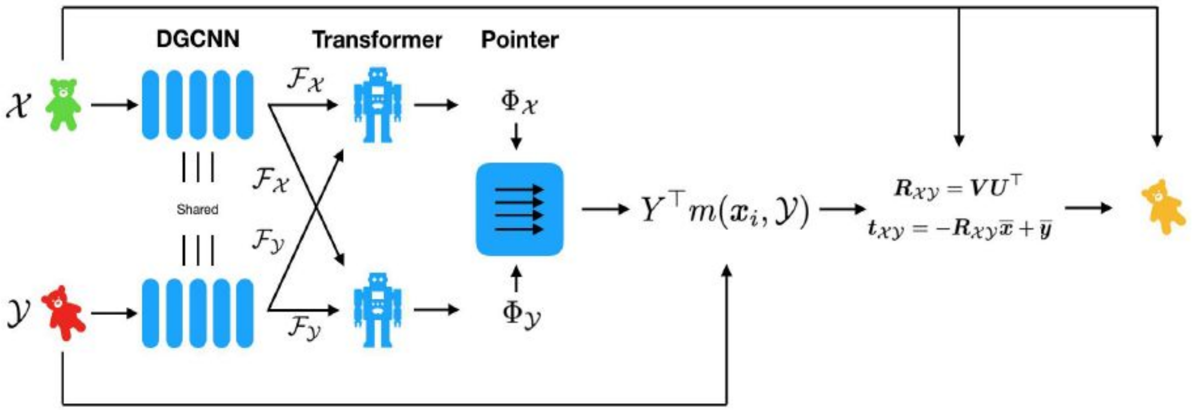


(a) Network architecture (b) Transformer module

Source: [Wang and Solomon, 2019]

DGCNN

- Generate features per input point
- DGCNN explicitly incorporates local geometry (compared to PointNet)
- Local features from DGCNN are critical for high quality matching



Transformer

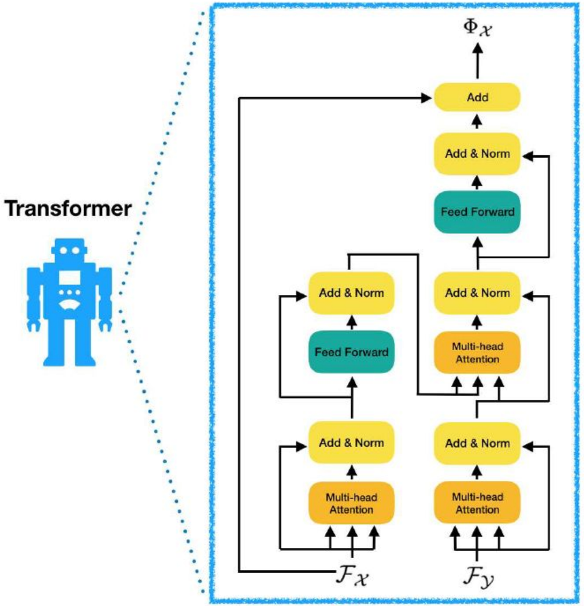
- Module to learn co-contextual information
- Attention model learns asymmetric function:

$$\phi : \mathbb{R}^{N \times P} \times \mathbb{R}^{N \times P} \rightarrow \mathbb{R}^{N \times P}$$

- Then this function is used as a residual on features:

$$\Phi_{\mathcal{X}} = \mathcal{F}_{\mathcal{X}} + \phi(\mathcal{F}_{\mathcal{X}}, \mathcal{F}_{\mathcal{Y}})$$
$$\Phi_{\mathcal{Y}} = \mathcal{F}_{\mathcal{Y}} + \phi(\mathcal{F}_{\mathcal{Y}}, \mathcal{F}_{\mathcal{X}})$$

- Essentially adds knowledge about other points



Pointer Module

- To avoid non-differentiable hard assignments
- Generate singly-stochastic "soft map" from one point cloud to another
- "m" is a soft pointer from each element of X into elements of Y

$$m(\boldsymbol{x}_i, \mathcal{Y}) = \mathrm{softmax}(\Phi_{\mathcal{Y}} \Phi_{\boldsymbol{x}_i}^\top)$$

SVD Module

- Use soft pointer to generate a matching averaged point in Y
- R, t extracted using SVD over pairings of X to predicted Y
- Differentiation of SVD as in Papadopoulo et al. (included in Pytorch)

$$\hat{y}_i = Y^\top m(x_i, \mathcal{Y}) \in \mathbb{R}^3$$

Approach: Additions

- Additions to DCP
  - Adding **point color** as an additional input signal
  - **Independent sampling** of source and target point clouds
  - **As-Rigid-As-Possible (ARAP) Regularization** as additional loss

Approach: Datasets

- **Mixamo** [Adobe, 2018]
  - Training Set
  - Validation Set

- **TUM RGBD** [Sturm et al., 2012]
  - Test Set

Additional Color Input

- Color as a **strong signal** for matching points
- Extend the input vector from **[X, Y, Z]** to **[X, Y, Z, R, G, B]**
- Keep the rest of the architecture **fixed**

Experiments: Additional Color Input

Mixamo	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP	51.460548	7.173	4.399	0.043476	0.208509	0.075243
DCP + Color	**0.171919**	**0.4114631**	**0.106271**	**0.000035**	**0.005910**	**0.001755**
DCP + Color + Noise	0.874181	0.934976	0.569145	0.000256	0.015988	0.007239

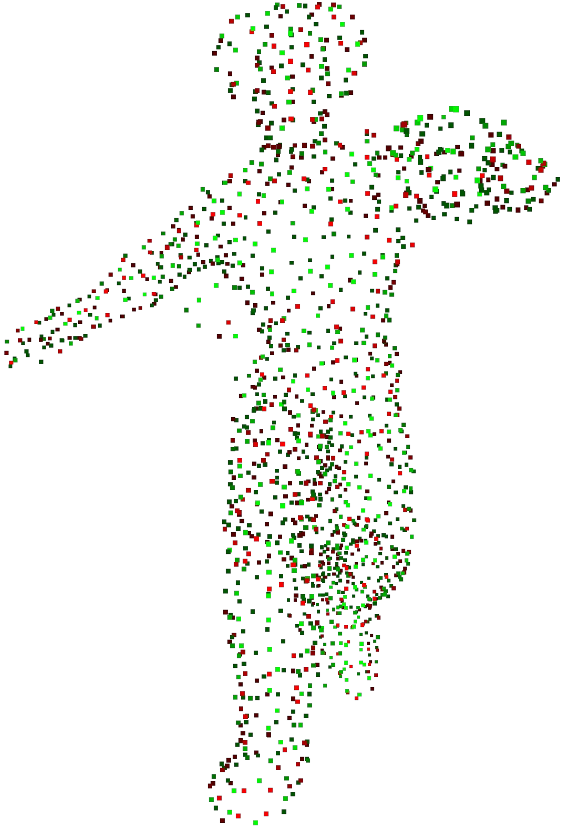
TUM RGBD	MSE(R)	RMSE(R)	MAE(R)	MSE(t)	RMSE(t)	MAE(t)
DCP	387.982239	19.697266	11.2857	0.029045	0.170427	0.113288
DCP + Color	**0.24021**	**0.490117**	**0.334967**	**0.000094**	**0.00967**	**0.007570**
DCP + Color + Noise	0.315961	0.562104	0.373782	0.000101	0.010030	0.008009

Original point sampling

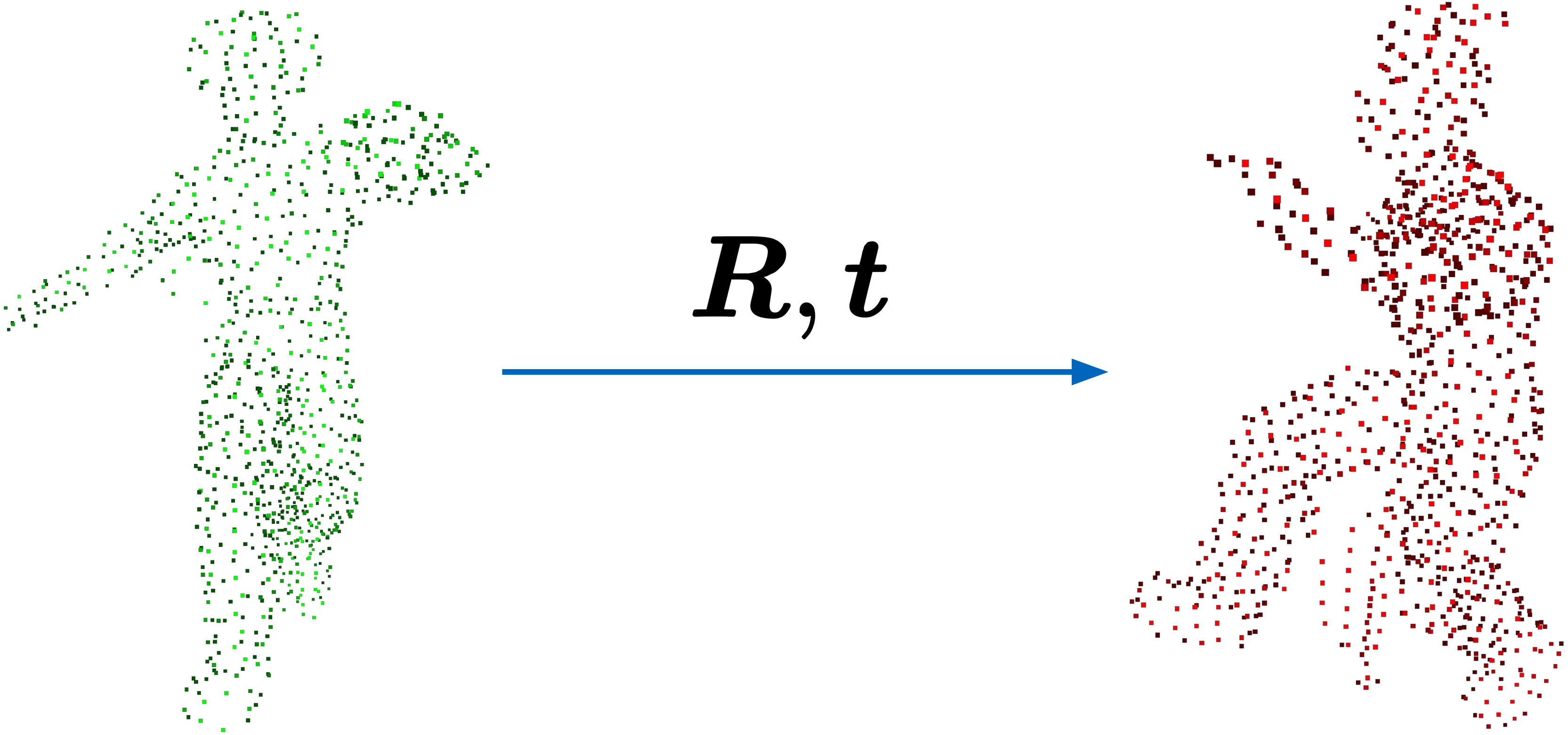
- Training data generation
  - Source: **Take N** points
  - Target: **Randomly apply rotation and translation**

- **One-to-one** correspondence of point

Independent Sampling



Independent Sampling



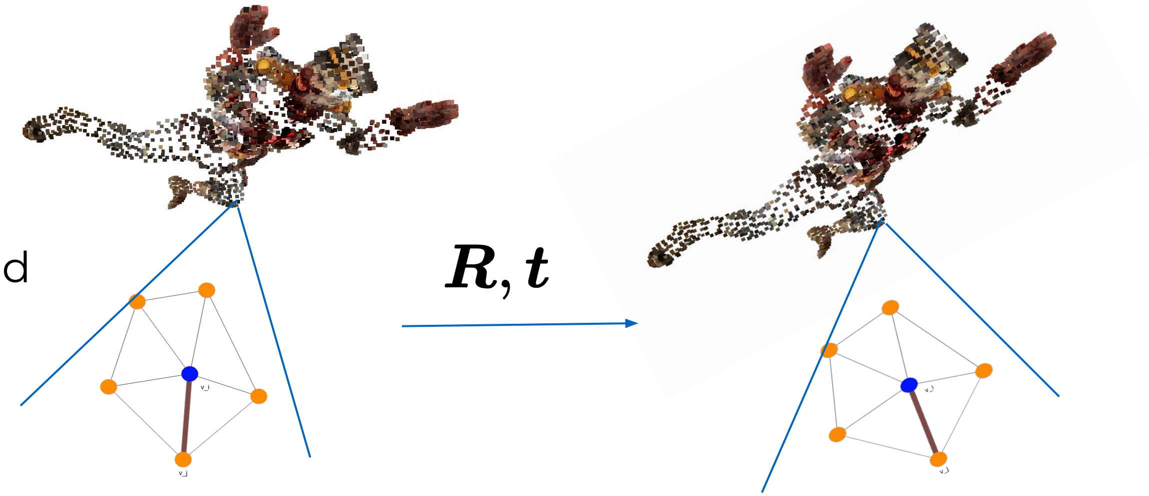
Experiments: Independent Sampling

Mixamo	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
DCP	53.342667	7.303607	4.438653	0.048586	0.220422	0.077097
DCP + Independent Sampling	**44.15807**	**6.645154**	**4.342775**	**0.030093**	**0.173474**	**0.067707**

TUM RGBD	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
DCP	390.456390	19.759970	11.853582	0.030235	0.173882	**0.119560**
DCP + Independent Sampling	**256.066223**	**16.002069**	**10.963472**	**0.027706**	**0.166452**	0.120926

ARAP Regularizer [Sorkine and Alexa, 2007]

- As-Rigid-As-Possible
- Punishes bad correspondences
- Neighbour distances should be preserved after transformation
- Additional loss



$$E\left(\mathcal{C}_i, \mathcal{C}_i'\right) = \sum_{j \in \mathcal{N}(i)} \left\| \left(\mathbf{p}_i' - \mathbf{p}_j'\right) - \mathbf{R}\left(\mathbf{p}_i - \mathbf{p}_j\right) \right\|^2$$

Experiments: ARAP Regularizer

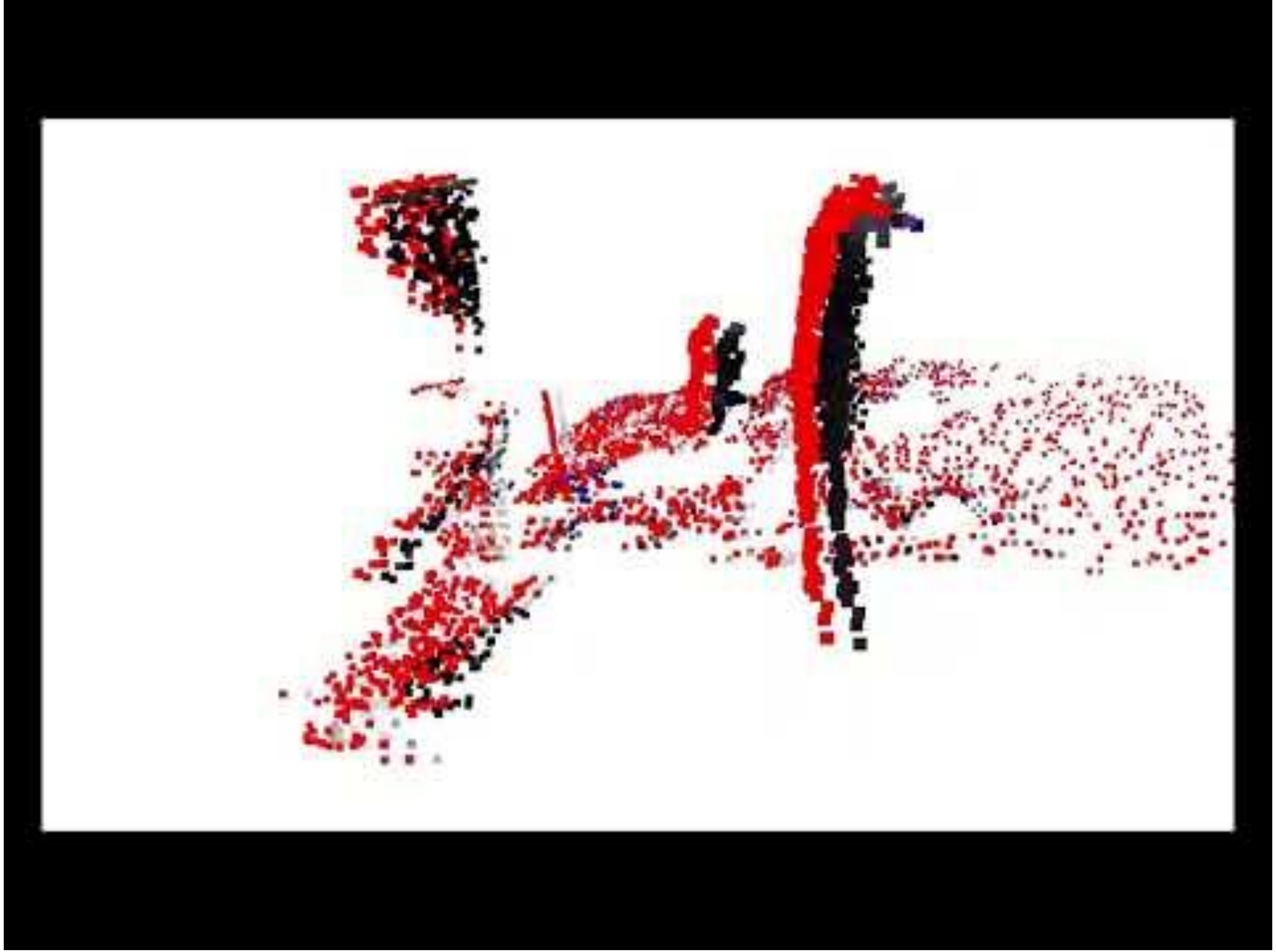
Mixamo	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
DCP	51.460548	7.173601	4.399128	**0.043476**	**0.208509**	**0.075243**
DCP + Arap	**48.080051**	**6.933978**	**4.212567**	0.050855	0.225510	0.076915

Mixamo	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
DCP + Color	0.171919	**0.414631**,	0.106271	**0.000035**	**0.004833**	0.001755
DCP + Color + ARAP	**0.171727**	0.451673	**0.083366**	0.000044	0.006611	**0.001589**

TUM RGBD	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
DCP + Color	0.240215	0.490117	0.334967	0.000094	0.009678	0.007570
DCP + Color + ARAP	**0.169369**	**0.411545**	**0.270199**	**0.000075**	**0.008643**	**0.007129**

Experiments: Final Results

TUM RGBD	MSE($\boldsymbol{R}$)	RMSE($\boldsymbol{R}$)	MAE($\boldsymbol{R}$)	MSE($\boldsymbol{t}$)	RMSE($\boldsymbol{t}$)	MAE($\boldsymbol{t}$)
Original DCP	1.573519	1.254400	0.890428	0.000277	0.016652	0.01259
DCP + Color + ARAP	**0.169369**	**0.411545**	**0.270199**	**0.000075**	**0.008643**	**0.007129**



Original

Ours

Conclusion & Future Work

- **Conclusions:**
  - Color input has the biggest impact
  - ARAP regularization slightly better, not a huge difference
  - Harder to learn if source and target independently sampled
- **Future Work:**
  - Can be made robust to lighting changes
  - Training with noise
  - Uniform sampling from mesh triangles for more even input points
  - Improving the Mixamo dataset by adding more variation and characters, occlusions

References

- Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. CoRR, abs/1905.03304, 2019.
- Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (tog), 38(5):1–12, 2019.
- Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net. ACM Transactions on Graphics, 38(6):1–15, Nov 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In Proc. of the International Conference on Intelligent Robot Systems (IROS), Oct. 2012.
- Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In Proceedings of the Fifth Eurographics Symposium on Geometry Processing, SGP ’07, page 109–116, Goslar, DEU, 2007. Eurographics Association.
- Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks, 2020.
- Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the ICP algorithm. In Proceedings third international conference on 3-D digital imaging and modeling, pages 145–152. IEEE, 2001.
- Pranav Kadam, Min Zhang, Shan Liu, and C. C. Jay Kuo. R-pointhop: A green, accurate and unsupervised point cloud registration method, 2021.
  Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration, 2020.
- Adobe. Mixamo model dataset, 2018.
- Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In Sensor fusion IV: control paradigms and data structures, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.
- Theodore Papadopoulo and Manolis IA Lourakis. Estimating the Jacobian of the singular value decomposition: Theory and applications. In European Conference on Computer Vision, pages 554–570. Springer, 2000

Thank you for your attention [Bahdanau et al., 2014]!