# Locally Rigid Registration for Structural Deformation Monitoring

Caner Korkmaz*[1], Nursena Köprücü†[1], Sinan Açıkgöz[2], and Fatma Güney[1]

[1]KUIS AI Center, Koç University
[2]Department of Engineering, University of Oxford

## Abstract

*Monitoring of large structures for deformations can be performed by aligning the 3D point clouds of the structure scanned at different times. Existing methods based on the Iterative Closest Point algorithm either ignore features due to high outlier ratios or require measurements to adjust the estimations. In this paper, we propose to use a deep registration framework, not only for learning discriminative features but also their associated confidences. The confidences are directly informed by the registration in an end-to-end manner, resulting in state-of-the-art performance without any extraneous measurements. Our work is the first to use learned features and deep registration models for deformation monitoring; therefore, we plan to share our data, code, and the models to benefit further research.*

## 1. Introduction

Undesired deformations of civil engineering structures such as historic assets may arise due to nearby construction activities. In current engineering practice, monitoring of slow (e.g. static) deformations is conducted using well-established surveying techniques. These techniques require the use of electronic distance measurement devices, such as total stations, which track the position of reflective targets installed on the structure over time. However, in some cases, it is not possible to place targets on the structure and non-contact techniques may be required.

Non-contact deformation monitoring can be achieved by comparing 3D point clouds of the structure, acquired at different times by using laser scanning. Comparing point clouds requires identifying their corresponding sections and estimating the deformations between them by registering them to one another. While structural deformations are globally non-rigid, they can be estimated point-wise by locally rigid registrations of small point cloud patches.

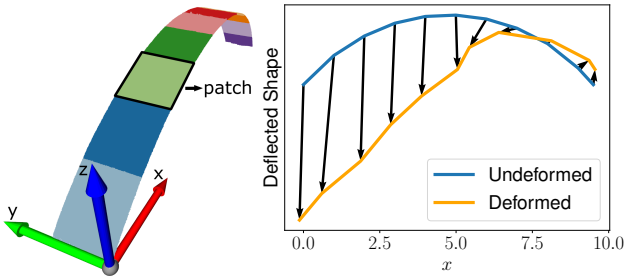


Figure 1: The target L1 arch **(left)** and the deflected shape **(right)** where deformations are scaled by $\times 100$.

Within this context, traditional algorithms, such as the Piecewise Alignment Method (PAM) [15], perform rigid registration of small point cloud segments using the Iterative Closest Point (ICP) algorithm. The ICP algorithm is based on establishing correspondences between point clouds using point-to-point distances. As such, this technique suffers from the arbitrary matching of points and predicts deformation fields that are very different from the total station measurements that can be considered as ground truth. The Iterative PAM (IPAM) [3] introduces handcrafted constraints to PAM to prevent arbitrary matching of points. IPAM results in a successful registration; however, it relies on control points with known deformations.

While traditional algorithms for structural health monitoring are based on ICP, there is a paradigm shift for point cloud registration in computer vision. Deep learning methods provide not only better correspondences with reliable confidence measures [7] but also registrations free from local optima problem [4], typically encountered in ICP methods and variants. These methods have not been utilized for deformation monitoring in structures despite their success in registering CAD models of objects [16], point clouds of room layouts [4], and outdoor scenes [11]. Deformation monitoring presents new challenges for these methods. First, instead of a single global transformation, in-

*ckorkmaz16@ku.edu.tr (equal contribution)
†nkoprucu16@ku.edu.tr (equal contribution)

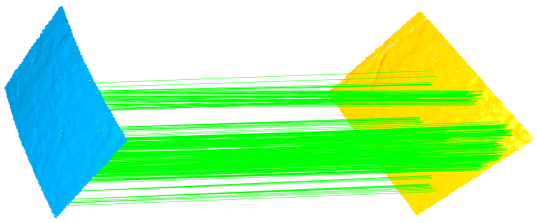

Figure 2: **Visualization of Correspondences.** This figure shows correspondences between two patches with high confidence values ($> 0.9$) as predicted by our registration method. Despite little information, our algorithm produces accurate correspondences distributed throughout the patch.

dependent transformations need to be determined for local regions. Second, expected deformations tend to be significantly smaller than other applications, typically in the order of millimeters or centimeters.

In this paper, we propose a learning-based method to address these challenges. We process the point cloud of the structure by extracting small patches. We assume deformations within a patch to be the result of the same rigid transformation and propose to learn these transformations independently in a sliding window manner. We first extract deep geometric features on patches [7] and identify a set of initial correspondences based on these features. We then learn the reliability of these correspondences together with the registration [4]. Focusing on these small patches allows us to achieve a high deformation sensitivity that would otherwise be hard to detect. We evaluate our method on georeferenced point cloud data from arched viaducts in London Bridge Station, which experienced structural deformations. Our method achieves better results than PAM and comparable results to IPAM without using any control points.

## 2. Methodology

We perform pairwise registration of point clouds scanned at different times to estimate structural deformations that occurred between these times with high accuracy.

**Feature Extraction:** For a successful registration, the first step is to establish reliable correspondences based on some features of the point cloud. Recently, learning-based methods like Fully Convolutional Geometric Features (FCGF) [7] have shown remarkable performance by processing the entire input point cloud with fully convolutional layers. Similarly, we use a 3D fully-convolutional network for extracting geometric features from 3D point clouds. By adapting the sparse tensor representation and metric learning losses introduced by Choy et al. [7], we can extract accurate features in a fast and memory-efficient way as shown in Fig. 2. These point-wise features summarize the geometric context around each point into a low-dimensional feature vector to ensure the efficiency of the following steps.

Our goal is to register two point clouds $\mathcal{X}^t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_N^t\} \in \mathbb{R}^{3\times N}$ and $\mathcal{X}^{t'} = \{\mathbf{x}_1^{t'}, \dots, \mathbf{x}_M^{t'}\} \in \mathbb{R}^{3\times M}$, taken at time $t$ and $t'$ with $N$ and $M$ points, respectively. We establish an initial set of correspondences based on the nearest neighbor search in the learned feature space. This step already provides a set of correspondences that can be filtered out for noise and used for registration as shown in the baselines in Section 3.2. However, as in Deep Global Registration (DGR) [4], we train a confidence network to predict the probability of being an inlier for a pair of points $(\mathbf{x}_i^t, \mathbf{x}_j^{t'})$. We use the same U-Net architecture with residual blocks as proposed by Choy et al. [4] and train it with a Binary Cross Entropy (BCE) loss. Positive correspondences are defined according to point-to-point distances up to a threshold. As the ratio of inliers over outliers is very small, we experiment with balancing the positives and negatives by weighting the BCE loss which results in better performance.

### 2.1. Locally Rigid Registration

In an end-to-end registration pipeline, we optimize for the weights $w_{i,j}$ for each pair of points $(\mathbf{x}_i^t, \mathbf{x}_j^{t'})$ in the initial set of correspondences $\mathcal{C}$. In particular, we use the Weighted Procrustes method that minimizes the weighted mean squared error to find the optimal rotation $\hat{\mathbf{R}}$ and the translation $\hat{\mathbf{t}}$ corresponding to the best alignment:

$$\begin{aligned} \hat{\mathbf{R}}, \hat{\mathbf{t}} &= \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i,j\in\mathcal{C}} w_{ij} \, ||\mathbf{x}_i^t - \mathbf{x}_j^{t'}||^2 \\ &= \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i,j\in\mathcal{C}} w_{ij} \, (\mathbf{x}_j^{t'} - (\mathbf{R}\,\mathbf{x}_i^t + \mathbf{t}))^2 \end{aligned} \tag{1}$$

There is a closed-form solution to find the optimal $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ in Eq. 1 as a weighted least squares transformation [14, 9, 4]. Since Weighted Procrustes is differentiable, gradients can be passed to the weights to learn confidence values informed by the registration. In addition to the BCE for learning confidence values, there is a differentiable rotation and translation error in the final loss function as in DGR.

We use a sliding window to extract patches with matching center locations from the target arch pairs at different dates and calculate the registration for each pair independently. This is different from IPAM, which sequentially processes neighboring patches based on the results of the previous patches. Our approach is parallelizable and avoids introducing dependence by the order of processing.

**Robust Registration and Fine-tuning:** We improve the results of the Weighted Procrustes by first applying the robust registration method from DGR and then fine-tuning with ICP. Starting from the result of the Weighted Procrustes, the robust registration minimizes an energy func-

tion based on the sum of weighted robust Huber losses using a continuous representation of rotations to remove discontinuities. We also experiment with ICP for further refinement by using a small maximum correspondence distance, i.e. twice the voxel size as proposed in DGR.

### 2.2. From Registration to Deformations

Given the result of registration as the rotation matrix $\hat{\mathbf{R}}$ and the translation vector $\hat{\mathbf{t}}$, we calculate the displacement vector $\Delta\mathbf{x} = [\Delta x, \Delta y, \Delta z]^T$ according to the movement of patch centers [15, 3] as follows:

$$\begin{bmatrix} \Delta\mathbf{x} \\ 0 \end{bmatrix} = \left( \mathbf{I}_4 - \begin{bmatrix} \hat{\mathbf{R}} & \hat{\mathbf{t}} \\ \mathbf{0} & 1 \end{bmatrix} \right) \begin{bmatrix} \bar{\mathbf{x}} \\ 1 \end{bmatrix} \tag{2}$$

where $\bar{\mathbf{x}}$ is the spatial average of the points in the patch containing deformations and $\mathbf{I}_4$ is the $4 \times 4$ identity matrix. Then, the first ($\Delta x$) and the last ($\Delta z$) components of the displacement vector correspond to the lateral and vertical displacements (left and middle in Fig. 3, respectively).

## 3. Experiments

**Model Details:** Our models are based on the original FCGF and DGR [5, 7, 4]. For feature learning and confidence estimation, we respectively use 3D and 6D versions of the ResUNet architecture from [6], a sparse U-shaped network [12] with generalized convolutions [5, 7] and residual connections [10] in the form of short and long skip connections. We use 7 channels in the first layers of the models, 32 for the feature dimension and $0.01$m for the voxel size. The feature model was trained using the hardest contrastive loss with normalized features.

### 3.1. Experimental Setup

We evaluated our approach on 3D point cloud scans of E57 London Bridge Station [3]. We used five additional bridge scans for training the feature model. We compare our approach to the previous work on the L1 Arch (Fig. 1) and exclude it from training. When training the feature model, we obtained the input pairs by first extracting a randomly oriented cube of size $1\text{m}^3$ from a random location in one of the scans and then extracting random one of its 18-connected neighbors to obtain a second cube of the same size. We found the sampling distance between the two cubes, *the interval*, to be an important parameter as it affects the overlap ratio between the cubes. Like FCGF, we aimed for at least $30\%$ overlap ratio and experimented with intervals of $0.5$m and $0.25$m in the direction of axes when using the $1\text{m}^3$ cubes. For training, we used random rotations between $-45$ and $45$ degrees around a random axis.

For training DGR, we experiment with two patch extraction methods; DGR-Temporal (DGR-T) similar to evaluation, at the same random location from different times, and DGR-Spatial (DGR-S) similar to FCGF, from different spatial locations on the same scan. For evaluation and DGR-T training, we used geo-referenced undeformed and deformed point clouds with the same X and Y coordinates.

| **Feature** | **Registration** | **Fine-tuning** | **RMSE** ($\Delta z$) |
|---|---|---|---|
| FPFH [13] | Robust | - | 29.52 |
| FCGF [7] | Robust | - | 19.5185 |
| FPFH [13] | FGR [17] | - | 16.7135 |
| FCGF [7] | FGR [17] | - | 15.0478 |
| - | PAM [15] | - | 9.7728 |
| - | PAM (Ours) | - | 7.4449 |
| FCGF [7] | DGR-T | - | 2.6986 |
| | DGR-S | - | 2.1518 |
| | DGR-T | ✓ | 2.062 |
| | DGR-S | ✓ | **1.5822** |

Table 1: **Comparison to Baselines.** This table compares our methods DGR-T and DGR-S to several baselines with different feature representations and registration methods.

### 3.2. Results

We perform comparisons according to the vertical displacements $\Delta z$, lateral displacements $\Delta x$, and rotation $\Delta\theta_y$. We also plot the deflected shapes on the L1 arch as shown on the right in Fig. 1. We report the RMSE results using only $\Delta z$, which is the dominant direction of movement. Although there are total station measurements that can be regarded as ground truth, they are very scarce, i.e. only on three locations on the arch. Therefore, we compute the RMSE of the methods with respect to IPAM, which is conditioned by and agrees with total station measurements. We report the results of both the original and our version of PAM with our extracted patches and Open3D implementation of ICP, which results in slightly better performance.

In Table 1, we first evaluate the performance of our learned features FCGF in comparison to Fast Point Feature Histogram (FPFH) features [13] by using two registration algorithms, Fast Global Registration (FGR) [17] and a robust iterative re-weighted least-square optimization-based registration (Robust) [7]. Both features perform better with FGR, and learned features FCGF outperforms FPFH in each case. Feature-based methods without any outlier filtering perform worse than PAM due to ambiguities in matching. FCGF features within the DGR framework perform significantly better due to learned outlier filtering via confidence measures. DGR-S, in which the pairs are extracted from the same scan, performs better than the DGR-T due to exact correspondences in the overlapping area. ICP fine-tuning with a maximum correspondence distance of $0.02$m, i.e. twice the voxel size as in DGR, improve the results for both DGR-based methods.

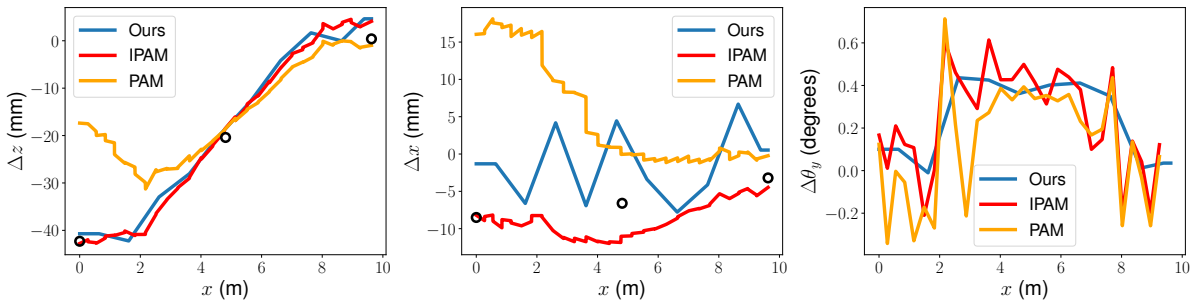


Figure 3: **Comparison to Existing Methods.** In this plot, we compare our displacement results to PAM and IPAM on the L1 Arch in terms of vertical displacements **(left)**, lateral displacements **(middle)** and in-plane rotations **(right)**. Our results are close to the total station measurements marked as circles, in agreement with IPAM which relies on measurement values.

| Interval | Fine-tuning | RMSE ($\Delta z$) |
|---|---|---|
| 0.5 | - | 5.7856 |
| | ✓ | 7.2675 |
| 0.25 | - | 3.0792 |
| | ✓ | **2.6290** |

Table 2: **The Effect of Interval.** This table compares the two values of the interval parameter using the DGR-T.

In Figure 3, we compare our method to PAM and IPAM by plotting displacements. Our $\Delta z$ displacements are consistently closer to IPAM than PAM without imposing any constraints. Moreover, our $\Delta x$ displacements are overall better than PAM and $\Delta\theta_y$ rotation values mostly agree for all methods. As shown on the left in Fig. 3, the sharp gradient changes in $\Delta z$ vs. $x$ plot around the ordinates $x = 2$m and $x = 8$m is captured well by our method. These indicate hinge formations, where structural damage is concentrated.

### 3.3. Ablation Study

In Table 2, we compare two different values for the interval parameter as 0.5m and 0.25m. While both still perform better than the PAM results reported in Table 1, a smaller interval value leads to a higher overlap ratio and results in significantly better results, both with and without fine-tuning. The task becomes easier with high overlap ratios, but also, results are better because the train and test time conditions align more. During testing, we compare patches from the same location on the arch but from different times.

We perform an ablation study with other important hyper-parameters including balancing in the BCE loss and the weight of Procrustes loss. The results are shown in Table 3 for both DGR-T and DGR-S with and without fine-tuning with ICP. Due to the high ratio of outliers in the feature matching, balancing in the BCE loss improves the results noticeably both for DGR-S and DGR-T. Doubling the weight of Procrustes further improves the results by shifting the focus from confidences to the quality of registration,

| DGR | BCE Balanced | Procrustes Weight | Fine -tuning | RMSE ($\Delta z$) |
|---|---|---|---|---|
| T | - | $\times 1$ | | 2.9468 |
| | ✓ | $\times 1$ | - | 2.6986 |
| | ✓ | $\times 2$ | | 3.0792 |
| | - | $\times 1$ | | 2.4895 |
| | ✓ | $\times 1$ | ✓ | 2.0262 |
| | ✓ | $\times 2$ | | 2.6290 |
| S | - | $\times 1$ | | 3.0225 |
| | ✓ | $\times 1$ | - | 2.1911 |
| | ✓ | $\times 2$ | | 2.1518 |
| | - | $\times 1$ | | 3.0160 |
| | ✓ | $\times 1$ | ✓ | 2.0400 |
| | ✓ | $\times 2$ | | **1.5822** |

Table 3: **The Effect of Loss Parameters.** This table shows the effect of balancing the BCE loss and increasing the weight of Procrustes loss for both DGR-T and DGR-S.

leading to the best performance with DGR-S. Fine-tuning with ICP consistently improves the results in all cases.

## 4. Conclusion

We adapted an end-to-end deep registration framework for deformation monitoring. We first extract geometric features on the point cloud and then learn their confidences together with the registration. Our method outperforms a traditional method based on ICP and performs similarly to the state-of-the-art method that requires extraneous information from other measurements. Although high accuracy levels can be achieved by focusing on small regions, we suspect that errors arise due to the independent processing of local patches. In future, we plan to relate our local estimations to each other in a global optimization framework to capture global non-rigid deformation patterns. We also plan to apply our algorithm to other data which features different geometries and deformation patterns [1, 8, 2].

## References


[1] Sinan Acikgoz, Antonio Luciano, Mark Dewhirst, Matthew J. Dejong, and Robert Mair. Innovative monitoring of the response of a heritage masonry building to nearby tunnelling in london clay. *Géotechnique*, pages 1–16, 2021. 4

[2] Sinan Acikgoz, Loizos Pelecanos, Giorgia Giardina, James Aitken, and Kenichi Soga. Distributed sensing of a masonry vault during nearby piling. *Structural Control and Health Monitoring*, 24(3):e1872, 2017. 4

[3] Sinan Acikgoz, Kenichi Soga, and Jim Woodhams. Evaluation of the response of a vaulted masonry structure to differential settlements using point cloud data and limit analyses. *Constr. Build. Mater.*, 150:916–931, 2017. 1, 3

[4] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3

[5] Christopher Choy, Junyoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[6] Christopher Choy, Junha Lee, René Ranftl, Jaesik Park, and Vladlen Koltun. High-dimensional convolutional networks for geometric pattern recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2, 3

[8] K. Deniz Dalgic, D. Burcu Gulen, Sinan Acikgoz, Harvey Burd, Max A. N. Hendriks, Giorgia Giardina, and Alper Ilki. Large scale experimental settlement tests to evaluate structural models for tunnelling-induced damage analysis. In *Challenges and Innovations in Geomechanics*, pages 164–171, Cham, 2021. Springer International Publishing. 4

[9] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[11] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, Cham, 2015. Springer International Publishing. 3

[13] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, pages 3212–3217, 2009. 3

[14] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1, 2017. 2

[15] G. Teza, A. Galgaro, N. Zaltron, and R. Genevois. Terrestrial laser scanner to detect landslide displacement fields: a new approach. *International Journal of Remote Sensing*, 28(16):3425–3446, 2007. 1, 3

[16] Yue Wang and Justin M. Solomon. Deep closest point: Learning representations for point cloud registration. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1

[17] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Proc. of the European Conf. on Computer Vision (ECCV)*. Springer, 2016. 3