Learning Compact Geometric Features
Supplementary Material

Marc Khoury
UC Berkeley

Qian-Yi Zhou
Intel Labs

Vladlen Koltun
Intel Labs

A. Additional Controlled Experiments

There are a number of hyperparameters and design choices that affect performance. We present a set of controlled experiments that validate each choice and provide intuition on how these choices affect the results. These experiments are performed on the validation set of the laser scan data (the Dancing Children model).

A.1. Dimensionality

The most important parameter is the dimensionality of the feature space. Figure 1 shows the performance of the learned feature for five different settings of dimensionality. As expected, increasing the dimensionality improves performance. However, we observe diminishing returns after $n = 32$.

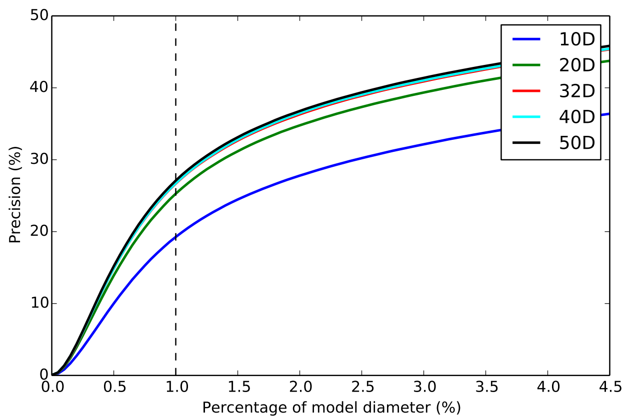


Figure 1. Precision of our learned feature as we increase the dimensionality of the embedding space.

A.2. Input Parameterization

Recall that our proposed input parameterization is $R = 17$ subdivisions in the radial direction, $E = 11$ in the elevation direction, $A = 12$ in the azimuth direction, and a search radius of 17% of the diameter of the model. The resultant dimensionality of the input histogram is $N = 2{,}244$. In each experiment we select one of these parameters and consider the parameterization resulting from changing this value by ± 2.

Figure 2 shows the precision for different values of the number of radial subdivisions. The number of radial subdivisions is the only parameter where precision does not increase as we increase the value of the parameter. As Figure 2 shows, $R = 17$ performs slightly better than the other two settings.

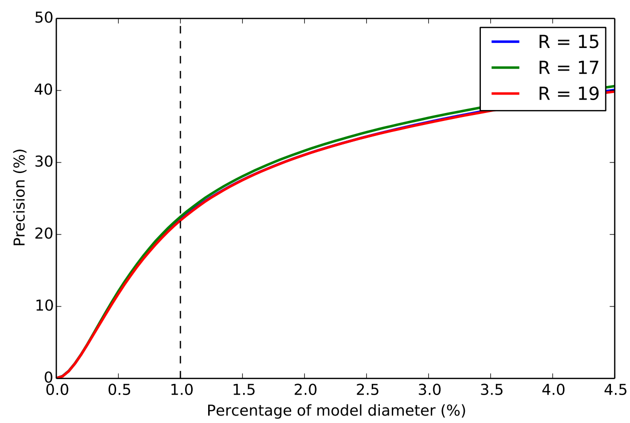


Figure 2. Precision of our learned feature as we increase the number of radial subdivisions.

If instead we increase both the number of radial subdivisions and the search radius in tandem, as shown in Figure 3, then precision continues to increase. The gain of going from $R = 17$ with a search radius of 17% to $R = 19$ with a search radius of 19% is smaller than the gain of going from $R = 15$ with a search radius of 15% to $R = 17$ with a search radius of 17%.

Figure 4 shows the precision for different values of the number of elevation subdivisions. There is a gain of 1.7 percentage points in going from $E = 9$ to $E = 11$, and negligible gain thereafter.

Figure 5 shows the precision for different values of the number of azimuth subdivisions. Precision increases as the number of azimuth subdivisions increases, but the gains are negligible.

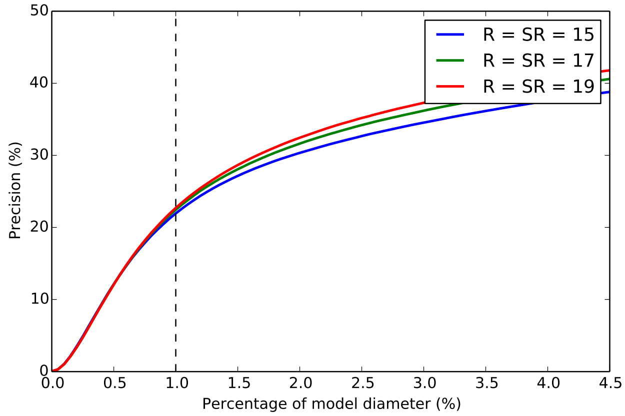


Figure 3. Precision of our learned feature as we increase the number of radial subdivisions and the search radius in tandem.

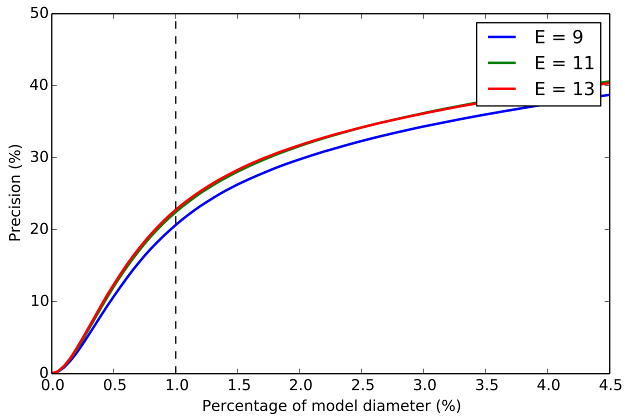


Figure 4. Precision of our learned feature as we increase the number of elevation subdivisions.

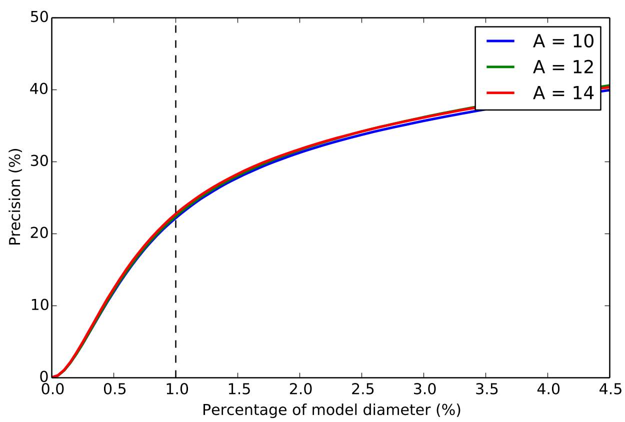


Figure 5. Precision of our learned feature as we increase the number of azimuth subdivisions.

A.3. Architecture Design

The architecture of our model consists of $L = 5$ hidden layers, each with $H = 512$ hidden units. As Figure 6 shows, $L = 5$ performs slightly better than using a model architecture with $L = 4$ or $L = 6$ hidden layers. Furthermore there is a gain of 1.5 percentage points in going from $H = 256$ to $H = 512$, with $L = 5$.

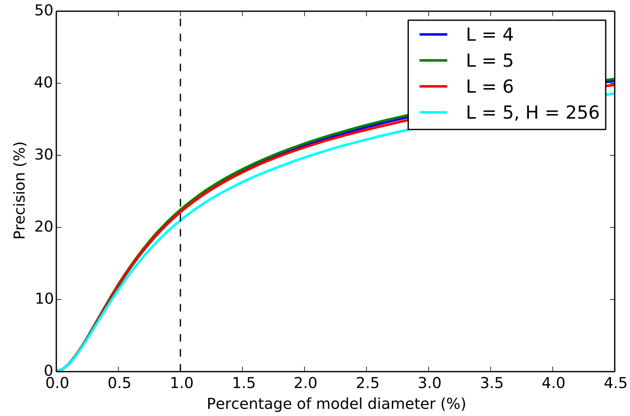


Figure 6. Precision of the learned feature as we vary the depth and width of our model.

A.4. Other Design Choices

Alternatively, we could consider different loss functions or different input parameterizations. Figure 7 shows the effect of two such modifications. The first uses SHOT features as the input to our model, instead of our input parameterization. The second uses the contrastive loss for training [1] instead of the triplet loss. Substituting SHOT features for our input parameterization reduces precision from 22.4% to 12%. Substituting the contrastive loss for the triplet loss reduces precision to 2.75%.

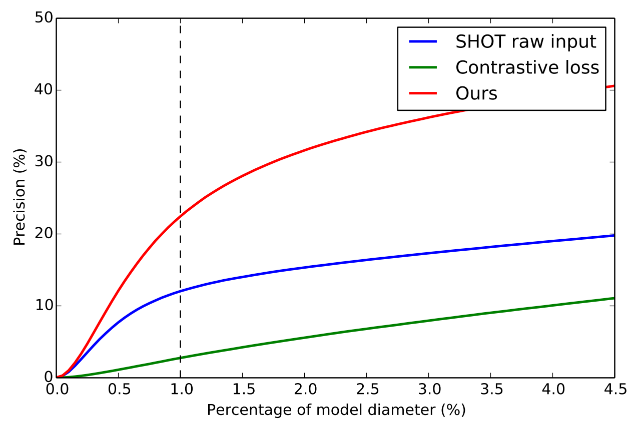


Figure 7. Effect of the input parameterization and the embedding objective. Our input parameterization and loss are compared to using SHOT features as input (everything else held fixed) and using a contrastive loss instead of the triplet loss (everything else fixed).

B. Additional Baselines

As an additional set of baselines we apply Principal Components Analysis (PCA) to each of the baseline feature descriptors to embed them into a 32-dimensional space.

Laser scan data. Figure 8 shows the precision of different feature descriptors after applying PCA to embed each into a 32-dimensional space on the laser scan test set. The precision values for each feature after applying PCA are 27.9% for PCA USC, 26.5% for PCA SI, 21.2% for PCA FPFH, 15.7% for PCA PFH, 14.8% for PCA RoPS, and 12% for PCA SHOT. These values can be compared to the precision achieved by the original descriptors: 31.5% for USC, 32.2% for SI, 21.2% for FPFH, 15.8% for PFH, 15.3% for RoPS, and 14.8% for SHOT. Most prior features lose a few percentage points of precision when projected into the lower-dimensional space. Both the original descriptors and their lower-dimensional versions are considerably less discriminative than our learned low-dimensional descriptor.

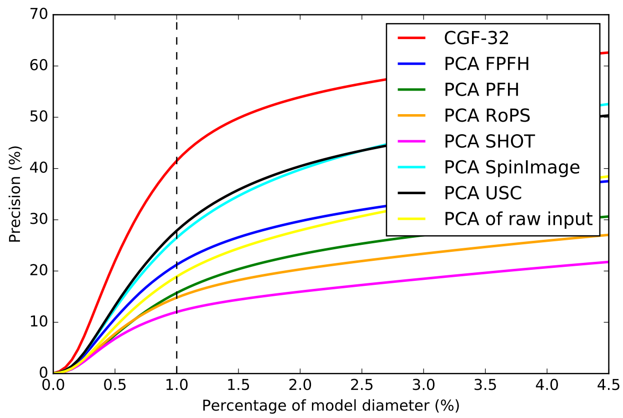


Figure 8. Precision of prior feature descriptors embedded into a 32-dimensional space using PCA, compared to CGF-32. Results on the laser scan test set.

SceneNN data. Figure 9 shows the precision of different feature descriptors after applying PCA to embed each into a 32-dimensional space on the SceneNN test set. The precision values for each feature after applying PCA are 22.4% for RoPS, 21% for PCA PFH, 20.7% for PCA FPFH, 20.6% for PCA USC, 18.2% for PCA SHOT, and 6.6% for PCA SI. These values can be compared to the precision achieved by the original descriptors: 22.7% for RoPS, 21.1% for PFH, 20.7% for FPFH, 29.8% for USC, 20.2% for SHOT, and 8.2% for SI. Most prior features lose a few percentage points of precision when projected into the lower-dimensional space. Both the original descriptors and their lower-dimensional versions are considerably less discriminative than our learned low-dimensional descriptor.

C. Approximate Nearest Neighbors

Approximate nearest neighbor algorithms accelerate nearest neighbor queries in high-dimensional spaces by loosening the constraint that the exact nearest neighbor

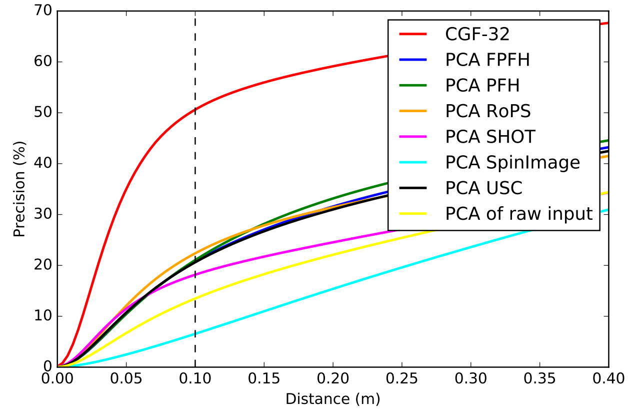


Figure 9. Precision of prior feature descriptors embedded into a 32-dimensional space using PCA, compared to CGF-32. Results on the SceneNN test set.

must be returned. Given a query point $\mathbf{q}$, an approximate nearest neighbor query returns a point $\mathbf{p}$ at distance within a factor of K of the nearest-neighbor distance. The tradeoff between speed and accuracy is controlled by the parameter K.

We demonstrate that CGF-32 is robust to approximation. Thus, in practice, our reported query times can be sped up even further using approximate nearest neighbor queries, at almost no loss in precision. The results for different values of K are shown in Figure 10. For $K = 20$, CGF-32 loses only 0.8 percentage points of precision at 1% of the model diameter.

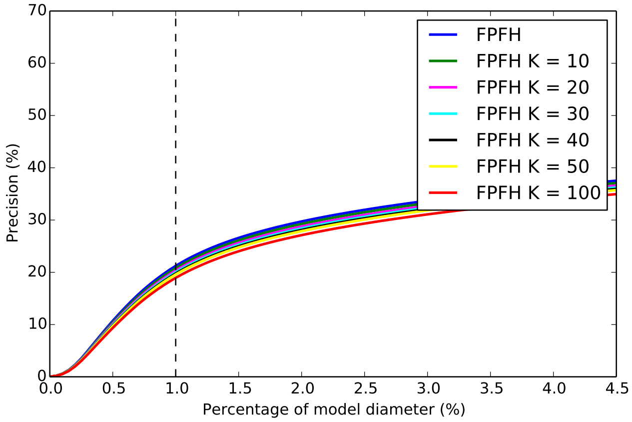
For the baseline features, at $K = 20$, FPFH loses 0.8 percentage points, PFH loses 1 percentage point, SHOT loses 1.6 percentage points, SI loses 2.3 percentage points, and USC loses 9.9 percentage points.

D. Geometric Registration

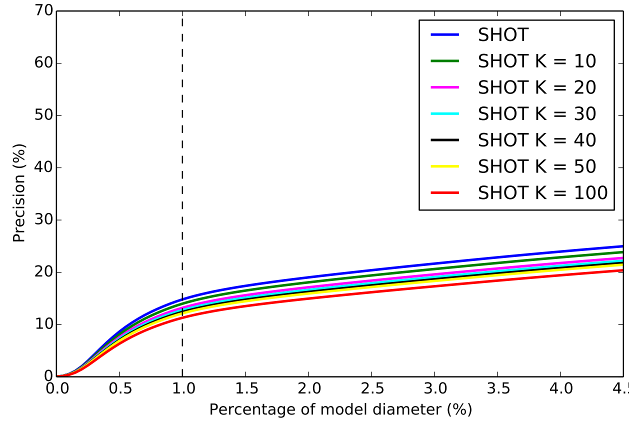
Figure 11 (multiple pages) shows 20 *randomly sampled* fragment pairs from the SceneNN test set and corresponding alignments produced by FGR with CGF-32. This illustrates the quantitative results presented in Section 7.5 in the paper.

References

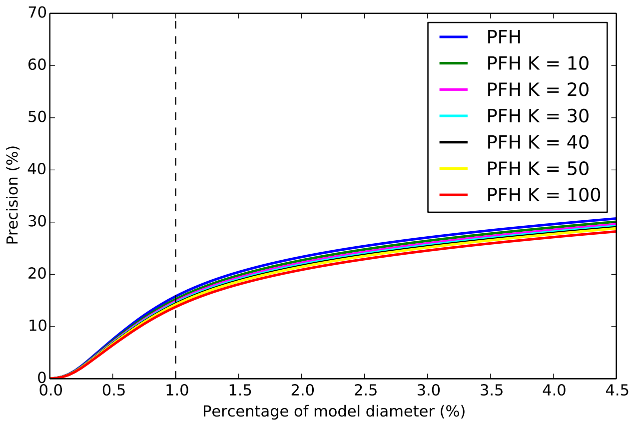
[1] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2



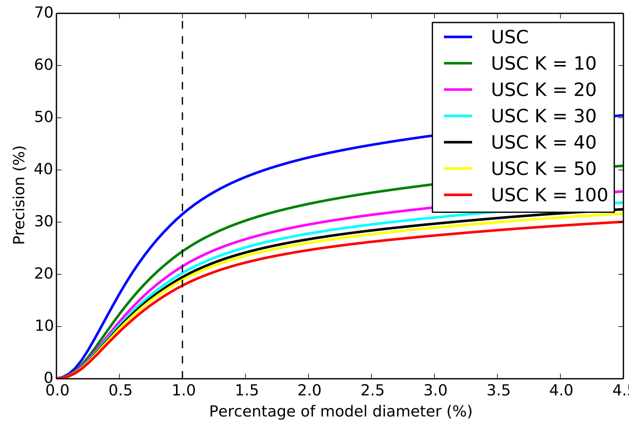
(a) FPFH



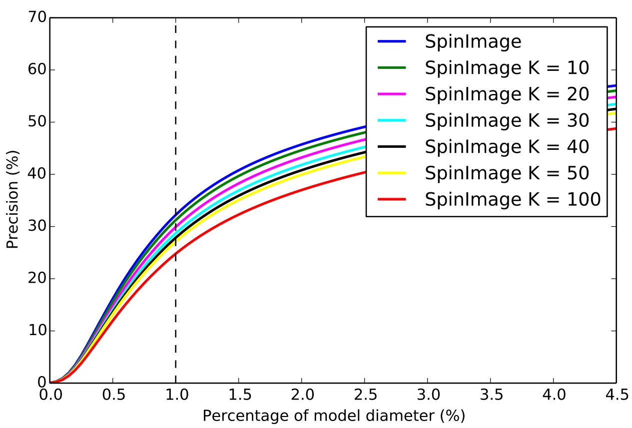
(b) SHOT



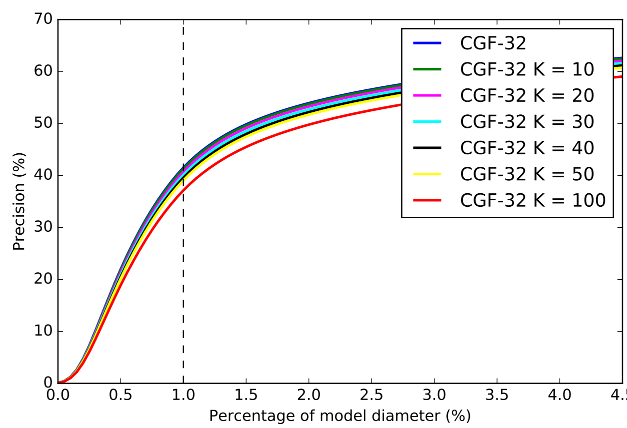
(c) PFH



(d) USC



(e) SI



(f) CGF-32

Figure 10. Robustness of different feature spaces to approximate nearest neighbor search. As the approximation factor K increases, the precision of retrieved matches decreases. Some feature spaces are more robust than others and the decline in precision as a factor of K is smaller. For $K = 20$, our feature space loses only 0.8 percentage points in precision. USC is the least robust feature space, losing 9.9 percentage points for $K = 20$.

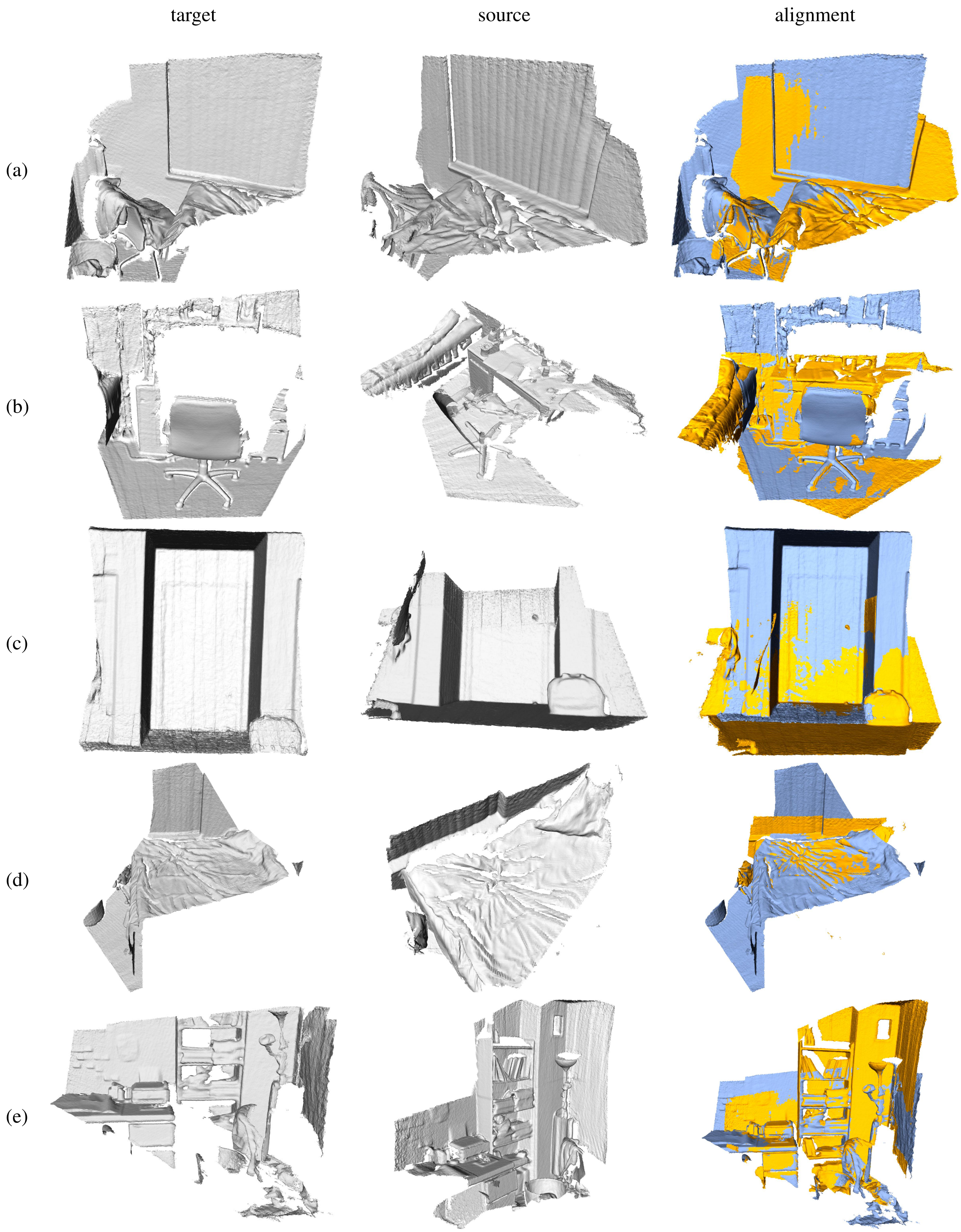


Figure 11. *Randomly sampled* fragment pairs from the SceneNN test set (left, middle) and corresponding alignments produced by FGR with CGF-32 (right).

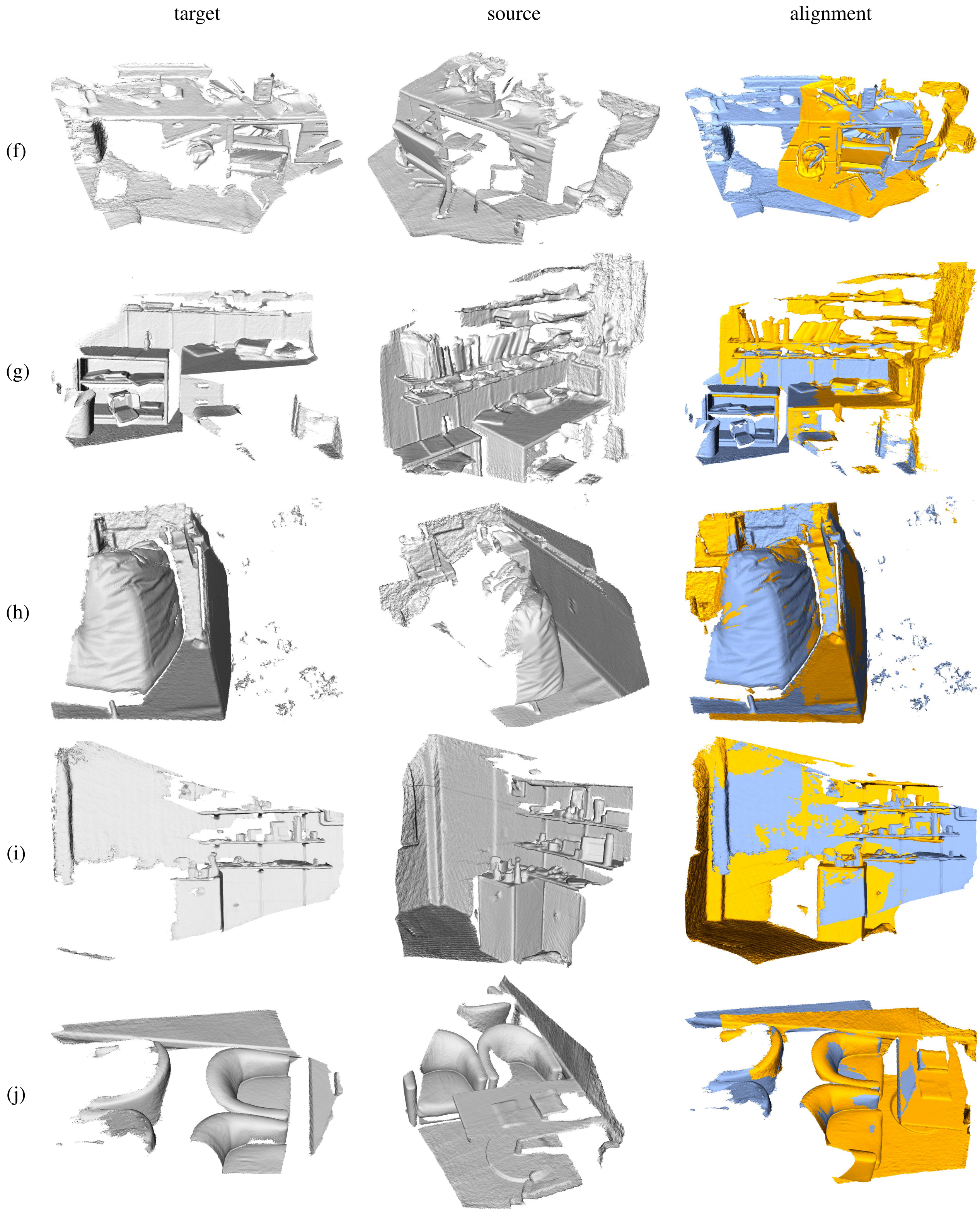


Figure 11 (cont.). *Randomly sampled* fragment pairs from the SceneNN test set (left, middle) and corresponding alignments produced by FGR with CGF-32 (right).

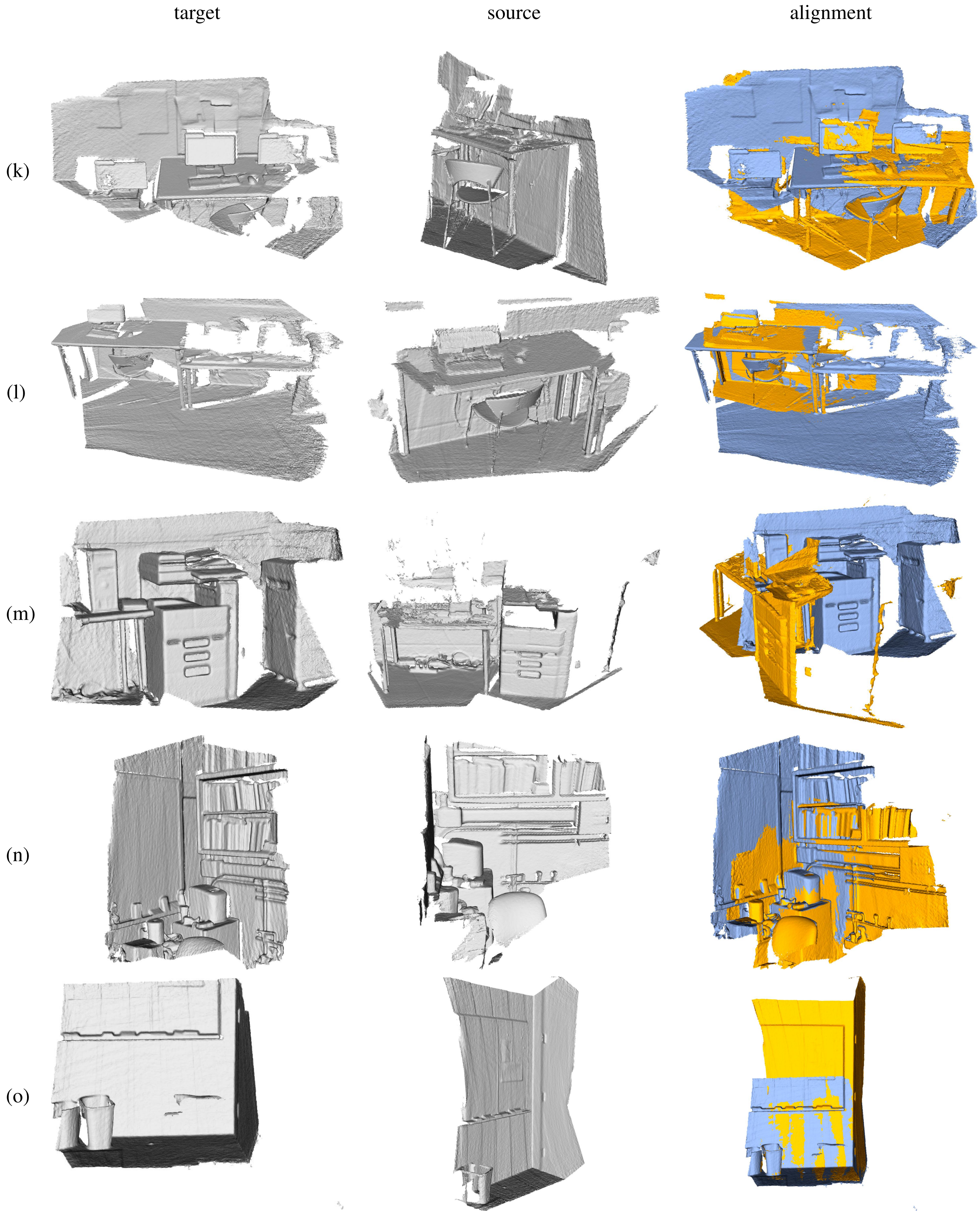


Figure 11 (cont.). *Randomly sampled* fragment pairs from the SceneNN test set (left, middle) and corresponding alignments produced by FGR with CGF-32 (right).

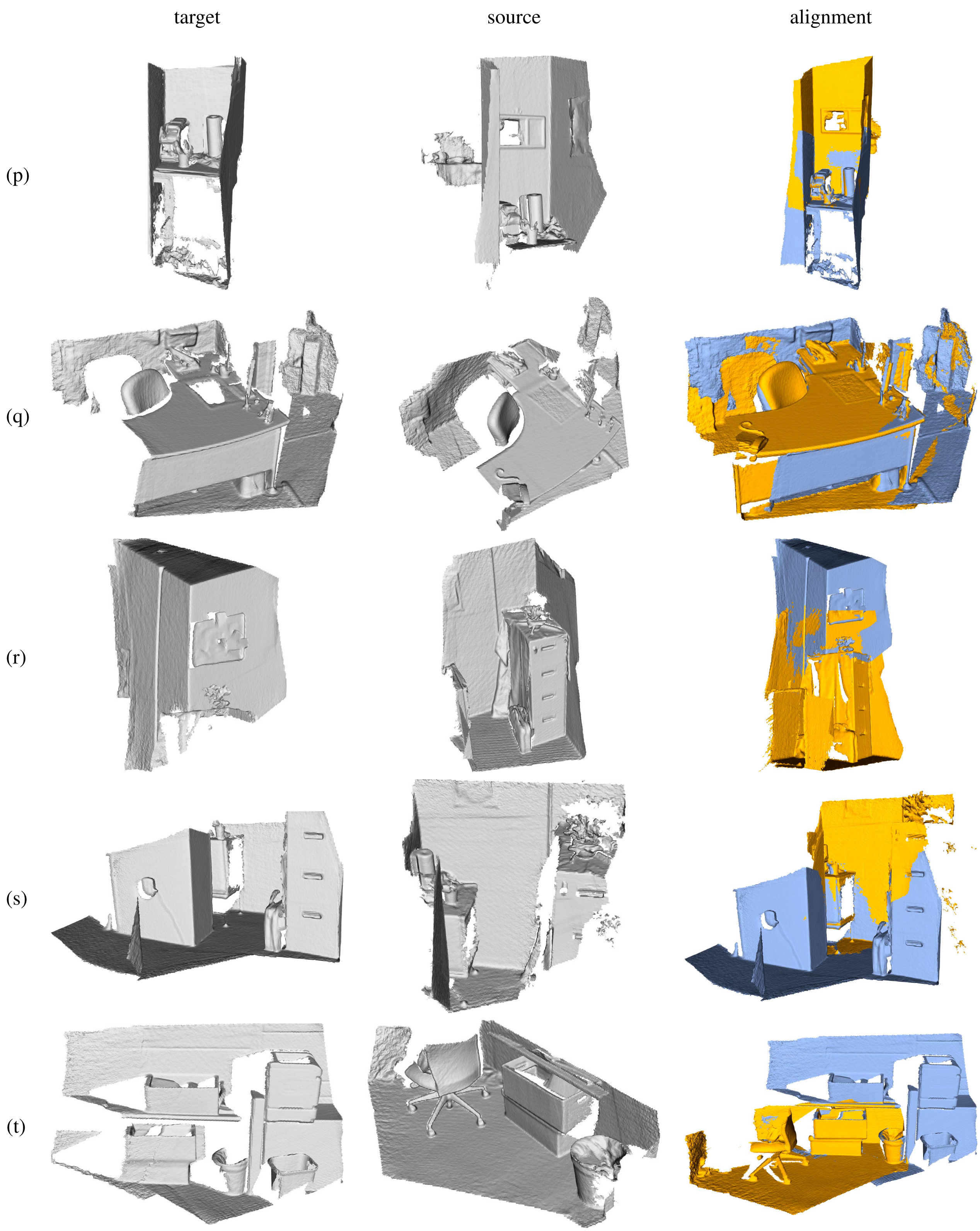


Figure 11 (cont.). *Randomly sampled* fragment pairs from the SceneNN test set (left, middle) and corresponding alignments produced by FGR with CGF-32 (right).