Single-Image Depth Prediction Makes Feature Matching Easier

Carl Toft[1*], Daniyar Turmukhambetov[2], Torsten Sattler[1],
Fredrik Kahl[1], and Gabriel J. Brostow[2,3]

[1]Chalmers University of Technology [2]Niantic [3]University College London
www.github.com/nianticlabs/rectified-features

Abstract. Good local features improve the robustness of many 3D relocalization and multi-view reconstruction pipelines. The problem is that viewing angle and distance severely impact the recognizability of a local feature. Attempts to improve appearance invariance by choosing better local feature points or by leveraging outside information, have come with pre-requisites that made some of them impractical. In this paper, we propose a surprisingly effective enhancement to local feature extraction, which improves matching. We show that CNN-based depths inferred from single RGB images are quite helpful, despite their flaws. They allow us to pre-warp images and rectify perspective distortions, to significantly enhance SIFT and BRISK features, enabling more good matches, even when cameras are looking at the same scene but in opposite directions.

Keywords: Local Feature Matching, Image Matching

1 Introduction

Matching local features between images is a core research problem in Computer Vision. Feature matching is a crucial step in Simultaneous Localization and Mapping (SLAM) [18, 55], Structure-from-Motion (SfM) [66, 70], and visual localization [63, 64, 71]. By extension, good feature matching enables applications such as self-driving cars [30] and other autonomous robots [42] as well as Augmented, Mixed, and Virtual Reality. Handling larger viewpoint changes is often important in practice, *e.g.*, to detect loop closures when revisiting the same place in SLAM [25] or for re-localization under strong viewpoint changes [65].

Traditionally, local features are computed in two stages [47]: the feature detection stage determines salient points in an image around which patches are extracted. The feature description stage computes descriptors from these patches. Before extracting a patch, the feature detector typically accounts for certain geometric transformations, thus making the local features robust or even invariant against these transformations. For example, aligning a patch to a dominant direction makes the feature invariant to in-plane rotations [47,53]; detecting salient points at multiple scales introduces robustness to scale changes [43]; removing

[*] Work done during an internship at Niantic.

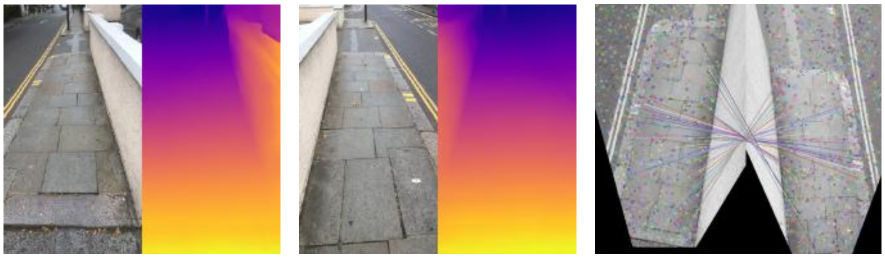
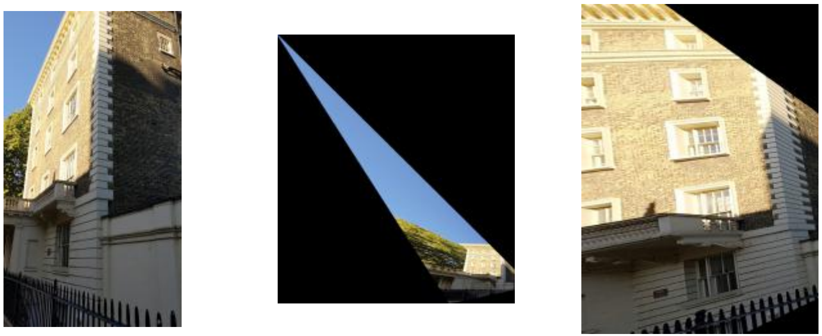


Fig. 1. Left to Right: Two input images, capturing the same section of a sidewalk, but looking in opposite directions. Each RGB input image is shown together with its depth map predicted by a single-image depth prediction network. Based on the predicted depth map, we identify planar regions and remove perspective distortion from them before extracting local features, thus enabling effective feature matching under strong viewpoint changes.

the effect of affine transformations [49, 50] of the image makes the extracted features more robust against viewpoint changes [1, 24]. If the 3D geometry of the scene is known, *e.g.*, from 3D reconstruction via SfM, it is possible to undo the effect of perspective projection before feature extraction [35, 76, 83, 84]. The resulting features are, in theory, invariant to viewpoint changes.

Even without known 3D scene geometry, it is still possible to remove the effect of perspective distortion from a single image [4, 58, 61]. In principle, vanishing points [10, 39, 69, 87] or repeating structural elements [58, 59, 77] can be used to rectify planar regions prior to feature detection [16]. However, this process is cumbersome in practice: it is unclear which pixels belong to a plane, so it is necessary to unwarp the full image. This introduces strong distortions for image regions belonging to different planes. As a result, determining a good resolution for the unwarped image is a challenge, since one would like to avoid both too small (resulting in a loss of details) and too high resolutions (which quickly become hard to handle). Fig. 2 shows an example of this behaviour on an image from our dataset. This process has to be repeated multiple times to handle multiple planes.

Prior work has shown the advantages of removing perspective distortion prior to feature detection in tasks such as visual localization [61] and image retrieval [4, 5, 12]. Yet, such methods are not typically used in practice as they are hard to automate. For example, modern SfM [66, 75], SLAM [55], and visual localization [27, 63, 64] systems still rely on classical features without any prior removal of perspective effects. This paper shows that convolutional neural networks (CNNs) for single-image depth estimation provide a simple yet effective solution to the practical problems encountered when correcting perspective distortion: although their depth estimates might be noisy (especially for scene geometry far away from the camera), they are typically trained to produce smooth depth gradients [28,40]. This fact can be used to estimate normals, which in turn define planes that can be rectified. Per-pixel normals provide information about



Input image
(1920 x 1080)

Rectified with VP
(1349 x 1188)

Rectified with Ours
(1500 x 1076)

Fig. 2. Perspective rectification of a challenging example. From left to right: input image, perspective rectification with a vanishing point method [11] and Ours. The vanishing point method can heavily distort an image and produce an image where the region of interest only occupies a small part. An output image may become prohibitively large in order to preserve detail. Our method does not have these artifacts as it rectifies planar patches, not the full image.

which pixels belong to the same plane, thus avoiding the problems of having to unwarp the full image and to repeatedly search for additional planes. Also, depth information can be used to avoid strong distortions by ignoring pixels seen under sharp angles. As a result, our approach is significantly easier to use in practice.

Specifically, this paper makes the following contributions: **1)** We propose a simple and effective method for removing perspective distortion prior to feature extraction based on single-image depth estimation. **2)** We demonstrate the benefits of this approach through detailed experiments on feature matching and visual localization under strong viewpoint changes. In particular, we show that improved performance does not require fine-tuning the depth prediction network per scene. **3)** We propose a new dataset to evaluate the performance of feature matching under varying viewpoints and viewing conditions. We show that our proposed approach can significantly improve matching performance in the presence of dominant planes, without significant degradation if there are no dominant planes.

2 Related Work

Perspective undistortion via 3D geometry. If the 3D geometry of the scene is known, *e.g.*, from depth maps recorded by RGB-D sensors, laser scans, or multi-view stereo, it is possible to remove perspective distortion prior to feature extraction [9, 35, 76, 83, 84]. More precisely, each plane detected in 3D defines homographies that warp perspective image observations of the plane to orthographic projections. Extracting features from orthographic rather than perspective views makes the features (theoretically) invariant to viewpoint changes[1]. This enables feature matching under strong viewpoint changes [84]. Given known 3D geometry, a single feature match between two images or an image and a 3D model can be sufficient to estimate the full 6-degree-of-freedom (relative) camera pose [4, 76]. Following similar ideas, features found on developable surfaces can be made more robust by unrolling the surfaces into a plane prior to feature detection. Known 3D geometry can also be used to remove the need for certain types of invariances [35], *e.g.*, predicting the scale of keypoints from depth [33] removes the need for scale invariance.

Previous work assumed that 3D data is provided together with an image, or is extracted from multiple images. Inspired by this idea, we show that perspective distortion can often be removed effectively using single-image depth predictions made by modern convolutional neural networks (CNNs).

Perspective undistortion without 3D geometry. Known 3D geometry is not strictly necessary to remove the effect of perspective foreshortening on planar structures: either vanishing points [10, 17, 39, 69, 87] or repeating geometrical structures [57–60, 77] can be used to define a homography for removing perspective distortion for all pixels on the plane [16, 41]. In both cases, an orthographic view of the plane can be recovered up to an unknown scale factor and an unknown in-plane rotation, *i.e.*, an unknown similarity transformation. Such approaches have been used to show improved performance for tasks such as image retrieval [4, 5, 12], visual localization [61], and feature matching [81]. However, they are often brittle and hard to automate for practical use: they do not provide any information about which pixels belong to a given plane. This makes it necessary to warp the full image, which can introduce strong distortion effects for regions that do not belong to the plane. This in turn leads to the problem of selecting a suitable resolution for the unwarped image, to avoid loosing details without creating oversized images that cannot be processed efficiently. As a result, despite their expected benefits, such methods have seen little use in practical applications, *e.g.*, modern SfM or SLAM systems. See Fig. 2 (as well as Sec. 8 in the supplementary material) for an example of this behaviour as seen on an image from our dataset.

In this paper, we show that these problems can easily be avoided by using single-image depth predictions to remove the effect of perspective distortion.

Pairwise image matching via view synthesis. An alternative to perspective undistortion for robust feature matching between two images taken from different viewpoints is view synthesis [46, 52, 54, 56]. Such approaches generate multiple affine or projective warps of each of the two images and extract and match features for each warp. Progressive schemes exist, which first evaluate small warps and efficient features to accelerate the process [52]. Still, such approaches are computationally very expensive due to the need to evaluate a large number of potential warps. Methods like ours, based on removing perspective distortion,

[1] In practice, strong viewpoint changes create strong distortions in the unwarped images, which prevent successful feature matching [35].

avoid this computational cost by determining a single warp per region. Such warps can also be estimated locally per region or patch [3,32]. This latter type of approach presupposes that stable keypoints can be detected in perspectively distorted images. Yet, removing perspective effects prior to feature detection can significantly improve performance [9,35].

Datasets. Measuring the performance of local features under strong viewpoint changes, *i.e.*, the scenario where removing perspective distortion could provide the greatest benefit, has a long tradition, so multiple datasets exist for this task [2, 3, 6, 15, 50, 52, 68]. Often, such datasets depict nicely textured scenes, *e.g.*, graffiti, paintings, or photographs, from different viewpoints. Such scenes represent "failure cases" for single-image depth predictions as the networks (not unreasonably) predict the depth of the elements shown in the graffiti *etc.* (*cf.* Fig. 4). This paper thus also contributes a new dataset for measuring performance under strong viewpoint changes that depicts regular street scenes. In contrast to previous datasets, *e.g.*, [6], ours contains both viewpoint and appearance changes occuring at the same time.

Single-image depth prediction. Monocular depth estimation aims at training a neural network to predict a depth map from a single RGB image. Supervised methods directly regress ground-truth depth, acquired with active sensors (LiDAR or Kinect) [22, 44, 45]; from SfM reconstructions [40]; or manual ordinal annotations [13, 40]. However, collecting training data is difficult, costly, and time-consuming. Self-supervised training minimizes a photometric reprojection error between views. These views are either frames of videos [29, 34], and/or stereo pairs [26, 28, 29, 74]. Video-only training also needs to estimate the pose between frames (up to scale) and model moving objects [86]. Training with stereo provides metric-accurate depth predictions if the same camera is used at test time. Supervised and self-supervised losses can be combined during training [36, 79].

CNNs can be trained to predict normals [20,21,31,73], both depth and normals [38,80,82,85], or 3D plane equations [44,45]. However, normals are either used to regularize depth, or trained exclusively on indoor scenes because of availability of supervised data, which is difficult to collect for outdoor scenes [14].

The approach presented in this paper is not tied to any specific single view depth prediction approach, and simply assumes that approximate depth information is available. Normal estimation networks could also be used in our pipeline, however depth estimation networks are more readily available.

3 Perspective Unwarping

We introduce a method for performing perspective correction of monocular images. The aim is to perform this prior to feature extraction, leading to detection and description of features that are more stable under viewpoint changes. That stability, can for example, establish more numerous correct correspondences between images taken from significantly different viewpoints.

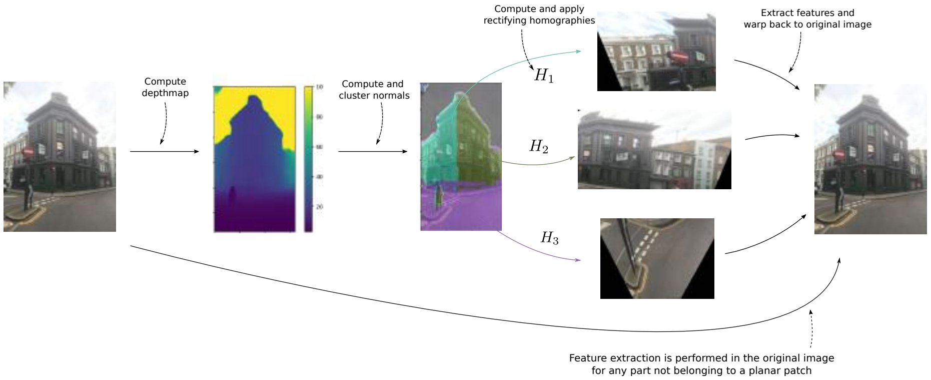

Compute depthmap
Compute and cluster normals
Compute and apply rectifying homographies
H_1
H_2
H_3
Extract features and warp back to original image
Feature extraction is performed in the original image for any part not belonging to a planar patch

Fig. 3. Proposed pipeline for extracting perspectively corrected features from a single image: Depths are computed using a single-image depth estimation network. The intrinsic parameters of the camera are then used to backproject each pixel into 3D space, and a surface normal is estimated for each pixel. The normals are clustered into three orthogonal directions, and a homography is computed for each cluster to correct for the perspective distortion. Local image features are then extracted from each rectified patch using an off-the-shelf feature extractor and their positions are warped back into the original image. Regular local features are extracted from the parts of the image that do not belong to a planar patch.

The method is inspired by, and bears close resemblance to, the view-invariant patch descriptor by Wu *et al*. [76]. The main difference is that while Wu's method was designed for alignment of 3D point clouds, our method can be applied to single, monocular images, allowing it to be utilized in applications such as wide-baseline feature matching, single image visual localization, or structure from motion.

A schematic overview of the method is shown in Fig. 3. The central idea is that given a single input image, a network trained for single image depth estimation is used to compute a corresponding dense depth map of the image. Using the camera intrinsics, the depth map gets backprojected into a point cloud, given in the camera's reference frame. From this point cloud, a surface normal vector is estimated for each point. For any given point in the point cloud (and correspondingly, in the image), a rectifying homography H can now be computed. H transforms a patch centered around the point in the image to a corresponding patch, which simulates a virtual camera looking straight down on the patch, *i.e.* a camera whose optical axis coincides with the patch's surface normal. As shown by several experiments in Sec. 5, by performing both feature detection and description in this rectified space, the obtained interest points can sometimes be considerably more robust to viewpoint differences.

In principle this method could be applied to each point independently, but typically a large number of points will share the same normal. Consider for example points lying on a plane, such as points on the ground, points on the same wall, or points on different but parallel planes, such as opposing walls.

These points will all be rectified by the same homography. This is utilized in the proposed method by identifying planar regions in the input image, which is done by clustering all normals on the unit sphere. This yields a partitioning of the input image into several connected components, which are then rectified individually. Each input image is thus transformed into a set of rectified patches, consisting of perpendicular views of all dominant planes in the image. Image features can then be extracted, using any off-the-shelf detector and descriptor. In the experiments, results are provided for SIFT [47], SuperPoint [19], ORB [62], and BRISK [37] features.

Note that the rectification process is not dependent on all planes being observed. If only one plane is visible, that may still be rectified on its own. Parts of the image that are not detected to be on a plane are not rectified, but we still extract features from these parts and use them for feature matching. This way, we do not ignore good features just because they are not on a planar surface. For complex, non-planar geometries, large parts of an image may not have planar surfaces. For such images our approach gracefully resorts to standard (non-rectified) feature matching for regions not belonging to the identified planes.

Below, we describe each of the above steps in more detail.

3.1 Depth Estimation

The first step in the perspective correction process is the computation of the depth map. In this paper we use MonoDepth2 [29] which was trained with a Depth Hints loss [74] on several hours of stereo video captured in one European city and three US cities. In addition to stereo, the network was also trained on the MegaDepth dataset [40] and Matterport [8] datasets (see supplementary materials for details). This network takes as input a single image resized to 512×256, and outputs a dense depth map. Under ideal conditions, each pixel in the depth map tells us the calibrated depth in meters. In practice, any method that provides dense depth estimates may be used, and the depths need not be calibrated, $i.e.$ depths estimated up to an unknown scale factor may also be used, since the depth map is only used to compute surface normals for each point.

3.2 Normal Computation and Clustering

With the depth map computed, the next step is normal computation. A surface normal is estimated for each pixel in the depth map by considering a 5×5 window centered on the pixel, and fitting a plane to the 25 corresponding back-projected points. The unit normal vector of the plane is taken as an estimate of the surface normal for that pixel.

With the normals computed, they are then clustered to identify regions in the image corresponding to planar surfaces. Since all points on the same plane share the same normals, these normals (and the pixels assigned to them) may be found by performing k-means clustering.

Since the depth map, and by extension the surface normals, are subject to noise, we found that clustering the normals into three clusters or dominant directions, while also enforcing orthogonality between these clusters, gave good results. Each cluster also includes its antipodal point (this means, for example, that two opposing walls would be assigned to the same dominant direction). This assumption seems to correspond to the 3D structure of many scenes: if at least one dominant planes is visible, such as the ground or a building wall, this method will produce satisfactory results. If two are visible, the estimated normals, and thus also the estimated homographies, tend to be more accurate. If no planes are visible, the method gracefully reduces to regular feature extraction. Note also that several different patches in the image can be assigned to the same cluster, but rectified separately as different planes: examples include opposing walls or parallel flat surfaces.

In non-Manhattan world geometries, where several non-perpendicular planes are visible, the estimated normals may not be completely accurate. Thus, our method would apply a homography that would render the planar surface not from a fronto-parallel view, but at a tilt. In most cases, this rectification still removes some effects of perspective distortion.

3.3 Patch Rectification

With the normals clustered into three dominant clusters, each pixel is assigned its normal's respective cluster. Each of these subsets may be further subdivided into their respective connected components. The input image is thus partitioned into a set of patches, each consisting of a connected region of pixels in the image, together with a corresponding estimate of the surface normal for that patch. In Fig. 3, the patches are shown overlaid on the image in different colors.

A rectified view of each patch is now computed, using the estimated patch normal. The patch is warped using a homography, computed as the homography which maps the patch to the patch as it would have been seen in a virtual camera sharing the same camera center as the original camera, but rotated such that its optical axis is parallel to the surface normal (*i.e.* it is facing the patch straight on). The smallest rotation that brings the camera into this position is used.

Lastly, not the entire patch is rectified, since if the plane corresponding to a given patch is seen at a glancing angle in the camera, most of the rectified patch would be occupied by heavily distorted, or stretched, regions. As such, a threshold of 80° is imposed on the maximum angle allowed between the viewing ray from the camera, and the surface normal, and the resulting patch is cropped to only fully contain the region of the patch seen at not too glancing an angle.

3.4 Warping Back

When matching features, the image may now be replaced with its set of rectified patches and patches from non-planar parts of the image. Alternatively, feature extraction may be performed in the non-planar parts of the original image, and in all rectified patches, and the 2D locations of the features in the rectified

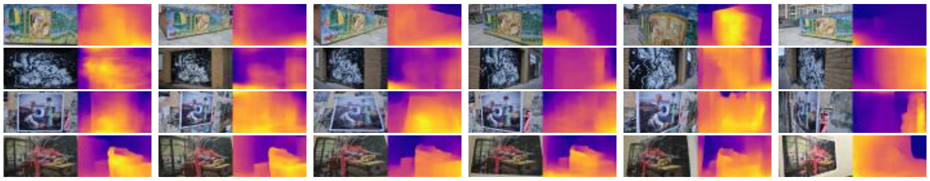


Fig. 4. "Failure" cases of applying a modern single-image depth prediction network [29, 74] on images from the HPatches dataset [6] (visualized as inverse depth): the network predicts the depth of the scenes depicted in the graffiti and images rather than understanding that these are drawings / photos attached to a planar surface.

patches may then be warped back into the original image coordinate system, but with the descriptors unchanged. A perspectively corrected representation of the image has then been computed. The final description thus includes perspectively corrected features for all parts of the image that were deemed as belonging to a plane, and regular features extracted from the original image, from the parts that were deemed non-planar.

4 Dataset for Strong Viewpoint Changes

A modern, and already well-established, benchmark for evaluating local features is the HPatches dataset [6]. HPatches consists of 116 sequences of 6 images each, where sequences have either illumination or viewpoint changes. Similar to other datasets such as [51], planar scenes that can be modeled as homographies are used for viewpoint changes. Most of the sequences depict paintings or drawings on flat surfaces. Such scenes are ideal for local features as they provide abundant texture. Interestingly, such scenes cause single-image depth prediction to "fail": as shown in Fig. 4, networks predict the depth of the structures shown in the paintings and drawings rather than modeling the fact that the scene is planar. We would argue that this is a rather sensible behavior as the scene's planarity can only be inferred from context, *e.g.*, by observing a larger part of a scene rather than just individual drawings. Still, this behavior implies that standard datasets are not suitable for evaluating the performance of any type of method based on singe-image depth prediction. This motivated us to capture our own dataset that, in contrast to benchmarks such as HPatches, intentionally contains non-planar structures.

In this paper, we thus present a new dataset for evaluating the robustness of local features when matching across large viewpoint variations, and changes in lighting, weather, *etc.* The dataset consists of 8 separate scenes, where each scene consists of images of one facade or building captured from a wide range of different viewpoints, and in different weather and environmental conditions. Each scene has been revisited up to 5 times, see Tab. 1.

All images included in the dataset originated from continuous video sequences captured using a consumer smartphone camera. These video sequences were then

Table 1. Statistics for the scenes in our dataset

scene #	1	2	3	4	5	6	7	8
# img. pairs	3590	3600	3600	2612	3428	3031	2893	3312
# sequences	4	4	3	4	3	5	3	3

reconstructed using the Colmap software [66, 67], and the poses for a subset of the images were extracted from this reconstruction. Colmap also provides an estimate of the intrinsic parameters for each individual image frame, which are included in the dataset. These are necessary since the focal length of the camera may differ between the images due to the camera's autofocus. Fig. 5 shows a set of example images from two of our 8 scenes.

Since our scenes are not perfectly planar, measuring feature matching performance by the percentage of matches that are inliers to a homography, as done in [6,51], is not an option for our dataset. Inspired by CVPR 2019 workshops "Image Matching: Local Features and Beyond" and "Long-Term Visual Localization under Changing Conditions", we evaluate feature matching on downstream tasks as opposed to measuring the number of recovered feature matches or repeatability, *etc*. Hence, we evaluate the performance of local features through the task of accurately estimating the relative pose between pairs of images. This allows us to judge if improvements in feature matching lead to meaningful improvements in practical applications such as localization and structure from motion.

For each scene, a list of image pairs is thus provided. Each image pair has been assigned to one of eighteen different difficulty categories, depending on the distance between the centres of the cameras that captured the images, and the magnitude of their relative rotation. The difficulty categories span the range of almost no difference in rotation up to almost 180° relative rotation. So, the image pairs in the k-th difficulty category have a relative rotation in the range of $[10k, 10(k+1)]$ degrees in one of the axes.

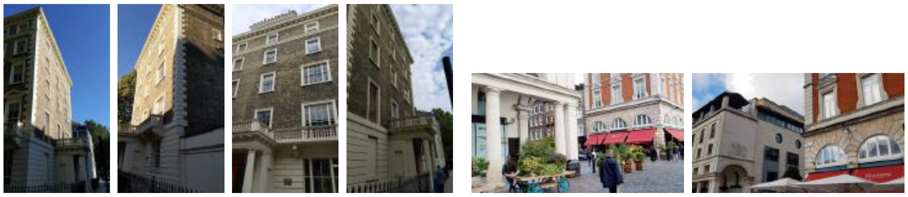


Fig. 5. Six example images, showing two different scenes of the presented benchmark dataset. The dataset contains several scenes, each consisting of over 1000 images of an urban environment captured during different weather conditions, and covering a large range of viewing angles. This dataset permits the evaluation of the degradation of local feature matching methods for increasing viewpoint angle differences. Please see the supplementary material for example images from each of the eight scenes.

The dataset is publicly available on the project webpage www.github.com/nianticlabs/rectified-features.

5 Experiments

This section provides two experiments: Sec. 5.1 shows that perspective unwarping based on the proposed approach can significantly improve feature matching performance on our proposed dataset. Sec. 5.2 shows that our approach can be used to re-localize a car under 180° viewpoint changes, *e.g.*, in the context of loop closure handling for SLAM. We use the SIFT [47] implementation provided by OpenCV [7] for all of our experiments.

5.1 Matching Across Large Viewpoint Changes

First, we evaluate our method on the 8 scenes of our proposed dataset. As a baseline we evaluate the performance of traditional and recently proposed learned local image features. To demonstrate the benefit of perspective rectification, we perform image matching with the same set of local features on the same set of image pairs.

For all image pairs in the dataset, feature matching was performed using SIFT [47], SuperPoint [19], ORB [62] and BRISK [37] features. For each feature type, feature matching was performed between features extracted from the original images, as well as between the perspectively corrected features, as explained in Sec. 3. Our unoptimized implementation performs image rectification in around 0.8 seconds per image. Using the established matches and the known intrinsics of the images, an essential matrix was computed, and the relative camera pose was then retrieved from this. This relative pose was compared to the ground truth relative pose (computed by Colmap as described in Sec. 4). An image pair was considered successfully localized if the difference between the estimated relative rotation and the ground truth relative rotation was smaller than 5°, where the difference between two rotations is taken as the magnitude of the smallest rotation that aligns one rotation with the other. Also included is a curve showing the performance of SIFT features extracted from images rectified using a vanishing point-based rectification method [11].

Fig. 6 shows the performance of feature matching directly on the image pairs, *vs*. matching after perspective rectification. The 18 difficulty classes as described in Sec. 4 are listed along the $x-$axis, and the fraction of image pairs successfully localized in that difficulty class is shown on the $y-$axis.

As can be seen in the figures, extracting perspectively corrected features can improve the pose estimation performance for planar scenes, particularly for SIFT and BRISK features. Overall, SuperPoint features seem to be more robust to viewpoint changes, which is natural since the SuperPoint feature is trained by performing homographic warps of patches. The ORB features show less improvement from using perspectively corrected images. This may have to do with the fact that these are not scale-invariant, and thus only correcting for

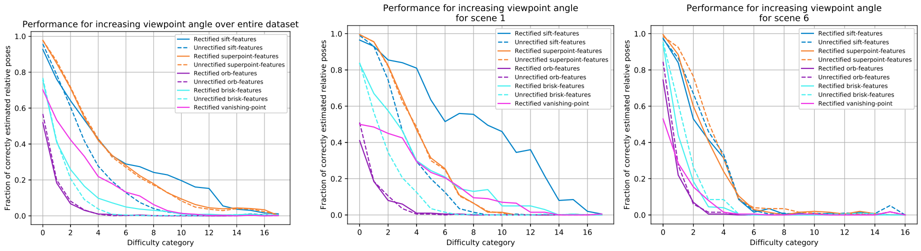


Fig. 6. Performance degradation due to increasing viewpoint difference. For each of the difficulty categories (labelled from 0 to 17 on the x-axis), the y-value shows the fraction of image pairs for which the relative rotation was estimated correctly to within 5 degrees. Feature matching in the original images was compared with our rectification approach, for a variety of local features. Depth-based rectification is helpful overall, particularly for scenes with dominant planes, and more or less reduces to regular feature matching for scenes where no planes can be extracted. *Left:* Results for all images, over all scenes, in the entire dataset. *Middle:* A scene where most image pairs show the same plane, and this plane takes up a large portion of the images. *Right:* A scene containing many small facades, each often occupying a small part of the image, and some non-planar scene structures.

the projective distortion, but not for the scale, may not be sufficient for obtaining good feature matching performance for these features.

In the supplementary material, localization rate graphs, like the middle and right figures in Fig. 6, can be found for all eight scenes.

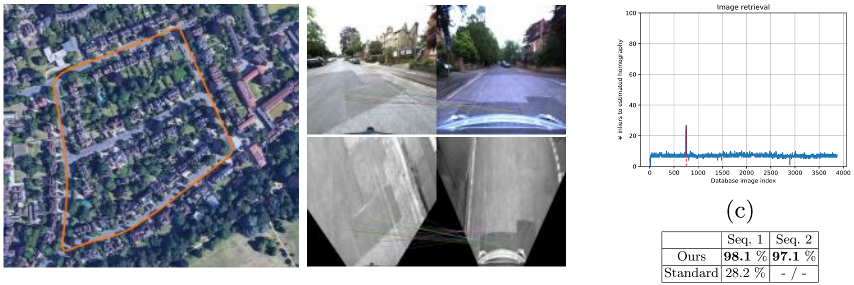
5.2 Re-localization from Opposite Viewpoints

For our next experiment, we consider a re-localization scenario for autonomous driving. More precisely, we consider the problem of re-localizing a car driving down streets in the opposite direction from its first visit to the scene. Such a problem occurs, for example, during loop closure detection inside SLAM.

We use a subset of the Oxford RobotCar dataset [48], shown in Fig. 7, namely the “Alternate Route” dataset already used in [72]. The dataset consists of two traversals of the scene. We use 3,873 images captured by the front-facing camera of the RobotCar during one traversal as our database representation. 729 images captured by the car’s rear-facing camera during the same traversal as well as an additional 717 images captured by the rear camera during a second traversal (captured around 10 minutes after the first one) are used as query images. As a result, there is a 180° viewpoint change between the query and database images.

We determine the approximate location from which the query images were taken by matching SIFT features between the query and database images. We compare our approach, for which we only unwarp the ground plane and use SIFT features, against a baseline that matches features between the original images[2]. For both approaches, we use a very simple localization approach that

[2] For the baseline, we only use approximately every 2nd query image.



(a)*

(b)

(c)

	Seq. 1	Seq. 2
Ours	**98.1** %	**97.1** %
Standard	28.2 %	- / -

(d)

Fig. 7. Re-localization from Opposite Viewpoints. (a) Satellite imagery of the area covered in the RobotCar "Alternate Route" dataset. We show the trajectory of one of the sequences overlaid in orange.(b) Example of feature matching between rear and frontal images in the RobotCar dataset. Feature matching is performed in the rectified space (bottom), and then visualised in the original images (top). (c) Results of the search for the best front-facing database image for the rear-facing image from (b). (d) Localization results on the two sequences of the RobotCar dataset as the percentage of localized images. We compare our approach based on unwarping the ground plane with matching features in the original images. * Image taken from Google Maps. Imagery ©2020 Google, Map data ©2020

exhaustively matches each query image against each database image. For our approach, we select the database image with the largest number of homography inliers, estimated using RANSAC [23]. For the baseline, we select the database image with the largest number of fundamental matrix inliers, since we noticed that most correct matches are found on the buildings on the side of the road, and the corresponding points in the two images are thus not generally related by a homography. Due to the 180° change in viewpoint, the query and its corresponding database image might be taken multiple meters apart. Thus, it is impossible to use the GPS coordinates provided by the dataset for verification. Instead, we manually verified whether the selected database image showed the same place or not (see also the supp. video).

Tab. 7(d) shows the percentage of correctly localized queries for our method and the baseline. As can be seen, our approach significantly ourperforms the baseline. A visualization for one query image and its corresponding database image found by our method is shown in Fig. 7(b), while Fig. 7(c) shows the number of homography inliers between this query and all database images. As can be seen, there is a clear peak around the correctly matching database image. This result is representative for most images localized by our approach (*cf*. the supp. video), though the number of inliers in the figure is on the lower end of what is common.

6 Conclusion

The results from Sec. 5 show that our proposed approach can significantly improve feature matching performance in real-world scenes and applications in dominantly planar scenes without a significant degradation in other environments. They further demonstrate that our approach is often easier to use than classical vanishing point-based approaches, which was one of the main motivations for this paper. Yet, our approach has its limitations.

Limitations. Similar to vanishing point-based methods, our approach requires that the planar structures that should be undistorted occupy a large-enough part of an image. If these parts are largely occluded, *e.g.*, by pedestrians, cars, or vegetation, it is unlikely that our approach is able to estimate a stable homography for unwarping. Further, the uncertainty of the depth predictions increases quadratically with the distance of the scene to the camera (as the volume projecting onto a single pixel grows quadratically with the distance). As a result, unwarping planes too far from the camera becomes unreliable. In contrast, it should be possible to relatively accurately undistort faraway scenes based on vanishing points or geometrically repeating elements. This suggests that developing hybrid approaches that adaptivley choose between different cues for perspective undistortion is an interesting avenue for future research.

Another failure case results from the fact that all training images seen by the depth prediction network have been oriented upright. As such, the network fails to produce meaningful estimates for cases where the images are rotated. However, it will be easy to avoid such problems in many practical applications: it is often possible to observe the gravity direction through other sensors or to pre-rotate the image based on geometric cues [78].

Future work. We have shown that using existing neural networks for single-image depth prediction to remove perspective distortion leads to a simple yet effective approach to improve the performance of existing local features. A natural direction for further work is to integrate the unwarping stage into the learning process for local features. Rather than assuming that perspective distortion is perfectly removed, this would allow the features to compensate for inaccuracies in the undistortion process. Equally interesting is the question whether feature matching under strong viewpoint changes can be used as a self-supervisory signal for training single-image depth predictors: formulating the unwarping stage in a differentiable manner, one could use matching quality as an additional loss when training such networks.

Acknowledgements The bulk of this work was performed during an internship at Niantic, and the first author would like to thank them for hosting him during the summer of 2019. This work has also been partially supported by Swedish Foundation for Strategic Research (Semantic Mapping and Visual Navigation for Smart Robots) and the Chalmers AI Research Centre (CHAIR) (VisLocLearn). We would also like to extend our thanks Iaroslav Melekhov, who has captured some of the footage.

References

1. Aanæs, H., Dahl, A., Steenstrup Pedersen, K.: Interesting interest points. International Journal of Computer Vision **97**, 18–35 (2012) 2
2. Aanæs, H., Dahl, A.L., Pedersen, K.S.: Interesting interest points. International Journal of Computer Vision **97**(1), 18–35 (2012) 5
3. Altwaijry, H., Trulls, E., Hays, J., Fua, P., Belongie, S.: Learning to Match Aerial Images With Deep Attentive Architectures. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 5
4. Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M.: Handling Urban Location Recognition as a 2D Homothetic Problem. In: European Conference on Computer Vision (ECCV). pp. 266–279 (2010) 2, 4
5. Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M.: Leveraging 3D City Models for Rotation Invariant Place-of-Interest Recognition. International Journal of Computer Vision (IJCV) **96**(3), 315–334 (Feb 2012). https://doi.org/10.1007/s11263-011-0458-7, `https://doi.org/10.1007/s11263-011-0458-7` 2, 4
6. Balntas, V., Lenc, K., Vedaldi, A., Mikolajczyk, K.: Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5173–5182 (2017) 5, 9, 10
7. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000) 11
8. 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) 7
9. Changchang Wu, Fraundorfer, F., Frahm, J., Pollefeys, M.: 3d model search and pose estimation from single images using vip features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (2008) 3, 5
10. Chaudhury, K., DiVerdi, S., Ioffe, S.: Auto-rectification of user photos. In: IEEE International Conference on Image Processing (ICIP) (2014) 2, 4
11. Chaudhury, K., DiVerdi, S., Ioffe, S.: Auto-rectification of user photos. In: 2014 IEEE International Conference on Image Processing (ICIP). pp. 3479–3483. IEEE (2014) 3, 11
12. Chen, D.M., Baatz, G., Kser, K., Tsai, S.S., Vedantham, R., Pylvninen, T., Roimela, K., Chen, X., Bach, J., Pollefeys, M., Girod, B., Grzeszczuk, R.: City-scale landmark identification on mobile devices. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011) 2, 4
13. Chen, W., Fu, Z., Yang, D., Deng, J.: Single-image depth perception in the wild. In: Advances in neural information processing systems. pp. 730–738 (2016) 5
14. Chen, W., Xiang, D., Deng, J.: Surface normals in the wild. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1557–1566 (2017) 5
15. Cordes, K., Rosenhahn, B., Ostermann, J.: High-resolution feature evaluation benchmark. In: International Conference on Computer Analysis of Images and Patterns. pp. 327–334. Springer (2013) 5
16. Criminisi, A., Reid, I., Zisserman, A.: Single view metrology. International Journal of Computer Vision (IJCV) **40**(2), 123–148 (Nov 2000) 2, 4
17. Criminisi, A.: Single-view metrology: Algorithms and applications. In: Van Gool, L. (ed.) Pattern Recognition (DAGM) (2002) 4

18. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. PAMI **29**(6), 1052–1067 (2007) 1
19. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 224–236 (2018) 7, 11
20. Dhamo, H., Navab, N., Tombari, F.: Object-driven multi-layer scene decomposition from a single image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5369–5378 (2019) 5
21. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. pp. 2650–2658 (2015) 5
22. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in neural information processing systems. pp. 2366–2374 (2014) 5
23. 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) 13
24. Fraundorfer, F., Bischof, H.: A novel performance evaluation method of local detectors on non-planar scenes. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops (2005) 2
25. Gálvez-López, D., Tardós, J.D.: Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics **28**(5), 1188–1197 (October 2012). https://doi.org/10.1109/TRO.2012.2197158 1
26. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision. pp. 740–756. Springer (2016) 5
27. Germain, H., Bourmaud, G., Lepetit, V.: Sparse-to-Dense Hypercolumn Matching for Long-Term Visual Localization. In: International Conference on 3D Vision (3DV) (2019) 2
28. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 270–279 (2017) 2, 5
29. 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) 5, 7, 9
30. Heng, L., Choi, B., Cui, Z., Geppert, M., Hu, S., Kuan, B., Liu, P., Nguyen, R.M.H., Yeo, Y.C., Geiger, A., Lee, G.H., Pollefeys, M., Sattler, T.: Project AutoVision: Localization and 3D Scene Perception for an Autonomous Vehicle with a Multi-Camera System. In: 2019 International Conference on Robotics and Automation (ICRA) (2019) 1
31. Hickson, S., Raveendran, K., Fathi, A., Murphy, K., Essa, I.: Floors are flat: Leveraging semantics for real-time surface normal prediction. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019) 5
32. Hinterstoisser, S., Lepetit, V., Benhimane, S., Fua, P., Navab, N.: Learning real-time perspective patch rectification. International Journal of Computer Vision **91**(1), 107–130 (Jan 2011) 5
33. Jones, E.S., Soatto, S.: Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. The International Journal of Robotics Research (IJRR) **30**(4), 407–430 (2011) 4

34. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning sfm from sfm. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 698–713 (2018) 5
35. Koser, K., Koch, R.: Perspectively Invariant Normal Features. In: IEEE International Conference on Computer Vision (ICCV) (2007) 2, 3, 4, 5
36. Kuznietsov, Y., Stuckler, J., Leibe, B.: Semi-supervised deep learning for monocular depth map prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6647–6655 (2017) 5
37. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: 2011 IEEE international conference on computer vision (ICCV). pp. 2548–2555. Ieee (2011) 7, 11
38. Li, B., Shen, C., Dai, Y., Van Den Hengel, A., He, M.: Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1119–1127 (2015) 5
39. Li, H., Zhao, J., Bazin, J.C., Chen, W., Liu, Z., Liu, Y.H.: Quasi-Globally Optimal and Efficient Vanishing Point Estimation in Manhattan World. In: The IEEE International Conference on Computer Vision (ICCV) (2019) 2, 4
40. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Computer Vision and Pattern Recognition (CVPR) (2018) 2, 5, 7
41. Liebowitz, D., Criminisi, A., Zisserman, A.: Creating Architectural Models from Images. Computer Graphics Forum **18**(3), 39–50 (1999) 4
42. Lim, H., Sinha, S.N., Cohen, M.F., Uyttendaele, M.: Real-time image-based 6-dof localization in large-scale environments. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012) 1
43. Lindeberg, T.: Scale-space theory: A basic tool for analysing structures at different scales. Journal of Applied Statistics **21**(2), 224–270 (1994) 1
44. Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J.: Planercnn: 3d plane detection and reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4450–4459 (2019) 5
45. Liu, C., Yang, J., Ceylan, D., Yumer, E., Furukawa, Y.: Planenet: Piece-wise planar reconstruction from a single rgb image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2579–2588 (2018) 5
46. Liu, W., Wang, Y., Chen, J., Guo, J., Lu, Y.: A completely affine invariant image-matching method based on perspective projection. Machine Vision and Applications **23**(2), 231–242 (Mar 2012) 4
47. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004) 1, 7, 11
48. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research **36**(1), 3–15 (2017) 12
49. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing **22**(10), 761 – 767 (2004) 2
50. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Gool, L.V.: A comparison of affine region detectors. International Journal of Computer Vision **65**(1), 43–72 (Nov 2005) 2, 5
51. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence (2005) 9, 10
52. Mishkin, D., Matas, J., Perdoch, M.: MODS: Fast and robust method for two-view matching. Computer Vision and Image Understanding **141**, 81–93 (2015) 4, 5

53. Moo Yi, K., Verdie, Y., Fua, P., Lepetit, V.: Learning to Assign Orientations to Feature Points. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1
54. Morel, J.M., Yu, G.: Asift: A new framework for fully affine invariant image comparison. SIAM journal on imaging sciences **2**(2), 438–469 (2009) 4
55. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017). https://doi.org/10.1109/TRO.2017.2705103 1, 2
56. Pang, Y., Li, W., Yuan, Y., Pan, J.: Fully affine invariant surf for image matching. Neurocomputing **85**, 6 – 10 (2012) 4
57. Pritts, J., Chum, O., Matas, J.: Rectification, and Segmentation of Coplanar Repeated Patterns. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 4
58. Pritts, J., Kukelova, Z., Larsson, V., Chum, O.: Rectification from Radially-Distorted Scales. In: Asian Conference on Computer Vision (ACCV) (2018) 2, 4
59. Pritts, J., Kukelova, Z., Larsson, V., Chum, O.: Radially-Distorted Conjugate Translations. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2, 4
60. Pritts, J., Rozumnyi, D., Kumar, M.P., Chum, O.: Coplanar Repeats by Energy Minimization. In: Proceedings of the British Machine Vision Conference (BMVC) (2016) 4
61. Robertson, D.P., Cipolla, R.: An Image-Based System for Urban Navigation. In: BMVC (2004) 2, 4
62. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.R.: Orb: An efficient alternative to sift or surf. In: ICCV. vol. 11, p. 2. Citeseer (2011) 7, 11
63. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 2
64. Sattler, T., Leibe, B., Kobbelt, L.: Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(9), 1744–1756 (2017) 1, 2
65. Schönberger, J.L., Pollefeys, M., Geiger, A., Sattler, T.: Semantic Visual Localization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1
66. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1, 2, 10
67. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016) 10
68. Shao, H., Svoboda, T., Gool, L.V.: ZuBuD — Zürich buildings database for image based recognition. Tech. Rep. 260, Computer Vision Laboratory, Swiss Federal Institute of Technology (April 2003) 5
69. Simon, G., Fond, A., Berger, M.O.: A-Contrario Horizon-First Vanishing Point Detection Using Second-Order Grouping Laws. In: The European Conference on Computer Vision (ECCV) (2018) 2, 4
70. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: Exploring photo collections in 3D. In: SIGGRAPH (2006) 1
71. Svärm, L., Enqvist, O., Kahl, F., Oskarsson, M.: City-Scale Localization for Cameras with Known Vertical Direction. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(7), 1455–1461 (2017) 1

72. Toft, C., Olsson, C., Kahl, F.: Long-term 3d localization and pose from semantic labellings. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 650–659 (2017) 12
73. Wang, X., Fouhey, D., Gupta, A.: Designing deep networks for surface normal estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 539–547 (2015) 5
74. Watson, J., Firman, M., Brostow, G.J., Turmukhambetov, D.: Self-supervised monocular depth hints. In: IEEE International Conference on Computer Vision (ICCV) (2019) 5, 7, 9
75. Wu, C.: Towards Linear-Time Incremental Structure from Motion. In: International Conference on 3D Vision (3DV) (2013) 2
76. Wu, C., Clipp, B., Li, X., Frahm, J.M., Pollefeys, M.: 3d model matching with viewpoint-invariant patches (VIP). In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008) 2, 3, 4, 6
77. Wu, C., Frahm, J.M., Pollefeys, M.: Detecting large repetitive structures with salient boundaries. In: European Conference on Computer Vision (ECCV) (2010) 2, 4
78. Xian, W., Li, Z., Fisher, M., Eisenmann, J., Shechtman, E., Snavely, N.: UprightNet: Geometry-Aware Camera Orientation Estimation From Single Images. In: The IEEE International Conference on Computer Vision (ICCV) (2019) 14
79. Yang, N., Wang, R., Stuckler, J., Cremers, D.: Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 817–833 (2018) 5
80. Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R.: Lego: Learning edge with geometry all at once by watching videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 225–234 (2018) 5
81. Yanpeng Cao, McDonald, J.: Viewpoint invariant features from single images using 3d geometry. In: Workshop on Applications of Computer Vision (WACV) (2009) 4
82. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5684–5693 (2019) 5
83. Zeisl, B., Köser, K., Pollefeys, M.: Viewpoint Invariant Matching via Developable Surfaces. In: European Conference on Computer Vision Workshops (2012) 2, 3
84. Zeisl, B., Köser, K., Pollefeys, M.: Automatic Registration of RGB-D Scans via Salient Directions. In: The IEEE International Conference on Computer Vision (ICCV) (2013) 2, 3, 4
85. Zhan, H., Weerasekera, C.S., Garg, R., Reid, I.: Self-supervised learning for single view depth and surface normal estimation. 2019 International Conference on Robotics and Automation (ICRA) (2019) 5
86. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 5
87. Zhou, Y., Qi, H., Huang, J., Ma, Y.: NeurVPS: Neural Vanishing Point Scanning via Conic Convolution. In: Conference on Neural Information Processing Systems (NeurIPS) (2019) 2, 4

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-

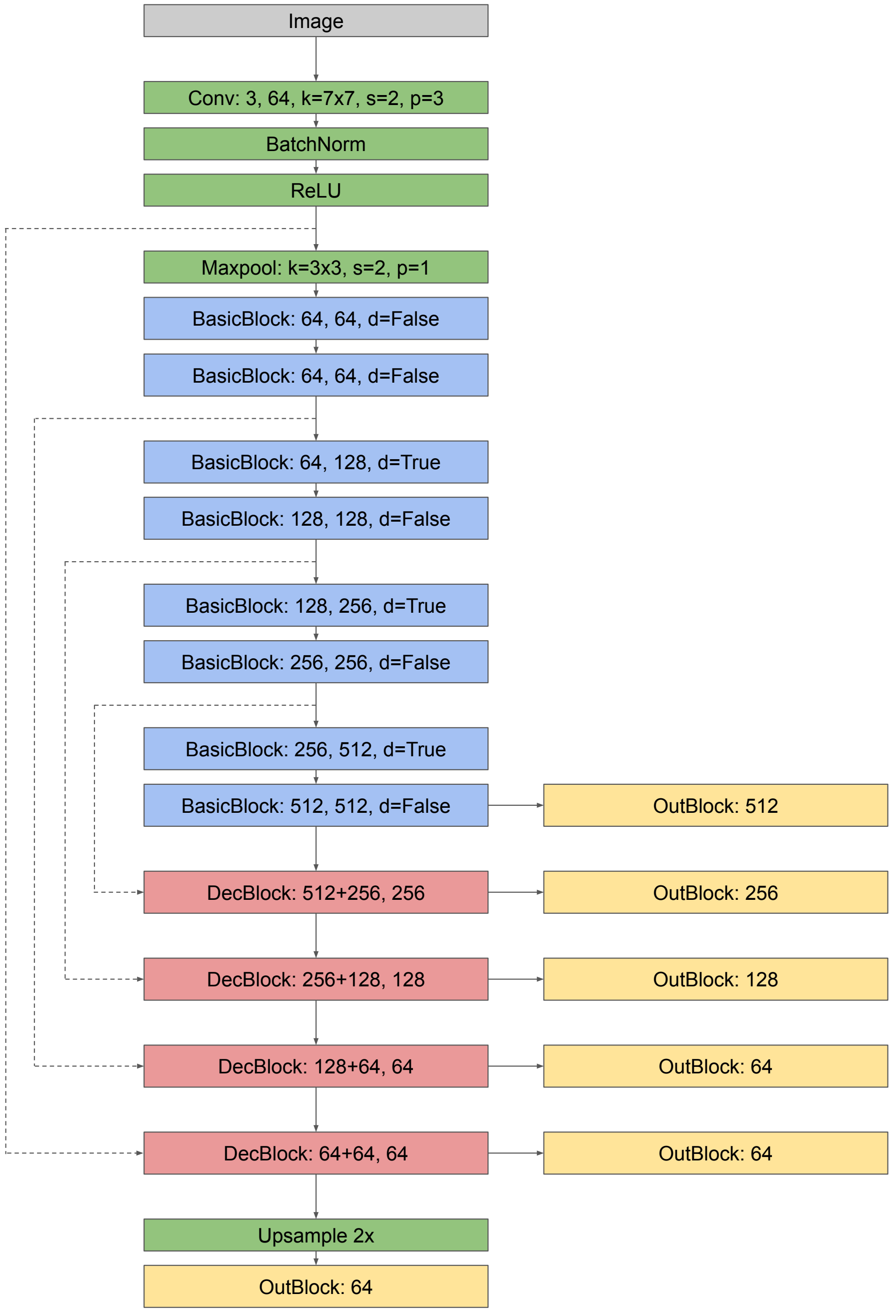


Image
Conv: 3, 64, k=7x7, s=2, p=3
BatchNorm
ReLU
Maxpool: k=3x3, s=2, p=1
BasicBlock: 64, 64, d=False
BasicBlock: 64, 64, d=False
BasicBlock: 64, 128, d=True
BasicBlock: 128, 128, d=False
BasicBlock: 128, 256, d=True
BasicBlock: 256, 256, d=False
BasicBlock: 256, 512, d=True
BasicBlock: 512, 512, d=False
OutBlock: 512
DecBlock: 512+256, 256
OutBlock: 256
DecBlock: 256+128, 128
OutBlock: 128
DecBlock: 128+64, 64
OutBlock: 64
DecBlock: 64+64, 64
OutBlock: 64
Upsample 2x
OutBlock: 64

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

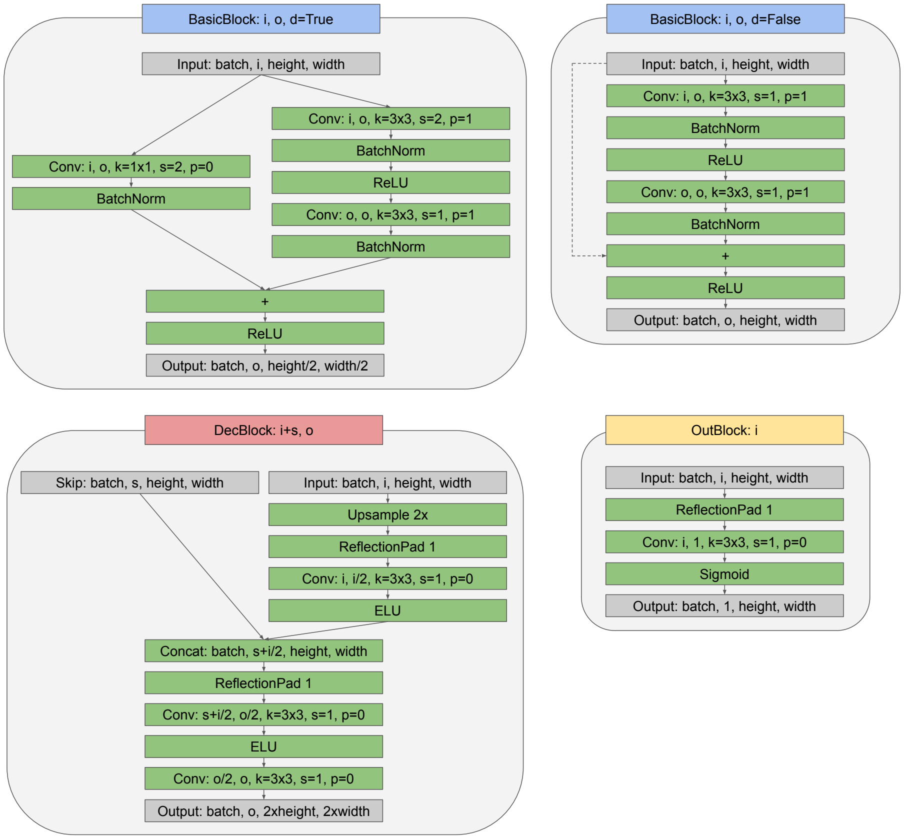

BasicBlock: i, o, d=True
Input: batch, i, height, width
Conv: i, o, k=3x3, s=2, p=1
BatchNorm
ReLU
Conv: o, o, k=3x3, s=1, p=1
BatchNorm
Conv: i, o, k=1x1, s=2, p=0
BatchNorm
+
ReLU
Output: batch, o, height/2, width/2
BasicBlock: i, o, d=False
Input: batch, i, height, width
Conv: i, o, k=3x3, s=1, p=1
BatchNorm
ReLU
Conv: o, o, k=3x3, s=1, p=1
BatchNorm
+
ReLU
Output: batch, o, height, width
DecBlock: i+s, o
Skip: batch, s, height, width
Input: batch, i, height, width
Upsample 2x
ReflectionPad 1
Conv: i, i/2, k=3x3, s=1, p=0
ELU
Concat: batch, s+i/2, height, width
ReflectionPad 1
Conv: s+i/2, o/2, k=3x3, s=1, p=0
ELU
Conv: o/2, o, k=3x3, s=1, p=0
Output: batch, o, 2xheight, 2xwidth
OutBlock: i
Input: batch, i, height, width
ReflectionPad 1
Conv: i, 1, k=3x3, s=1, p=0
Sigmoid
Output: batch, 1, height, width

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.

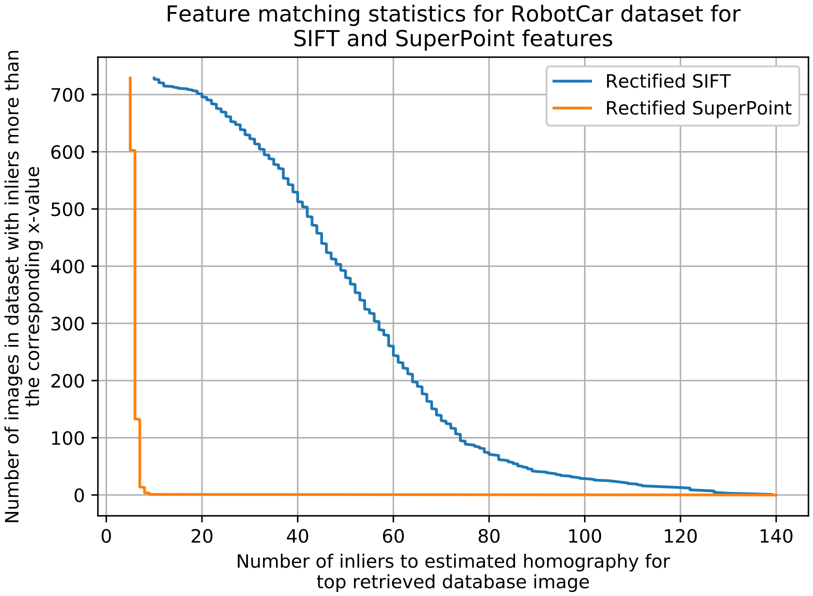


Fig. 3. Number of inliers to the homography estimated during pairwise matching between the query images and the top-retrieved database images for the first sequence of the RobotCar dataset. The y-values denote the number of images whose homography has at least the number of inliers specified on the x-axis.

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.

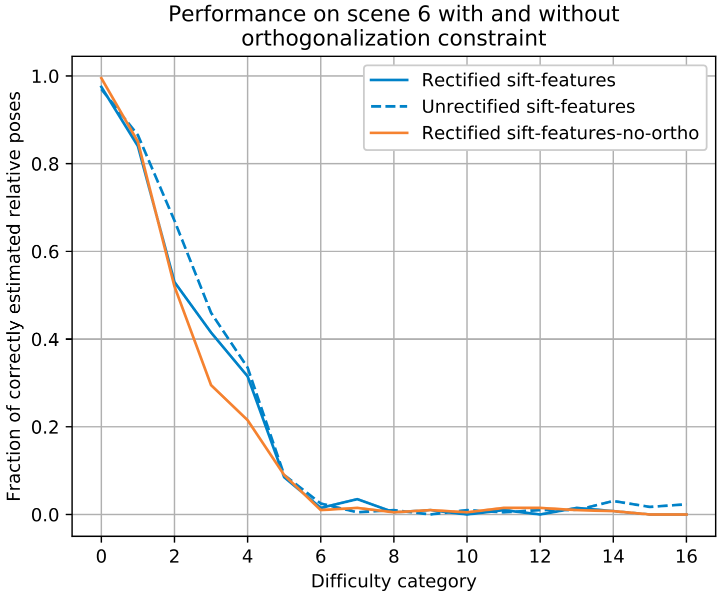


Fig. 4. Performance on scene 6 with and without enforcing the normal clusters to be orthogonal.

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.

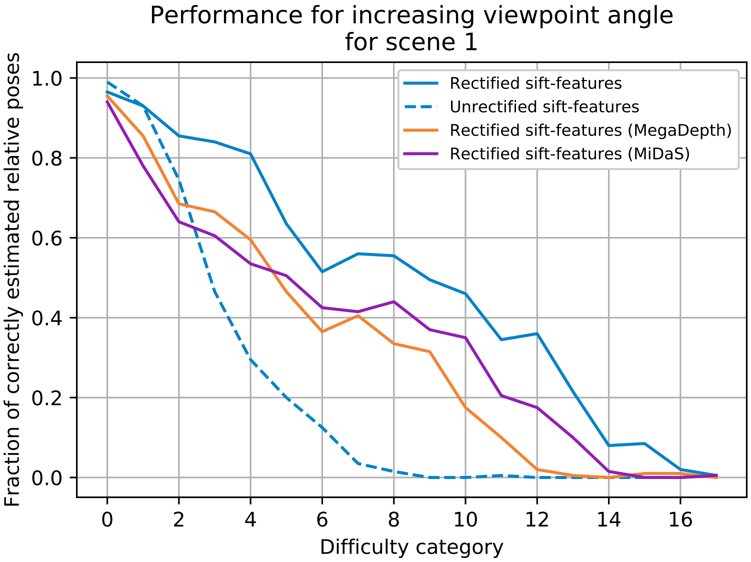


Fig. 5. Performance on scene 1 using two other monocular depth prediction networks. Only the depth prediction network has changed, the rest of the pipeline remains unchanged.

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.

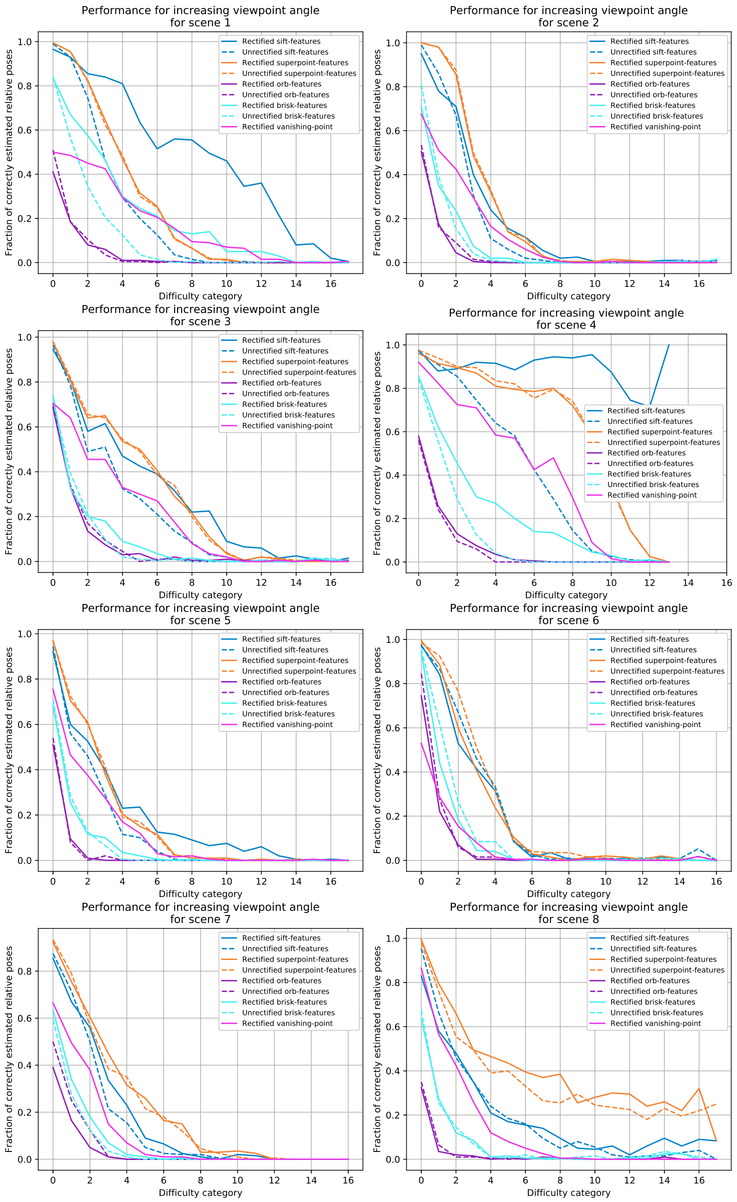


Fig. 6. Detailed results on the presented local feature matching dataset, showing the performance on each scene individually.

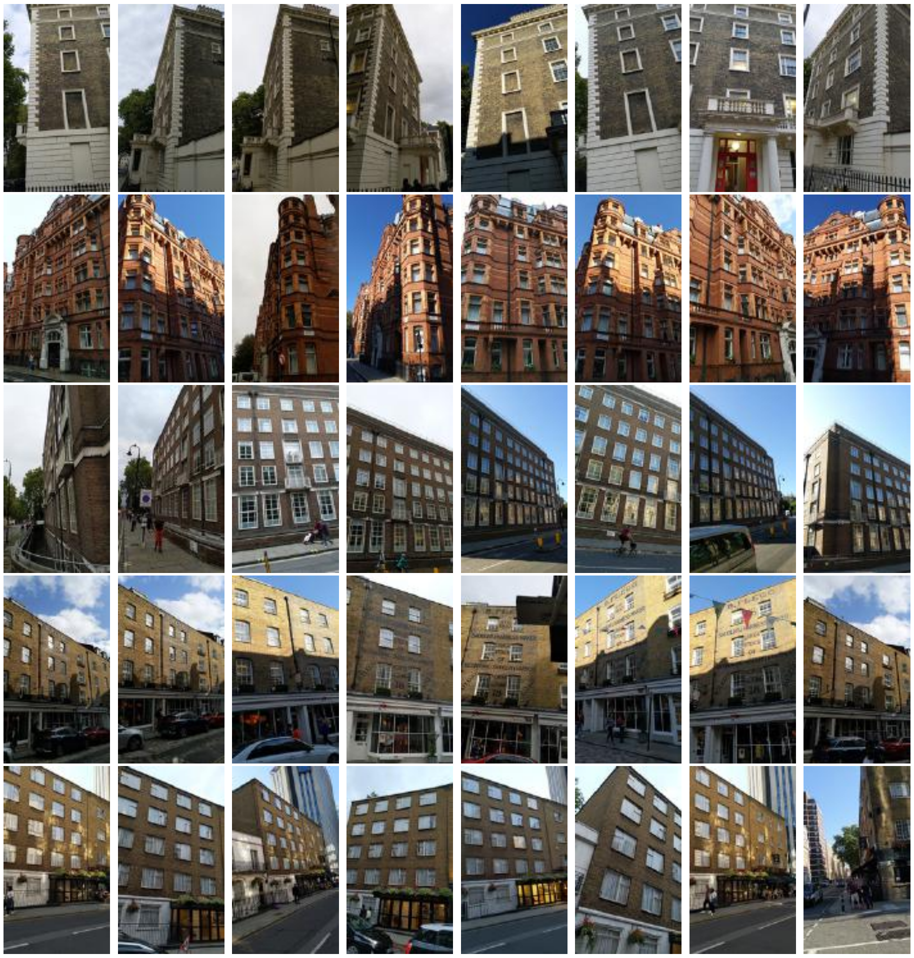


Fig. 7. Example images from our dataset for Strong Viewpoint Changes. Each row shows a sample of images showing scenes 1 to 5.

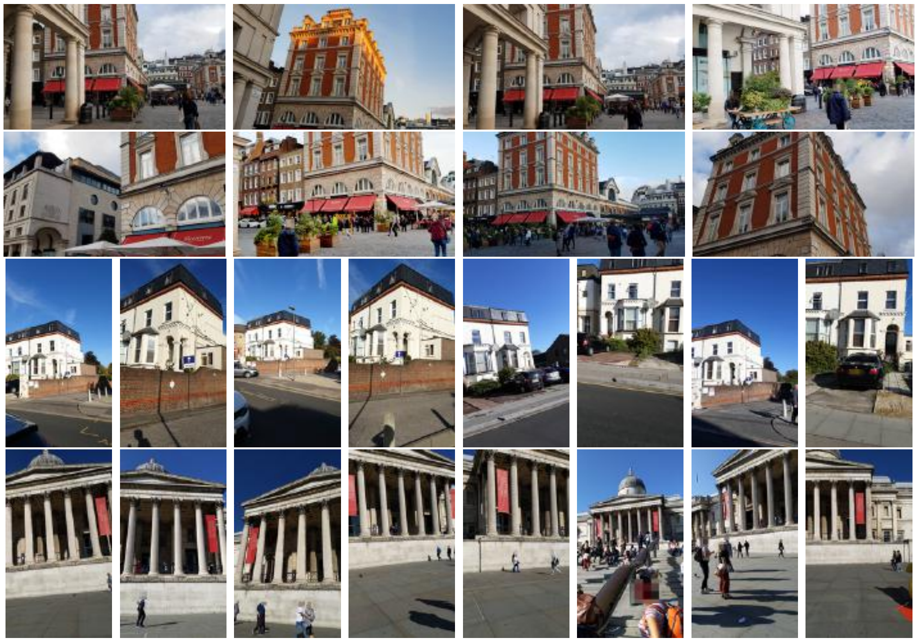


Fig. 8. Example images from our dataset for Strong Viewpoint Changes. Top two rows show a sample of images for scene 6. The following rows show images of scene 7 and 8, respectively.

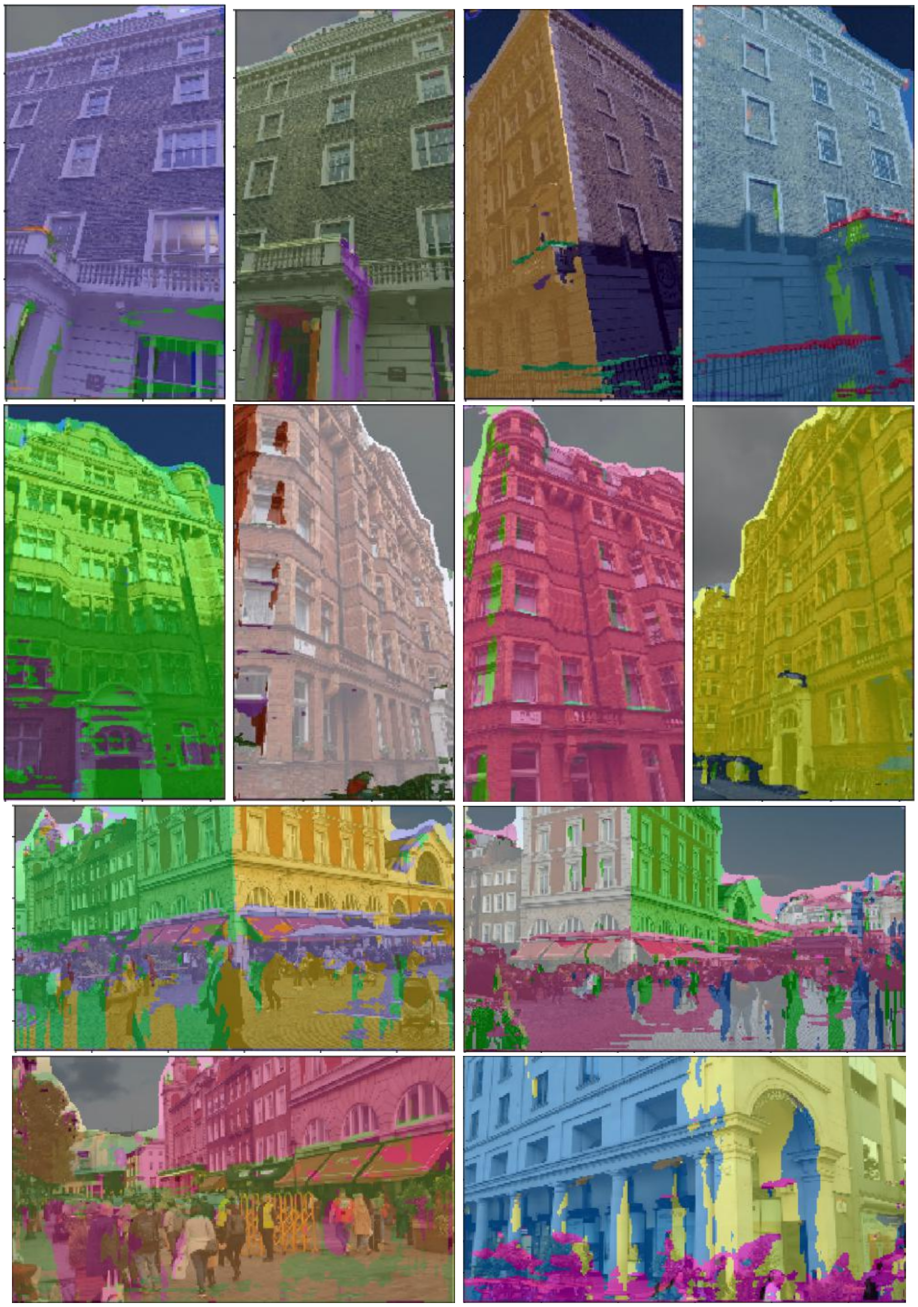


Fig. 9. Normal clustering results on four images from three of the scenes in our dataset.

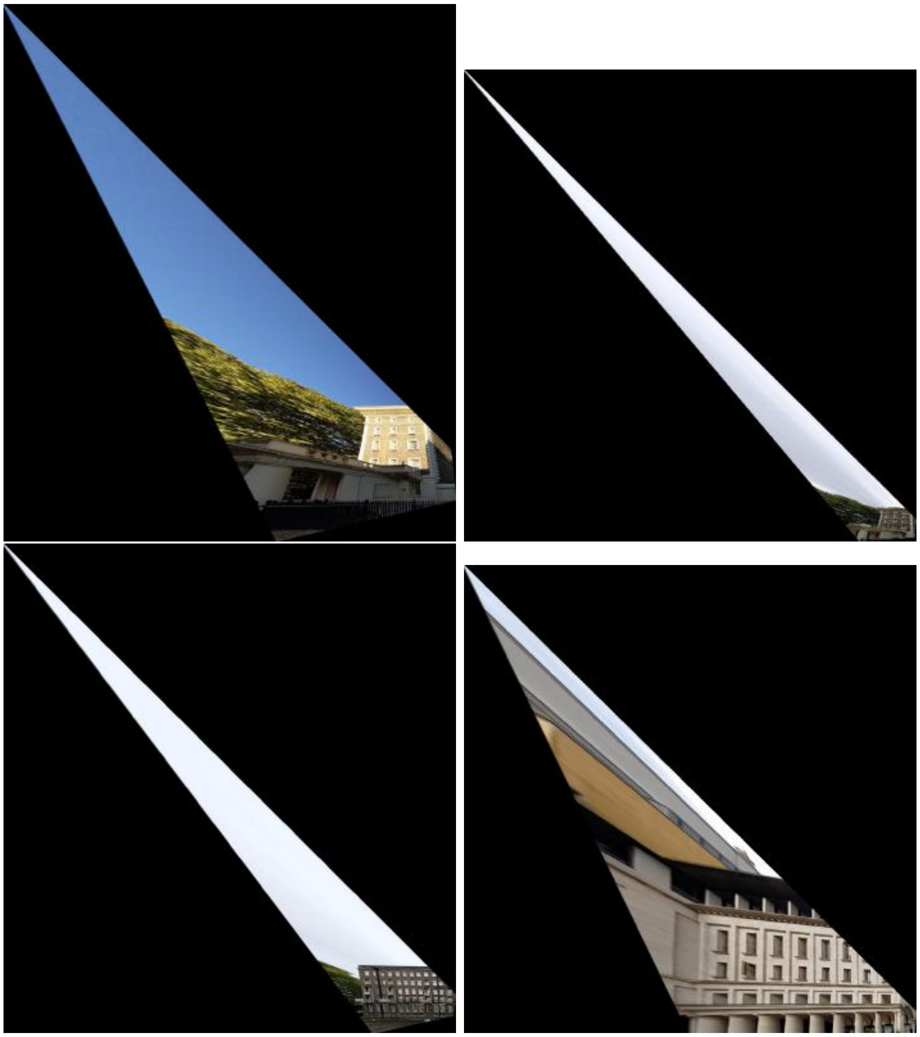


Fig. 10. Examples of heavily distorted images that have been rectified using a vanishing point based rectification method.

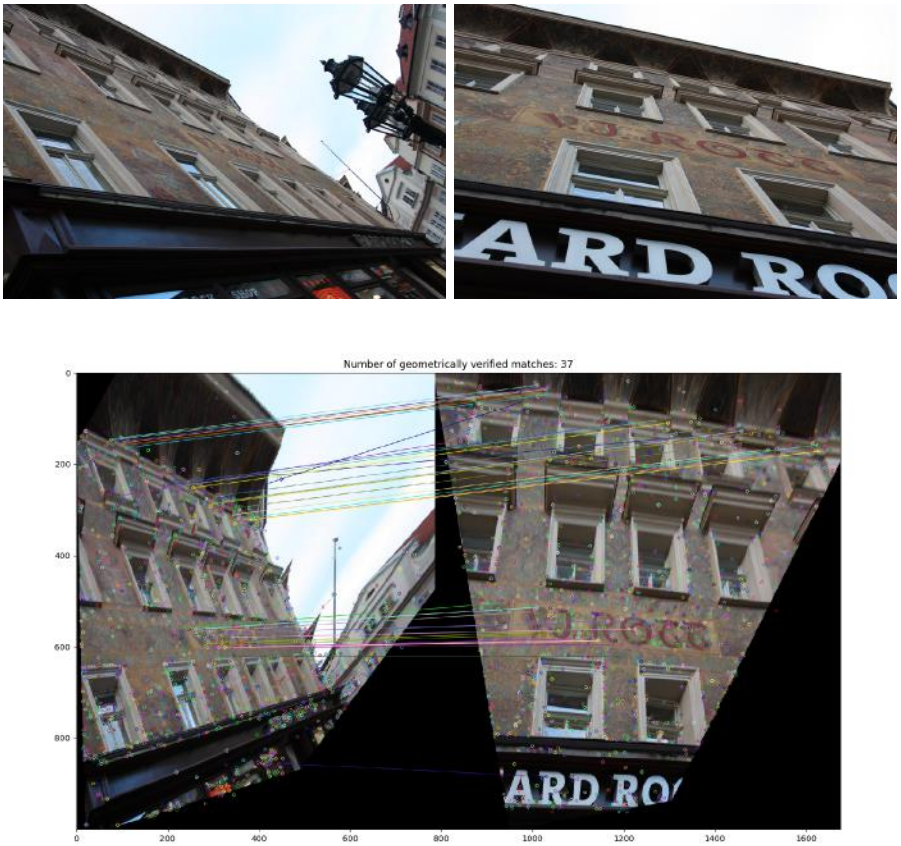


Fig. 11. Results on the Café scene in the EVD dataset. Top row: Original images. Bottom row: Geometrically consistent matches between the rectified patches.

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