Enhancing pose estimation through efficient patch synthesis

Pierre Rolin, Marie-Odile Berger, Frédéric Sur

Enhancing pose estimation through efficient patch synthesis

Pierre Rolin
https://members.loria.fr/PRolin/
Université de Lorraine

Marie-Odile Berger
https://members.loria.fr/MOBerger/
INRIA

Frédéric Sur
https://members.loria.fr/FSur/
Université de Lorraine

Abstract

Estimating the pose of a camera from a scene model is a challenging problem when the camera is in a position not covered by the views used to build the model, because feature matching is difficult. Several viewpoint simulation techniques have been recently proposed in this context. They generally come with a high computational cost, are limited to specific scenes such as urban environments or object-centred scenes, or need an initial pose guess. This paper presents a viewpoint simulation method well suited to most scenes and query views. Two major problems are addressed: the positioning of the virtual viewpoints with respect to the scene, and the synthesis of geometrically consistent patches. Experiments show that patch synthesis dramatically improves the accuracy of the pose in case of difficult registration, with a limited computational cost.

1 Introduction

Camera pose estimation from a single query view and an unstructured scene model, typically made of a 3D point cloud endowed with local photometric descriptors, is encountered in many computer vision applications. These applications include, for instance, augmented reality applications [4], vision-based robot positioning [6] and aerial image georegistration [25]. In many applications, the scene model is built from a collection of images (called here *construction views*) with a structure-from-motion (SfM) algorithm. The local descriptors of the 3D points are extracted from the construction views as, e.g., SIFT features [16]. Afterward, these descriptors are used to match interest points of the query view and 3D points, which makes it possible to solve the perspective-n-point (PnP) problem [9] and estimate the pose. This approach presents a major issue when the construction views do not cover the whole set of potential viewpoints. Indeed, a query view taken from an uncovered viewpoint is likely to give too few reliable point correspondences because of the limited invariance of the photometric descriptors to viewpoint changes [19]. A good example of such a situation is described by the authors of [25] who aim at registering a view from an aerial drone to a model built from ground-level construction views.

To make the matching step easier, several recent works propose to generate synthetic views from the construction views through some geometric transformations corresponding

to uncovered viewpoints, for instance [17, 20] in the context of image matching and [23] in the context of pose estimation. The existing approaches are generally dedicated to specific scene types or do not scale up well. The objective of the present work is to propose a view synthesis method that is tractable for most scenes and makes pose estimation possible for any query view in the scene. The following section discusses the related literature.

1.1 View synthesis for pose estimation

Two views of a plane of equation $n^T X + d = 0$ in the 3D scene, taken from cameras with projection matrices $P_i = K_i[R_i|T_i]$ ($i \in \{1,2\}$, where K_i is the intrinsic parameter matrix and $[R_i|T_i]$ is the camera pose) can be mapped by a homography of equation

$$H = K_2(R - Tn^T/d)K_1^{-1} \tag{1}$$

where $R = R_2R_1^T$ and $T = T_2 - RT_1$, see [8].

It is therefore possible to generate, for any virtual camera P_2, a synthetic view of a locally planar part of the 3D scene, from a construction view corresponding to the real camera P_1. Photometric descriptors can be extracted from such a synthetic view to enrich the scene model and make it easier to match a query view. An open question is, however, to select appropriate virtual positions with respect to the observed scene.

The authors of [14, 15, 30] generate fronto-parallel views of planar structures, which comes down to choosing a single virtual position in front of the considered scene planes. Robustness to viewpoint changes is improved but still limited in case of slanted views of the plane. In [28], pose estimation in a urban environment is addressed. The virtual positions lie on a dense grid at street level and a rough 3D planar model of the scene is used. Synthetic views, generated in [28] by ray-tracing, are matched to the query view, which gives reliable place recognition. Synthetic views from street level are also used in [11] to improve image registration to urban models. The preceding papers focuses on images taken by a pedestrian in a urban environments, which justifies the eye-level view assumption. The authors of [25] address the ground-to-aerial registration problem where this simplifying assumption does not hold. Nevertheless, they assume that an estimation of the aerial position is available from GPS tags. This makes it possible to generate a synthetic view from a dense reconstruction of the scene corresponding to the GPS position, which can be accurately registered to the query view. A similar idea is exploited in [6] in the context of vision-based robot localization where it is assumed that an estimation of the pose is available to drive view synthesis. The same assumption is used in some simultaneous localization and mapping (SLAM) applications to generate synthetic patches, after [18], or in tracking-by-synthesis [27].

These works require either a dense scene model (or a multiplanar textured reconstruction of the scene) [11, 15, 28, 30], or an initial guess for the pose [6, 25, 27]. In [23], no initial guess is available and the scene model is an unstructured 3D point cloud. It is assumed, however, that virtual viewpoints are regularly distributed on a sphere centered on the model. This restricts the applicability to relatively small object-centered scenes. In addition, all viewpoints have to be simulated to produce synthetic patches for all 3D points, making the algorithm quite demanding in terms of computing time.

As a conclusion of this short survey, and to the best of our knowledge, it seems that existing view synthesis approaches generally need some prior information on the scene or do not simply scale up to larger scenes.

In this paper, we consider pose estimation from a query view, based on a SfM model of the scene, without initial pose guess. For instance, such a problem has to be solved when

initializing a tracking process. Each 3D point of the model is endowed with the collection of the corresponding SIFT descriptors matched in the SfM step. SIFT keypoints from the query view are matched to the model points by nearest-neighbour matching followed by PnP-RANSAC [7]. Our goal is to add SIFT descriptors coming from synthesized patches in order to facilitate keypoint matching when the query view is not covered by the construction views. The additional SIFT descriptors are extracted around the reprojected scene points in the synthesized patches. Two problems have to be solved in a computationally efficient way: the positioning of the virtual cameras with respect to the scene in order to cover all potential viewpoints, and the simulation of realistic views from these virtual cameras.

1.2 Contributions

Our contributions are twofold. In Section 2, we propose a method to position the virtual viewpoints with respect to a segmentation of the scene in planar parts. An adapted measure for viewpoint changes, introduced in [20], ensures that the existing viewpoints are completed with relatively few virtual viewpoints. This positioning is generic and does not require any limiting assumption on the sought pose. In Section 3, we propose an efficient scheme for viewpoint simulation. In [11, 25, 28], an image is generated for any virtual viewpoint thanks to the dense scene model, and subsequently matched to the query view. In [6, 23], local patches are generated for any interest point thanks to a local planarity assumption. The first approach fails in cases where some parts of the scene are not correctly densified, and the latter is computationally demanding without any pose guess. We propose an intermediate approach consisting in synthesizing semi-local planar patches of the scene and enriching the scene model with descriptors from these synthesized patches, using a visibility constraint. Section 4 shows that this approach is sound and tractable for scenes ranging from small objects to complete buildings.

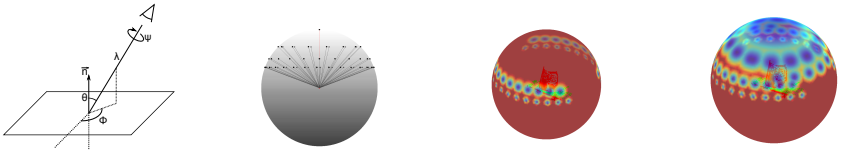
2 Virtual viewpoint positioning

We position virtual viewpoints in the scene, in order to simulate, in a subsequent stage, the appearance of the scene viewed from these viewpoints. The proposed method is based on the assumption that the scene is piecewise planar, which is not restrictive in most human-made environments. The following subsections discuss how to sample virtual viewpoints around a planar patch, and how to segment a point model into a set of planar patches.

2.1 View direction sampling

Considering a planar patch, we want any potential view of the patch to be close enough either to one of the simulated viewpoints or to one of the construction views, in order that SIFT features extracted from them can be matched. With affine cameras, the transition tilt, defined in [20], is a good indication of how easy it is to match SIFT features. Although it has been shown in [23] that homographic synthesis yields better results than affine synthesis, the affine model, as a first order approximation, is sufficient to position synthetic viewpoints. If A is the affine map between two images I_1 and I_2 of a planar scene (that is, $I_2 = AI_1$), then A has a unique decomposition:

$$A = \lambda\, R(\psi)\, T_t\, R(\phi) = \lambda \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \tag{2}$$



(1) (2) (3) (4)

Figure 1: In (1), parameterization of an affine camera pointing to a planar patch: λ is a scale factor, ψ the rotation around the optical axis, θ the latitude and ϕ the longitude. In (2), distribution of the sampled virtual viewpoints on a half-sphere lying on a planar patch. Map of the transition tilts as defined in Equation 2 to the closest viewpoint (blue is 0 and red is greater than the $\sqrt{2}$) for a planar patch of the pot dataset: with respect to the real viewpoints only (3) and with respect to the additional virtual viewpoints (4). The centres of the blue patches correspond to the viewpoint positions. We can see in (4) that most potential viewpoints are within a limited tilt of a real or synthetic viewpoint, making it possible to match SIFT features.

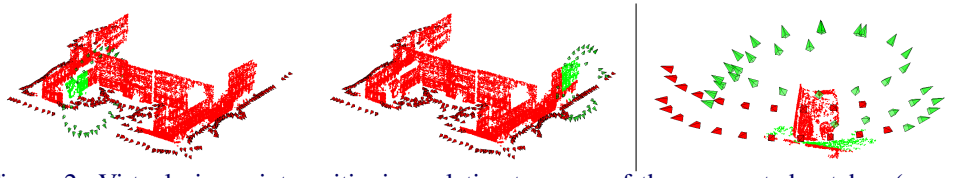


Figure 2: Virtual viewpoint positioning relative to some of the segmented patches (green points). In the left scene, only two rings of virtual viewpoints (in green) are added as the other potential viewpoints would have been close to existing viewpoints (in red).

where $R(\psi)$ and $R(\phi)$ are rotation matrices, and $t \geq 1$ is the transition tilt between the two views. If one of the view is fronto-parallel, the parameters correspond to the notations of Figure 1 (1), with $t = 1/\cos(\theta)$. Parameter t expresses how much the view is flattened out. Assuming SIFT features invariant to similarities, Equation (2) shows that, at fixed t and ϕ, any λ and ψ give the same features. This motivates to position the virtual viewpoints around the planar patch similarly to [20], that is, at (t,ϕ) such that $t = 2^{m/2}$ ($m \in \{1,2,3\}$) and $\phi = n72^\circ/t$ (with n such that ϕ spans $[0,360^\circ]$). The resulting sampling can be seen in Figure 1 (2). It should be noted that only affine cameras are considered in ASIFT [20]. This justifies that ASIFT limits ϕ to $[0,180^\circ]$ (for symmetry reasons) and does not consider the distance to the scene. Since we consider pinhole cameras, we have to set the distance of the virtual camera to the planar patch. We use the average distance of the real cameras to limit interpolation artefacts during synthesis. Since we are not interested in adding virtual viewpoints if a real viewpoint is already available, in order to limit redundant information, virtual viewpoints are added only if the transition tilt to one of the real viewpoints is larger than $\sqrt{2}$. This is illustrated in Figure 1 (3-4).

The following section explains how to segment the scene into planar patches, each one of them being associated with virtual viewpoints through the preceding process, as in Figure 2.

2.2 Planar segmentation

Segmentation is not an easy task because the SfM point cloud is noisy and non-uniformly sampled. We first estimate the normals at each point, via PCA on the neighbouring points

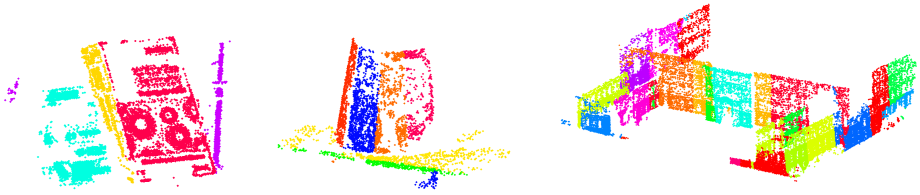


Figure 3: Three segmentation examples on an object-centred scene (left and middle) and a building (right). The datasets are the book, pot and CAB datasets that can be seen in Figure 5. Each planar patch is in a different colour. We can see that the curved surface of the pot is correctly approximated by a set of planar patches.

as in [10]. A simple iterative RANSAC scheme is used: RANSAC (with a fitting criterion based on both the distance between the points and the plane and the consistency of the normals at each point, described in [24]) gives points lying on a plane, which are iteratively removed until 90% of the model points are associated with a plane. Note that the fitting criterion eliminates points around the edges of the scene, the normal of these points being not consistent [5]. This robustifies the estimation of n in Equation (1).

Synthesizing the appearance of a patch far away from a virtual camera is likely to suffer from image quantization. This typically happens when synthesizing the appearance of large planes such as building façades. We therefore segment further the planes into smaller sets of points included in square cells oriented along the two principal directions of the plane, and of width equal to the average distance between a point and the cameras that reconstructed it, in order to ensure that the local scale change induced by a homography does not vary too much across the synthesized patch. Note that these cells are not necessarily aligned with the scene edges. The pieces of planes obtained by this segmentation are called *planar patches*.

Examples of segmentations are shown in Figure 3. Virtual viewpoints are positioned around the planar patches as in Figure 2. See also the videos `pot_positioning`, `tower_positioning` and `CAB_positioning` available as supplementary materials.

3 Patch synthesis

This section describes the simulation process. The scene model is supposed to be segmented into planar patches, each one of them being associated with a set of virtual viewpoints.

3.1 Image transformation

For each virtual viewpoint, the aspect of each planar patch is simulated with the homography given by (1). SIFT descriptors are extracted from the simulated views and associated with the corresponding 3D points. This in an intermediate approach between [11] where synthetic views come from full images, and [23] (additional experiments available in [22]) where many small overlapping patches are produced.

The remaining problem is to define from which construction view the synthetic patches should be simulated. As a patch may not have been fully observed in a single construction view, we may have to use several construction views. The views are selected using a greedy approach, by iteratively selecting the construction viewpoints in which the largest number of

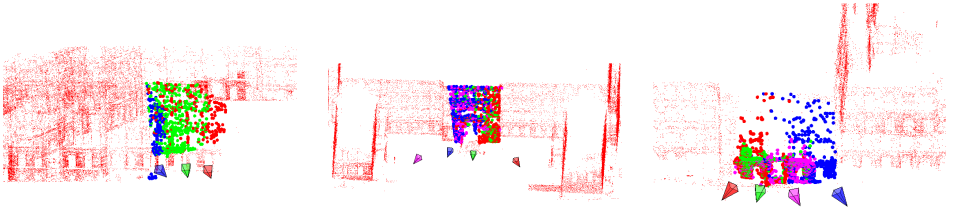


Figure 4: Examples of real viewpoint selection to be used in patch synthesis.

model points belonging to the considered planar patch are visible. The stopping criterion is that 90% of the patch points are visible from at least one of the viewpoints. A point of the model is considered visible in a construction view if there is a descriptor extracted from this view associated with this point in the SfM model. In all our experiments we needed at most five views to cover 90% of the points on a planar patch.

Figure 4 shows the set of points in a planar patch and the associated construction view in the same colour. SIFT descriptors extracted from simulated views based on the construction view are associated with these 3D points.

3.2 Visibility from virtual viewpoints

The preceding procedure simulates the aspect of planar shapes from virtual viewpoints, but it does not take into account potential occlusions from other parts of the scene. This means that it could simulate the appearance of some parts of a patch from a position where they are actually not visible. The resulting descriptors would not only unnecessarily increase the complexity of the model, but would also increase the outlier rate in the matching stage.

We therefore impose an additional visibility constraint. The authors of [13] propose an efficient method that only relies on 3D point location (no meshing is needed) and performs well on point clouds with a non uniform density. The set of model points visible from a point O is computed as follows. The model points are transformed using the so-called flipping transformation given by:

$$p' = p + 2(R - ||p||)\, p/||p|| \tag{3}$$

where p is a vector from O to a point of the model and R is a smoothing parameter. Let P' be the set of the transformed points. The set of points visible from O belongs to the preimage of the convex hull of $P' \cup O$. It is shown in [13] that the number of visible points increases with R. This method was originally designed for point models with noise levels much lower than in an SfM reconstruction, which imposes us to set R carefully. To choose R, we isolate each planar patch and run visibility tests from a fronto-parallel viewpoint for increasing R. Noise-free points lying on a plane should all be visible; this is not the case here. We choose R as the smallest value such that at least 90% of the points are visible.

It is worth noting that we can tolerate a few mislabelling as pose estimation is performed with RANSAC. The gain from this step is contextual, depending on the presence of occlusions in the scene. It may, however, be significant. For instance, in the pot dataset, the size of the model goes from 160,000 descriptors to 134,000 when using this visibility constrain, see Figure 5. All of these discarded descriptors would incorporate spurious data in the model.

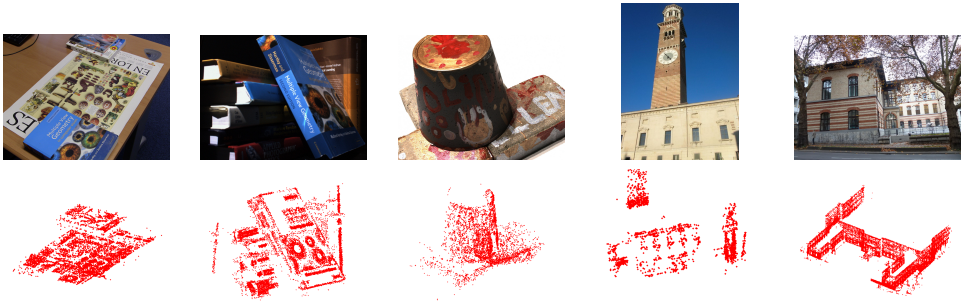


Figure 5: Sample images of five datasets and the reconstructed 3D point clouds. From left to right: poster (17 images), book (53 images), pot (21 images), tower (21 images) and CAB (300 images).

4 Experimental results

The experiments show that a model enriched with the proposed synthesis method leads to more accurate poses, and even gives accurate poses when pose estimation simply fails without synthesis. In addition the time needed for pose computation is reduced. The experimental setup consists in estimating the pose of a query view independent from the construction views, the model being built with VisualSfM [29]. The query view is typically chosen far away from the construction views. Poses are computed from the model using approximate nearest neighbour matching of the descriptors [21], followed by RANSAC filtering of the query/model correspondences, the pose being eventually estimated by Direct PnP [9]. RANSAC stopping criterion is based on an online estimation of the inlier ratio as proposed in [8],

Datasets go from a small object to a full building. The size of the scenes is limited to a few objects or buildings, which is a realistic assumption even in city-scale environments if a rough localization is available (through GPS for instance). Poster is a simple planar scene. Pot and book are small object-centred scenes from [3]. Tower is a relatively simple outdoor scene from [2] that essentially consists in two planar façades. CAB is a larger outdoor scene from [1]. This model is significantly larger (49,000 points and 325,000 descriptors, reconstructed from 300 images). On this latter dataset, the query views come from Google Street View, the acquisition conditions (camera, weather, viewpoint) are thus significantly different from the one of the construction views. In this experiment, the intrinsic camera parameters are estimated using the method described in [26]. Figure 5 displays sample images of the datasets (approximately 1,000 pixels wide) and the associated models.

As an illustration, Figure 6 shows 100 runs of the pose estimation for the representative pot and tower datasets. We can see that pose estimation is significantly improved by patch synthesis. Table 1 gives computing time (obtained on an Intel i7 quad core 2.7 GHz with 16 Gb memory). In all these experiments, the computing time for adding descriptors through patch synthesis was smaller than the reconstruction time, in spite that our implementation of patch synthesis is a Matlab code, while SfM is a compiled software. Note that both steps can be done offline, when building the scene model.

While the enriched models are significantly larger than the initial ones (for instance, it grows from 32,000 descriptors to 134,000 descriptors in pot), the pose estimation is not proportionaly longer. Table 1 shows that in our experiments the computation times for matching

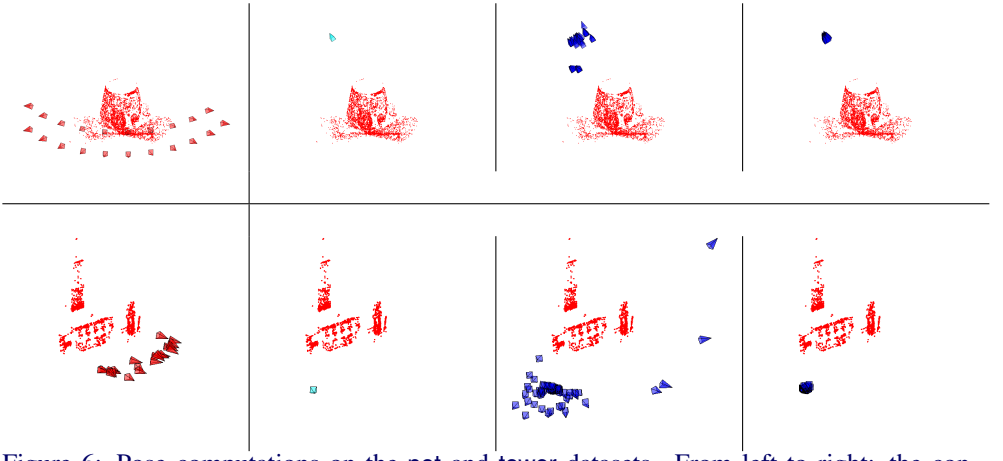


Figure 6: Pose computations on the pot and tower datasets. From left to right: the construction viewpoints, the query viewpoint, 100 tentative pose estimations without synthetic patches, 100 tentative pose estimations with synthetic patches.

are only a few seconds longer when using an enriched model, with a Matlab implementation. The reason is that the SIFT descriptors coming from synthesized patches make the inlier ratio to increase, and consequently the number of RANSAC iterations to decrease. In all our experiments, the tentative correspondences obtained using the enriched model has always a higher inlier ratio than the ones from the initial model, this difference ranging from a 7% increase in CAB to a 37% increase in poster.

We evaluate the accuracy of the estimated poses visually in Figure 6, and numerically in Table 2. To measure pose accuracy we rely on the reprojection error of 3D scene edges in the query view, which can be seen in Figure 7. These edges have been obtained by manually extracting them in the construction views and reconstructing them using multi-view stereo. The accuracy gain ranges from slight to considerable improvements, depending on the relative poses of the query and construction views. In Figure 7, we can see that the reprojected edges are almost superposed when using patch synthesis, which shows the improved accuracy of the pose. In case of strong viewpoint changes between the query and construction views (as in poster and tower), pose estimation simply fails without patch synthesis. Additional information is available in the supplementary file `viewpoint_changes.pdf`.

5 Conclusion

In this paper we proposed a method to add descriptors to a SfM model using patch synthesis, in order to facilitate pose estimation from viewpoints not covered by construction views. This method is not specific to the scene and is still tractable for scenes as large as buildings. Compared to an exhaustive approach as ASIFT [20], the computational burden is limited thanks to two ingredients. First, we add carefully selected virtual viewpoints with respect to the geometry of the scene. Second, we only transform parts of images that can yield useful SIFT descriptors. Experiments show that the proposed algorithms facilitate point matching by reducing the outlier rate, and dramatically increase pose accuracy.

A continuation to this proof-of-concept study could be within tracking initialization in

	poster	book	pot	tower	CAB
SfM time	6min	11min	15min	5min	18min
synthesis time	3min	6min	10min	4min	9min
	SfM model				
# of descriptors	47643	225207	32568	7774	324360
matching time	2.53s	3.15s	2.65s	1.48s	7.55s
pose time	35.2	15.64s	10.17s	35.16s	8.29s
	enriched model				
# of descriptors	664848	887216	134484	85949	1523298
matching time	5.51s	4.60s	4.38s	3.72s	13.76s
pose time	0.06s	0.80s	0.44s	0.38s	1.30s

Table 1: Computing times for synthesis and for the different steps of pose estimation. Matching times are slightly higher when using an enriched model but pose estimation is substantially faster because of the higher inlier ratio in the tentative correspondences.

	poster	book	pot	tower	CAB
SfM model	1175±1.72	3.47±2.31	19.79±26.70	32.92±50.27	26.94±17.23
enriched model	1.21±0.97	2.32±1.26	4.39±4.50	6.72±3.27	15.53±13.72

Table 2: Average pixel reprojection error of 3D scene edges in the query view, plus/minus the standard deviation. The large errors for poster and tower correspond to situation where the RANSAC/PnP step did not actually converge to a reasonable pose, as shown in Figure 6.

augmented reality applications. It would also be interesting to use the information from all available views when synthesizing patches, using, e.g., super-resolution. We also intend to reduce the enriched model size, using a more compact representation in the same spirit as the visual vocabularies proposed in [11] or [12].

References

[1] URL https://cvg.ethz.ch/research/symmetries-in-sfm/.

[2] URL http://www.diegm.uniud.it/fusiello/demo/samantha/.

[3] H. Aanæs, A.L. Dahl, and K.S. Pedersen. Interesting interest points. *International Journal of Computer Vision*, 97(1):18–35, 2012.

[4] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272, 2015.

[5] A. Boulch and R. Marlet. Fast normal estimation for point clouds with sharp features using a robust randomized Hough transform. *Computer Graphics Forum*, 31(5):1765–1774, 2012.

[6] B. Charmette, E. Royer, and F. Chausse. Vision-based robot localization based on the efficient matching of planar features. *Machine Vision and Applications*, 27(4):415–436, 2016.

Figure 7: poster, book, pot, and tower datasets. From left to right: query camera and hand-picked scene edges; reprojection of these edges with 100 pose estimations without synthetic patches; reprojection of these edges with 100 pose estimations when using synthetic patches.

[7] M. Fischler and R. Bolles. 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.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[9] J.A. Hesch and S.I. Roumeliotis. A direct least-squares (DLS) method for PnP. In *Proc. International Conference on Computer Vision (ICCV)*, pages 383–390, 2011.

[10] H. Hoppe, T. DeRose, T. Duchamp, J.McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proc. SIGGRAPH*, volume 26, pages 71–78, 1992.

[11] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2599–2606, 2009.

[12] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[13] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. *ACM Transactions on Graphics (TOG)*, 26(3):24, 2007.

[14] K. Köser and R. Koch. Perspectively invariant normal features. In *Proc. International Conference on Computer Vision (ICCV)*, 2007.

[15] M. Kushnir and I. Shimshoni. Epipolar geometry estimation for urban scenes with repetitive structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2381–2395, 2014.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141:81–93, 2015.

[18] N. Molton, A. J Davison, and I. Reid. Locally planar patch features for real-time structure from motion. In *Proc. British Machine Vision Conference (BMVC)*, 2004.

[19] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3D objects. *International Journal of Computer Vision*, 73(3):263–284, 2007.

[20] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[21] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

[22] P. Rolin, M.-O. Berger, and F. Sur. Simulation de point de vue pour la mise en correspondance et la localisation. *Traitement du Signal*, 32(2-3):169–194, 2015.

[23] P. Rolin, M.-O. Berger, and F. Sur. Viewpoint simulation for camera pose estimation from an unstructured scene model. In *Proc. International Conference on Robotics and Automation (ICRA)*, pages 6320–6327, 2015.

[24] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. International Conference on Robotics and Automation (ICRA)*, 2011.

[25] Q. Shan, C. Wu, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz. Accurate geo-registration by ground-to-aerial image matching. In *Proc. International Conference on 3D Vision (3DV)* vol. 1, pages 525–532, 2014.

[26] G. Simon, A. Fond, and M.-O. Berger. A simple and effective method to detect orthogonal vanishing points in uncalibrated images of man-made environments. In *Proc. Eurographics*, 2016.

[27] G. Simon. Tracking-by-synthesis using point features and pyramidal blurring. In *Proc. International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 85–92. 2011.

[28] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. 24/7 place recognition by view synthesis. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1808–1817, 2015.

[29] C. Wu. VisualSFM: A visual structure from motion system. `http://homes.cs.washington.edu/~ccwu/vsfm/`, 2011.

[30] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.