Guided Capturing of Multi-view Stereo Datasets

Fabian Langguth and Michael Goesele

TU Darmstadt

Abstract

We present an application for mobile devices, that allows any user, even without background in computer vision, to capture a complete set of images, that is suitable for a multi-view stereo reconstruction. Compared to related tasks, such as panorama capture, this setting is much harder, as the camera needs to move unrestricted in 3D space. Our system uses structure from motion to register captured images and generates a sparse reconstruction of the scene. The dataset is built in an incremental procedure, where the next best view is computed with a novel view planning strategy, that aims for a good coverage of the scene. The user is then guided towards the new view, and the image is captured automatically at the right position. The next iteration starts after the reconstruction has been updated. The quality of the resulting dataset is on par with datasets captured by an expert user.

Categories and Subject Descriptors (according to ACM CCS): I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—Scanning

1. Introduction

With the constant development of computer graphics 3D content is becoming an essential part of visual applications. Today, even small mobile devices can render complex models. Nevertheless, an average user rarely experiences this in an everyday environment. While online platforms such as YouTube or Flickr are a great way to view, explore, and share videos or photos, there exists no such platform for 3D content. One limiting factor is probably the actual creation of appropriate models. While it is trivial to capture images, panoramas, or videos with mobile devices, there is no easy way to also capture 3D data. Some applications such as Autodesk's 123D Catch are pushing casual creation of content via image-based multi-view reconstruction. But their capture method is not really comfortable: Since the reconstruction system is purely cloud-based, all images must be uploaded to a central server, and a final model is then delivered some time later. The user therefore lacks immediate feedback about the quality of the dataset. If the images are too loosely connected, the reconstruction may fail. [HKR*12] show, that direct feedback about the reconstruction can improve the capturing process even for experienced users. However, they rely on a nearby workstation to perform all computations and expect the user to capture the right images manually. We want to provide a convenient way of mobile capturing for average users that is independent of a network connection, and does not require complex interactions. In an ideal scenario one would just move the device around an object while good images are captured automatically.

Elaborate applications that actively support the capturing process already exist for special purposes such as panoramas. In our setting, however, the camera moves freely in 3D space and does not just rotate around its center. Therefore the registration of captured images, as well as user guidance is more complicated. Recent approaches such as [KM09] use camera phones for localization and mapping in the context of augmented reality. Moving one step further, we present an application that registers all captured images, creates a sparse reconstruction, and uses this data to estimate new views and to guide the capturing process. The whole system is automated and the only user interaction needed is to actually move the device.

2. Related work

Image registration techniques, that operate in realtime and also provide an approximate reconstruction of the scene, are mostly based on a simultaneous localization and mapping (SLAM) approach [LDW91, DRMS07, KM09]. These methods are built for frame-to-frame video tracking, do not necessarily provide a globally optimal model for wide baseline view changes, and need special techniques for loop clos-

ing [New05]. In contrast, we require a certain reconstruction quality as we also want to compute new viewpoints, that explore the capture target. Therefore, we use incremental structure from motion (SfM) similar to [SSS06], which provides an accurate global registration and reconstruction. This also allows us to capture high quality still images (instead of a video stream), which are more suitable for multi-view stereo algorithms [FP10, HKLP09, GSC*07]. Previous works on next best view estimation are often restricted to a specific setting such as polyhedral objects [WDAN07]. Other methods require a complete 3D reconstruction at each iteration [DF09], or a specific lab setting without occlusions [TMD10]. We show a new technique, that works with any textured object, and relies only on the sparse SfM output for view planning. [BAD10] already explored how to guide a user in the context of rephotography. We solve a similar problem with the key difference, that we do not know what the new image actually looks like.

3. Guided capture

We propose a complete system for multi-view capture, designed to run on current smartphones, specifically the iPhone 5. To capture a dataset the user first acquires 6 images, which are used to initialize the SfM reconstruction. We then estimate a new view that extends the reconstruction and assist the user in capturing this view. The reconstruction is then updated and the next iteration starts with the view planning.

Structure from motion Our SfM approach is based on typical algorithms with some modifications to speed up computation on mobile devices. First, we search for point correspondences between the images using ORB image features [RRKB11], as they currently provide the best runtime performance. ORB extracts binary image features, that can be matched efficiently with multi-probe locality-sensitive hashing [LJW*07]. Once correspondences are established, we compute the camera projection matrices and the 3D position of all corresponding features using standard techniques. To establish a globally optimal model we run a nonlinear least squares optimization [AM12] over all parameters.

View planning Previous view planning approaches often work in a laboratory environment, which is not as runtime focused as our application. We propose a technique, that computes an approximate next best view (NBV) in less than a second. As we cannot build a view planning for general scenes, we restrict our capture setting to one or a few objects, that are captured by moving the camera around the scene.

We approximate the scene by discretizing its volume into a voxel grid. The purpose of this grid is to identify space, that has already been observed and does not need be seen by another view again. Every NBV is intended to observe as much unknown volume as possible. We thus ensure, that by adding new views we capture additional scene geometry. To

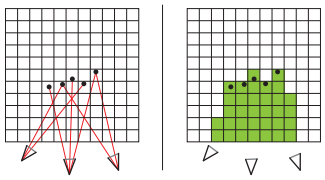

Figure 1: *Left*: rays are traced from every camera, to their corresponding 3D points. *Right*: all *observed* voxels are marked green.

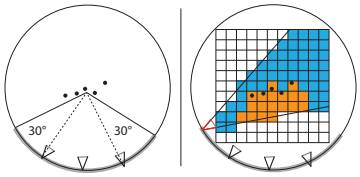

Figure 2: *Left*: A new view lies on the sphere and has to be inside the bounds. The resulting search region is marked gray. *Right*: The NBV (red) is chosen to maximize potential new volume $U(v)$ (blue) and to minimize already observed volume $O(v)$ (orange)

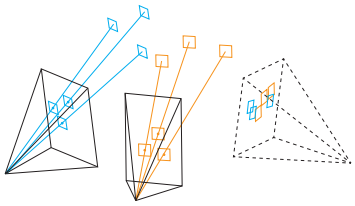

Figure 3: Generation of the preview image corresponding to the NBV. Patches around reconstructed features in neighboring views are projected into the new image (dashed outline) according to the NBV's parameters.

build the volume we use the 3D SfM points. We select only points, that are close to the center of mass (in terms of standard deviation), and assume that this point cloud resembles at least parts of the object. We build our volume around the points as an axis-aligned voxel grid with N voxels in each dimension. For further processing we found $N = 13$ to give a good tradeoff between runtime and accuracy. As the scene will extend over time we increase the volume by a factor of 1.5 in all dimensions. Next, we determine which parts of this volume are already covered by the views captured so far. The idea here is, that if a 3D point was reconstructed from a set of cameras, they have seen the point and therefore all space between the camera and this point must be empty. The camera consequently has observed all voxels, that lie along the ray from the camera to the particular point, at least partly. To find these observed voxels we intersect a ray from each camera to each of its reconstructed points with all voxels of the volume as shown in Figure 1 using [EGMM07]. If the ray hits a voxel before the actual point, we label the voxel

as observed, which results in an approximation of the scene volume that is already captured with our current views.

As we want to capture objects, the views will be distributed roughly on a surrounding sphere estimated from the camera positions of the current SfM reconstruction. The NBV is required to lie on this sphere, and to look at the object center. To find a specific position on the sphere, we use spherical coordinates. We also define bounds to the parameters, as the viewing angle between the new view and previous ones must be smaller than 30° in order to ensure, that enough feature matches can be found (see Figure 2). We finally formulate the search for the best position inside the bounds as an energy minimization problem. Because of the restriction to only two parameters, the search can be done very quickly. We define the energy of a view v depending on the number of newly observed voxels $U(v)$, the number of already observed voxels $O(v)$, and the distance d to previous views in the set P. As some views might be computed at a physically infeasible position, the user will have the option to reject a NBV during the capturing process. To make sure, that we do not visit rejected views R again, we also maximize the distance towards this set. The final energy is

$$E(v) = 2 \cdot O(v) - U(v) + \alpha \sum_{c \in P} \frac{1}{d(v,c)} + \beta \sum_{r \in R} \frac{1}{d(v,r)}. \quad (1)$$

Where the parameters $\alpha = 4 \cdot \xi$ and $\beta = 12 \cdot \xi$ are determined based on the radius ξ of the sphere, since we always have an arbitrary scaling of the SfM reconstructed scene. Therefore we need to ensure, that the energy of the distance is compatible to the energy of the voxels. To compute $O(v)$ and $U(v)$ we determine the projection matrix of the view from the current parameter set and estimate which voxels are inside a potential camera frustum or not. Due to the discrete structure of the volume we cannot compute the gradient of the energy function. Therefore we minimize the energy using the complex method [Box65] which does not use gradients and also deals with bounds during the optimization.

Guiding the user In order to actually capture a computed view, we need the user to move the device into the correct position. We use the phone display to provide guiding in two ways: First, we present an estimate of how the new image has to look like, in order to give a global impression of the camera position and viewing angle. Second, we show an approximation of the direction towards the exact 3D position.

To generate a preview of the current NBV we use the reconstructed 3D positions of image features in neighboring views. As we know the depth of the features, we can project small patches from the neighbors into the NBV. The overall process is sketched in Figure 3. The approximation is then shown as a semitransparent overlay over the current camera preview and the user simply needs to align the two images. As this alignment can be somewhat ambiguous, we additionally render an arrow that shows the direction towards the NBV. To generate the arrow we track the camera position by matching the current view with the reconstructed features and project the direction towards the computed NBV into the image plane. We also show the normalized distance towards the targeted position to give an impression of the amount of movement needed. An image is captured automatically if the current position is close enough to the planned view to approximate a viewing direction difference of less than 10 °.

4. Results

We show results of the capture process for a simple object in Figure 4. User navigation to a new view typically takes 5 s to 30 s depending on the quality of the generated preview and the users experience. It usually takes longer if the preview is too vague, as the user has to rely on the arrow. An example alignment is shown in Figure 5. We also experienced situations where the application was not able to track the device towards the NBV position due to bad feature matches. This is of course undesired, but also indicates that the NBV is not suitable for a multi-view reconstruction. After a view has been captured successfully the integration into the reconstruction takes 2 s to 20 s: The feature matching and pose estimation is done constantly in about 1 s but the global optimization gets more expensive when the number of images and 3D points increases. 25 to 30 images are usually the limit, where the runtime becomes unreasonable. The NBV planning itself is done in about 900 ms.

The final dataset contains high quality images, that are all suitable for further processing. As shown in Figure 6, a final multi-view stereo reconstruction is comparable to one obtained from a dataset captured by an expert. Results for an outdoor statue are shown in Figure 7. As the view planning is intended to capture the scene from all possible directions, it also generates positions, that are way above the ground and cannot be captured casually. Therefore the ability to reject views is important for the system. The final scene of this experiment is shown in Figure 8.

5. Conclusion

We presented a first approach to guided capturing of multi-view stereo datasets on mobile devices. Our technique does not require previous knowledge about the scene and relies exclusively on captured visual information. It explores the scene automatically and can therefore assist any user, especially unexperienced ones.

The main deficit of the system is currently the computation time of the reconstruction optimization, which limits a smooth usability for a high number of images. We also want to incorporate additional sensors of mobile devices in order to improve the position tracking. Finally, to evaluate and improve the view planning and the guiding interface a user study is part of future work.

Figure 4: *Left-to-right:* An initially captured image, a generated preview, and the captured view for the preview.

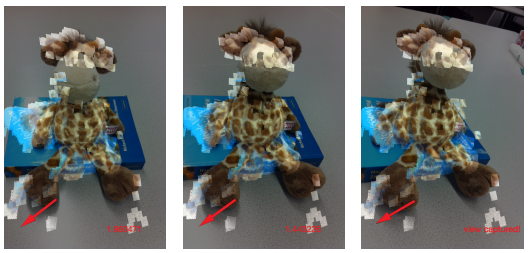

Figure 5: *Left-to-right:* The alignment process for a specific view of the giraffe scene. The distance decreases while the device is moved according to the arrow.

Figure 6: *Left*: Multi-view stereo reconstruction with just the initial set of images. *Middle*: Final reconstruction with 22 images. *Right*: Reconstruction from a dataset of 22 images captured by an expert without explicit view planning.

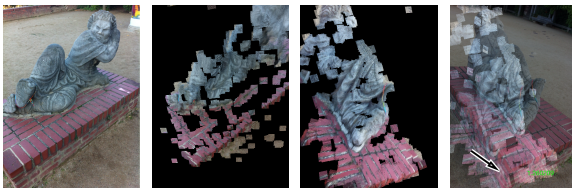

Figure 7: *Left-to-right:* Captured image, two generated previews, preview overlay in the capture view for the statue.

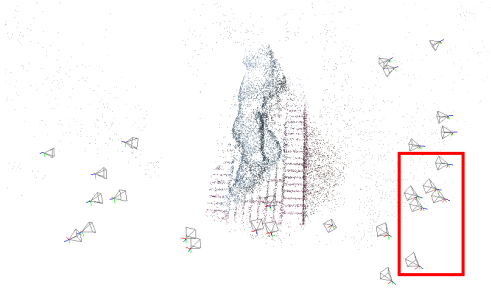

Figure 8: Visualization of the captured dataset (initial set marked red). All images are registered and can be used for a dense multi-view stereo reconstruction.

References

[AM12] AGARWAL S., MIERLE K.: *Ceres Solver: Tutorial & Reference*. Google Inc., 2012. 2

[BAD10] BAE S., AGARWALA A., DURAND F.: Computational rephotography. *ACM Trans. Graph. 29*, 3 (July 2010). 2

[Box65] BOX M. J.: A new method of constrained optimization and a comparison with other methods. *The Computer Journal 8* (Apr. 1965), 42–52. 3

[DF09] DUNN E., FRAHM J. M.: Next best view planning for active model improvement. In *Proc. of BMVC* (2009). 2

[DRMS07] DAVISON A., REID I., MOLTON N., STASSE O.: MonoSLAM: real-time single camera SLAM. In *Proc. of IEEE PAMI* (2007). 1

[EGMM07] EISEMANN M., GROSCH T., MÜLLER S., MAGNOR M.: Fast ray/axis-aligned bounding box overlap tests using ray slopes. *Journal of Graphics Tools 12*, 4 (2007), 35–46. 2

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multi-view stereopsis. *IEEE PAMI 32*, 8 (2010), 1362–1376. 2

[GSC*07] GOESELE M., SNAVELY N., CURLESS B., HOPPE H., SEITZ S. M.: Multi-view stereo for community photo collections. In *IEEE ICCV* (2007). 2

[HKLP09] HIEP V., KERIVEN R., LABATUT P., PONS J.-P.: Towards high-resolution large-scale multi-view stereo. In *IEEE CVPR* (2009). 2

[HKR*12] HOPPE C., KLOPSCHITZ M., RUMPLER M., WENDEL A., KLUCKNER S., BISCHOF H.: Online feedback for structure-from-motion image acquisition. In *BMVC* (2012). 1

[KM09] KLEIN G., MURRAY D.: Parallel tracking and mapping on a camera phone. In *Proc. of IEEE ISMAR* (2009). 1

[LDW91] LEONARD J., DURRANT-WHYTE H.: Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ IROS* (1991), pp. 1442–1447 vol.3. 1

[LJW*07] LV Q., JOSEPHSON W., WANG Z., CHARIKAR M., LI K.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. In *Proc. of VLDB* (2007). 2

[New05] NEWMAN P.: Slam–loop closing with visually salient feature. In *Proc. of IEEE ICRA* (2005). 1

[RRKB11] RUBLEE E., RABAUD V., KONOLIGE K., BRADSKI G.: ORB: an efficient alternative to SIFT or SURF. In *IEEE ICCV* (2011), pp. 2564–2571. 2

[SSS06] SNAVELY N., SEITZ S. M., SZELISKI R.: Photo tourism: exploring photo collections in 3D. In *ACM Transactions on Graphics* (2006), vol. 25, pp. 835–846. 2

[TMD10] TRUMMER M., MUNKELT C., DENZLER J.: Online next-best-view planning for accuracy optimization using an extended e-criterion. In *Proc. of the IEEE ICPR* (2010). 2

[WDAN07] WENHARDT S., DEUTSCH B., ANGELOPOULOU E., NIEMANN H.: Active visual object reconstruction using d-, e-, and t-optimal next best views. In *IEEE CVPR* (2007). 2