Single-Image Depth Prediction Makes Feature Matching Easier Supplementary Material

Paper ID 2583

In this document we present some additional results and expand on some of the topics in the main paper. Specifically, we provide results on the Aachen Day-Night dataset, which evaluates localization of nighttime query images against a 3D model build from daytime images. We also provide more detailed information on the MonoDepth model used and how it was trained (cf. Sec. 3.1 in the main paper), pose estimation results for eight individual scenes of the dataset (cf. Sec. 5.1 in the main paper), as well as example images from each scene (cf. Fig. 4 in the main paper), and a comparison of SIFT and SuperPoint features in the RobotCar experiments (cf. Sec. 5.2 in the main paper), as well as an evaluation on three scenes from the Extreme View Dataset.

We also provide a supplementary video showing the performance of our approach on the RobotCar dataset.

1 Additional Results on Aachen-Day Night

In addition to the experiments on the RobotCar dataset, we also evaluated our approach on the nighttime queries of the Aachen Day-Night dataset [10, 11]. We follow the experimental setup for the local feature challenge of the CVPR 2019 workshop on "Long-Term Visual Localization under Changing Conditions": each each nighttime query image is matched against a pre-defined set of daytime database images. Similarly, daytime database images are matched with each other. The known poses and intrinsics of the database images, as well as the feature matches between them, are then used to triangulate the 3D scene structure in COLMAP [12]. Finally, the matches between the nighttime queries and the database images, together with known intrinsics for the queries, are used to estimate the camera poses of the query images in COLMAP [12]. We build on the code provided by the organizers[1], with one small difference: the original code performs mutual nearest neighbor matching whereas we use a Lowe ratio test [7] with a threshold of 0.8 as we observed better results when using the ratio test.

For this experiments, we extracted SIFT features using OpenCV, both on the original images and on the rectified versions obtained by our approach. Following [10], we report the percentage of query images localized within (0.5m, 2°), (1m, 5°), and (5m, 10°) of the reference pose (using the evaluation server

[1] https://github.com/tsattler/visuallocalizationbenchmark/tree/master/local_feature_evaluation

provided at https://www.visuallocalization.net/. Using the original images, we obtain 23.5%, 35.7%, and 48.0%, respectively. Extracting features on images rectified by our approach improves the performance to 26.5%, 40.8%, and 53.1%, respectively. As can be seen, our approach is able to significantly improve localization performance. This clearly shows that removing perspective distortion before feature extraction improves pose estimation accuracy under changing viewpoints. Furthermore the results indicate that our method does not cause degradation despite challenges such as day-night changes.

2 Our Depth Prediction Network

In this section we expand on the singe-image depth prediction network used in our method.

Architecture The network architecture is a U-Net similar to the Resnet18-based architecture in Monodepth2 [3], but with double convolutions in the decoder. Please see Figure 1 for a visualization of the network architecture used.

Training We trained our network with several datasets: Our own stereo video footage, Megadepth [6], and Matterport [1]. The network was trained with a 512×256 resolution as input (similarly 256×512 for portrait data).

We scale the sigmoid prediction of the network to be in the range $(0.5, 100)$ meters.

Stereo data Our stereo data consists of several hours of stereo video captured in one European city and three US cities. The footage was captured with a landscape orientation of the cameras as well as a portrait orientation of the cameras. The cameras were calibrated so that the network predictions are metric. The cameras were re-calibrated at each capture session.

The network was trained with the Depth Hints loss [13] on stereo data in addition to a Monodepth2 reprojection-based loss and a sky segmentation prior (see below). However, our results in the paper for Robotcar dataset (only) used a network that was trained without the Depth Hints loss and instead used a Monodepth2 reprojection-based SSIM+L1 loss for the training loss for the stereo data.

Megadepth Megadepth [6] has depth estimates that are scale-ambiguous. So, we use a scale-invariant loss (Equation 2 in [6]) for the images with dense depth estimates in Megadepth. The images that have ordinal labels are also used with a robust ordinal depth loss (equation 4 in [6]). During training the images and depth maps were cropped to the target aspect ratio 512/256 or 256/512 (randomly chosen as landscape or portrait) and isotropically scaled to 512×256 to be fed as input to the network.

Matterport The Matterport dataset provides images with metric depth captured with Kinect-like cameras. We follow [4] for supervised training from Matterport data, using the loss function $\log(1 + |d - t|)$, where d is the network prediction and t is the target depth (Equation 3 in [4]) as well as a depth gra-

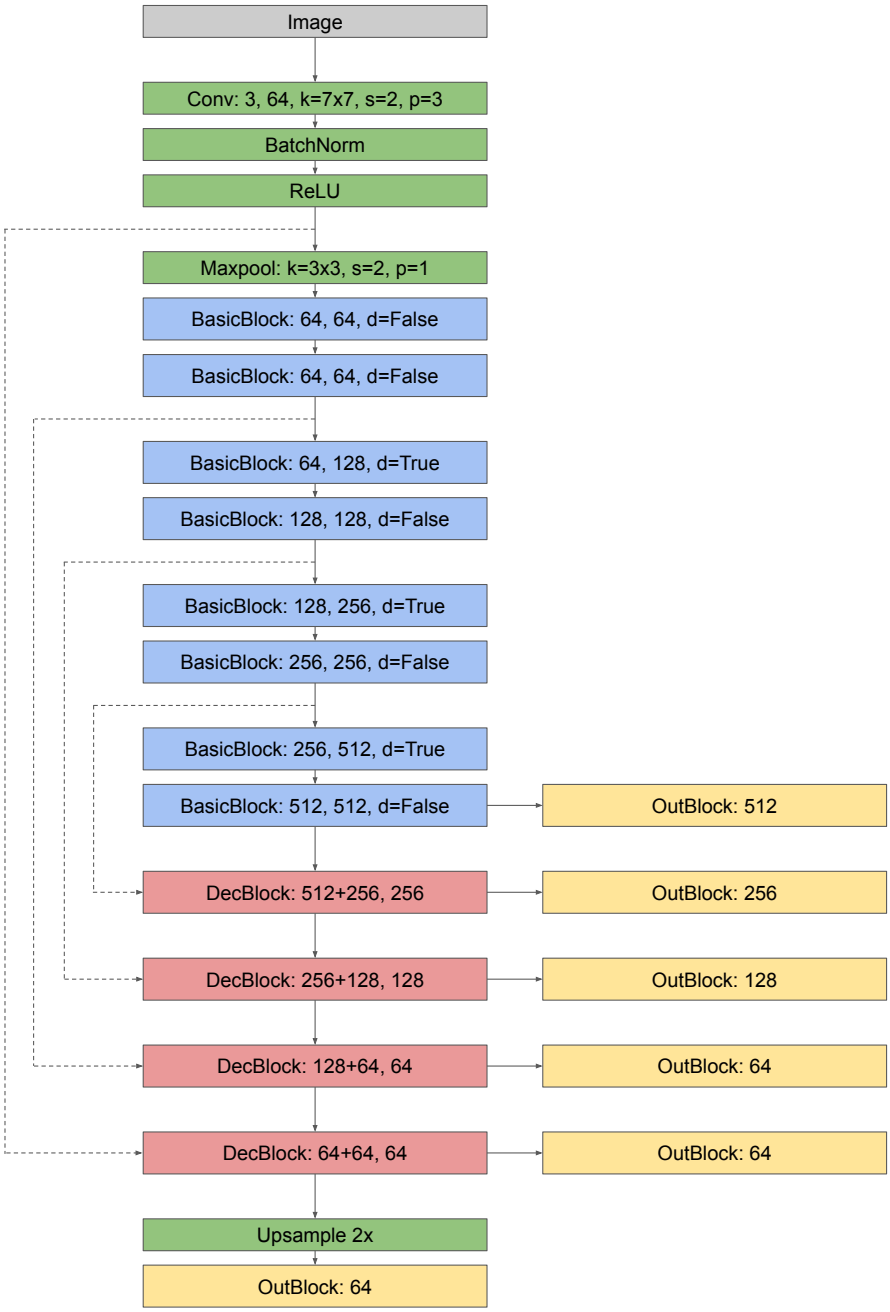

Fig. 1. Architecture of our network. Please see Figure 2 for details on building blocks.

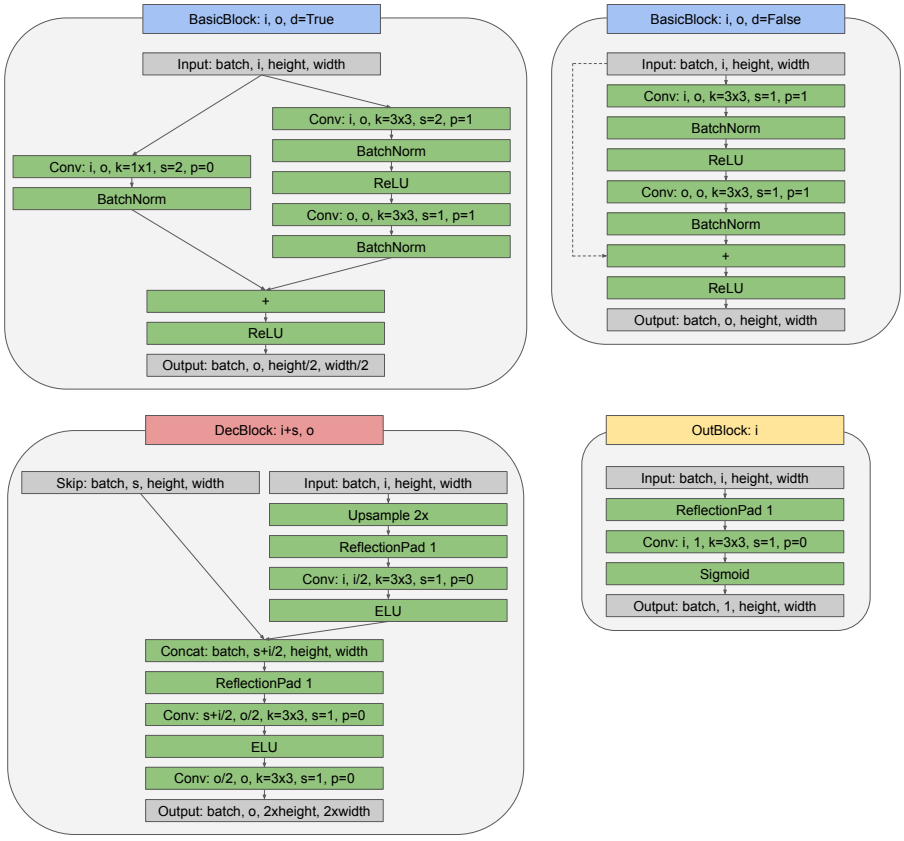

Fig. 2. Building blocks used in the architecture (Figure 1) of our network.

dient loss (Equation 4). Similarly to the Megadepth dataset, we crop and scale images during training.

Sky loss We also trained a segmentation network using the ADE20K dataset [14] that predicts if the pixels belongs to the sky or not. During training we use the predicted sky segmentation mask to have a small regualarization loss (weight 0.04) that forces masked pixels to have maximum depth (100 meters in our model) with L1 loss on depth values.

3 Robotcar with Superpoint

In this section we elaborate and motivate more on the choice of SIFT features for the RobotCar experiments. One of the main reasons for using SIFT is its invariance to in-plane rotations, a property not possessed by the SuperPoint or D2-Net features. This rotational invariance is crucial to the presented localization experiments, since unlike in an upright photo, there is no clear preferred direction in a top-down view of the road. We may thus expect the rectified query and database images to have any possible relative rotation.

SuperPoint features are trained by applying homographic warps to patches to obtain correspondences. These warps include rotations, but the publicly available model has been trained on only small rotations, leading to a reduced robustness to rotations. In this section we present an experiment that demonstrates this, illustrating that there are still some applications where SIFT continues to be an appropriate choice.

For pairwise matching, the same procedure is followed as in the main paper: features are extracted from the rectified patches, and features close to the warped image border are discarded. Pairwise matching is performed between the images using approximate nearest neighbour matching [9], and the obtained matches are then geometrically verified by fitting a homography to them using RANSAC [2] with a 10 pixel inlier threshold. Lastly, the number of inliers to the homography is saved for this query. Specifically, we go through each of the 729 query images in the first sequence of the RobotCar dataset used in the main paper. For each query image, we retrieve the top-ranked database image from the experiments in Sec. 5.2 of the main paper, and we check whether SuperPoint is able to establish matches between these images. Since the image retrieval failed for a few images, we do not expect all of these query-database image pairs to match. However, since the success rate was larger than 98% for this dataset, performing pairwise feature matching between the query image and the top retrieved database image should indicate whether or not SuperPoint features are suitable for this task at all.

Fig. 3 shows the number of inliers to the estimated homography from both the SIFT matching, as well as the SuperPoint matching. For each value on the x-axis, the corresponding y-value shows the number of query images (out of the 729) whose final estimated homography had that number of inliers or more. A "higher" curve is thus better.

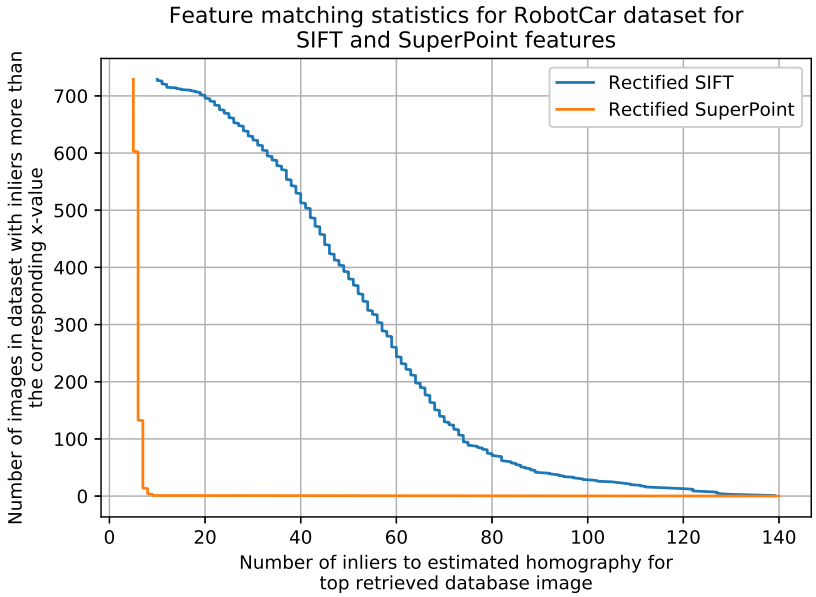

Fig. 3. Number of inliers to the homography estimated during pairwise matching between the query images and the top-retrieved database images for the first sequence of the RobotCar dataset. The y-values denote the number of images whose homography has at least the number of inliers specified on the x-axis.

As expected, due to the rotational variance of SuperPoint, it fails to reliably match essentially all image pairs. No query image had more than nine inliers to the estimated homography.

4 Results with and without enforcing orthogonal normals during clustering

Experiments were performed on scene 6 of our dataset when not enforcing orthogonality between the normal clusters. Instead, planes were found by histogramming the normals into 200 bins on the unit sphere. Thresholding and non-maximum suppression were then performed to obtain a set of plane hypotheses. Otherwise the pipeline was the same as in the main experiments. Results using this method is shown in Fig. 4.

As can be seen in the figure, enforcing the orthogonality improves the performance. The decreased performance without rectification is most likely due to inaccuracies in the monocular depth estimation network. Enforcing orthogonality is thus a way to reduce the noise in the depth predictions.

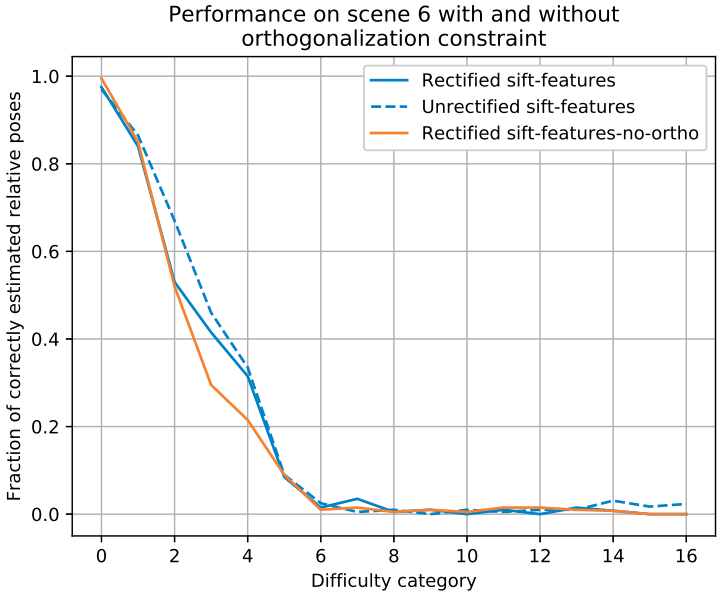

Fig. 4. Performance on scene 6 with and without enforcing the normal clusters to be orthogonal.

5 Performance using different monocular depth estimation networks

Fig. 5 shows the results when replacing the depth prediction network used in the main paper (described in Sec. 2) with the MegaDepth network [6] and MiDaS [5].

For both MiDaS and MegaDepth, we used the official implementations available on the project webpages. In MiDaS, the images are rescaled such that their largest axis equals 384, and the smaller axis is chosen as the multiple of 32 that best preserves the aspect ratio of the original image. For MegaDepth, we similarly rescale the images to have a maximum dimension of 512, with the other dimension chosen as the multiple of 32 that best preserves the original aspect ratio.

The reason MonoDepth performs better on this scene seems to be that MiDaS and MegaDepth sometimes have difficulty separating a building facade and a cloudy gray sky, whereas the MonoDepth network does not seem to have trouble distinguishing between these. This leads to noisier estimates of the surface normal of the plane. This may perhaps be attributed to the different training data the three networks have been trained on.

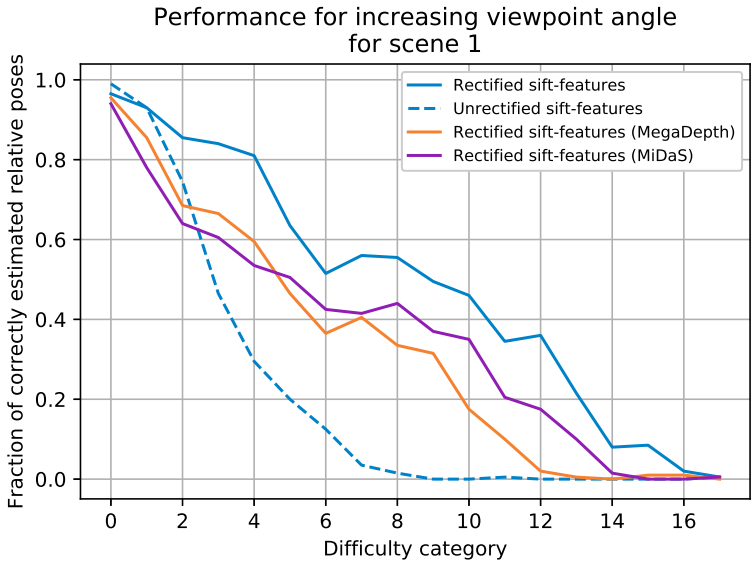

Fig. 5. Performance on scene 1 using two other monocular depth prediction networks. Only the depth prediction network has changed, the rest of the pipeline remains unchanged.

6 Detailed results on all scenes of our dataset

Fig. 6 presents individual results for each of the eight scenes in the dataset. Figs. 7 and 8 show example images from each of the eight scenes in our dataset. Note that our dataset contains scenes of varying difficulty for our approach, ranging from scenes dominated by a planar surface (scenes 1, 3, 4, 5), roughly planar scenes (scene 2), over scenes with multiple planar surfaces (scenes 6, 7), to scenes with little dominant planes (scene 8).

We note that the proposed method of extracting features from rectified patches achieves the best performance for the datasets where a large portion of the image is taken up by one dominant plane, and the performance seems to drop as the viewed planes become smaller. This is likely due to the estimated normals getting more noisy, leading to less accurate rectifications. Since all normals assigned to a given plane are used to estimate the plane normal, fewer pixels per plane lead to fewer measurements of the plane normal, and thus a more noisy estimate.

As a result, our method performs the best on scenes 1 to 5, where an estimate of the plane normal can be extracted fairly reliably, whereas for example in scene 8, where there are very few planar surfaces to rectify, the method more or less reduces to feature matching using regular features.

The performance on scene 4 is especially good. This is most likely due to the depth predictions being very accurate: some of the data used to train the

depth network was captured from the surrounding areas (though none of the images in the scene have been seen during training), which may result in more accurate depths for this scene. The results may thus be indicative of what might be achieved as monocular depth estimation networks get better.

7 Example normal clusterings

Figure 9 shows some examples of the normal clusters obtained on images from 3 of the scenes in our dataset.

8 Heavily distorted vanishing-point rectified images

Fig. 10 shows heavily distorted images that have been rectified using a vanishing point based rectification method. Since the vanishing point based method does not provide information about which pixels belong to the plane, the entire image is rectified, which can cause strong distortions, and since the entire image is warped, the area of interest may only occupy a small portion of the rectified image.

9 Experiments on EVD

We also ran experiments on three of scenes from the challenging the extreme view dataset (EVD) [8]. The scenes tested were Café, Dum, Grand. Our method was able to successfully match the Café scene, but was unable to estimate the homography between the image pairs of the two other scenes. This is likely mainly due to two reasons. First, our method needs the camera intrinsics in order to compute the surface normals from the depthmap, and the dataset does not provide camera calibration information. Secondly, the other scenes contain some non-planar parts, which may cause the estimated plane normals to not be completely accurate. We note that regular feature matching on the original image pairs fails for all three pairs.

Fig. 11 shows the results on the Café scene.

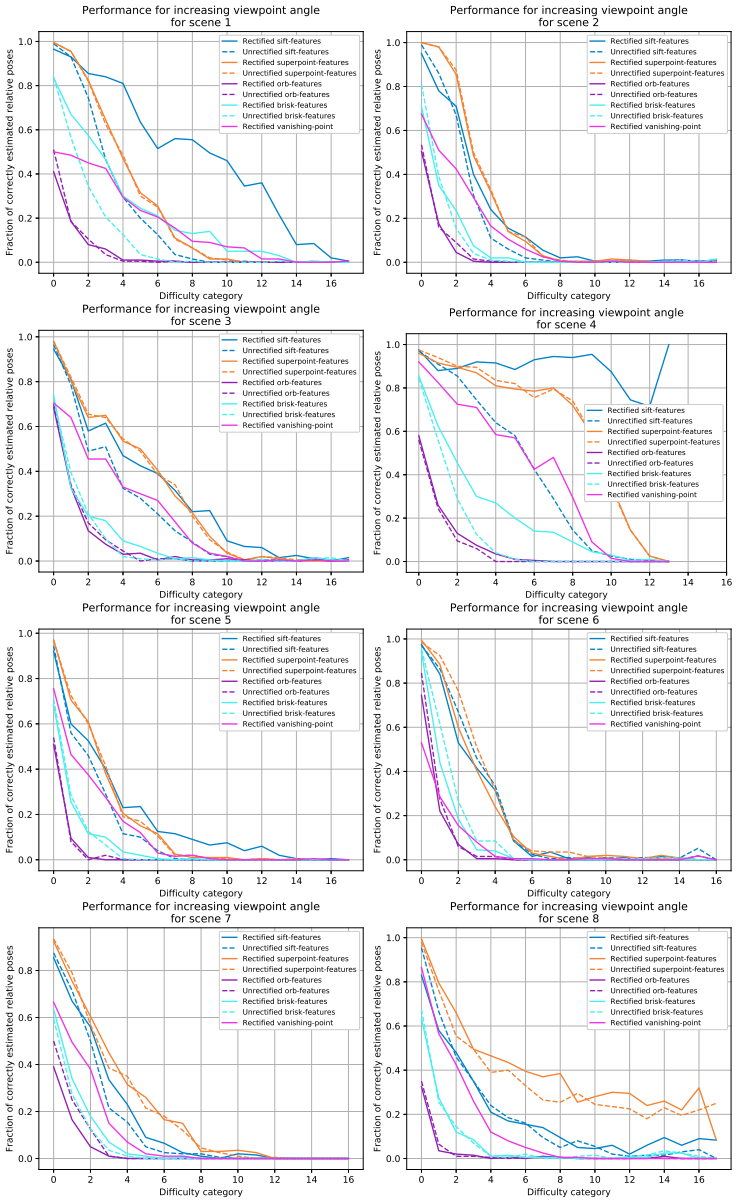

Fig. 6. Detailed results on the presented local feature matching dataset, showing the performance on each scene individually.

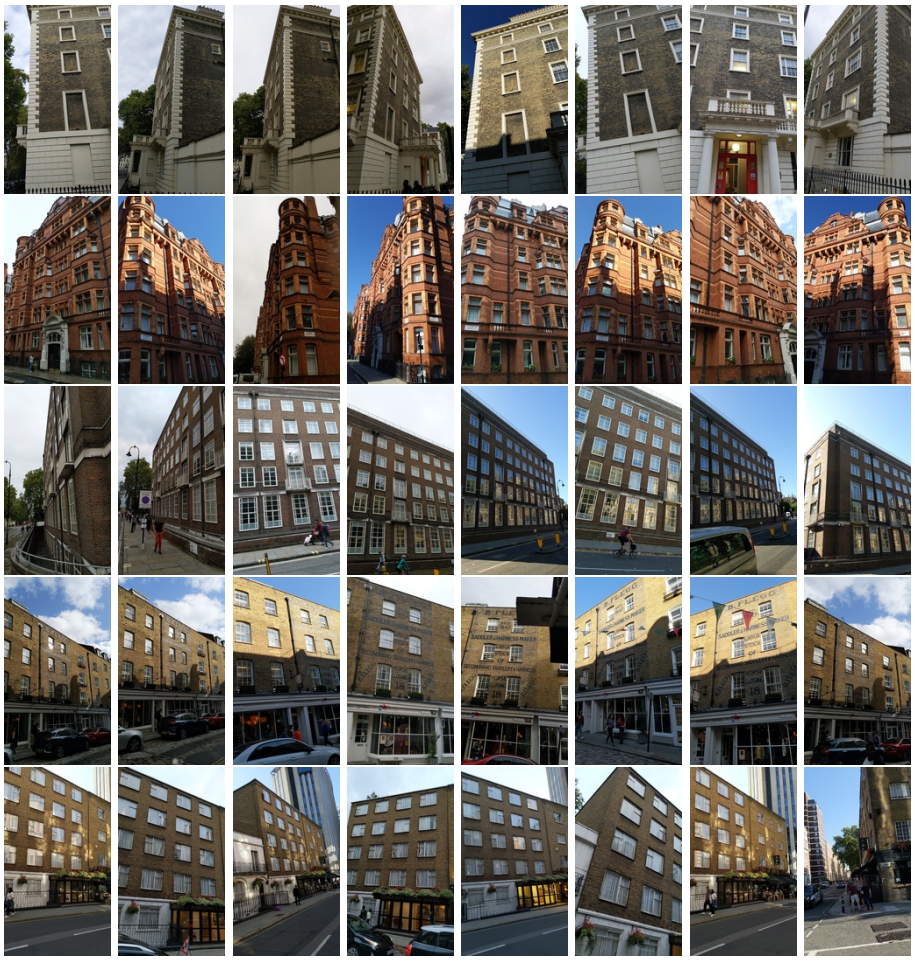

Fig. 7. Example images from our dataset for Strong Viewpoint Changes. Each row shows a sample of images showing scenes 1 to 5.

Fig. 8. Example images from our dataset for Strong Viewpoint Changes. Top two rows show a sample of images for scene 6. The following rows show images of scene 7 and 8, respectively.

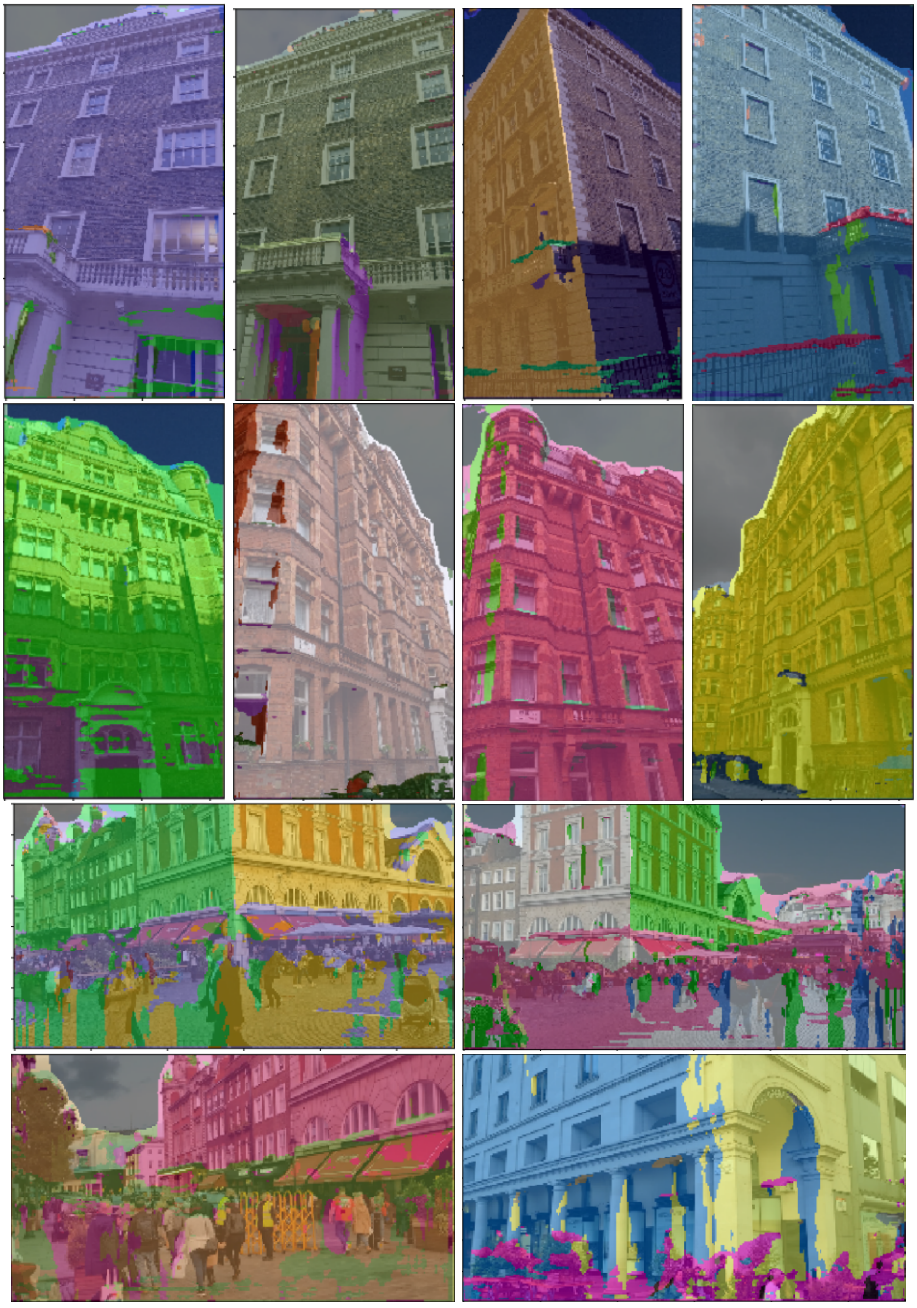

Fig. 9. Normal clustering results on four images from three of the scenes in our dataset.

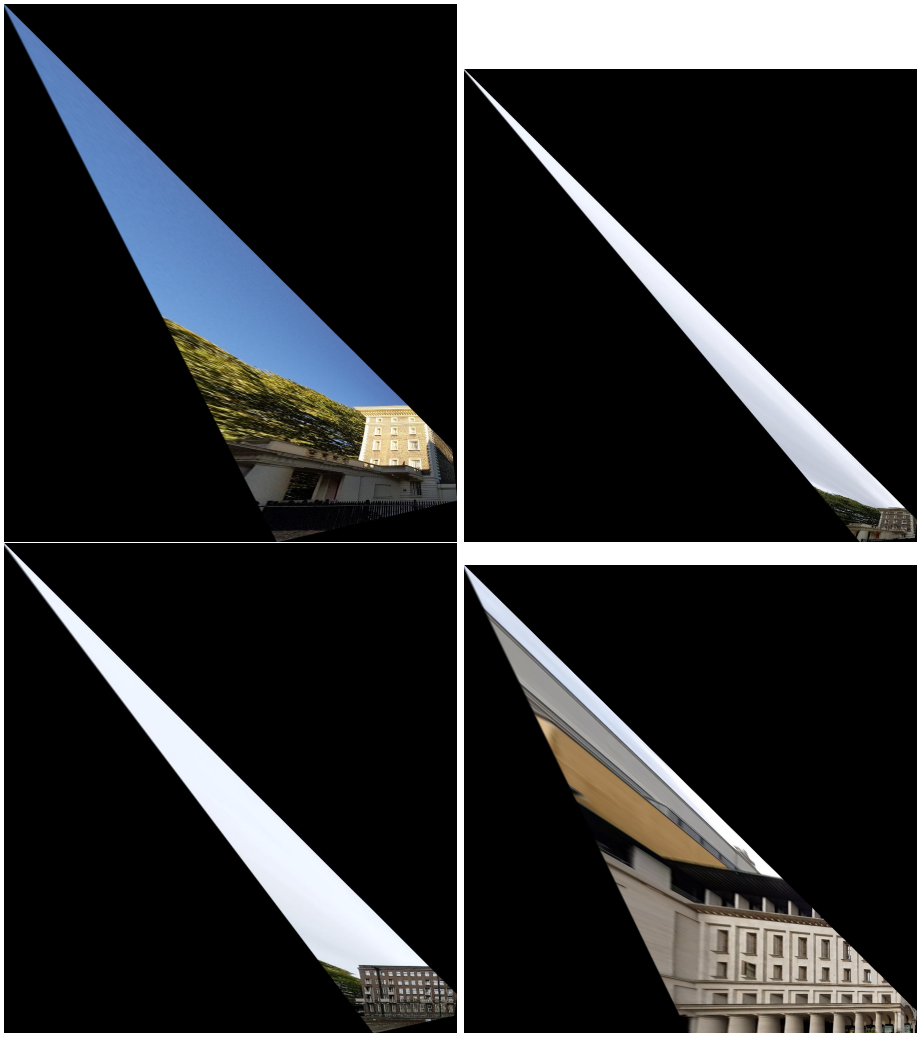

Fig. 10. Examples of heavily distorted images that have been rectified using a vanishing point based rectification method.

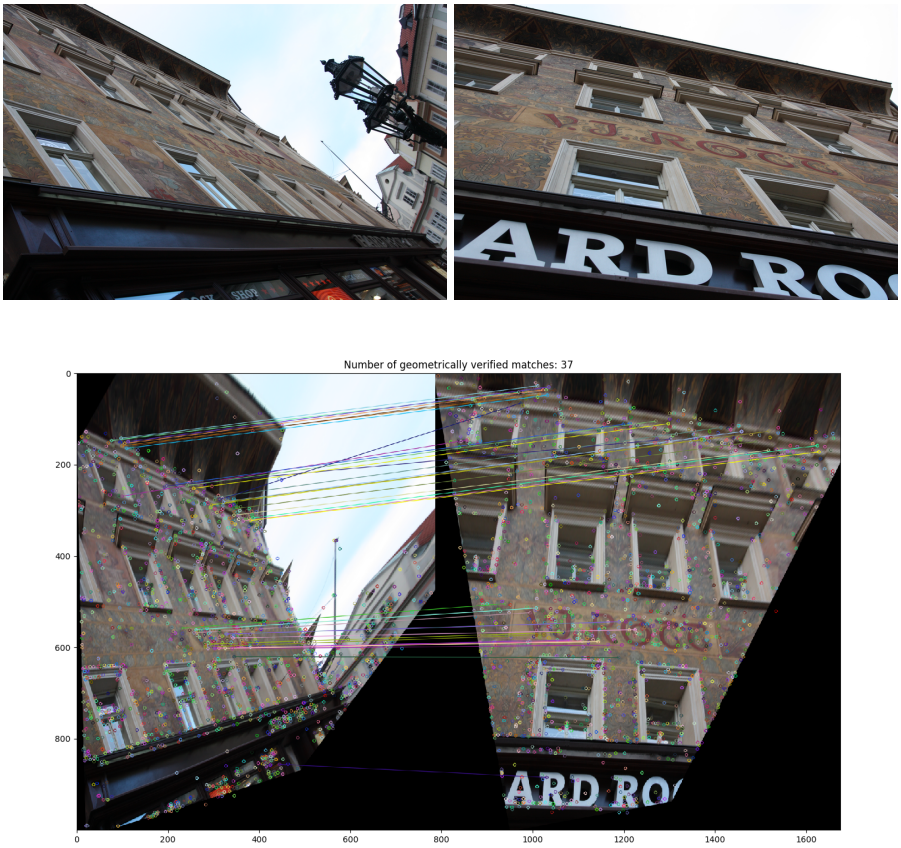

Fig. 11. Results on the Café scene in the EVD dataset. Top row: Original images. Bottom row: Geometrically consistent matches between the rectified patches.

References

1. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from RGB-D data in indoor environments. International Conference on 3D Vision (3DV) (2017) 2
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981) 5
3. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth prediction. The International Conference on Computer Vision (ICCV) (October 2019) 2
4. Hu, J., Ozay, M., Zhang, Y., Okatani, T.: Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In: IEEE Winter Conf. on Applications of Computer Vision (WACV) (2019) 2
5. Lasinger, K., Ranftl, R., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019) 7
6. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Computer Vision and Pattern Recognition (CVPR) (2018) 2, 7
7. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004) 1
8. Mishkin, D., Perdoch, M., Matas, J.: Two-view matching with view synthesis revisited. In: 2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013). pp. 436–441. IEEE (2013) 9
9. Muja, M., Lowe, D.: Flann-fast library for approximate nearest neighbors user manual. Computer Science Department, University of British Columbia, Vancouver, BC, Canada (2009) 5
10. Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., Kahl, F., Pajdla, T.: Benchmarking 6dof outdoor visual localization in changing conditions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1
11. Sattler, T., Weyand, T., Leibe, B., Kobbelt, L.: Image Retrieval for Image-Based Localization Revisited (2012) 1
12. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1
13. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: IEEE International Conference on Computer Vision (ICCV) (2019) 2
14. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017) 5