

# Self-supervised Correspondence Estimation via Multiview Registration

Mohamed El Banani<sup>1\*</sup>  
 mbanani@umich.edu

Natalia Neverova<sup>2</sup>  
 nneverova@meta.com

Ignacio Rocco<sup>2</sup>  
 irocco@meta.com

Justin Johnson<sup>1,2</sup>  
 justincj@umich.edu

David Novotny<sup>2</sup>  
 dnovotny@meta.com

Andrea Vedaldi<sup>2</sup>  
 vedaldi@meta.com

Ben Graham<sup>2</sup>  
 benjammingraham@meta.com

<sup>1</sup>University of Michigan      <sup>2</sup>Meta AI

## Abstract

*Video provides us with the spatio-temporal consistency needed for visual learning. Recent approaches have utilized this signal to learn correspondence estimation from close-by frame pairs. However, by only relying on close-by frame pairs, those approaches miss out on the richer long-range consistency between distant overlapping frames. To address this, we propose a self-supervised approach for correspondence estimation that learns from multiview consistency in short RGB-D video sequences. Our approach combines pairwise correspondence estimation and registration with a novel  $SE(3)$  transformation synchronization algorithm. Our key insight is that self-supervised multiview registration allows us to obtain correspondences over longer time frames; increasing both the diversity and difficulty of sampled pairs. We evaluate our approach on indoor scenes for correspondence estimation and RGB-D pointcloud registration and find that we perform on-par with supervised approaches.*

## 1. Introduction

Consider the couch in Fig. 1. While the start and end frames depict overlapping regions in space, the large viewpoint change makes them appear significantly different. The ability to establish correspondence across views lies at the core of scene understanding and visual tasks such as SLAM and structure-from-motion. The common approach to learning correspondence estimation relies on correspondence supervision; *i.e.*, telling the model which pixels belong to the same point in space. However, we commonly learn about the world by moving and observing how appearance changes without such explicit supervision. *Could we learn correspondence estimation directly from video?*

Modern correspondence estimation approaches rely heavily on supervision. This is typically obtained by ap-

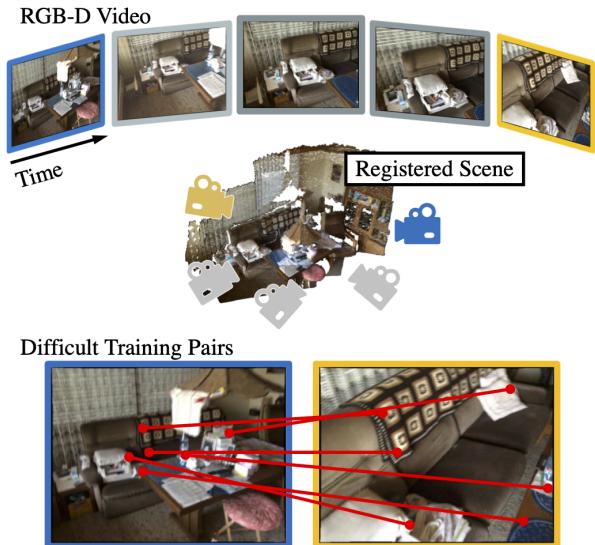


Figure 1. Multiview RGB-D registration allows us to learn from wide-baseline view pairs which exhibit more appearance change than adjacent frames.

plying classical 3D reconstruction algorithms on large image collections [41, 61, 66] or carefully captured indoor scans [11, 15, 16], and then sampling overlapping view pairs for training. This has been widely successful, as it provides correspondence supervision with large viewpoint and lighting changes. While current methods benefit from supervision, they are limited to learning from carefully captured videos that can already be constructed using standard algorithms. Recently, there has been a rise in self-supervised correspondence approaches that rely on close-by frames in video [20, 21, 32, 38]. This sampling strategy limits the appearance change, as shown in Fig. 1, resulting in poor performance on image pairs with large viewpoint changes. Ideally, we would leverage the temporal consistency within the video to learn from distant overlapping frames while ignoring non-overlapping pairs.

\* Work done during an internship at Meta AI.

To this end, we propose SyncMatch: a self-supervised approach for learning correspondence estimation through synchronized multiview pointcloud registration. Our approach bootstraps itself, generating wide-baseline view pairs through registering and synchronizing all pairwise transformations within short RGB-D video clips. Our core insight is that through synchronizing transformations across longer time frames, we can detect and learn from difficult pairs with large viewpoint changes. Despite only relying on geometric consistency within RGB-D videos, we achieve comparable performance to fully-supervised approaches.

Our approach is inspired by self-supervised pointcloud registration [20, 21] and transformation synchronization [3, 23]. The core insight in self-supervised pointcloud registration is that randomly initialized networks provide sufficiently good features for narrow-baseline pointcloud registration. This allows them to provide good pseudolabels for self-supervised training. Meanwhile, transformation synchronization allows us to estimate an accurate camera trajectory from a potentially noisy set of pairwise relative camera poses. Our approach combines both ideas for self-supervised multiview registration; allowing us to learn correspondence estimation across large viewpoint changes.

We evaluate our approach on RGB-D indoor scene videos. We train our model on RGB-D videos from ScanNet and ETH-3D, and evaluate it on correspondence estimation and RGB-D pointcloud registration. Despite only learning from RGB-D video, our approach achieves a similar performance to supervised approaches with more sophisticated matching algorithms. Furthermore, we provide a comprehensive analysis of our approach to understand the impact of the training data and the model components.

In summary, our contributions are as follows:

- A self-supervised correspondence estimation approach based on multiview consistency in RGB-D video.
- A novel SE(3) transformation synchronization algorithm that is fast, and numerically stable during training.

## 2. Related Work

**Correspondence Estimation.** Correspondence estimation is the task of identifying points in two images that correspond to the same physical location. The standard approach for establishing correspondence has two distinct steps: feature extraction and feature matching. Early work exploited hand-crafted feature detectors [43, 46] to extract normalized image patches across repeatable image points, combined with hand-crafted descriptors based on local statistics to obtain features with some robustness to illumination changes and small translations [1, 6, 43]. These features are matched via nearest neighbor search and filtered using heuristics; *e.g.* ratio test [43] or neighborhood consensus [58]. With the advent of deep learning, learnt keypoint detectors [37, 40], descriptors [5, 63], and correspondence estimators [54, 57, 69]

have been proposed. These models are trained using correspondence supervision from traditional 3D reconstruction algorithms [16, 59] on image collections of tourist landmarks [41, 61, 66] or indoor scene video scans [11, 15, 70]. Other approaches have explored self-supervision using synthetic data [18, 45, 53, 73], traditional descriptors [67], or RGB-D pairs from video [20, 21]. Our work shares the motivation of self-supervised approaches and extends it to learning from multiview consistency to better exploit the rich signal in video.

**Pointcloud Registration.** Pointcloud registration is the task of finding a transformation that aligns two sets of 3D points. Early work assumes access to perfect correspondence and devised algorithms to estimate the rigid body transformation that would best align them [4, 34, 64]. Later work proposed robust estimators that can handle correspondence errors and outliers that arise from feature matching [22, 71]. More recently, learned counterparts have been proposed for 3D keypoint descriptors [14, 17, 25], correspondence estimation [8, 9, 13, 23, 30, 51, 68], as well as models for directly registering a pair of pointclouds [17, 26, 39, 44]. Closest to our work are approaches that use RGB-D video to learn correspondence-based pointcloud registration [20, 21, 65]. Similar to our approach, they learn from the geometric consistency between RGB-D frames in a video. However, unlike those approaches, we train on short sequences of videos instead of frame pairs, allowing us to train on view pairs with larger camera changes.

**SE(3) Transformation Synchronization** Given a set of pairwise estimates, synchronization estimates that set of latent values that explains them. Transformation synchronization refers to this problem applied to SO(3) and SE(3) transformations, as it commonly arises in SLAM settings [10, 36, 47, 56]. For video, one could naively only consider adjacent pairs to construct a minimum spanning tree and aggregate the transformations. However, this only works if all pairwise estimates are accurate, since a single bad estimate can be detrimental. More robust approaches have been proposed that can leverage the information from multiple (or all) edges in the graph [7, 28, 29, 31, 36, 55]. Most relevant to our work is Arrigoni *et al.* [2, 3] and Gojcic *et al.* [23]. Arrigoni *et al.* [2, 3] propose a closed-form solution to SO(3) and SE(3) synchronization based on the eigendecomposition of a pairwise transformation matrix. Gojcic *et al.* [23] builds on those ideas and integrates transformation synchronization with a supervised end-to-end pipeline for multiview registration. We are inspired by this work and propose a different approach to SE(3) synchronization based on iterative matrix multiplication, which allows for accurate synchronization while being more numerically stable. Furthermore, unlike prior work [23, 29, 55], we use transformation synchronization for learning without supervision.

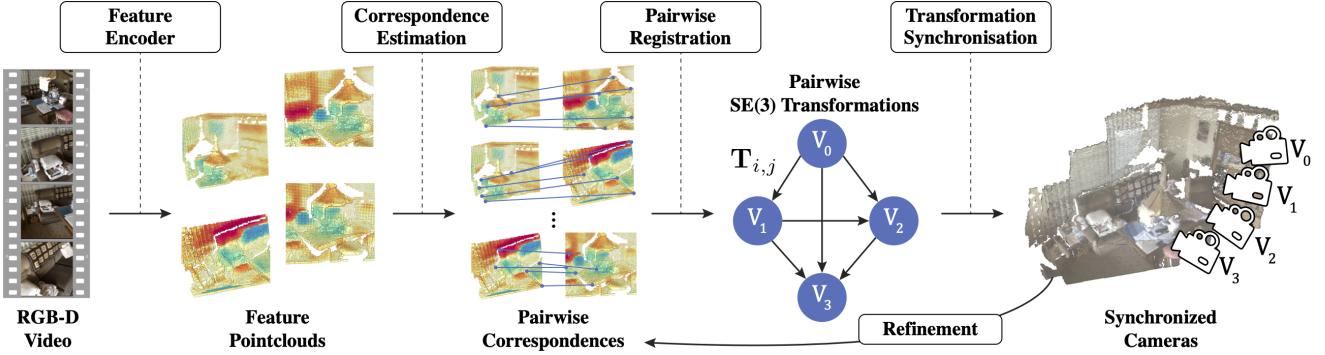


Figure 2. **SyncMatch.** Given a sequence of  $N$  RGB-D video frames, we extract features for each image and project them to a 3D pointcloud using input depth. We extract all pairwise correspondences to estimate pairwise SE(3) transformations. We then synchronize the pairwise transformations to register the scene. Finally, we use the estimated registration to refine our correspondence and transformation estimates. We compute correspondences losses for both the initial and refined registrations, and backpropagate them to the feature encoder.

### 3. Approach

We learn correspondence estimation from multiview registration of short RGB-D sequences without relying on pose or correspondence supervision. We first provide a high-level sketch of our approach, shown in Fig. 2, before discussing each component in detail.

**Approach Sketch.** Given  $N$  RGB-D frames, we extract features for each RGB image and project them onto a 3D pointcloud using input depth and camera intrinsics. We then extract correspondences between all pointcloud pairs and estimate pairwise SE(3) transformations. Given  $\binom{N}{2}$  pairwise transformations, we apply transformation synchronization to find the  $N$  camera extrinsic parameters in a shared global frame. Given this coarse alignment, we resample correspondences based on both feature and spatial proximity. We repeat the registration using the updated correspondences. Finally, we compute the loss using the estimated correspondences and SE(3) transformations and backpropagate it to the feature encoder.

#### 3.1. Feature Pointcloud

We use a randomly-initialized ResNet-18 for feature extraction. While correspondence estimation methods often rely on keypoint detection, our approach is detector free and generates dense features uniformly across the image. Similar to Sun *et al.* [62], we generate the feature grid at a lower resolution ( $1/4$ ) than the input image. For each frame  $i$ , we use the input depth map and camera intrinsics to project the features into a pointcloud  $\mathcal{P}_i$  where each point  $p \in \mathcal{P}_i$  has a feature  $\mathbf{f}_p$  and a 3D coordinate  $\mathbf{x}_p$ .

#### 3.2. Correspondence Estimation

We estimate feature correspondences for all view pairs  $(i, j)$ . We first generate an initial set of correspondences by finding for each point in image  $i$  the point in image  $j$  with the closest matching feature vector. The initial correspon-

dence set will include mismatches due to poor matching, repetitive textures, or occlusion.

Correspondences can be filtered using measures of unique matches or geometric consistency. Heuristics such as Lowe’s ratio test [43] prefer unique matches and have been extended to self-supervised pointcloud registration [20, 21] and attention-based matching [27, 57, 62]. Geometric consistency relies on the idea a geometrically consistent set of correspondences is likely correct. This can be done by estimating the transformation similar to RANSAC [22] or directly estimating the inlier scores [13, 51, 69]. We use the ratio test for initial alignment and leverage geometric consistency for refinement (Sec. 3.5). Specifically, we compute a ratio between the cosine distances in feature space as follows:

$$w_{p,q} = 1 - \frac{D(p, q)}{D(p, q')}, \quad (1)$$

where  $D(p, q)$  is the cosine distance between the features, and  $q$  and  $q'$  are the first and second nearest neighbors to point  $p$  in feature space. We use the weights to rank the correspondences and only keep the top  $k$  correspondences. This results in a correspondence set  $\mathcal{C}_{i,j}$  for each pair of frames. The correspondences  $(p, q, w_{p,q}) \in \mathcal{C}_{i,j}$  consists of the two matched points and the match weight.

#### 3.3. Pairwise Alignment

For each pair of frames, we can identify a transformation  $\mathbf{T}_{i,j} \in \text{SE}(3)$  that minimizes the weighted mean-squared error between the aligned correspondences across the images:

$$\mathbf{T}_{i,j} = \arg \min_{\mathbf{T} \in \text{SE}(3)} \sum_{(p,q,w) \in \mathcal{C}_{i,j}} w \|\mathbf{x}_q - \mathbf{T}(\mathbf{x}_p)\|_2^2. \quad (2)$$

A differentiable solution is given by the Weighted Procrustes (WP) algorithm [13] which adapts the classic Umeyama pointcloud alignment algorithm [34, 64].

**WP-RANSAC.** While the WP algorithm can handle small errors, it is not robust against outliers. El Banani *et al.* [20] propose combining the alignment algorithm with random sampling to increase robustness. However, a single large outliers can still perturb the solution since solutions are still ranked according to the average residual error on all matches. We modify the WP algorithm to more closely resemble classic RANSAC [22]. We randomly sample  $k$  correspondence subsets, estimate  $k$  transformations, and compute an inlier score based on each transformation. We choose the transformation that maximizes the inlier score instead of minimizing the weighted residual error [20]; making us more robust to large outliers. Finally, we update the correspondence weights with the inlier scores which can zero out large outliers. We compute the final registration using the WP algorithms with the updated weights; maintaining differentiability with respect to the correspondence weights.

### 3.4. SE(3) Transformation Synchronization

Given estimates for the  $\binom{N}{2}$  camera-to-camera transformations, we want to find the  $N$  world-to-camera transformations that best explain them. Arrigoni *et al.* [2, 3] propose a closed-form solution to SE(3) synchronization using spectral decomposition. This approach was latter extended to end-to-end learning pipelines [23, 29]. The approach operates by constructing a block matrix of pairwise transformations where block  $(i, j)$  corresponds to the transformation between camera  $i$  to camera  $j$ . The core insight in that line of work is that the absolute transformations constitute the basis of the pairwise transformations matrix, and hence, can be recovered using eigendecomposition.

While those approaches are successful for inference, they suffer from numerical instabilities during training. This is caused by the backward gradient of the eigendecomposition scaling with  $\frac{1}{\min_{i \neq j} \lambda_i - \lambda_j}$  where  $\lambda$  are the eigenvalues. Given that the rank of a perfect SE(3) pairwise matrix is 4,  $\min_{i \neq j} \lambda_i - \lambda_j$  approaches 0 for an accurate pairwise matrix which results in exploding gradients. We observed this instability during training. To avoid this instability, we compute the relevant part of the eigendecomposition by power iteration, similar to PageRank [49]. We observe that this converges quickly while being stable during training. We refer the reader to the appendix for more details.

**Pairwise confidence.** Synchronization is more effective when provided with confidence weights for each of the pairwise estimates. While prior approaches train separate networks to estimate pairwise confidence [24, 29], we instead opted for a simpler approach. We observe that the mean confidence weight is well correlated with view overlap as shown in Fig. 3, where pairwise confidence is computed as  $\hat{c}_{i,j} = \frac{1}{|\mathcal{C}_{i,j}|} \sum_{w \in \mathcal{C}_{i,j}} w$ .

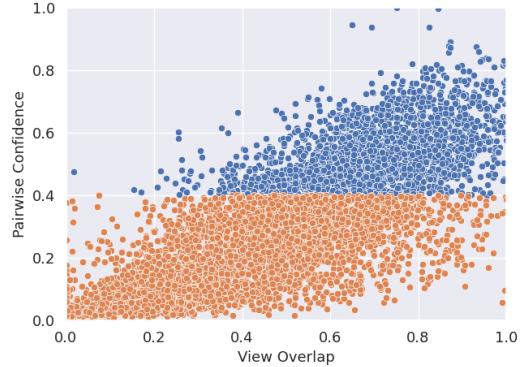


Figure 3. **Mean correspondence confidence is an effective overlap filter.** While confidence is not perfectly correlated with view overlap, simple thresholding can accurately filter out low- and no-overlap view pairs.

While confidence is correlated with view overlap, non-overlapping frames still receive a non-zero confidence. Incorrect transformations from non-overlapping pairs can negatively affect both synchronization and learning. We address this by rescaling the confidence values based on a threshold to ignore any non-overlapping pairs. While this criteria is simple, it has many false negatives as shown in Fig. 3. To ensure that synchronization is always possible, we exclude adjacent pairs from rescaling since we know they most likely overlap. The pairwise confidence terms are adjusted as follows:

$$c_{i,j} = \begin{cases} \max(0, \hat{c}_{i,j} - \gamma)/(1 - \gamma) & \text{if } |i - j| > 1, \\ \hat{c}_{i,j} & \text{otherwise.} \end{cases} \quad (3)$$

where  $\gamma$  is the confidence threshold. We only estimate pairwise correspondences for pairs  $(i, j)$  where  $i < j$  to avoid repeated calculations. For pairs where  $j > i$ , we set  $c_{j,i} = c_{i,j}$  and  $T_{j,i} = T_{i,j}^{-1}$ .

**Pairwise Transformation Matrix.** We form a block matrix  $\mathbf{A}$  using the weighted transformations as follows:

$$\mathbf{A} = \begin{bmatrix} c_1 \mathbf{I}_4 & c_{1,2} \mathbf{T}_{1,2} & \cdots & c_{1,N} \mathbf{T}_{1,N} \\ c_{2,1} \mathbf{T}_{2,1} & c_2 \mathbf{I}_4 & \cdots & c_{2,N} \mathbf{T}_{2,N} \\ \vdots & \ddots & & \vdots \\ c_{N,1} \mathbf{T}_{N,1} & c_{N,2} \mathbf{T}_{N,2} & \cdots & c_N \mathbf{I}_4 \end{bmatrix}, \quad (4)$$

where  $c_{i,j}$  is the pairwise confidence,  $\mathbf{T}_{i,j}$  is the estimated pairwise transformation, and  $c_i = \sum_{k \in N} c_{i,k}$ . We perform  $t$  matrix multiplications to calculate  $\mathbf{A}^{2^t}$  and extract the synchronized transformations by taking the first block column and normalizing each transformation by its confidence (bottom right element). This results in  $N$  SE(3) transformations in the first view’s frame of reference.



Figure 4. **Geometry-aware sampling greatly improves our correspondence set.** GART allows us to extract accurate correspondences even if the initial feature matches are very noisy.

### 3.5. Refinement

While feature-based matching is powerful, it can produce false positive feature matches that could be easily filtered out through geometric consistency. To this end, we use the predicted scene alignment to refine our correspondences by filtering matches that are not geometrically consistent with the estimated transformation. We resample the correspondences but compute the ratio test based on both feature similarity and spatial proximity. We update our correspondence filtering criteria by changing the distance function in the ratio test to:

$$D_{refine}(p, q) = D_C(\mathbf{f}_p, \mathbf{f}_q) + \lambda \|\mathbf{x}_p - \mathbf{x}_q\|_2, \quad (5)$$

where  $D_C(x, y)$  is cosine distance,  $\mathbf{f}_p$  is the feature vector belonging to point  $p$ ,  $\lambda$  is a weighing constant, and  $\mathbf{x}_p$  is the aligned 3D coordinate of point  $p$  in a common global frame. We refer to this updated ratio test as a Geometry-Aware Ratio Test (GART).

### 3.6. Losses

We emphasize that our approach is self-supervised. Hence, we only rely on the consistency within the video for training. We use the registration loss proposed by El Banani and Johnson [21] which minimizes the weighted residual error of the estimated correspondences using the estimated alignment. For a given pair  $(i, j)$ , we compute a registration loss as follows:

$$\mathcal{L}_{reg}(i, j) = \sum_{(p, q, w) \in \mathcal{C}_{i, j}} w \|\mathbf{T}_j^{-1} \mathbf{x}_q - \mathbf{T}_i^{-1} \mathbf{x}_p\|_2, \quad (6)$$

where  $p$  and  $q$  are the corresponding points,  $w$  is their weight, and  $\mathbf{T}_i$  is the synchronized transformation (Sec. 3.4). We compute this loss for both the initial and refined correspondence sets and predicted transformations for all view pairs.

## 4. Experiments

We evaluate our approach on two indoor scene datasets: ScanNet [15] and ETH3D [60]. Our experiments address the following questions: (1) does multiview training improve over pairwise training?; (2) can multiview self-supervision replace full-supervision?; (3) can we reconstruct scenes from RGB-D sequences?; (4) can we learn from videos that cannot be reconstructed using standard approaches?

**Training Dataset.** ScanNet provides RGB-D videos of 1513 scenes and camera poses computed using BundleFusion [16]. We use the train/valid/test scene split from ScanNet v2. While automated reconstruction models like BundleFusion are able to reconstruct ScanNet scenes, ETH3D [60] videos are more challenging to such systems with BundleFusion failing on most of those sequences. As a result, ETH3D offers us with an interesting set of videos which cannot be currently used by supervised training.

We emphasize that we only use RGB-D video and camera intrinsics for training and any provided camera poses are only used for evaluation. We generate view pairs by sampling views that are 20 frames apart. For longer sequences, we combine adjacent pairs to get N-tuples.

**Training Details.** We train our model with the AdamW [35, 42] optimizer using a learning rate of  $10^{-3}$  and a weight decay of  $10^{-3}$ . We train for 100K iterations with a batch size of 16. Unless otherwise stated, we use 6 views for training. Our implementation is in PyTorch [50], with extensive use of PyTorch3D [52], FAISS [33], PyKeOps [12], and Open3D [72]. We make our code publicly available.<sup>1</sup>

### 4.1. Correspondence Estimation

We evaluate our model on correspondence estimation. While our model is trained on adjacent pair sequences, the primary challenge is how it performs for wide-baseline correspondence estimation. We evaluate this on the test set proposed by SuperGlue [57] of 1500 view pairs. This dataset includes difficult pairs with large camera motions; *e.g.*, images from opposite sides of a room.

**Evaluation Metrics.** We evaluate the estimated correspondences based on their 2D and 3D errors. We lift the estimated correspondences into 3D using known depth and intrinsics and only consider keypoints with valid depth values. We use groundtruth transformations to align the keypoints and compute the 3D error and the 2D reprojection error. We extract 500 correspondences for all methods to allow for a meaningful comparison between precision values.<sup>2</sup>

<sup>1</sup><https://github.com/facebookresearch/SyncMatch>

<sup>2</sup>LoFTR and SuperGlue use a mutual check heuristic for matching which can produce fewer correspondences. In such cases, we use all the correspondences they produce.

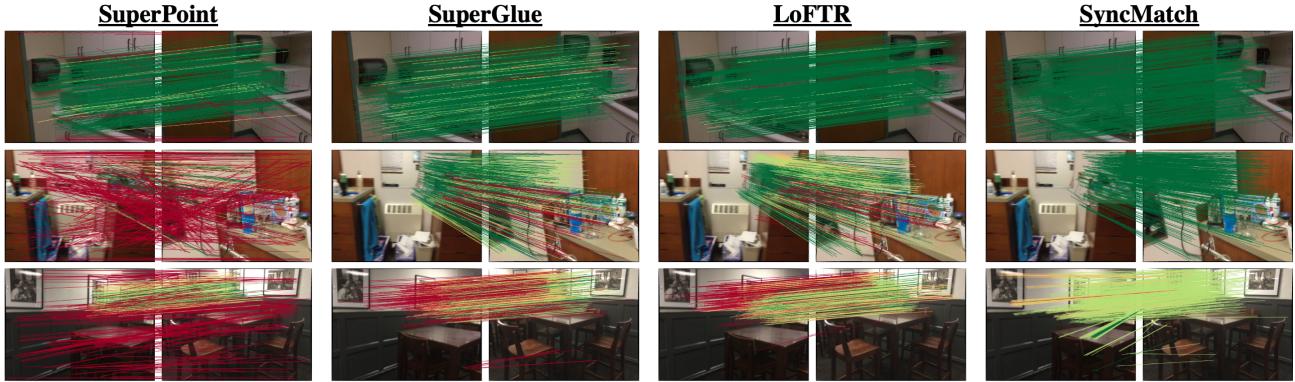


Figure 5. **Correspondence Estimation on ScanNet.** Our model extracts accurate correspondences for large viewpoint change. Through combining both strong feature descriptors and geometric refinement, we can successfully handle cases where prior approaches fail. Correspondences are color-coded by 3D error.

**Baselines.** We compare our approach against classic, self-supervised, and supervised correspondence estimation methods. First, we compare against two commonly used feature descriptors: RootSIFT [1] and SuperPoint [18]. RootSIFT is a hand-crafted feature that is still used in modern pipelines, while SuperPoint is a self-supervised descriptor trained on synthetic data and affine transformations of web images. We report the performance of these features for image-only matching using the ratio test [43] as well as depth-refined matching using our proposed GART.

We consider three end-to-end approaches: SuperGlue [57], LoFTR [62], and BYOC [21]. SuperGlue is an attention-based matching algorithm built on top of SuperPoint. LoFTR and BYOC are both detector-free approaches which train their own features from scratch. SuperGlue and LoFTR both use transformers for matching that are trained with correspondence supervision. BYOC is self-supervised and uses a variant of the ratio test for matching.

We take several measures to ensure a comprehensive and fair comparison. First, We update BYOC’s visual backbone from ResNet-5 to ResNet-18. This results in a stronger and fairer baseline which we refer to as BYOC<sup>†</sup>. Second, SuperGlue and LoFTR are both fully supervised image-based approaches which use transformers for feature matching. Hence, it was unclear how to adapt their matching algorithm to use depth information. Instead of adapting their matching, we use GART to re-rank their proposed matches and only retain the top set of matches. This resulted in large performance improvements as seen in Tab. 1

**How does heuristic matching perform?** We find that well-tuned matching allows hand-crafted features to achieve a strong performance against learned features, as observed by Efe *et al.* [19]. Nevertheless, self-supervised feature descriptors still retain a performance advantage. Furthermore, our proposed approach outperforms both hand-crafted and self-supervised descriptors regardless of whether depth is used at test time for refinement.

Table 1. **Wide-Baseline Correspondence Estimation on ScanNet.** SyncMatch extracts accurate wide-baseline correspondences; performing on-par with supervised methods. Our proposed GART uses estimated alignment to sample more accurate correspondences regardless of the underlying feature descriptor.

| Method  | 3D Corres.  |             |             | 2D Corres. |             |             |
|---|-------------|-------------|-------------|------------|-------------|-------------|
|   | 1cm         | 5cm         | 10cm        | 1px        | 2px         | 5px         |
| <b><i>Unsupervised Features with Heuristic Matching</i></b> |             |             |             |            |             |             |
| RootSIFT [1]  | 7.1         | 29.2        | 35.1        | 2.8        | 8.6         | 22.9        |
| RootSIFT [1] + GART   | 14.1        | 72.4        | 84.3        | 3.8        | 12.8        | 42.8        |
| SuperPoint [18]   | 7.5         | 41.4        | 51.3        | 2.5        | 8.6         | 29.5        |
| SuperPoint [18] + GART                                      | 16.8        | 73.7        | 84.3        | 4.7        | 15.5        | 47.9        |
| BYOC <sup>†</sup> [21]                                      | 12.8        | 53.3        | 63.0        | 4.5        | 14.6        | 41.9        |
| BYOC <sup>†</sup> [21] + GART                               | 22.8        | 73.1        | 81.4        | 6.0        | 19.6        | 54.0        |
| SyncMatch (Ours)  | 13.1        | 55.1        | 65.4        | 4.6        | 15.3        | 43.9        |
| SyncMatch (Ours) + GART                                     | <b>26.8</b> | 76.5        | 84.4        | <b>7.5</b> | <b>23.5</b> | <b>59.7</b> |
| <b><i>Supervised Features with Trained Matching</i></b>     |             |             |             |            |             |             |
| SuperGlue [57]  | 8.7         | 62.4        | 78.7        | 2.5        | 9.0         | 36.9        |
| SuperGlue [57] + GART                                       | 13.8        | 74.8        | 87.7        | 3.3        | 11.7        | 44.4        |
| LoFTR [62]  | 16.0        | 72.2        | 84.6        | 5.6        | 18.5        | 55.5        |
| LoFTR [62] + GART   | 21.4        | <b>80.8</b> | <b>90.2</b> | 6.5        | 21.2        | 59.0        |

**Can self-supervision replace correspondence supervision?** While our approach outperforms classic and self-supervised approaches, it still underperforms the strongest supervised approaches. This is expected since we use pseudocorrespondence labels from short sequences, whereas supervised approaches are trained with view pairs that were sampled the same way as the test pairs [57]. Nevertheless, we argue that our approach is still promising, as it can match SuperGlue supervised features and matching despite being self-supervised and using the ratio-test for matching.

**Does geometry based refinement help?** We find that our proposed geometry-based refinement improved the performance for all methods. Furthermore, the improvement is most pronounced for our method, which performs on-par with supervised methods when using depth, and outperforms them for some thresholds.

Table 2. **Pairwise Registration on ScanNet.** SyncMatch outperforms all approaches on narrow-baseline registration, while underperforming some methods for wide-baseline registration. WP-RANSAC results in large performance gains for all methods across metrics.

|   | Narrow Baseline   |                     | Wide Baseline     |                     |
|---|-------------------|---------------------|-------------------|---------------------|
|   | AUC <sub>5°</sub> | AUC <sub>10cm</sub> | AUC <sub>5°</sub> | AUC <sub>10cm</sub> |
| <b><i>Unsupervised Features with Heuristic Matching</i></b> |                   |                     |                   |                     |
| RootSIFT [1] + RANSAC                                       | 36.7              | 28.9                | 15.5              | 11.1                |
| SuperPoint [18] + RANSAC                                    | 50.3              | 39.8                | 24.9              | 17.0                |
| RootSIFT [1] + <b>Ours</b>                                  | 84.3              | 77.9                | 64.4              | 52.3                |
| SuperPoint [18] + <b>Ours</b>                               | 83.8              | 77.0                | 61.7              | 49.2                |
| BYOC <sup>†</sup> [21]                                      | 84.6              | 77.6                | 60.4              | 48.4                |
| SyncMatch ( <b>Ours</b> )                                   | 85.3              | 78.8                | 63.4              | 50.5                |
| <b><i>Supervised Features with Trained Matching</i></b>     |                   |                     |                   |                     |
| SuperGlue [57] + RANSAC                                     | 65.7              | 54.0                | 47.6              | 33.2                |
| LoFTR [62] + RANSAC   | 75.0              | 64.6                | 57.2              | 41.8                |
| SuperGlue [57] + <b>Ours</b>                                | 82.3              | 75.0                | 66.0              | 51.2                |
| LoFTR [62] + <b>Ours</b>                                    | 84.5              | 78.1                | 70.5              | 56.2                |

## 4.2. Pointcloud Registration

We next evaluate pairwise and multiview pointcloud registration performance. We evaluate pairwise registration using view pairs extracted from ScanNet. We also evaluate our model’s ability to scale up to large sequences by reconstructing the challenging sequences in the ETH-3D dataset.

**Pairwise Registration.** We evaluate the approaches on pairwise registration for both narrow-baseline and wide-baseline view pairs, as shown in Tab. 2. We evaluate narrow baseline view pairs similar to BYOC and wide-baseline view pairs similar to SuperGlue. We report the area under the curve for pose errors with a threshold of 5° and 10cm for rotation and translation errors, respectively. For RootSIFT and SuperPoint, we compute correspondences using the ratio test, while SuperGlue and LoFTR provide us with matches. We use either Open3D’s [72] RANSAC or our proposed WP-RANSAC for alignment. For SyncMatch and BYOC, we use the method’s estimated transformation. We report numbers without depth refinement to avoid confounding the evaluation.

Our approach outperforms all baselines for narrow-baseline registration but underperforms several in wide-baseline registration. This is surprising given our model’s strong performance in wide-baseline correspondence estimation, but probably arises from the domain shift from the mostly narrow-baseline pairs used for training. Furthermore, we note that our proposed alignment algorithm greatly improves the performance of all baselines, especially RootSIFT. We observe this improvement from supervised models with trained correspondence estimators, suggesting that their predicted correspondences still contain significant structured error that benefit from robust estimators that can utilize the match confidence weights.



Figure 6. **RGB-D Scene Reconstruction.** SyncMatch can scale at inference time to longer videos to reconstruct longer sequences.

**Scaling up to longer sequences.** Computing pairwise correspondence for  $N$  frames scales as  $O(N^2)$ , which is problematic for longer videos. However, with minor reformulation, SyncMatch can be adapted to significantly reduce its run-time. Instead of considering all pairs, we only consider adjacent frames in the first step to give us an approximate camera trajectory from  $N-1$  pairs. We use this trajectory to refine the correspondences and then consider all frames within a specific window  $w$ ; *i.e.*, only consider frames  $(i, j)$  if  $|i - j| < w$ . We can then run the synchronization step with the confidence of all other pairs set to 0. This allows us to scale the model’s run-time linearly with number of frames, instead of quadratically, which allows us to handle hundreds of frames. We visualize two reconstructions from ScanNet in Fig. 6.

We apply our adapted model to the challenging ETH3D dataset. ETH-3D sequences are challenging to traditional RGB-D reconstruction models, with BundleFusion failing on nearly 75% of the sequences. SyncMatch can reconstruct 33/61 training scenes and 16/35 test scenes. This outperforms standard systems such as BundleFusion (14/61 and 7/35) and ORB-SLAM [48] (25/61 and 16/35). Since such systems are often used to automatically generate annotations for RGB-D video datasets, SyncMatch’s strong performance against them shows the promise of self-supervised approaches to use videos that currently are missing from large-scale RGB-D scene datasets. We emphasize that our model was not designed for full-scene reconstruction; this evaluation is only intended to showcase our model’s performance against existing methods for automatically generating pose annotation for RGB-D videos.

## 4.3. Analysis

We analyze our model through a series of experiments aimed at answering some key questions regarding its performance. The analysis experiments are aimed at understanding the impact of the multiview setting, training data, as well as the impact of our model’s components during both training and inference.

Table 3. **Impact of number of training views.** Training with more views improves correspondence estimation performance.

| Num. of Views | 3D Corres. |      |      | 2D Corres. |      |      |
|---------------|------------|------|------|------------|------|------|
|               | 1cm        | 5cm  | 10cm | 1px        | 2px  | 5px  |
| 2             | 24.7       | 73.9 | 81.6 | 6.2        | 19.8 | 54.0 |
| 3             | 26.2       | 76.8 | 85.2 | 7.0        | 22.6 | 58.8 |
| 4             | 26.8       | 75.4 | 83.5 | 7.5        | 23.3 | 59.2 |
| 5             | 26.9       | 75.9 | 83.6 | 7.6        | 23.5 | 59.7 |
| 6             | 26.8       | 76.5 | 84.4 | 7.5        | 23.5 | 59.7 |

Table 4. **Ablation Experiments.** While WP-RANSAC is crucial to model performance, the impact of other components depends on downstream tasks.

|                    | Ablation |      | PW Corr.            |                     | MV Reg.           |                     |
|--------------------|----------|------|---------------------|---------------------|-------------------|---------------------|
|                    | Train    | Test | AUC <sub>10cm</sub> | AUC <sub>10px</sub> | AUC <sub>5°</sub> | AUC <sub>10cm</sub> |
| Full Model         |          |      | 65.9                | 48.1                | 83.4              | 77.8                |
| Naive Sync.        | ✗        |      | 65.7                | 48.9                | 82.8              | 76.2                |
| No Depth Refine    | ✗        |      | 66.0                | 49.0                | 82.9              | 76.3                |
| No Conf Threshold  | ✗        |      | 35.2                | 20.4                | 17.9              | 14.5                |
| No WP-RANSAC       | ✗        |      | 1.2                 | 0.2                 | 1.1               | 3.3                 |
| Naive Sync.        | ✗        |      | 65.9                | 48.1                | 82.6              | 76.9                |
| No Depth Refine    | ✗        |      | 29.7                | 19.4                | 83.4              | 77.7                |
| No Conf. Threshold | ✗        |      | 65.9                | 48.1                | 76.9              | 70.9                |
| No WP-RANSAC       | ✗        |      | 42.2                | 27.7                | 74.4              | 66.0                |

### What’s the impact of the number of training views?

We train our model with a variable number of training views to understand the impact of multiview training. We observe that increasing the number of training views results in progressively better performance. However, the performance gains saturate after 5 views. This could be explained how ScanNet videos were captured: often, the camera is moving laterally and after 5 frames (3 seconds), there is often no overlap. Our results suggest that using more views for training will not help until enough frames are used to provide loop closure for the model to learn from.

**How do the different components contribute to the model performance?** We analyze the impact of various model components through a series of ablations that are applied either during training or testing. We report the performance for both pairwise correspondence estimation (3D and 2D correspondence error) as well as multiview registration with 6 views (rotation and translation error) in Tab. 4. Similar to prior work [20], we observe that ablations that make it harder to register the pointcloud can boost correspondence performance when ablated during training; *e.g.*, using naive synchronization based only on adjacent views or training without depth refinement. However, replacing WP-RANSAC with WP prevents the model from learning due to inaccurate registration early during training. We also observe that almost all test-time ablations result in worse performance. One surprising exception is removing depth refinement which greatly reduces wide-baseline correspondence estimation accuracy, while not impacting multiview registration. This could be explained by the performance saturating for narrow-baseline registration such that depth refinement is not needed there.

Table 5. **Training on ETH-3D data.** Our model is capable of learning for the more challenging videos in ETH-3D.

| Training Set | 3D Corres. |      |      | 2D Corres. |      |      |
|--------------|------------|------|------|------------|------|------|
|              | 1cm        | 5cm  | 10cm | 1px        | 2px  | 5px  |
| ETH-3D       | 17.4       | 66.4 | 74.5 | 4.8        | 16.1 | 47.3 |
| ScanNet Mini | 18.5       | 67.0 | 75.6 | 4.8        | 16.2 | 47.6 |

**Can we learn from more challenging sequences?** While ScanNet offers a large set of videos, these videos are carefully captured to ensure successful downstream 3D reconstruction. We investigate whether our approach can be trained on more challenging videos, such as ETH-3D. ETH-3D has fewer and more erratic videos, forcing us to reduce the view stride during training. However, our model can still learn without supervision on ETH-3D videos and the features can be used for wide-baseline correspondence estimation as shown in Tab. 5. For comparison, we train on a subset of ScanNet to match number of instances and view spacing. Both models achieve similar performance. This suggests that our approach could scale to challenging videos beyond carefully captured ScanNet sequences.

## 5. Conclusion

We present SyncMatch: a self-supervised approach to learning correspondence estimation that relies on multiview registration to learn from difficult view pairs. Our core insight is that multiview registration allows us to leverage the consistency within short video sequences to obtain difficult view pairs for learning. To this end, we propose a series of components that integrate self-supervised correspondence estimation with multiview registration in a single end-to-end differentiable pipeline. Our approach follows the conventional registration pipeline with several technical contributions that improve registration and correspondence estimation performance while maintaining differentiability.

Our goal is not to beat supervised methods, but rather to show that supervision might not be needed. While standard 3D reconstruction systems like COLMAP and Bundle-Fusion provide us with good reconstructions, learned approaches trained with their outputs are starting to rival their accuracy. This raises an interesting question: how can we exceed standard pipelines when our training data is limited by what they can handle and our supervision is limited by their error? Through proposing self-supervised pipelines that learn directly from videos, we take a step towards enabling the development of approaches that go beyond conventional setups and scale to more uncurated data, allowing us to both achieve better 3D reconstruction and tackle difficult challenges like dynamic scene reconstruction.

**Acknowledgments** We thank Karan Desai, Mahmoud Azab, David Fouhey, Richard Higgins, Daniel Geng, and Menna El Bannani for feedback and edits to early drafts of this work.

## References

[1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. [2](#), [6](#), [7](#)

[2] Federica Arrigoni, Luca Magri, Beatrice Rossi, Pasqualina Fragneto, and Andrea Fusiello. Robust absolute rotation estimation via low-rank and sparse matrix decomposition. In *3DV*, 2014. [2](#), [4](#)

[3] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in se (3). *SIAM Journal on Imaging Sciences*, 2016. [2](#), [4](#)

[4] KS Arun, TS Huang, and SD Blostein. Least square fitting of two 3-d point sets. In *TPAMI*, 1987. [2](#)

[5] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, 2016. [2](#)

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006. [2](#)

[7] Tolga Birdal, Michael Arbel, Umut Şimşekli, and Leonidas Guibas. Synchronizing probability measures on rotations via optimal transport. In *CVPR*, 2020. [2](#)

[8] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. DSAC: Differentiable RANSAC for Camera Localization. In *CVPR*, 2017. [2](#)

[9] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, 2019. [2](#)

[10] Luca Carlone, Roberto Tron, Kostas Daniilidis, and Frank Dellaert. Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization. In *ICRA*, 2015. [2](#)

[11] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *3DV*, 2017. [1](#), [2](#)

[12] Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *JMLR*, 2021. [5](#)

[13] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *CVPR*, 2020. [2](#), [3](#)

[14] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *ICCV*, 2019. [2](#)

[15] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. [1](#), [2](#), [5](#)

[16] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *ACM ToG*, 2017. [1](#), [2](#), [5](#)

[17] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *CVPR*, 2019. [2](#)

[18] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshops*, 2018. [2](#), [6](#), [7](#)

[19] Ufuk Efe, Kutalmis Gokalp Ince, and A Aydin Alatan. Effect of parameter optimization on classical and learning-based image matching methods. In *CVPR*, 2021. [6](#)

[20] Mohamed El Banani, Luya Gao, and Justin Johnson. UnsupervisedR&R: Unsupervised Point Cloud Registration via Differentiable Rendering. In *CVPR*, 2021. [1](#), [2](#), [3](#), [4](#), [8](#)

[21] Mohamed El Banani and Justin Johnson. Bootstrap Your Own Correspondences. In *ICCV*, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#)

[22] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981. [2](#), [3](#), [4](#)

[23] Zan Gojcic, Caifa Zhou, Jan D. Wegner, Leonidas J. Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *CVPR*, 2020. [2](#), [4](#)

[24] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *CVPR*, 2019. [4](#)

[25] Benjamin Graham. Sparse 3D convolutional neural networks. In *BMVC*, 2015. [2](#)

[26] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. PointGMM: A Neural GMM Network for Point Clouds. In *CVPR*, 2020. [2](#)

[27] Shengyu Huang, Zan Gojcic, Mikhail Usyatsov, and Konrad Schindler Andreas Wieser. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *CVPR*, 2021. [3](#)

[28] Xiangru Huang, Zhenxiao Liang, Chandrajit Bajaj, and Qixing Huang. Translation synchronization via truncated least squares. *NeurIPS*, 2017. [2](#)

[29] Xiangru Huang, Zhenxiao Liang, Xiaowei Zhou, Yao Xie, Leonidas J Guibas, and Qixing Huang. Learning transformation synchronization. In *CVPR*, 2019. [2](#), [4](#)

[30] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *CVPR*, 2020. [2](#)

[31] Daniel F Huber and Martial Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*. [2](#)

[32] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *NeurIPS*, 2020. [1](#)

[33] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2021. [5](#)

[34] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 1976. [2](#), [3](#)

[35] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. [5](#)

[36] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *ICRA*. IEEE, 2011. [2](#)

[37] Axel Barroso Laguna, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Key.Net: Keypoint detection by hand-crafted and learned cnn filters. In *ICCV*, 2019. [2](#)

[38] Zihang Lai and Weidi Xie. Self-supervised learning for video correspondence flow. In *BMVC*, 2019. [1](#)

[39] Huu M Le, Thanh-Toan Do, Tuan Hoang, and Ngai-Man Cheung. SDRSAC: Semidefinite-based randomized ap-

proach for robust point cloud registration without correspondences. In *CVPR*, 2019. 2

[40] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCV*, 2016. 2

[41] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 1, 2

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 5

[43] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 3, 6

[44] 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 *ICCV*, 2019. 2

[45] Iaroslav Melekhov, Zakaria Laskar, Xiaotian Li, Shuzhe Wang, and Juho Kannala. Digging into self-supervised learning of feature descriptors. In *3DV*, 2021. 2

[46] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004. 2

[47] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 2015. 2

[48] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 2017. 7

[49] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. 4

[50] Adam Paszke, Soumith Chintala, Ronan Collobert, Koray Kavukcuoglu, Clement Farabet, Samy Bengio, Iain Melvin, Jason Weston, and Johnny Mariethoz. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, may 2017. 5

[51] Rene Ranftl and Vladlen Koltun. Deep fundamental matrix estimation. In *ECCV*, 2018. 2, 3

[52] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv*, 2020. 5

[53] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 2

[54] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. 2

[55] Barbara Roessle and Matthias Nießner. End2end multi-view feature matching using differentiable pose optimization. *arXiv preprint arXiv:2205.01694*, 2022. 2

[56] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *CVPR*, 2013. 2

[57] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2, 3, 5, 6, 7

[58] Cordelia Schmid and Roger Mohr. Local grayvalue invariants for image retrieval. *TPAMI*, 1997. 2

[59] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2

[60] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In *CVPR*, 2017. 5

[61] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM SIGGRAPH*, 2006. 1, 2

[62] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021. 3, 6, 7

[63] Yurun Tian, Bin Fan, and Fuchao Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, 2017. 2

[64] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *TPAMI*, 1991. 2, 3

[65] Ziming Wang, Xiaoliang Huo, Zhenghao Chen, Jing Zhang, Lu Sheng, and Dong Xu. Improving rgb-d point cloud registration by learning multi-scale local linear transformation. In *ECCV*, 2022. 2

[66] Kyle Wilson and Noah Snavely. Robust global translations with 1dsfm. In *ECCV*, 2014. 1, 2

[67] Heng Yang, Wei Dong, Luca Carlone, and Vladlen Koltun. Self-supervised geometric perception. In *CVPR*, 2021. 2

[68] Zi Jian Yew and Gim Hee Lee. RPM-Net: Robust Point Matching using Learned Features. In *CVPR*, 2020. 2

[69] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, 2018. 2, 3

[70] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3DMatch: Learning local geometric descriptors from RGB-D reconstructions. In *CVPR*, 2017. 2

[71] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 1995. 2

[72] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv*, 2018. 5, 7

[73] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2