Local Image Feature Matching Improvements for Omnidirectional Camera Systems

Benjamin Resch
Computer Graphics Group
Tübingen University
72076 Tübingen, Germany
benjamin.resch@uni-tuebingen.de

Jochen Lang
EECS
University of Ottawa
Ottawa, On., K1N 6N5, Canada
jlang@eecs.uottawa.ca

Hendrik P. A. Lensch
Computer Graphics Group
Tübingen University
72076 Tübingen, Germany
hendrik.lensch@uni-tuebingen.de

***Abstract*—The matching of oriented local image feature descriptors like SIFT, SURF or ORB often includes the refinement and filtering of matches based on the relative orientation of the features. This is important since the computational cost for subsequent tasks like camera pose estimation or object detection increases dramatically with the number of outliers. Simple 2D orientation descriptions are unsuitable for omnidirectional images because of image distortions and non-monotonic mapping from camera rotations to image rotations. In this work we introduce 3D orientation descriptors which, unlike 2D descriptors, are suitable for match refinement on omnidirectional images and improve matching results on images from cameras and camera rigs with a wide field of view. We evaluate different match refinement strategies based on 2D and 3D orientations and show the fundamental advantages of our approach.**

I. Introduction

The extraction of local image features is a key task in many algorithms such as image classification, object recognition, camera pose estimation, camera calibration and many others. Many local feature extraction techniques like SIFT [1], SURF [2] or GLOH [3] achieve rotation invariance in the image plane by assigning an orientation to each feature before extracting a feature descriptor from the image patch relative to this orientation.

The feature orientations are not only useful for feature extraction but also for refining the matches that are found for a pair of images based on the feature descriptors: For "normal" 2D images with a decent field of view (FoV) (usually $< 40°$), it is a legitimate assumption that all features in an image pair have the same rotation behavior: inverse to the camera roll.

For wider FoVs, the scene is mapped to the image plane quite differently in different image regions (see Figure 1). This effect gets even worse for strong image distortions.

Fig. 1. Inconsistent feature rotations on wide field of view images. Note how the orientation of the feature at the bottom left/right of the left/right image changes while the central image areas keep their orientation.

Omnidirectional camera rigs can provide FoVs of virtually 360°, which are usually handled as a set of individual images or mapped to a cylindrical, spherical or other representation for feature extraction. In all those types of images, the rotation of matched features does not monotonically map to the camera's orientation (see Figure 2): Tilting and rolling of the rig results in features rotating in opposite directions.

To overcome both issues, we propose a 3D orientation descriptor that can be inferred from its 2D counterpart and the intrinsic camera calibration (per pixel view direction in camera space): In principle, our orientation descriptor defines an orthonormal space for each feature whose base axes are given by (1) the 2D feature orientation transformed to camera space, (2) the view direction towards the feature in camera space and (3) the cross product of (1) and (2) (see Figure 3).

This 3D orientation frame allows for significantly more robust matching of features between omnidirectional and wide-angle camera images. The number of spurious matches is significantly reduced using our 3D orientation while positive matches are essentially detected unhindered. Subsequent processing tasks such as Structure from Motion (SfM) become significantly more robust and more efficient after this simple outlier removal based on 3D orientation.

In the following sections, we introduce related work considering the handling of wide FoVs as well as omnidirectional cameras with local image features. We present our 3D ori-

Fig. 2. Omnidirectional images with straight (top) and rolled (bottom) camera. Note how the image areas in the back facing part rotate inverse to the front, so there is no common 2D feature rotation.

entation descriptor in detail, describe our match refinement strategies and evaluate and compare our work to the common use of 2D feature orientations.

II. Related Work

All local image feature frameworks discussed in this section contain the following steps: during the extraction of features from the images, remarkable feature keypoints are detected and some basic properties including orientation and scale are determined. At the keypoints, feature descriptors are calculated from image patches relative to the keypoints' orientation and scale to achieve scale and rotation invariance. Feature descriptors may not always be extracted from the image data directly but also from an image pyramid, e.g., for better performance, or filtered versions of the image, e.g., for higher distinctiveness.

Matching the features of an image pair starts with finding the best matches based on some similarity metric on the feature descriptor. Usually, thresholding is applied to accept only matches deemed more certain for the subsequent processing. This matching usually treats all features *individually*. Frameworks such as SIFT [1] refine the set of matches by analyzing keypoint properties, e.g. feature orientations, of all matches *in common* for consistency. By dropping inconsistent matches, the number of outliers is reduced. This match refinement stage is what we analyze and improve in this work.

We can apply our improvements to all image features based matching frameworks that contain an orientation: SIFT [1], SURF [2] and GLOH [3] determine feature orientations based on the (smoothed) gradient at the feature's location. Newer binary descriptors assign orientations more sophistically and robustly. ORB [4] uses intensity centroids [5]: it calculates moments for all pixels of a patch and obtains the orientation based on their gradients. BRISK [6] and FREAK [7] use the sum of gradients obtained from special sample pairs of patches around a keypoint. Scaramuzza et al. [8] extract vertical line features which are oriented by definition. Lu and Wu [9] do quasi-dense matching of omnidirectional and perspective images using an affine model for local transformations which can be reduced to a rotation.

There are several SIFT extensions which target wide FoV and omnidirectional cameras during feature extraction: Arican and Frossard [10] focus on calculating the scale space pyramid used for keypoint detection correctly for parabolic mirrors by solving the heat equation using Riemannian geometry. Hansen et al. [11] and Cruz-Mota et al. [12] make features invariant against distortions by projecting the images to a sphere and then back to planes tangential to the sphere at the feature's location. Lourenço et al. [13] implement scale space construction and feature extraction without image resampling by using adaptive filtering that compensates image distortion. All those publications focus on feature extraction and do not concern match refinement. Our approach of match refinement based on feature orientation is orthogonal to the above techniques and can be applied on top of those for further improvement.

A. Applications

Our match refinement provides advantages on all cameras with a large field of view or distorted images, e.g., currently emerging action cams like the GoPro Hero Series.

The greatest improvements can however be observed on omnidirectional camera systems, e.g. custom camera rigs, professional devices such as the various Point Grey's Ladybugs, commercial omnidirectional camera systems like the 360° camera in the new Mercedes E-Class or the Panono Panoramic Ball Camera. Beside one shot solutions, many smartphones are able to capture panoramas, e.g. using Photo Sphere (Google) or Photosynth (Microsoft).

III. 3D Orientation Descriptor

Our 3D orientation descriptor V which can be calculated for each feature defines an orthonormal 3D space. For each match with its features' 3D orientation descriptors V_1, V_2, one can find the rotation $r^{3D} = V_2 * V_1^{-1}$ which transforms from one feature space to the other. The orientation descriptors are designed in a way that the match rotation r^{3D} is roughly the inverse of the camera rotation and therefore similar for all correct matches of an omnidirectional image pair.

The 3D orientation descriptor is calculated based on the 2D orientation ω of a feature in the image plane, its position $p \in \mathbb{R}^2$ in the image and the intrinsic calibration of the camera (or camera rig) $(C(p) = d_c) \in \{\mathbb{R}^2 \rightarrow \mathbb{R}^3\}$ that maps image coordinates to directions in camera space. Note that calculating the 3D descriptor works the same way for single images, spherical or cylindrical mappings or sets of images that represent an omnidirectional view.

The 3D orientation descriptor is defined by $V = (\vec{v_1}\vec{v_2}\vec{v_3})$ (see also Figure 3):

$$\vec{v_1} = \frac{C(p_x + \cos(\omega), p_y + \sin(\omega)) - C(p_x, p_y)}{\|C(p_x + \cos(\omega), p_y + \sin(\omega)) - C(p_x, p_y)\|} \quad (1)$$

$$\vec{v_2} = C(p_x, p_y) \quad (2)$$

$$\vec{v_3} = \vec{v_1} \times \vec{v_2} \quad (3)$$

A. Properties

By design, the rotation r^{3D} obtained from two matched features is the inverse of the camera (or camera rig) rotation (see Figure 4). In contrast to 2D orientations, all our 3D orientation descriptors rotate the same way under arbitrary camera transformations. This allows for analyzing a set of feature matches for consistent rotations and discarding inconsistent matches.

Another benefit is that we incorporate the features' viewing directions d_c into the orientation descriptors. This enables us

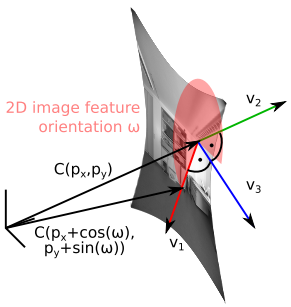

Fig. 3. 3D orientation descriptor. The red, green and blue vectors represent the axes of the orthonormal descriptor frame defined by the orientation and position of each feature.

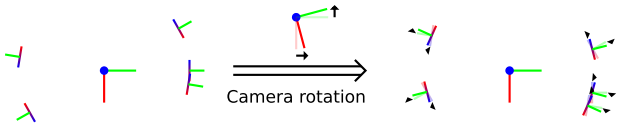

Fig. 4. Descriptor behavior under camera rotation. If the camera is rotated, all 3D feature orientations perform an inverse rotation.

to distinguish between features with similar appearance and 2D orientation but different viewing direction.

IV. Feature Matching

The proposed 3D orientation helps in refining matches by analyzing the rotations between their features for consistency and removing inconsistent matches. We do not propose any new method on how to find matches. Instead our approach can work with any pre-selection of matches.

The general procedure for analyzing matches given 3D feature orientations is the same as for 2D feature orientations. The steps are as follows:

1) Calculate a rotation r_i from the first to the second feature's orientation of each match.
2) Find a representative rotation $\hat{r}$ for the rotations r_i of all matches.
3) Remove all matches whose feature's rotations r_i differ from the representative $\hat{r}$ by more than a certain threshold.

The next two subsections for 2D and 3D descriptor match refinement first describe how to obtain a rotation from an orientation (step 1) and how to calculate differences in rotation (step 3). Then we present different strategies for finding a representative rotation $\hat{r}$ (step 2) in both subsections.

A. 2D Descriptor Match Refinement

For 2D feature orientations ω_1, ω_2 which are scalar rotation angles, we obtain the scalar rotation for a match $r^{2D} = \omega_2 - \omega_1$ by simple subtraction. The results are mapped to the range $(-\pi, \pi)$ to have the identity rotation in the middle of the interval.

We implemented and tested the following strategies for obtaining $\hat{r}^{2D}$ from a set of rotations r_i^{2D}:

1) Histograms: This idea is borrowed from SIFT [1] and sorts all r_i^{2D} into histogram bins. To avoid problems at the bin boundaries, the bins overlap by 50%, so each r_i^{2D} is assigned to two bins. The center of the bin with the most rotations is used as $\hat{r}^{2D}$.

2) Mean: Using the average of all rotations is expected to lead to good results for evenly distributed outlier rotations but is prone to error for multimodal distributions which can arise from rotated omnidirectional cameras (see Figure 11).

3) Cosine fit: We select the $\hat{r}^{2D}$ that fits a shifted $\cos^2$ function to the relative rotations r_i^{2D} by maximizing

$$E(\hat{r}^{2D}) = \sum_i \left(\frac{1}{2} \cos(r_i^{2D} - \hat{r}^{2D}) + \frac{1}{2} \right)^2 \quad (4)$$

with a gradient ascent method using a Levenberg-Marquardt [14] inspired step size. This method should nicely fit to a peak in the rotation distribution - it might however not be the global optimum.

4) K-Means: K-Means clustering is applied to all r_i^{2D}. K-Means cluster centers will converge to the peaks in the rotation distribution, so the center of the largest cluster can be selected as $\hat{r}^{2D}$. Based on tests with 2-10 clusters we decided that three clusters lead to the best k-Means results for all of our test scenes. This corresponds to up to trimodal 2D rotation distributions of our data (see Figure 11).

B. 3D Descriptor Match refinement

For a match's 3D feature orientations V_1, V_2 which are orthonormal matrices, we obtain the match's rotation matrix $r^{3D} = V_2 * V_1^{-1}$. This r^{3D} can be interpreted as the inverse camera rotation matrix (compare with Section III-A). As distance associated with a rotation we will use the shortest rotation angle $\alpha(r^{3D}) = \arccos(\mathrm{Tr}(r^{3D}) - 1)$ according to [15].

We implemented the following strategies for obtaining $\hat{r}^{3D}$ from a set of rotations r_i^{3D}:

1) Mean: Using the average of all matches' rotations is promising for 3D orientations since the rotation distribution won't get multimodal for omnidirectional setups by design (compare to Section IV-A2) and outlier rotations can be expected to be more randomly distributed since we incorporate scene information *plus* viewing direction information into the feature orientations. The mean of rotations for all matches is obtained by the orthonormalized sum of all r_i^{3D}:

$$(U, \Sigma, V^T) = \mathrm{SVD}\left(\sum_i r_i^{3D} \right) \quad (5)$$

$$\hat{r}^{3D} = U * V^T \quad (6)$$

Singular Value Decomposition (SVD) is employed for orthonormalization to keep the rotational parts U, V^T while setting the scaling part Σ to identity. This generates the optimal solution in a least squares sense [16].

2) Cosine fit: $\hat{r}^{3D}$ is selected like in the 2D case (see Section IV-A3) by maximizing

$$E(\hat{r}^{3D}) = \sum_i \left(\frac{1}{2} \cos(\alpha((\hat{r}^{3D})^{-1} * r_i^{3D})) + \frac{1}{2} \right)^2 \quad (7)$$

where $\alpha((\hat{r}^{3D})^{-1} * r_i^{3D})$ is the angular difference between the representative rotation $\hat{r}^{3D}$ and the rotation r_i^{3D} of a match. As in Section IV-A3, this will fit to a peak in the rotation distribution which will typically be the global optimum since we expect an unimodal distribution.

3) K-Means: We proceed as in the 2D case (see Section IV-A4) and use $\alpha(r_j^{3D} * (r_k^{3D})^{-1})$ as distance metric for a pair of rotations. As before, three clusters already lead to the best k-Means results for all our test scenes. This strategy was mainly implemented to have a counterpart to the 2D version.

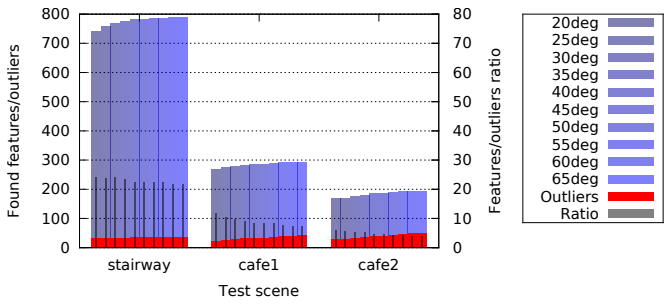

Fig. 5. Selecting a good rotation threshold for 2D features using the cosine fit strategy. We chose $e^{2D} = 35°$ because this threshold selects almost all correct feature matches while the match/outlier ratio is still high.

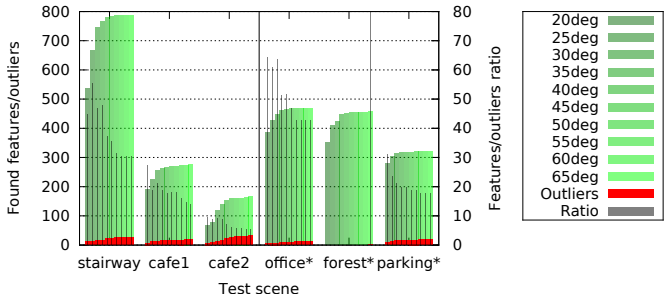

Fig. 6. Selecting a good rotation threshold for 3D features using the cosine fit strategy. We chose $e^{3D} = 35°$ because this threshold selects almost all correct feature matches while the match/outlier ratio is still high.

C. Match Selection

All matches whose deviation from $\hat{r}$ is smaller than a certain error threshold e are *selected*; the others are *discarded*. For a fair comparison, we selected a common threshold e.

To select a good e, we sampled the thresholds in steps of five degrees on our test scenes using the cosine fit strategies from Section IV-A3 and IV-B2 (see Figures 5 and 6). It turns out that setting $e = 35°$ is high enough to select most good matches while a higher threshold gives overproportional rise to the outliers.

Note that the original SIFT matching [1] uses 30° bins for 2D features. For the 2D case, our higher threshold is justified by the fact that our cameras have more radial distortion. The 3D match refinement strategies are invariant to image distortion because we use the intrinsic camera calibration, but we incorporate the viewing direction into the descriptors which will cause more variance (Section III).

V. Evaluation

Our proposed 3D orientation descriptor and all match refinement strategies are evaluated on datasets captured with a Point Grey Ladybug 3 omnidirectional camera system and a GoPro Hero 3 wide angle camera.

The Ladybug 3 system has 5 equatorially aligned and one up-facing cameras, each shipped with calibration data. The six individual images have a resolution of 1616x1232 pixels each. Six scenes were captured with the omnidirectional camera taking two shots each (see Figure 7): for *stairway*, *cafe1* and *cafe2* the camera was moved but not rotated. In *office**, *forest** and *parking**, the camera was rolled and tilted, so that there is no uniform 2D image feature rotation. Features were extracted by a standard SIFT-like implementation [1] as the FoV for each camera is moderate.

Fig. 7. Ladybug test shots used for evaluation. From left to right: *stairway*, *cafe1* (small baseline), *cafe2* (wider baseline), *office**, *forest** and *parking** (cloudy non static sky blackened). * scenes include camera roll and tilt.

The two scenes *office'* and *parking'* were acquired with a wide angle GoPro Hero 3 camera (Figures 12 and 13 right part only) for which we performed intrinsic calibration. Images were scaled down to 1600x1200 pixels. For dealing with the wide FoV, SIFTS [12] was applied for feature extraction.

For all scenes, the feature descriptor based matches were classified manually to obtain ground truth for refinement.

A. Runtime Discussion

Let us first discuss the runtime to be expected by the different refinement strategies, namely histogram, mean, cosine fit and k-Means clustering.

The non-iterative histogram based and the mean strategies have $O(n)$ runtime complexity: Histogram bin assignment transforms each rotation to two bin indices (because we have overlapping bins) while the mean strategies basically sum up all rotations.

The iterative methods of cosine fitting and k-Means clustering exhibit $O(n)$ runtime *per iteration*: The cosine fit strategies compute the gradient contribution of each match's rotation with finite differences while the k-Means strategies obtain the distance of each rotation to the cluster centers.

Given that cosine fitting and k-Means clustering need a few iterations to converge, we can expect the mean strategy to be the fastest among the 3D orientation based methods.

B. Results

We first compare results of the de-facto standard SIFT-like histogram-based 2D refinement (Section IV-A1) with the fast mean-based 3D refinement (Section IV-B1). Figure 9 shows selected and discarded matches in the *office** scene. Note how our approach discards most false matches while keeping almost all inliers. Similar rotations of 2D feature orientations can only be found in a small image region. Therefore the 2D approach cannot discern positive and false matches except for one small image region. Figures 8 and 10 shows a random set of selected and discarded feature patches for both strategies. Note that the 2D strategy discards too many features because of inconsistent 2D image (and patch) rotations while our approach almost always classifies matches correctly.

In Figure 11, we visualize the rotations r_i of the matches for all scenes with all strategies. It is obvious that with

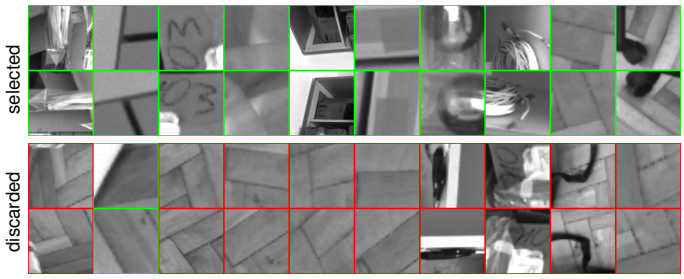

Fig. 8. Poor match refinement performance with 2D orientation based strategy. Randomly selected matched patches of the *office** dataset refined with the 2D histogram approach. Selected (top) and discarded (bottom) patches. Green borders indicate correct classification. Many correct matches are discarded because they don't exhibit a similar 2D rotation behavior.

Fig. 9. Failing 2D orientation vs. nicely separating 3D orientation based match refinement. 2D histogram based (top, Section IV-A1) vs. 3D mean (bottom, Section IV-B1) strategy. White lines represent selected, red lines discarded features. Our 3D refinement discards false matches selectively in the entire omnidirectional image while the 2D refinement only succeeds for a compact region of viewing directions, discarding lots of inliers.

the 2D strategies, the rotations are only similar for the first three scenes where the camera orientations don't change. The 3D strategies can handle all scenes correctly. In particular, the 2D mean strategy must fail by design in the * scenes (see Figure 2), while the 3D mean strategy produces similar rotations given the 3D feature orientations.

In Figure 11, we visualize the rotations r_i of the matches for all scenes with all strategies. It is obvious that with the 2D strategies, the rotations are only similar for the first three scenes where the camera orientations don't change. The 3D strategies can handle all scenes correctly. In particular, the 2D mean strategy must fail by design in the * scenes (see Figure 2), while the 3D mean strategy produces similar rotations given the 3D feature orientations.

We present the number of the selected matches compared to the number of included outliers for all our test sets in Figure 12. As a baseline, we also present all found matches

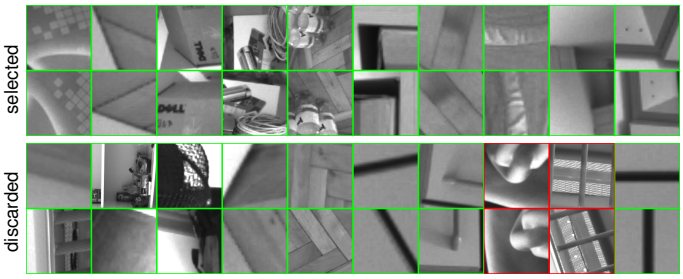

Fig. 10. Good match refinement performance with 3D orientation based strategy. Randomly selected matched patches of the *office** dataset refined with the 3D mean approach. Selected (top) and discarded (bottom) patches. Green borders indicate correct classification. This refinement classifies similarly looking patches correctly even if they don't exhibit a similar 2D rotation.

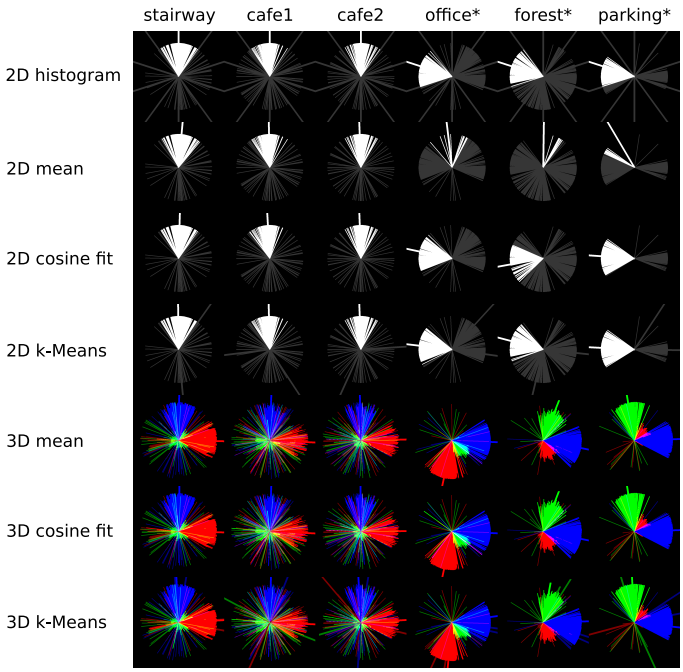

Fig. 11. Visualization of the relative feature rotations of matches. Long bars visualize the representatives $\hat{r}$. Bright bars show selected matches and representatives while dark bars show discarded ones. 3D visualizations show the axes of the 1^{st} camera in the 2^{nd} camera's coordinate system based on the feature rotations of the matches. Omnidirectional camera rotations lead to multimodal 2D rotation distributions that cannot be separated correctly. 3D rotations distribute unimodal and are therefore easy to separate into inliers and outliers.

including all outliers without refinement. It can be clearly seen that our 3D refinement strategies perform better in terms of outliers per match in all scenes. In the rolled and tilted * scenes almost all correct matches survive while the 2D strategies fail as they only select about 50% of the correct inlier matches. In the other scenes, the 2D strategies keep most inliers whereas our 3D approach removes significantly more outliers.

One possible subsequent task for feature matching is relative camera pose estimation using the 8-point algorithm [17] together with RANSAC [18] for robustness against the remaining outliers. The minimum number of iterations i necessary to pick m correct samples from a set of n samples polluted with

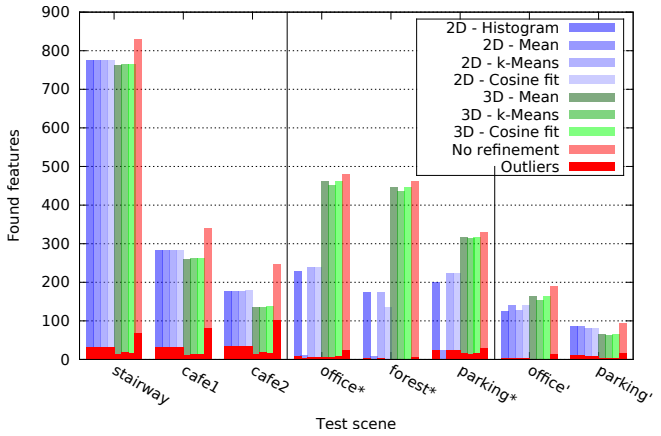

Fig. 12. Improvements from 3D orientation descriptors for outlier/match ratio and number of matches. 3D orientations lead to the fewest outliers per match in all test scenes. 3D strategies often keep less matches for fixed orientation omnidirectional image pairs (left part) and wide FoV image pairs (right part), which can mainly be explained by better outlier elimination. In the rolled/tilted * scenes, our 3D descriptor helps to discard false matches selectively while finding most of the correct matches which is about 100% more than the 2D orientation based methods can find.

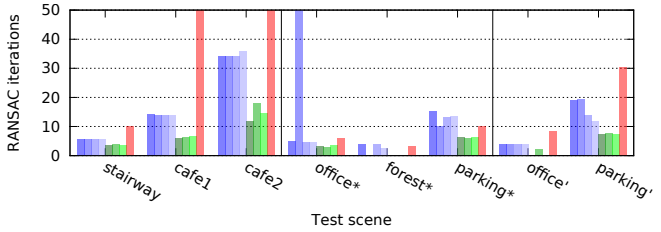

Fig. 13. Computational cost reduction by 3D orientations for exemplary subsequent RANSAC task. We show the number of iterations theoretically necessary for 8-point relative pose estimation desiring 99.9% success rate. 3D orientation based match refinement reduces the computational cost of this subsequent task by about 50%.

an outlier fraction of a_0 is approximately

$$i \geq \frac{\log(p_{max})}{\log(1-(1-a_o)^m)^n}, \tag{8}$$

given that error fraction $p_{max} = 0.001$ is tolerated. Figure 13 shows the number of RANSAC iterations that would theoretically be necessary for 8-point relative pose estimation based on the refined matches: our 3D orientation descriptor reduces the costs for that task by more than 50% on average by improving the input data.

VI. Conclusion

In this paper we propose a novel 3D orientation descriptor for local image features which is particularly designed for omnidirectional and wide FoV camera systems. Simple match refinement based on our 3D orientation descriptor outperforms refinement based on 2D orientations not only in cases where image features rotate differently in omnidirectional shots, but also for wide FoV and distorted images where we can exploit intrinsic camera calibration and viewing direction.

A limitation of our method is that an intrinsic camera calibration has to be provided. Beside that constraint, the 3D orientation descriptors can be built on top of any local image feature framework that extracts feature orientations. Our match refinement's computational overhead is low compared to other tasks in feature extraction, but the use of the 3D orientation descriptor can dramatically speed up subsequent tasks because of the reduced number of outliers while keeping almost all correct inliers. In some cases subsequent processing stages only succeed with the use of the 3D orientation based features.

Acknowledgment

This work has been partially funded by the DFG Emmy Noether fellowship (Le 1341/1-1).

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[4] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf." in *ICCV*. IEEE, 2011, pp. 2564–2571.

[5] P. L. Rosin, "Measuring corner properties." in *BMVC*. British Machine Vision Association, 1997.

[6] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2548–2555.

[7] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint." in *CVPR*. IEEE, 2012, pp. 510–517.

[8] D. Scaramuzza, N. Criblez, A. Martinelli, and R. Siegwart, "Robust feature extraction and matching for omnidirectional images," in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2008, vol. 42, pp. 71–81.

[9] L. Lu and Y. Wu, "Quasi-Dense Matching between Perspective and Omnidirectional Images," in *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008*. Marseille, France: Andrea Cavallaro and Hamid Aghajan, 2008.

[10] Z. Arican and P. Frossard, "Omnisift: Scale invariant features in omnidirectional images." in *ICIP*. IEEE, 2010, pp. 3505–3508.

[11] P. Hansen, P. Corke, and W. W. Boles, "Wide-angle visual feature matching for outdoor localization." *I. J. Robotic Res.*, vol. 29, no. 2-3, pp. 267–297, 2010.

[12] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J.-P. Thiran, "Scale invariant feature transform on the sphere: Theory and applications," *International Journal of Computer Vision*, pp. 1–25, 2012, 10.1007/s11263-011-0505-4.

[13] M. Loureno, J. P. Barreto, and F. Vasconcelos, "srd-sift: Keypoint detection and matching in images with radial distortion." *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 752–760, 2012.

[14] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly Journal of Applied Mathmatics*, vol. II, no. 2, pp. 164–168, 1944.

[15] A. S. Glassner, *Graphics Gems*. Orlando, FL, USA: Academic Press, Inc., 1990.

[16] W. D. Curtis, A. L. Janin, and K. Zikan, "A note on averaging rotations." in *VR*, 1993, pp. 377–385.

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, Mar. 2004.

[18] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.