Viewpoint simulation for camera pose estimation from an unstructured scene model

Pierre Rolin, Marie-Odile Berger, and Frédéric Sur

Abstract— **We consider the problem of camera pose estimation from a scene model obtained beforehand by a structure-from-motion (SfM) algorithm. The model is made of 3D points, each one of them being represented by its coordinates and a set of photometric descriptors such as SIFT, extracted from some of the input images of the SfM stage. Pose estimation is based on the matching of interest points from a test view with model points, using the descriptors. Descriptors having a limited invariance with respect to viewpoint changes, such an approach is likely to fail when the test view is far away from the images used to construct the model. Viewpoint simulation techniques, as ASIFT, have proved effective for wide-baseline image matching. This paper explores how these techniques can enrich a scene model by adding descriptors from simulated views, using either orthographic or pinhole virtual cameras. Viewpoint simulation permits pose estimation in situations where the approach based on the sole SIFT descriptors simply fails.**

I. INTRODUCTION

Pose estimation from a known environment is a problem of uttermost importance in, e.g., pose initialization before image-based refinement [1], re-localization in SLAM in case of tracking failure [2], and, more generally, geo-localization [3], or augmented reality applications [4]. The present paper deals with pose estimation from correspondences between interest points from a new view (called here *test view*) and points from an unstructured 3D scene model as in [2], [4], [5]. Here, the scene model basically consists of a point cloud built beforehand from a set of prior images through a structure-from-motion (SfM) algorithm [6], [7]. Such an algorithm first matches interest points between images, based on photometric descriptors. Chains of matched interest points from several images are then used to simultaneously estimate camera poses and 3D point coordinates using triangulation and bundle adjustment. The descriptors from a chain eventually give a set of image features to any 3D point. Several features are proposed in the literature, among them invariant patches [5] or visual words built upon the photometric descriptors [8], [9]. The present work makes use of the collection of the SIFT descriptors [10], as in [4]. Each 3D point of the model is therefore associated with the list of SIFT descriptors from the corresponding matching chain in the prior images used in SfM. The scene model is assumed small enough so that such an exhaustive representation is still realistic (as in [4]), and does not necessitate a compact representation such as in [9].

Estimating a camera pose from the test view based on the scene model consists in solving the Perspective-n-Points (PnP) problem [11], [12], [13] for a set of point correspondences between the test view and the scene model. This approach is limited by the invariance of the photometric descriptors, which is known to be typically limited to a 30° orientation change [14]. If the test view shows a too strong viewpoint change with respect to the prior images, SIFT matching is not reliable and too few consistent matches can be built to solve the PnP problem.

A. Proposed contribution

The goal of this work is to enrich the set of features associated with the 3D points by generating additional SIFT descriptors extracted from simulated viewpoints far away from the available actual ones. As a consequence, the invariance range of the description of the 3D points is enlarged, and matching (hence pose estimation) is facilitated when the scene presents a strong aspect change in the test view. Viewpoint simulation has proved to be efficient for matching interest points between two images under a strong viewpoint change in ASIFT [15] and methods derived from it [16], [17], or, to a lesser extent, FERNS [18]. In these approaches, viewpoint simulation is performed under affine transformations. In our case, assuming the scene to be locally planar, all views of a region surrounding a 3D point are linked by homographies in the pinhole model, or by affine transformations in the orthographic model. The additional descriptors will thus be generated from simulated views following one of these two models, in order to mimic a motion of the camera in positions not present in the prior images. This is illustrated in Figures 1 and 2. Note that a similar approach is developed in [19] or [20], but the simulated views are fronto-parallel only. In [21], viewpoint simulation is also envisaged in order to improve object recognition, and in [9] for the localization of a camera in a large environment. In these approaches, the simulated descriptors are embedded in a quantized visual vocabulary potentially affected by information loss (as pointed out in [22]). To the best of our knowledge, a dedicated study focused on the benefit of viewpoint simulation for pose estimation is still missing.

B. Paper organization

Section II explains viewpoint simulation using affine transformation or homographies. Section III gives the implementation details: how the 3D scene model is enriched with the generated descriptors, and how correspondences with the test view are built. Experiments are discussed in Section IV and Section V concludes.

The authors are with LORIA UMR 7503, Université de Lorraine, CNRS, INRIA project-team Magrit; Nancy, France.

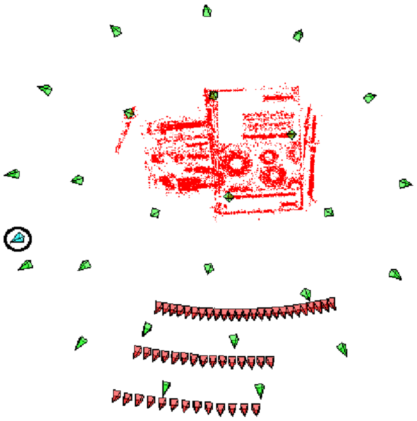
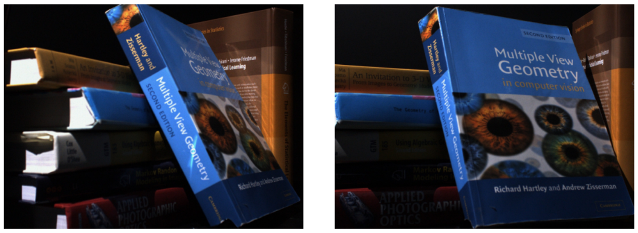


Fig. 1. The 3D model of the scene (red points), the camera pose corresponding to the prior images used as an input of SfM (in light red), a far-away camera whose pose is sought (in cyan, circled), and the virtual cameras (in green), here lying on a hemisphere and pointing to the barycenter of the scene. Virtual cameras permit to generate new SIFT descriptors for each 3D point of the model.



(a) (b)

Fig. 2. The view whose pose is sought (a) (in cyan in Figure 1) and the nearest view among the ones used to build the model (b). Note the strong viewpoint change, making image feature matching impracticable.

II. VIEWPOINT SIMULATION FOR A LOCALLY PLANAR SCENE

We assume that the scene is locally planar, and that it is possible to associate a normal vector with each 3D point of the model (giving a so-called *surfel*). The question addressed in this section is: a real view of a planar region around a 3D point being given, how to simulate a view from a novel camera position, in order to extract a new SIFT descriptor for this 3D point?

Within the pinhole camera model, two views of a plane are linked by a homography. Within the simplified affine (or orthographic) camera model, the two views are linked by an affine transformation. It is proven in [15], [18] that this simplification is often sufficient for wide-baseline matching with SIFT. Indeed, as an affine transformation is a first-order approximation of a homography, affine or homographic transformations of a small image patch are nearly indiscernible. However, SIFT descriptors are often extracted from circular patches with a diameter of several tens of pixels, for which the affine approximation is no more valid as soon as the view angle differs too much (typically beyond 30°).

A. Homographies

We consider two cameras represented by their projection matrices $P_1 = K_1[R_1|T_1]$ and $P_2 = K_2[R_2|T_2]$, where, for $i \in \{1,2\}$, R_i, T_i are the rotation and the translation pose components in a common coordinate system, and K_i is the matrix of the intrinsic parameters (square sensor pixels),

$$K_i = \begin{pmatrix} f_i & 0 & \alpha_i \\ 0 & f_i & \beta_i \\ 0 & 0 & 1 \end{pmatrix} \qquad (1)$$

with f_i the focal length and (α_i, β_i) represents the principal point. The transformation that maps the points from a plane of equation $n^T X + d = 0$ (where n is a normal vector) is the homography given by the homogeneous equation [23]:

$$H = K_2(R - Tn^T/d)K_1^{-1} \qquad (2)$$

where $R = R_2 R_1^T$ and $T = -R_2(C_2 - C_1)$ (the optical center C_i satisfying $C_i = -R_i^T T_i$, $i \in \{1,2\}$).

When the two cameras share the same optical axis, colinear to n, this homography reduces to a similarity.

If P_1 is the projection matrix of a real camera, and P_2 of a virtual camera, and I_1 and I_2 are the images of the plane through both cameras, then $HI_1 = I_2$, i.e.,

$$K_2 R_2 (R_1^T + (C_2 - C_1)n^T/d))K_1^{-1} I_1 = I_2. \qquad (3)$$

The rotation R_2 writes $R_2 = R_Z(\kappa)R_Y(\phi)R_X(\omega)$ where (X, Y, Z) is an orthonormal coordinate system such that Z is aligned with the optical axis of the camera and (κ, ϕ, ω) are the associated Euler angles. SIFT descriptors being invariant through planar similarities, any rotation around the optical center or any focal change in camera 2 (which, from (1), amounts to changing the scale in I_2) gives the same descriptors. Therefore, the pose of any virtual camera only needs to be fixed up to a rotation around its optical axis and the focal length is arbitrary. Of course, this is valid apart from image discretization issues and under a perfect, hypothetical, invariance of SIFT to similarities. That being said, the distance of the virtual camera to the scene is still important, since T_2 is present in (3).

A plane, a real camera pose, a virtual camera pose up to a rotation along the optical axis being given, it is possible to simulate with equation (3) a view from which we extract SIFT descriptors.

B. Affine transformations

With orthographic / affine cameras and with the same notations as in Figure 3, let $(\lambda_i, \psi_i, t_i, \phi_i)$ be the characteristic elements of the camera $i \in \{1,2\}$ in a coordinate system associated with the plane given by its normal vector n. The transformation mapping a fronto-parallel view of this plane and the view through camera i is given by the affine transformation [15]:

$$A_i = \lambda_i \begin{pmatrix} \cos(\psi_i) & -\sin(\psi_i) \\ \sin(\psi_i) & \cos(\psi_i) \end{pmatrix} \begin{pmatrix} t_i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{pmatrix}. \qquad (4)$$

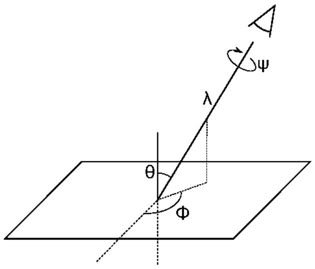


Fig. 3. Position of an affine camera with respect to the normal of a piece of plane, with the notation of equation (4) where $t = 1/\cos(\theta)$ as in [15].

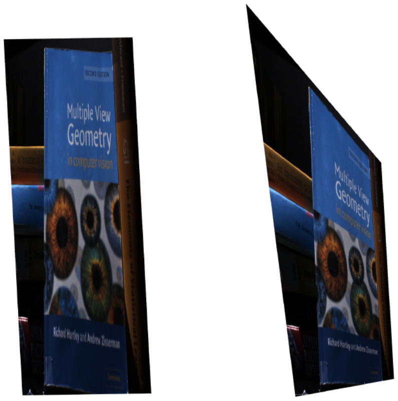


Fig. 4. A simulation example. These simulated views correspond to a view of the book cover which would be obtained from the camera whose pose is sought (in cyan in Figure 1), simulated from the far right real image. On the left: simulation with an affine transformation. On the right: simulation with a homography. Up to a slight rotation, the homography simulation of the cover is more like the actual view (in Figure 2 a).

The affine transformation induced by the plane between the two cameras is hence:

$$A = A_2 A_1^{-1}. \tag{5}$$

With the same notations as in the homography case, $AI_1 = I_2$, i.e., $A_1^{-1}I_1 = A_2^{-1}I_2$. Since SIFT descriptors are supposed invariant to similarities, any $\psi_1, \psi_2, \lambda_1, \lambda_2$ gives the same SIFT descriptors. It is hence possible to arbitrarily choose $\psi_1 = \psi_2 = 0$ and $\lambda_1 = \lambda_2 = 1$.

Consequently, the relative positions of real and virtual cameras, given by the (t_i, ϕ_i) defined by a planar part of the scene, make it possible to simulate a view through equation (5), from which SIFT descriptors are extracted.

C. *Summary*

For any surfel (3D point and normal vector), and for any virtual camera pose, a view is simulated (either with a homography or an affine transformation, depending on the method of interest), then a SIFT descriptor is extracted in this view and associated with the 3D point. A simulation example can be seen in Figure 4.

III. Practical Implementation

This section explains how the general principles of Section II are implemented. Some perspectives are also mentioned. A point model is built and its points associated with a set of SIFT descriptors and with a normal vector (Section III-A), then descriptors associated with simulated views are added (Section III-B). The pose of a test view is eventually estimated from this enriched scene model (Section III-C).

A. *Building the unstructured scene model*

The VisualSFM software [7] is used to generate a set of 3D points from prior images of the scene. Each point is associated to the set of SIFT descriptors from the chain of matched features used to feed the bundle adjustment step, called here the *class* of descriptors associated to a 3D point. VisualSFM also gives a dense reconstruction based on [24]. This dense model is used to generate at any 3D point an estimation of the local normal to the scene. The method is to keep the vector associated with the smallest eigen-value in a principal component analysis of the coordinates of its k nearest neighbours [25] (k between 50 and 100 according to the model density). The normal is oriented towards the cameras in which the point is seen.

B. *Enriching 3D point features from simulated views*

1) Position of the virtual cameras: The position of the virtual cameras is chosen to complete the available viewpoints (the real cameras). As mentioned in Section II, the affine case does not require to fix the distance of the camera to the scene. On the contrary, the homography case requires to also give this distance to properly define I_2 with (3).

In this paper, we place the virtual cameras in the same position in both homography and affine cases. We consider twenty-five positions regularly distributed on a hemisphere lying on a dominant plane (deduced from a PCA on the whole scene), its radius being equal to the distance of the nearest camera to the barycenter of the scene. Cameras are oriented towards the barycenter of the scene, as illustrated in Figure 1. If this strategy covers all viewpoints in the affine model (up to the discretization of the set of views), this does not simulate cameras getting closer or more far away from the scene in the homography case. Nevertheless, from the remark following (2), the invariance of SIFT to similarities embeds potential displacements along the optical axes of the cameras, for those planes of the scene orthogonal to these axes.

Positioning the virtual cameras should actually depend on the local geometry of the scene in order to cover the viewpoints of most surfels. This is still an open issue which is not within the scope of this work.

2) Choosing a real view for simulation and generating a SIFT descriptor: A 3D point being given together with SIFT descriptors from real views, a strategy has to be given to choose one of these views to generate a descriptor corresponding to a virtual view. (i.e., with the notations of Section II, to generate I_2 from I_1.) Here, the real view which has the closest angular distance to the virtual view is chosen. Of course, other strategies can be envisaged. In particular choosing the most fronto-parallel camera would reduce discretization effects. Each strategy have its pros and cons; the discussion is left for future works.

Simulation gives a patch centered on the 3D point projection in the virtual view. The SIFT algorithm then gives keypoints and associated descriptors in this patch. The descriptor of the nearest keypoint in the simulated patch to the theoretical projection of the considered 3D point is added to the list of the descriptors of the 3D points, provided the distance is below 10 pixels (which is a rough estimate of the reprojection distance in the SfM stage). This threshold is useful to get rid of cases where no SIFT keypoint is extracted at the expected position. A quite large threshold is needed since Gaussian scale-space does not commute with non-similarity transformations, hence the keypoint of interest cannot rigorously match the projection of the 3D point.

C. Estimating the camera pose

1) Image / model correspondences: A new view being given, SIFTs are first extracted. Second, any SIFT keypoint is associated to a 3D point if the ratio of the distances between the descriptor and the two nearest classes in the 3D model is below a threshold (0.6 in practice). Approximated nearest neighbor [26] speeds up the search. This is the same algorithm as in [4].

2) Perspective-n-Points: Pose estimation is performed through a robust estimation via RANSAC [27] based on the PnP algorithm proposed in [28]. We assume that intrinsic parameters are known, which is the case if the same camera as in the SfM stage is used. Of course, the smaller the outlier rate in the preceding step, the smaller the number of iterations in RANSAC.

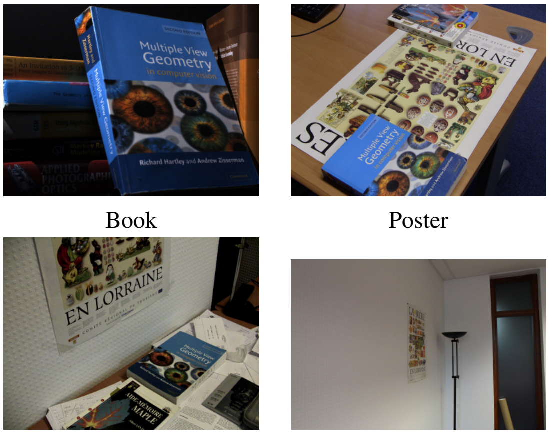
IV. EXPERIMENTS

The following experiments prove that, under wide differences of view direction or under strong depth variation, viewpoint simulation dramatically improves pose estimation. The pose can be estimated in situations where the standard algorithm (without simulation) fails. Generally speaking, for a fixed number N of RANSAC iterations, the pose is more accurately estimated than without simulation. Afterwards, we discuss computation times and potential improvements.

In the figures, the cameras giving the prior images for SfM are in red, virtual cameras are in green, test viewpoints are in cyan and computed poses are in blue.

A. Experimental setup

We evaluate the proposed method on four datasets : a series of images from the publicly available Robot Dataset [29] (the result of the SfM stage is presented in Figure 1) and three personal datasets, as depicted in Figure 5. These datasets feature 1600 × 1200 images and present mostly piecewise planar, object-centered scenes. All experiments are within the same setup. A 3D model of the scene is built with VisualSfM (Section III-A). The pose of a test view is estimated (Section III-C) under different scenarios: **S** where the model is the SfM reconstruction without simulation, **A** where the model of **S** is enriched with the additional descriptors from affine simulations, and **H** where the model of **S** is enriched with homography simulation (Section III-B).



Book

Poster

Desk

Wall

Fig. 5. Representative images from the four datasets. Book is from [29].

To compare the three scenarios, 100 poses are computed from the same test view in each case using the same number of RANSAC iterations for every run. The variability of these 100 poses are then visually compared. It is expected that all these poses are superposed. In this case, we also compute the standard deviation (see caption). In the Book dataset, a ground truth is available and the actual test pose is known.

As the inlier ratio among image/model correspondences is very different from a dataset to another (e.g., from 4% to 23% in scenario **S**), we consider different numbers of RANSAC iterations for each of the datasets. However, to make variability comparison easier, the same number of iterations is used for the three scenarios. As we shall see, simulation-based scenarios give a smaller pose variability than scenario **S**. This means that enriching the model with simulated features makes RANSAC to converge faster, i.e., the inlier ratio to increase.

B. Pose improvement using simulation-based models

1) Robustness of pose estimation to viewpoint changes: Viewpoint simulation is shown to significantly increase pose accuracy when the test view is taken far away from the SfM views, giving a strong aspect change of the scene.

We first present experiments with the Book dataset (Figure 1), where the actual test pose is available. It is hence possible to determine if a 2D/3D match is correct or not, by projecting the considered 3D point using the ground truth pose: If the reprojection distance is below 20 pixels the match is considered as correct (this threshold corresponds to $\mu+3\sigma$ where μ and σ are respectively the mean and the standard deviation of the SfM reprojection error). In this experiment the inlier ratio is found to be 23% in scenario **S**, 30% in **A**, and 37% in scenario **H**.

Figure 7 shows the repartition of the 2D/3D matches along the simulated and actual views in scenario **H**. The viewpoint that contributes the most to pose computation is here a virtual

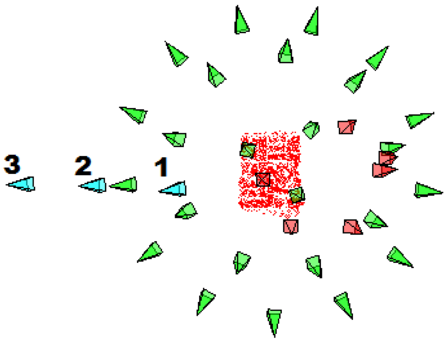


Fig. 6. Wall dataset : positions of the reconstruction cameras (red), virtual cameras (green) and test viewpoints (cyan) No. 1, 2, 3.

one, near the test camera. Overall, virtual viewpoints produce 85% of the RANSAC consensus set. These graphics illustrate the soundness of the proposed approach and the increased inlier ratio obtained through simulation.

Pose estimation results are illustrated in Figures 8 ($N = 500$) and 9 ($N = 1,000$). The estimated poses are visually more accurate in **A** or **H** than in **S**. With 500 RANSAC iterations, pose computation fails in **S**, and are superposed, hence correct, in **H**. Increasing the number of iterations to 1,000, pose variability is only marginally improved in **S**, whereas the poses cannot be distinguished in **H**. A remarkable phenomenon can be noted in **A** (it can also be seen in **S**). In this experiment, the calculated poses can be classified in three categories: most poses are close to the expected viewpoint, some poses are completely wrong, and a cluster of poses can be seen in front of the book cover. This cluster is actually caused by a repeated pattern in the scene, namely the eye on the book cover that also appears on the book side (see Figure 2). In this case, homography-based simulation gives additional correspondences outside this repeated pattern, which makes it possible to obtain a consistent pose in **H**. The influence of repeated pattern in pose estimation is discussed in, e.g., [17], [30], [31].

Simulation has been tested on the Poster and Desk datasets with similar results, see Figures 10 and 11.

2) Robustness of pose estimation to variations of the distance to the scene: As explained in Section II, viewpoint simulation in the affine camera model is independent from the distance of the virtual cameras to the scene. Although simulation using homography should depend on this distance, in this paper we use the same position for the virtual cameras in this case. The goal of the following experiment is to assess the influence of this tentative choice on pose computation when the test camera is significantly far away from the virtual and real cameras. A scene model is built from 6 cameras pointing to a poster (in red in Figure 6). This particular planar scene is chosen to highlight the potential benefit of simulation. We need a test camera not aligned with the optical axis of the cameras and such that the poster is not fronto-parallel. Indeed, we want the underlying transformation to be a homography not degenerated to a similarity (thus not factored out by SIFT invariance, as discussed in Section II-A).

TABLE I

FOR EACH DATASET: NUMBER OF PRIOR IMAGES (1), NUMBER OF 3D POINTS IN THE SFM MODEL (2), NUMBER OF DESCRIPTORS IN SCENARIO **S/A/H** (3), COMPUTATION TIME FOR IMAGE/MODEL MATCHING (4), NUMBER OF CORRESPONDENCES (5) AND COMPUTATION TIME FOR THE PNP STEP (NORMALIZED TO 1,000 RANSAC ITERATIONS) (6). COMPUTATION TIMES ARE IN SECONDS.

	book	poster
1	53	17
2	15,269	7,552
3	225,207 / 403,662 / 386,970	47,643 / 161,596 / 224,923
4	76.7 / 82.4 / 81.4	70.2 / 99.5 / 120.8
5	1,272 / 809 / 1,097	1,144 / 1,293 / 1,092
6	5.2 / 5.1 / 5.3	4.9 / 4.7 / 4.8
	desk	wall
1	17	6
2	3,525	2,527
3	15,109 / 33,396 / 45,393	10,765 / 59,325 / 61,690
4	11.0 / 16.9 / 22.3	3.0 / 10.2 / 10.2
5	892 / 779 / 657	322 / 338 / 266
6	5.5 / 4.9 / 5.1	5.2 / 5.1 / 5.7

The test views are taken with a moderate change in view direction but strong depth variations, see Figures 6 and 12. The number of RANSAC iterations is $N = 300$ for all these experiments. We only detail scenarios **S** and **H**, scenario **A** giving the same results as **S** (not shown because of space limitation). The reason is that affine simulation does not produce descriptors consistent with the actual transformation in the case of such an oblique displacement.

Figure 13 concerns **S**. We can see that an accurate pose estimation is possible only if the test view is not too far away from the virtual and actual cameras, as for test views 1 and 2. However the accuracy significantly decreases for test view 3. The results for **H** are in Figure 14. We can see that accurate poses can be estimated in the three cases, the estimated camera poses being visually superposed.

C. *Computation time and future improvements*

Table I gives computation times for each dataset. The code runs on Matlab on an Intel core i7 without particular optimizations. Computation times are reasonable for a prototype implementation; however, some heuristic speed-ups are under investigation. We give a few perspectives here.

Contrary to the instinctive thought, naively pruning the 3D model is not an appropriate approach. In this case the nearest-neighbour-based matching is indeed impaired by numerous erroneous correspondences due to keypoints from the test image which do not exist anymore in the model.

The 2D/3D matching is time consuming because of the large amount of descriptors in the model. We could use a more compact class representation, as proposed in [9]. Our experiments suggest that using a representative element for each class could be possible, in spite of the limitations of such an approach discussed in [22]. The representative element we use is the medoid of the class, which is the element of the class that minimizes its average distance to the other elements. Using this matching technique in the Robot dataset, the 2D/3D inlier ratio changes from 23% to 22% in

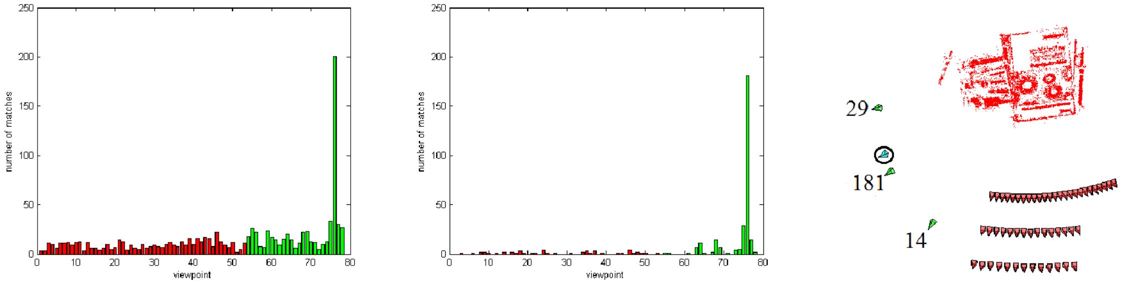


Fig. 7. Book dataset : number of matches associated with each viewpoint (real in red, simulated in green), for all 2D/3D matches (left) and in the RANSAC consensus set (middle). The top-contributing viewpoints remain the same, and correspond to viewpoints close to the actual pose (the three top-contributing views can be seen on the right).

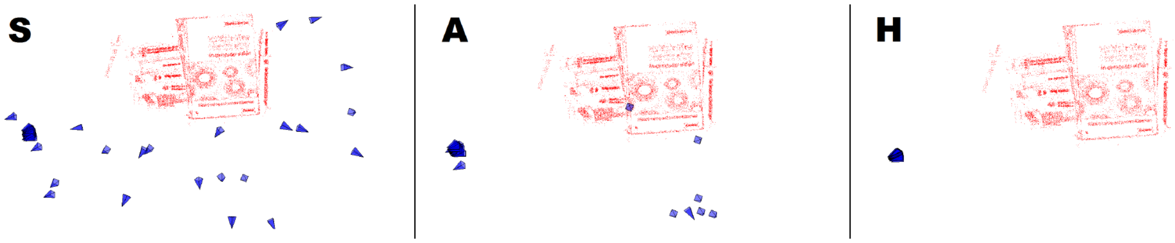


Fig. 8. Book dataset : 100 poses computed with $N = 500$ RANSAC iterations. In **H**, the standard deviation is 0.31% of the distance to the scene.

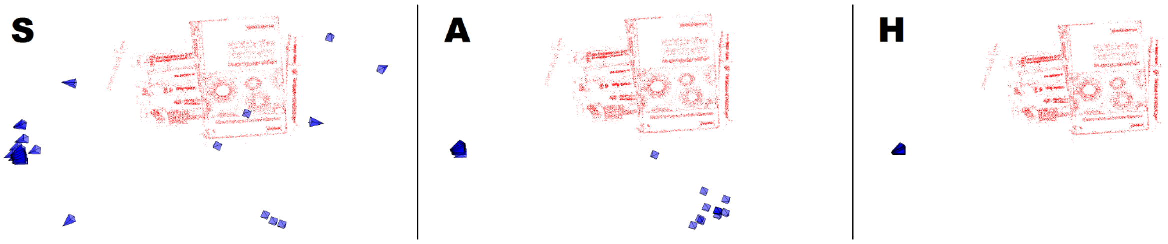


Fig. 9. Book dataset : 100 poses computed with $N = 1,000$ RANSAC iterations. In **H**, the standard deviation is 0.29% of the distance to the scene.

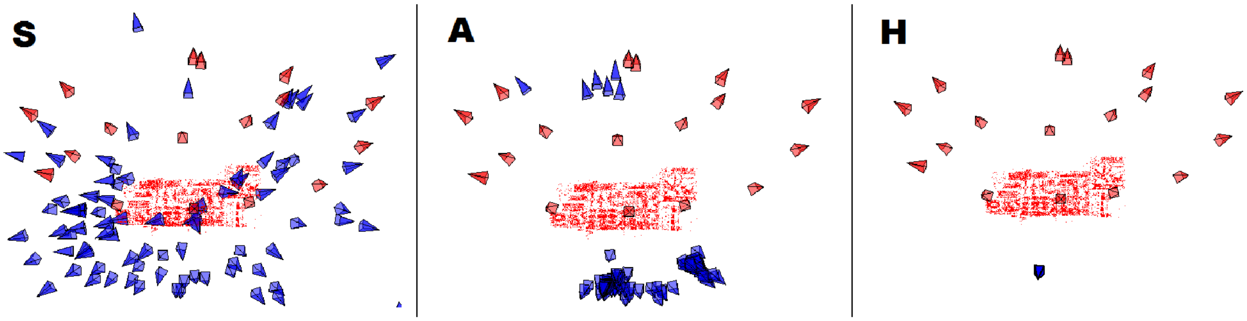


Fig. 10. Poster dataset : 100 poses computed with $N = 1,000$ RANSAC iterations. In **H**, the standard deviation is 0.07% of the distance to the scene.

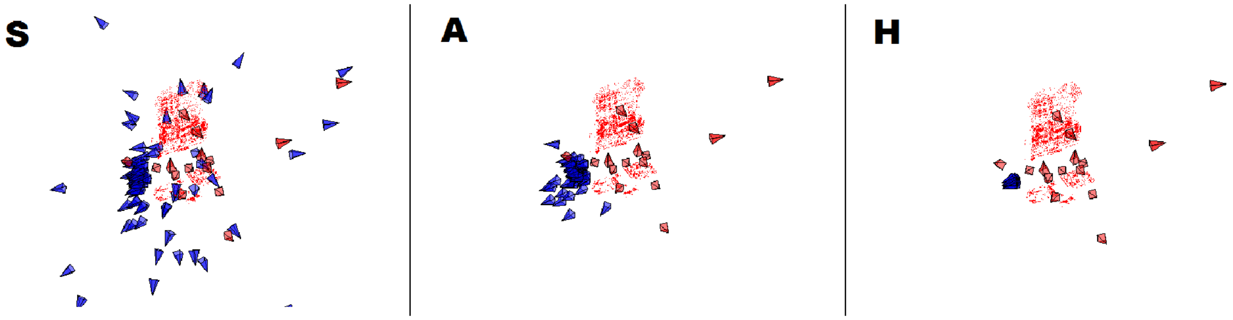


Fig. 11. Desk dataset : 100 poses computed with $N = 5,000$ RANSAC iterations. In **H**, the standard deviation is 3.04% of the distance to the scene

scenario **S**, 30% to 28% in scenario **A** and 37% to 35% in scenario **H**. Although these results are encouraging, choosing a unique representative element is paradoxical, since our approach consists in enriching descriptor classes. Reducing classes to a small number of representative elements is however to be further investigated.

Let us finish with a speed-up for RANSAC. Figure 7 is a representative example of the distribution of the 2D/3D correspondences along the virtual and actual views. It turns out that only a few views (either virtual or real) significantly contribute to the set of correspondences, and among these views the inlier ratio is quite high. Viewpoints close to the sought one are more likely to produce correct correspondences than the others. A straightforward speed-up for the RANSAC stage is thus to impose a prior distribution which favours correspondences from the images giving the largest numbers of image/model correspondences. This amounts to drawing correspondences from subsets with a higher inlier rate, as in PROSAC [32]. Although further investigations are needed, early tests indicate that this strategy actually improves RANSAC convergence speed.

V. CONCLUSION

This paper discusses viewpoint simulation to enrich an unstructured scene used in a pose estimation application. It presents both the theoretical model and an experimental setup. Although the present study is limited to relatively small scenes, it permits us to gain several insights. First, viewpoint simulation actually makes it possible to estimate a pose in situations where the standard SIFT-based matching simply fails, either because of a strong difference in the view angle, or in the distance to the scene. Second, viewpoint simulation also gives a more reliable pose when the standard approach would need a large number of RANSAC iterations. The homography model performs significantly better than the affine one : it produces sets of 2D/3D correspondences with higher inlier ratio and bigger RANSAC consensus sets, the two models requiring a similar computation time.

Further works are also needed in the definition of the 2D/3D matching, since the pose accuracy is not directly linked to the number of prior correspondences but also depends on their distribution in the scene. A heuristic criterion for two-view SfM is proposed in [33]. It would be interesting to extend such a criterion to the problem of interest.

REFERENCES

[1] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Proc. ICRA*, 2009, pp. 48–55.

[2] B. Williams, G. Klein, and I. Reid, "Real-time SLAM relocalisation," in *Proc. ICCV*, 2007.

[3] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Proc. CVPR*, 2007.

[4] I. Gordon and D. Lowe, "What and where: 3D object recognition with accurate pose," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds. Springer, 2006, vol. 4170, pp. 67–82.

[5] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[6] C. Wu, S. Agarwal, B. Curless, and S. Seitz, "Multicore bundle adjustment," in *Proc. CVPR*, 2011, pp. 3057–3064.

[7] C. Wu, "VisualSFM: A visual structure from motion system," http://ccwu.me/vsfm/, 2011.

[8] S. Bhat, M.-O. Berger, and F. Sur, "Visual words for 3D reconstruction and pose computation," in *Proc. 3DIMPVT*, 2011, pp. 326–333.

[9] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. CVPR*, 2009, pp. 2599–2606.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[12] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, 2005.

[13] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate $O(n)$ Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[14] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[15] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009.

[16] D. Mishkin, M. Perd́och, and J. Matas, "Two-view matching with view synthesis revisited," in *Proc. IVCNZ*, 2013, pp. 448–453.

[17] N. Noury, F. Sur, and M.-O. Berger, "How to overcome perceptual aliasing in ASIFT?" in *Proc. ISVC, Part. 1*, 2010, pp. 231–242.

[18] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, 2010.

[19] M. Kushnir and I. Shimshoni, "Epipolar geometry estimation for urban scenes with repetitive structures," in *Proc. ACCV*, 2012, pp. 163–176.

[20] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys, "3D model matching with viewpoint-invariant patches (VIP)," *Proc. CVPR*, 2008.

[21] E. Hsiao, A. Collet, and M. Hebert, "Making specific features less discriminative to improve point-based 3D object recognition," in *Proc. CVPR*, 2010, pp. 2653–2660.

[22] O. Boiman, E. Shechtman, and M. Irani, "In defense of Nearest-Neighbor based image classification," in *Proc. CVPR*, 2008.

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

[24] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[25] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Computer Graphics (SIGGRAPH '92 Proc.)*, vol. 26, 1992, pp. 71–78.

[26] D. Mount and S. Arya, "ANN: A library for approximate nearest neighbor searching," http://www.cs.umd.edu/%7Emount/ANN/, 2010.

[27] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[28] J. Hesch and S. Roumeliotis, "A direct least-squares (dls) method for pnp," in *Proc. ICCV*, Barcelona, Spain, 2011, pp. 383–390.

[29] H. Aanæs, A. Dahl, and K. Steenstrup Pedersen, "Interesting interest points," *International Journal of Computer Vision*, vol. 97, no. 1, pp. 18–35, 2012.

[30] F. Sur, N. Noury, and M.-O. Berger, "An a contrario model for matching interest points under geometric and photometric constraints," *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 1956–1978, 2013.

[31] R. Roberts, S. Sinha, R. Szeliski, and D. Steedly, "Structure from motion for scenes with large duplicate structures," in *Proc. CVPR*, 2011, pp. 3137–3144.

[32] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *Proc. CVPR*, vol. 1, 2005, pp. 220–226.

[33] Z. Liu, P. Monasse, and R. Marlet, "Match selection and refinement for highly accurate two-view structure from motion," in *Proc. ECCV*, 2014, pp. 818–833.

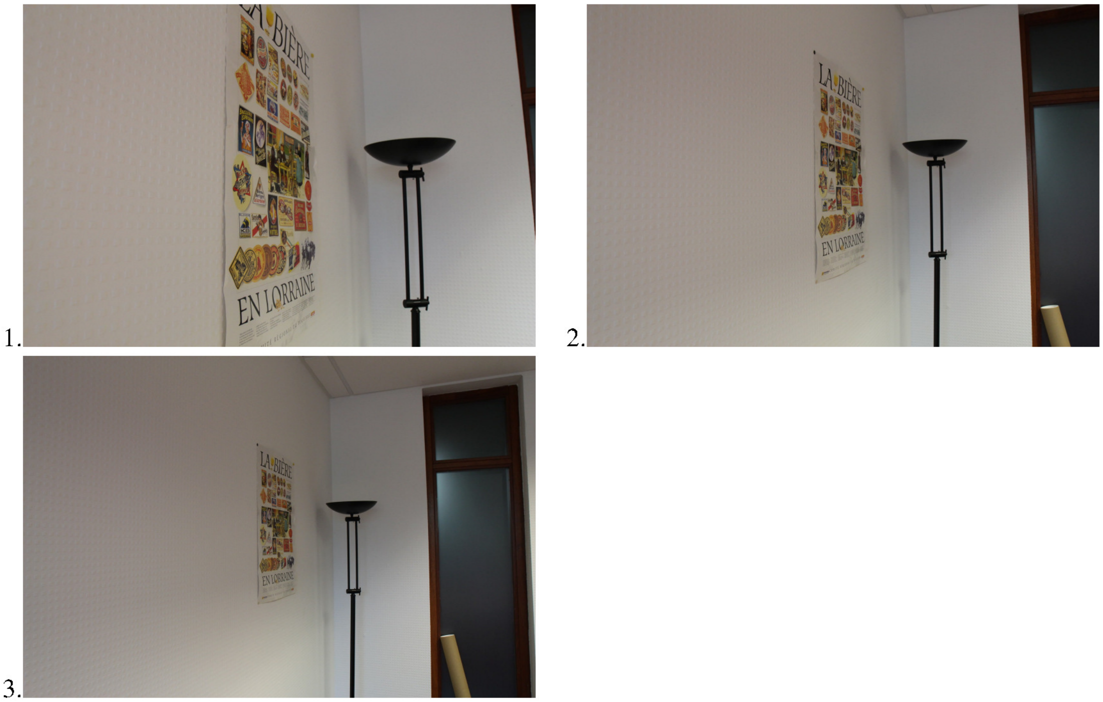


Fig. 12. Wall dataset : test views for robustness with respect to the distance to the scene, increasing from 1 to 3.

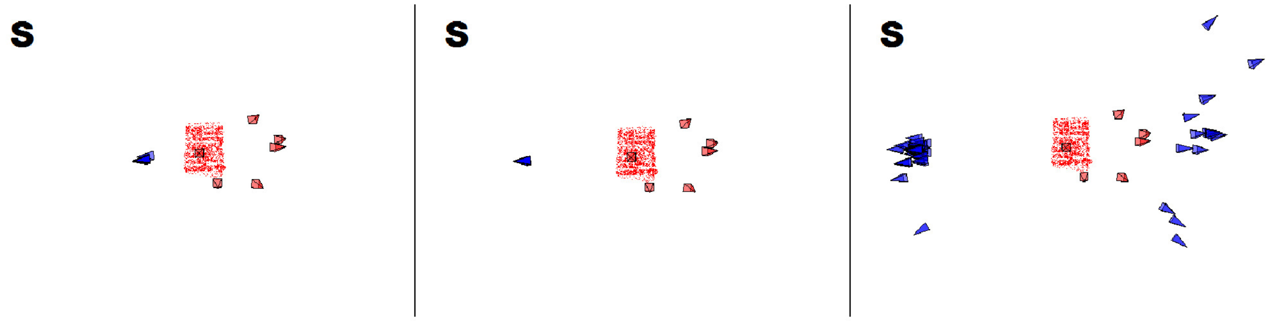


Fig. 13. Wall dataset : 100 pose computations with $N = 300$ RANSAC iterations for the three test views (see 12) in scenario **S**. From left to right: Test views 1 to 3. The standard deviation is 2.14% of the distance to the scene in view 1 and 0.12% in view 2.

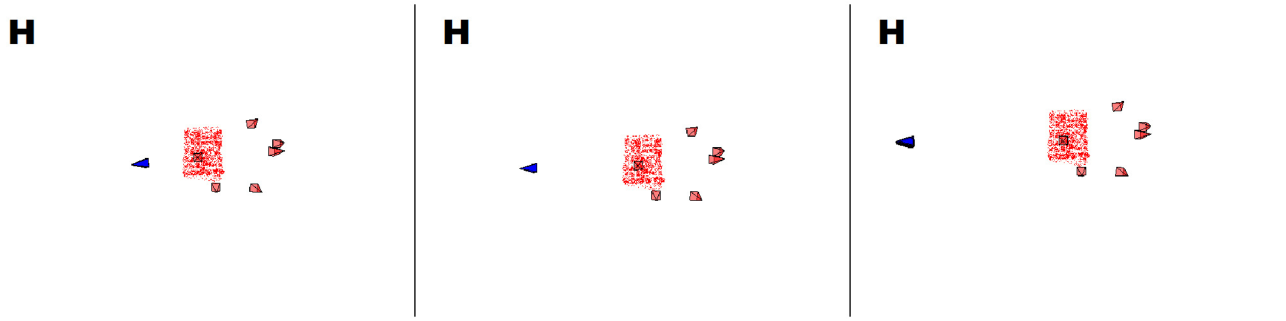


Fig. 14. Wall dataset : 100 pose computations with $N = 300$ RANSAC iterations for the three test views (see 12) in scenario **H**. From left to right: Test views 1 to 3. The standard deviation is 0.07% of the distance to the scene in view 1, 0.02% in view 2 and 0.28% in view 3.