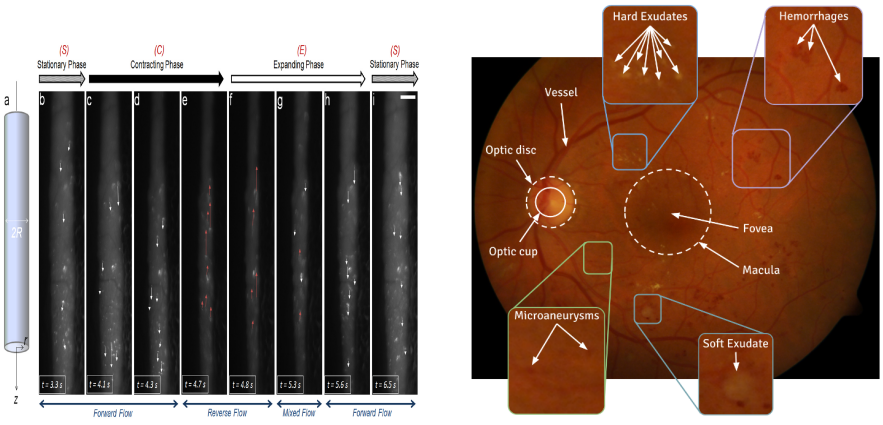
Simplicial Complex based Point Correspondence between Images warped onto Manifolds (supplementary material)

Charu Sharma and Manohar Kaul

Department of Computer Science & Engineering,
Indian Institute of Technology Hyderabad, India
{cs16resch11007,mkaul}@iith.ac.in

A Proof of Lemma 1



(a) (b)

Fig. 1. Example of (a) a nanofluid microflow pattern monitored by tracking the colloidal microparticles [3], (b) a fundus image with retinal morphologies and pathologies [2].

We begin this proof by focusing on the case where $k \geq 2$. We know that each k-clique contains $\binom{k}{2}$ edges in it.

Now, we study how many k-cliques an arbitrary edge (u, v) can belong to. Notice that if edge (u, v) belongs to a k-clique K, then all other vertices in the K must also be adjacent to u and vertex u can have at most $\delta - 1$ neighbors that are not v. Therefore, if K contains both vertices u and v, then K contains $k - 2$ other vertices and each of them must also be neighbors of u. So, combining all these observations, if a k-clique K must contain both u and v and $k - 2$ of the other $\delta - 1$ maximum allowed neighbors of u, then there must be at most $\binom{\delta-1}{k-2}$ such k-cliques.

Thus, G has at most $\frac{m\binom{\delta-1}{k-2}}{\binom{k}{2}}$ k-cliques. Then, the total number of k-cliques in G for $k \geq 2$ is

$$\sum_{k=2}^{h} \frac{m\binom{\delta-1}{k-2}}{\binom{k}{2}} = m \sum_{k=2}^{h} \frac{2}{k(k-1)} \frac{(\delta-1)!}{(k-2)!(\delta+1-k)!} \tag{1}$$

$$= m \sum_{k=2}^{h} \frac{2(\delta-1)!}{k!(\delta+1-k)!} \tag{2}$$

We need a $(\delta+1)!$ term in the numerator of Equation QAP, since we know that $h \leq (\delta+1)$. So,

$$m \sum_{k=2}^{h} 2 \left\{ \frac{(\delta+1)!}{(\delta+1)(\delta)} \right\} \frac{1}{k!(\delta+1-k)!} \tag{3}$$

$$= \frac{2m}{\delta(\delta+1)} \sum_{k=2}^{h} \frac{(\delta+1)!}{k!(\delta+1-k)!} \tag{4}$$

$$= \frac{2m}{\delta(\delta+1)} \left(\sum_{k=0}^{h} \frac{(\delta+1)!}{k!(\delta+1-k)!} - \frac{(\delta+1)!}{1!\delta!} - \frac{(\delta+1)!}{0!(\delta+1)!} \right) \tag{5}$$

$$= \frac{2m}{\delta(\delta+1)} \left(\left\{ \sum_{k=0}^{h} \binom{\delta+1}{k} \right\} - \delta - 2 \right) \tag{6}$$

$$\leq \frac{2m}{\delta(\delta+1)} \left[\min \left\{ (\delta+1)^h + 1, \left(\frac{e(\delta+1)}{h} \right)^h \right\} - \delta - 2 \right] \tag{7}$$

In Equation 6, the term in braces $\{\}$, is the partial sum of the first h binomial coefficients. Note that for $h = \delta + 1$, the term would reduce to $2^{\delta+1}$. Finally, we must also account for the n 1-cliques (vertices) in G. This completes the proof. □

B Additional Experiments

B.1 Datasets

Details of all six datasets used for warped image matching are as follows: (i) *Chessboard* dataset has 28 spherical images in which 16 are omnidirectional and 12 are fish-eye images. The images contain a view of a room with a chessboard, chair, table, people, and computer systems. (ii) *Kamaishi* dataset has 15 panoramic images of video frames from a moving car. (iii) We picked 10 panoramic images covering various scenes from *SUN360* dataset. (iv) *Desktop* dataset consists of two kinds of images (7 omnidirectional and 9 planar). Images represent a room in which a desktop is positioned on a table. (v) *Parking* dataset also has omnidirectional and planar images with a view outside a house near a parking area. (vi) *Table* dataset contains omnidirectional, fish-eye, and planar images. This dataset has few tables in the images with computer systems, boards, chairs, etc. placed in a room. All our annotated datasets will be made publicly available.

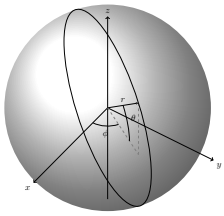


Fig. 2. Parameterized $\mathcal{S}^2$ (for Example 2 in main draft)

In addition to experiments in our main paper, we conducted additional experiments for different settings. In *multi-modal matching*, we performed matching on *unwrapped-unwrapped spherical/warped* images and *unwrapped spherical/warped vs planar* images. In *ablative studies*, we perfomed experiments considering two noise models and geometric verification and rectification. Additional experiments on rotation are performed for SUN360 dataset and Desktop dataset. We also analyzed the effect of missing points on four datasets (Chessboard, Desktop, Kamaishi and Table). Additionally, we also studied effects of varying k values in a neighborhood of landmark points, varying the radius of the sphere and manifold, and varying the number of landmark points for matching. Matching images for different cases are shown in Figures 11 and 12. We also created a video of frames of Kamaishi dataset where we match points of frames in a sequence which can be useful for tracking purposes. The video titled *Kamaishi_Panorama_Matching_Video.avi* is located in the folder of our supplementary material.

Table 1. Error (%) of pairwise matching between spherical (omnidirectional and fish-eye) and planar images of three datasets for different methods.

Algorithms	**Desktop**	**Parking**	**Table**
OurWarped_Cone	14.71 ± 0.0 %	4.41 ± 0.0 %	0.88 ± 0.0 %
OurWarped_Ellip	13.82 ± 0.0 %	5.94 ± 0.0 %	2.65 ± 0.0 %
OurSpherical	**15.16 ± 0.0** %	**5.53 ± 0.0** %	**0.78 ± 0.0** %
FGM + geodesic	97.94 ± 0.24 %	97.53 ± 0.57 %	96.98 ± 2.51 %
SPHORB	41.5 ± 0.0 %	53.85 ± 0.0 %	70.88 ± 0.0 %
BRISK	37.84 ± 0.0 %	51.05 ± 0.0 %	65.11 ± 0.0 %
ORB	34.82 ± 0.0 %	48.82 ± 0.0 %	62.78 ± 0.0 %
Tensor	93.85 ± 0.0 %	94.43 ± 0.22 %	56.94 ± 1.02 %
FGM	98.0 ± 0.0 %	97.50 ± 0.01 %	97.5 ± 0.01 %

B.2 Matching on Spherical/Warped vs. Planar

To understand the effects of matching spherical / warped images versus planar images, we conducted matching experiments between planar and warped images, whose results

are shown in the first two rows of Table 1. We find that the matching error is higher for warped-planar matching compared to warped-warped matching shown in Table in main paper because of the difference in geodesic distances between landmarks embedded on the warped versus the Euclidean distance between landmarks on planar images.

Next, planar images are compared to both fish-eye and omnidirectional images. The results are shown in Table 1 in which we clearly observe an increase in error for the Desktop dataset when compared to the omni-omni matching results in Table in main paper. As planar images cover a much smaller portion of the scenes as compared to omnidirectional images, they also contain much fewer landmark points with many occluded regions, which in turn affects our method due to the lack of higher-dimensional cliques. However, our method still beats other state-of-the-art methods.

B.3 Matching on Planar vs. Unwrapped Spherical/Warped

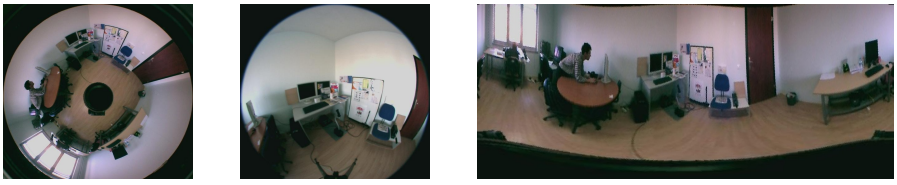
We observe that matching between spherical and planar is a challenge due to the difference in euclidean and geodesic distances in planar and spherical images, respectively, as is evidenced in Table 3 of our main paper. Therefore, to compare these kinds of images, we can perform matching between unwrapped equirectangular spherical/warped and planar images. In this way, we can also apply state-of-the-art graph matching methods which perform well on planar images. Thus, we consider three datasets with both spherical (omnidirectional and fish-eye) and planar images and conducted the experiment. Table 2 shows the results. We can compare the results with Table 3 in the main paper, we can clearly see that the results are almost same even after flattening of the image. Thus, comparing flattened and planar images does not reduce the error of matchings between spherical and planar images. However, our method outperforms the existing methods.

Table 2. Error (%) of pairwise matching between unwrapped equirectangular version of spherical (omnidirectional and fish-eye) images and planar images of three datasets for different methods including graph matching methods on flat surfaces.

Algorithms	**Desktop**	**Parking**	**Table**
OurWarped_Cone	14.26 ± 0.0 %	5.23 ± 0.0 %	1.09 ± 0.0 %
OurWarped_Ellip	15.16 ± 0.0 %	3.59 ± 0.0 %	2.96 ± 0.0 %
OurSpherical	**13.08** ± **0.0** %	**4.10** ± **0.0** %	**0.93** ± **0.0** %
RCC	30.85 ± 1.78 %	12.38 ± 1.53 %	11.69 ± 1.58 %
EigenAlign	99.94 ± 0.0 %	99.22 ± 0.0 %	99.25 ± 0.0 %
Tensor	97.88 ± 0.11 %	95.59 ± 0.35 %	81.04 ± 1.43 %
FGM	99.0 ± 0.0 %	100.0 ± 0.0 %	97.0 ± 0.0 %
SPHORB	41.33 ± 0.0 %	59.9 ± 0.0 %	71.15 ± 0.0 %
BRISK	37.67 ± 0.0 %	55.98 ± 0.0 %	67.8 ± 0.0 %
ORB	34.43 ± 0.0 %	45.03 ± 0.0 %	64.84 ± 0.0 %

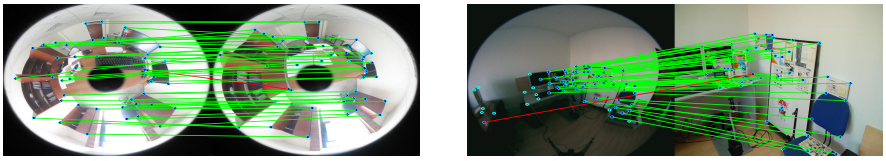
B.4 Effect of Noise Models

Our proposed matching method is subjected to two random noise models proposed by Feizi et. al. [1]. While matching two simplicial complexes $\mathcal{M}$ and $\mathcal{M}'$, noise is



(a) (b) (c)

Fig. 3. Example of (a) a parabolic omnidirectional image, (b) fish-eye, and (c) panoramic (unwrapped equirectangular) images of the same view.



(a) (b)

Fig. 4. Instances of matchings between (a) *Desktop* omnidirectional images and (b) *Table* fish-eye and planar images. Green/red lines show correct/incorrect matches respectively. Isolated points show no matches.

introduced in the first simplicial complex $\mathcal{M}$. We denote the *noisy simplicial complex* as $\widetilde{\mathcal{M}}$. Next, we attempt a matching between $\widetilde{\mathcal{M}}$ and $\mathcal{M}'$. The noisy complex $\widetilde{\mathcal{M}}$ is generated as follows:

$$\textbf{Noise Model I: } \widetilde{\mathcal{M}} = \mathcal{M} \odot (1 - P) + (1 - \mathcal{M}) \odot P$$

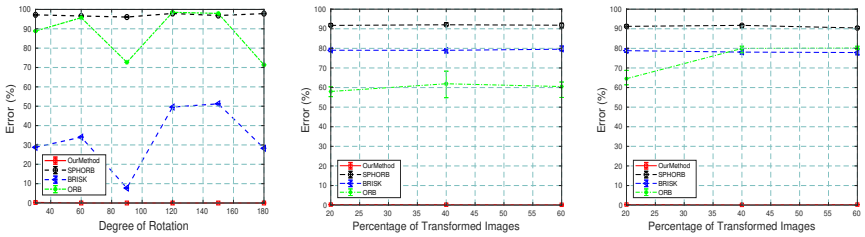
$$\textbf{Noise Model II: } \widetilde{\mathcal{M}} = \mathcal{M} \odot (1 - P) + (1 - \mathcal{M}) \odot Q$$

where, $\odot$ represents the *element-wise multiplication* with matrices P / Q which are

Table 3. Error (%) of pairwise matching between unwrapped spherical (omnidirectional and fish-eye) images of four datasets for different graph matching methods with two noise models.

Algorithms	**Chessboard**	**Desktop**	**Parking**	**Table**
		Noise Model I		
OurMethod	**4.17 ± 0.0 %**	**0.85 ± 0.0 %**	**0.0 ± 0.0 %**	**0.31 ± 0.0 %**
RCC	29.7 ± 0.68 %	17.1 ± 1.17 %	15.8 ± 7.89 %	16.2 ± 0.48 %
EigenAlign	99.5 ± 0.0 %	96.5 ± 0.0 %	97.7 ± 0.0 %	98.2 ± 0.0 %
FGM	93.0 ± 0.0 %	72.0 ± 0.0 %	65.0 ± 0.0 %	92.0 ± 0.0 %
		Noise Model II		
OurMethod	**4.36 ± 0.0 %**	**0.64 ± 0.0 %**	**0.0 ± 0.0 %**	**0.44 ± 0.0 %**
RCC	29.9 ± 0.52 %	17.0 ± 1.01 %	14.2 ± 5.26 %	16.1 ± 0.42 %
EigenAlign	98.9 ± 0.0 %	97.2 ± 0.0 %	98.5 ± 0.0 %	99.1 ± 0.0 %
FGM	94.0 ± 0.0 %	71.0 ± 0.0 %	60.0 ± 0.0 %	92.5 ± 0.0 %

binary random symmetric matrices drawn from a Bernoulli distribution with v vertices and p / q probabilities, respectively. Here, P *flips* the *vertex-vertex adjacency* with probability p and Q *adds new edges* between non-connected vertices with probability q to the underlying geometric graph $G = (V, E, g)$ of the simplicial complex $\mathcal{M}$. Experiments are performed between pairwise flattened (unwrapped) spherical images of four datasets against graph matching methods. Table 3 shows matching results after the application of both noise models and it shows that our method is the most robust.



(a) (b) (c)

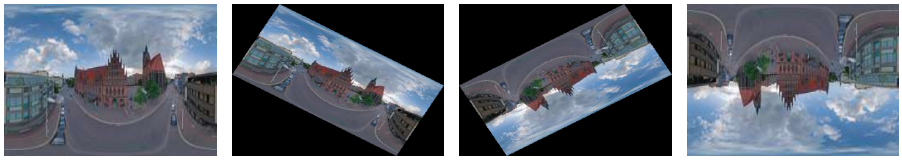
Fig. 5. Error(%) of matching (a) on SUN 360 dataset. We selected 10 spherical images and created their rotated versions. Matching pairs of images for different angles from 30° to 180° rotation. $(b) - (c)$ when varying the percentage (20% to 60%) of transformed images in the set of spherical images of *Desktop* for 20° and 60° rotation respectively.

B.5 Effect of Rotation

We study the effect of rotation (affine transformation) on a simulated dataset. We collected 10 different images from SUN360 dataset and annotated them. Therefore, the number of landmark points vary in all the images. The images are panoramic spherical images. We perform rotation on the images from 30° to 180° and compare the original image against the rotated images. Figure 6(b) to 6(d) are the rotated versions of the original image in Figure 6(a). Matching results are presented in Figure 5(a) against three feature descriptor methods. We note that, there is an increase in the error percentage for other methods as the degree of rotation increases from 30° to 180°, except in the 90° rotation case, as 90° rotation does not introduce distortions in the image and the pairwise-distances between points can be preserved. Whereas, our method is quite stable to transformation by rotation. We also rotated images (clockwise) by 20° and 60° for omnidirectional images of Desktop dataset and randomly transformed 20% to 60% of images in a set of spherical images. Results for both the rotations are shown in Figures 5(b) and 5(c).

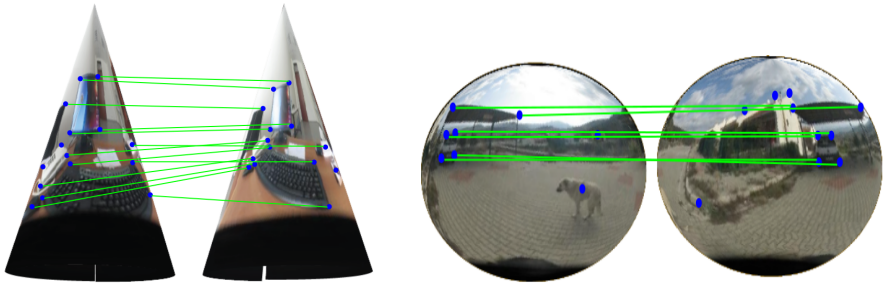
B.6 Effect of varying k-NN, Radius of surface and Number of points

We study the effect of parameters (k, R) and the effect of sparse / dense set of landmark points by performing few experiments for our method (spherical) and our warped on cone



(a) (b) (c) (d)

Fig. 6. (a) Original image from SUN360 dataset, rotated images after (b) 30° rotation, (c) 150° rotation, and (d) 180° rotation. Matching results are shown in Figure 5(a) with the original image (a).



(a) (b)

Fig. 7. Matching on warped images of (left) Cone and (right) Ellipsoid. Green lines show correct matches and isolated points show no matches.

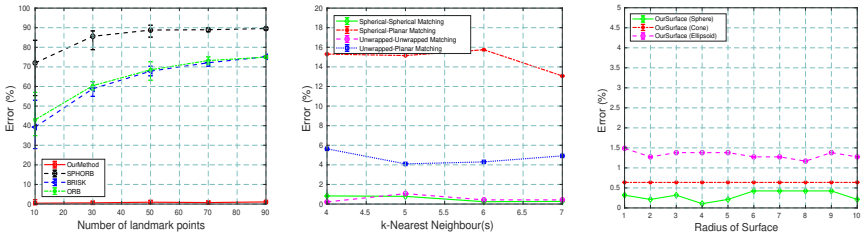
and ellipsoid methods and the existing feature descriptor based methods, respectively.

Vary k-Nearest Neighbour: We performed few experiments of matching between spherical vs. spherical (for Kamaishi dataset), spherical vs. planar (Desktop), unwrapped spherical vs. unwrapped spherical (Desktop) and unwrapped spherical vs. planar (Parking) images by varying k parameter of k-NN while generating a simplicial complex. Figure 8(b) shows the results for four different datasets with k varying from range 4 to 7 on x-axis. Although, the value of k changes the matching results, the error doesn't change significantly. Selection of k is important since it defines the local connectivity of a vertex. So, larger the value of k, larger are the adjacent vertices of a vertex in question. However, a large value of k can slow down the algorithm and hence this trade-off must be taken into consideration while choosing k.

Vary Radius of Surface: Projecting an image onto a sphere or manifold depends on the radius of the underlying manifold. Change in radius affects the rate of distortion of projection onto the manifold. Thus, we conducted an experiment on spherical (omnidirectional) images of Desktop dataset. Results for varying R ranging from 1 to 10 are shown in Figure 8(c). Matching is performed on our three methods including spherical (radius of 2-sphere), cone (radius of cone) and ellipsoid (radius of 2-sphere projected to an ellipse with a fixed major and minor axis). Results report that the change in radius doesn't affect the matching accuracy.

Sparse to Dense set of landmark points: We considered spherical (omnidirectional and fish-eye) images of Table dataset which consists of 93 landmark points originally. We randomly selected 10 to 90 landmark points on x-axis in Figure 8(a). This experiment

analyzes if the matching algorithm depends on sparsity / density of landmark points in an image. From the results, our method doesn't seem to be affected by *removal* or *addition* of points whereas there is a substantial increase in error for other methods.



(a) (b) (c)

Fig. 8. Error(%) of matching by (a) varying the number of landmark points in the images of Table dataset(Sparse to Dense), (b) varying the value of k (from $k = 4$ to 7) in k-NN ball of a landmark point, and (c) varying the radius of surface (from $r = 1$ to 10) of a sphere, cone and an ellipsoid for Desktop dataset.

B.7 Effect of Missing Points

In order to compare the robustness of our method versus the state-of-the-art methods under the effect of landmark points *missing completely at random (MCAR)*, we randomly remove 20% to 60% of the landmark points from 20% to 60% of images in a set of spherical images. We picked an arbitrary image as the original image and match it against the rest of the spherical images, affected by missing points, in the set. We conducted this experiment on four datasets mentioned in Figure 9. We have shown results for removal of points from 20%, 40% and 60% of images in Figure 9. We observe that there is no large increase in error when 20% to 60% of the landmark points are removed randomly for our method. Also, results for other methods are not affected by varying percentage of missing points and there is no increase in the error percentage since removal of points reduces the number of points in matching, which in turn also reduces error for feature descriptor methods because of decrease in probability to mismatch.

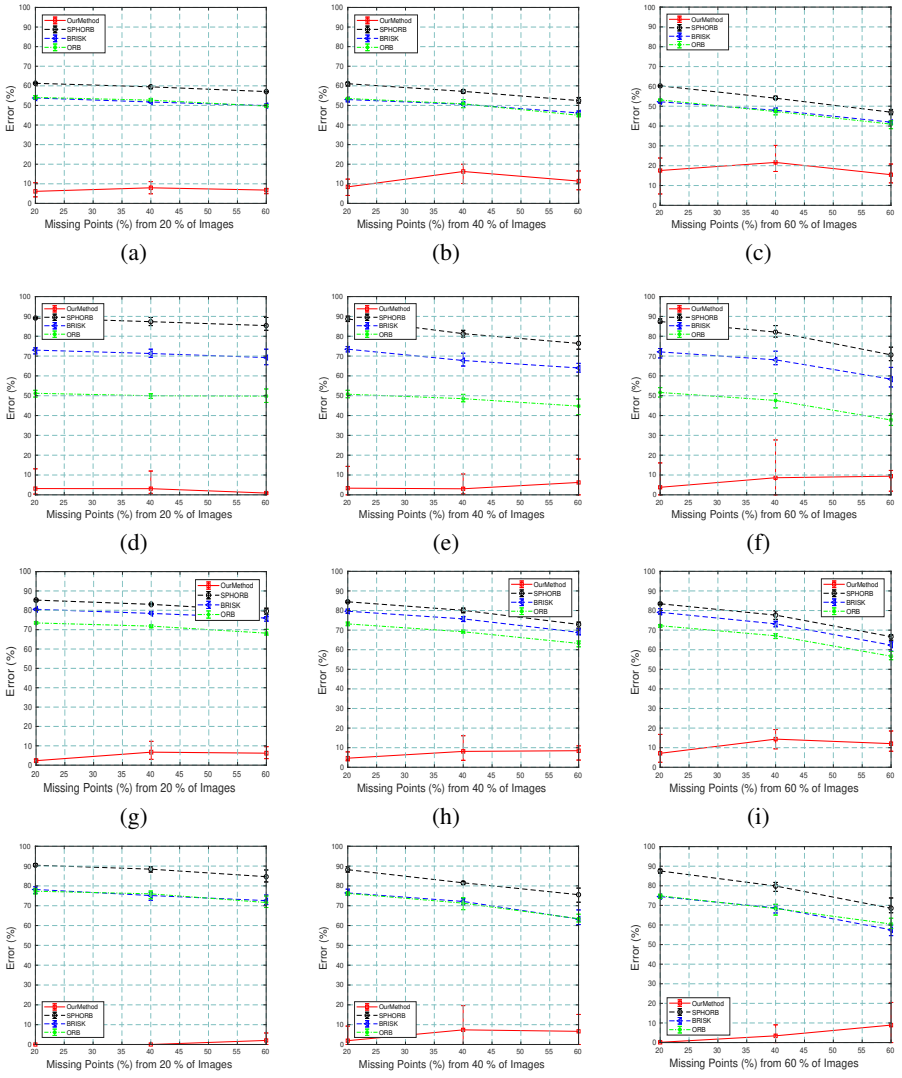
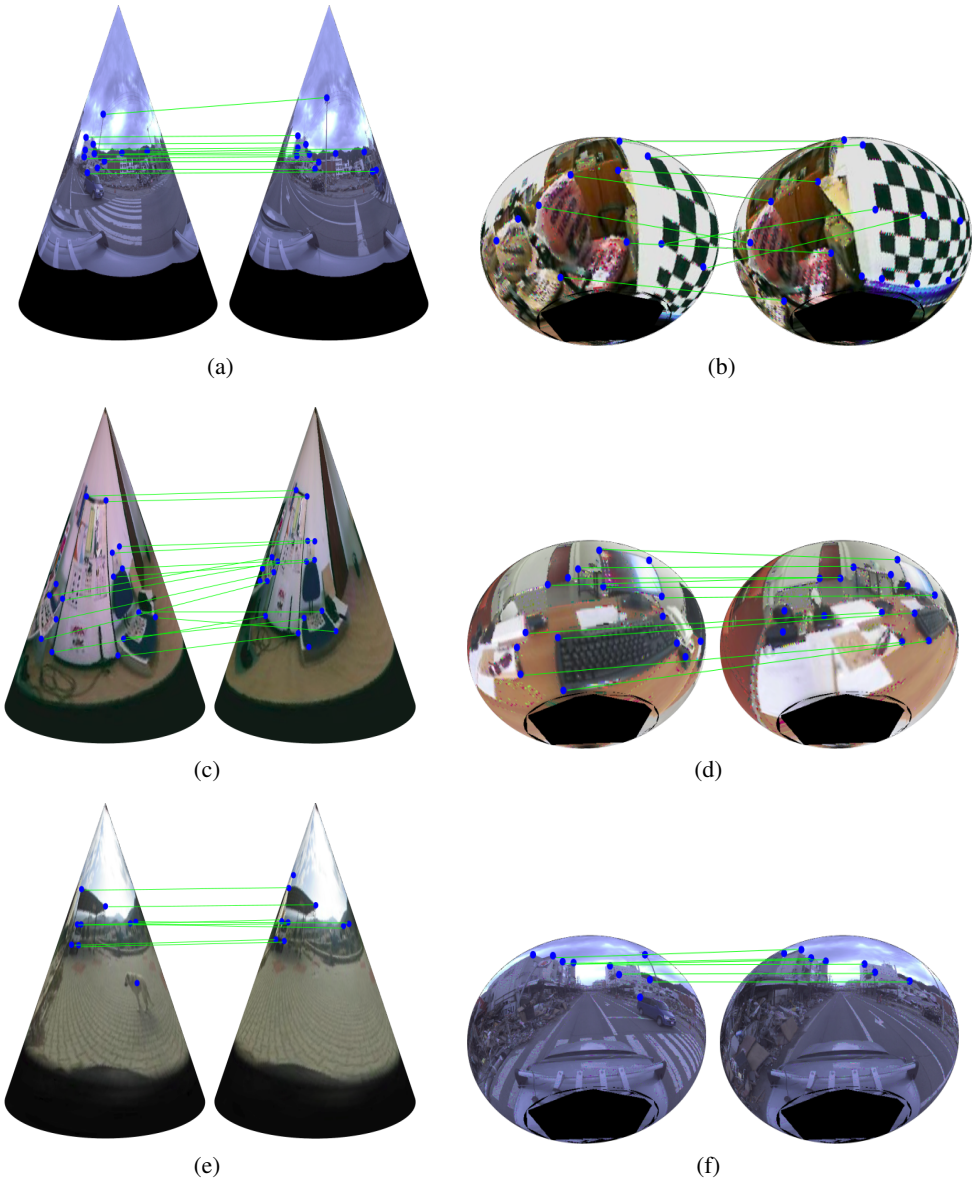


Fig. 9. Error (%) of matching between spherical images by randomly removing points ($20\% - 60\%$) from 20%, 40% and 60% of the images. Matching is computed for four datasets ((a)-(c) Chessboard , (d)-(f) Desktop, (g)-(i) Kamaishi, (j)-(l) Table) from 1^{st} frame to the other $N-1$ frames.



(a) (b) (c) (d) (e) (f)

Fig. 10. Instances of a matching of unwrapped images of different datasets on curved manifold for (a) Kamaishi, (b) Chessboard, (c) Table, (d) Desktop, (e) Parking and (f) Kamaishi. Green lines show correct matches respectively. Isolated points show no matches.

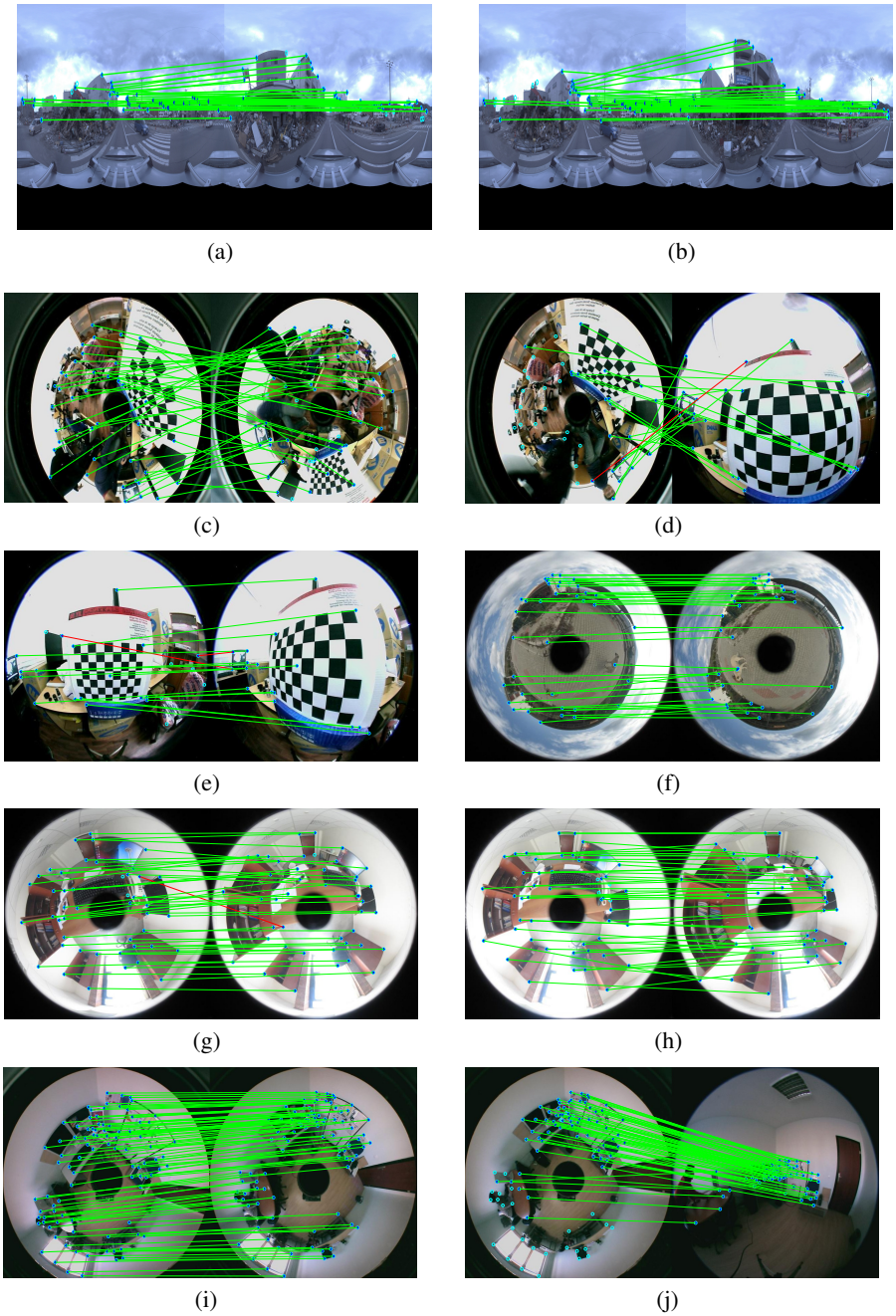


Fig. 11. Instances of matchings between spherical images of $(a) - (b)$ Kamaishi, $(c) - (e)$ Chessboard, (f) Parking, $(g) - (h)$ Desktop and $(i) - (j)$ Table datasets. Green/red lines show correct/incorrect matches respectively. Isolated points show no matches.

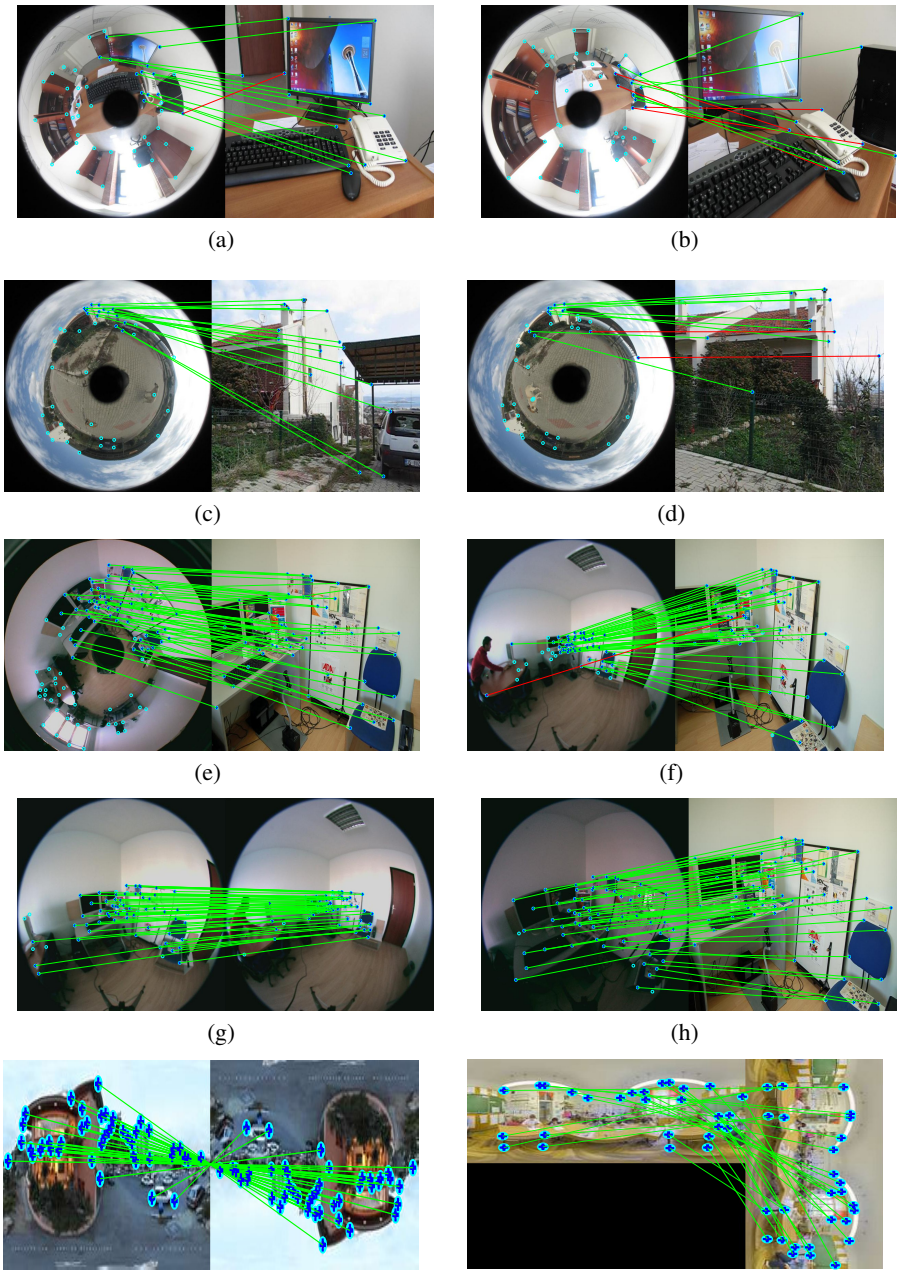


Fig. 12. Instances of matchings of spherical-spherical $((g), (i), (j))$ and spherical-planar $((a) - (f), (h))$ images of $(a) - (b)$ Desktop, $(c) - (d)$ Parking, $(e) - (h)$ Table and $(i) - (j)$ SUN360 datasets. Green/red lines show correct/incorrect matches respectively and isolated points show no matches.

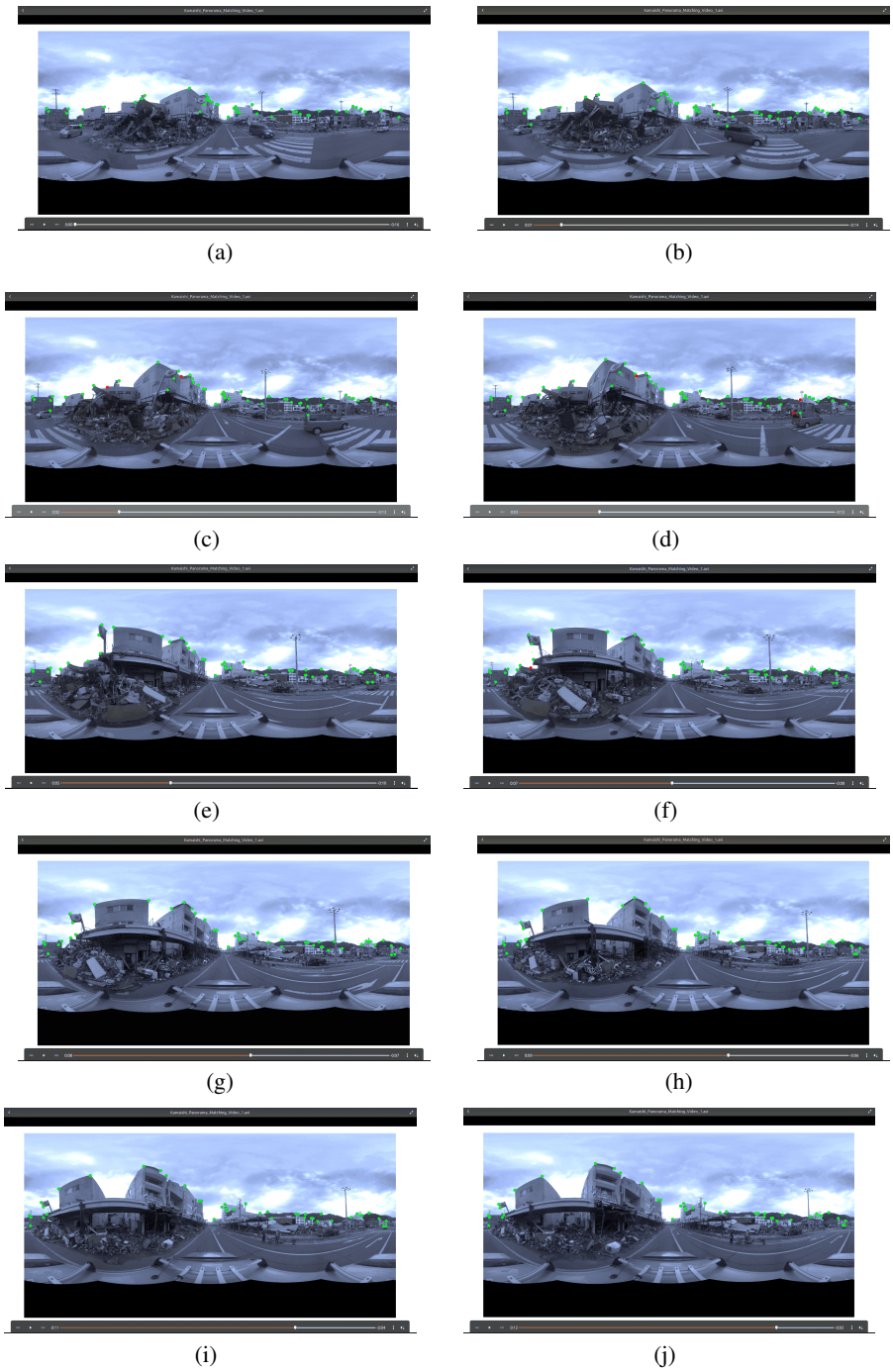


Fig. 13. Instances of a video of $Kamaishi$ dataset with correct matches as green landmark points and incorrect or zero matches as red landmark points.

References

1. Feizi, S., Quon, G., Recamonde-Mendoza, M., Médard, M., Kellis, M., Jadbabaie, A.: Spectral alignment of networks. arXiv preprint arXiv:1602.04181 (2016)
2. Sengupta, S., Singh, A., Leopold, H.A., Lakshminarayanan, V.: Ophthalmic diagnosis and deep learning–a survey. arXiv preprint arXiv:1812.07101 (2018)
3. Sung, B., Kim, S.H., Lee, S., Lim, J., Lee, J.K., Soh, K.S.: Nanofluid transport in a living soft microtube. Journal of Physics D: Applied Physics (2015)