Towards Rapid 3D Reconstruction on Mobile Phones from Wide Field-of-view Images

Qi Pan\(^*\)\hspace{1cm} Clemens Arth\(^*\)\hspace{1cm} Ed Rosten\(^\dagger\)\hspace{1cm} Gerhard Reitmayr\(^\ddot{\text{§}}\)
University of Cambridge\hspace{1cm} Graz University of Technology\hspace{1cm} University of Cambridge\hspace{1cm} Graz University of Technology

**Abstract**

Rapid 3D reconstruction of environments has many applications for mobile augmented reality, ranging from creating models for visual tracking systems, interaction models for placing annotations and environment maps for realistic rendering of virtual objects. Ideally, a single user should be able to rapidly generate a coarse model of their surroundings using only a handheld device. To achieve this goal, we propose an efficient reconstruction method using several wide field-of-view images instead of a large collections of images. Furthermore, we adapt a cheap online space carving approach based on Delaunay triangulation to obtain a dense, polygonal representation of the environment. Due to the reduced redundancy, all these steps are feasible on mobile devices.

1 INTRODUCTION

The creation of 3D reconstructions of scenes is essential for many Augmented Reality (AR) applications, but in traditional usage scenarios, the accuracy of the model is far more important than the time taken to reconstruct it. Improvements in mobile phone technology have allowed a device capable of delivering AR to be available in the pocket of the average consumer. To be able to deliver AR on this ubiquitous platform, however, requires 3D models of the scenes which require augmenting. As detailed 3D models for the whole world do not exist (and probably cannot exist due to environments which require augmenting), this leads to the need to rapidly create coarse models of scenes in-situ to track from and annotate, to form the foundation of AR applications on mobile phones.

Traditional 3D model reconstruction techniques [11] are unsuitable for use on mobile phones, as their computational and memory costs are too great. These algorithms generally take tens of minutes to hours on powerful desktop hardware, which are orders of magnitude more powerful than mobile phones. Computationally cheaper on-line reconstruction algorithms exist, but the majority of these produce only point clouds, rather than dense geometric models of a scene. Examples of on-line point cloud reconstruction systems include SLAM systems as well as PTAM [6], the latter of which has been successfully adapted to run on mobile platforms. Newcombe and Davison recently demonstrated dense reconstruction on live videos based on implicit surface calculations and dense optical flow [7]. This method is too expensive for the mobile devices we target, but provides an impressive outlook for future rapid reconstruction systems. Van den Hengel et al. [12] propose a video-based in-situ reconstruction system for laptops, although the technique requires human interaction and is not fully automated like the proposed algorithm. ProFORMA [9] is a video based on-line reconstruction system which produces dense geometric models from live video, and a cut-down more computationally efficient version would seem to be a sensible approach for on-line reconstruction on mobile phones. However, it was found in our field tests with the system running on a laptop, that this approach faces several difficulties compared to the standard set-up with a fixed camera. The jerky camera motion due to taking steps in the environment meant that many correspondences were not tracked due to camera blur, causing tracking to fail regularly. It was also found to be very inconvenient both trying to walk around in the environment and look at the screen whilst also trying to keep the camera steady. Passing cars and pedestrians would also cause the system to fail by occluding large parts of the scene, requiring the sequence to be recaptured.

These reasons mean that video-based on-line reconstruction techniques are not ideally suited for use on mobile phones. An alternative would be to take images using the mobile phone, but this also faces several difficulties. The narrow field of view on the mobile phone means that many overlapping images are required to be taken in order to cover a building/scene completely. However, the complexity of bundle adjustment grows with the number of cameras to be estimated, thus increasing the load on the device. Furthermore, the user has to ensure that there are no gaps in the image set, or a fully connected reconstruction is not possible. This is usually circumvented through the use of Internet image databases, e.g. flickr, which store images taken by many more people.

Therefore, based on these observations, we have designed a rapid reconstruction system suitable for use on mobile phones. At the start, the user stands in one place and uses the phone to generate a panorama instantly from live video. The user then moves to a further 2 (or more) positions and creates panoramas in the same way as the first location. Using a minimum of 3 wide field of view images, there are only 3 unknown camera poses to work out, whilst covering the whole of the object/scene to be modelled in high resolution (due to the use of stitched images). The use of three images is the minimum image number required to detect outliers in matching, which is necessary due to the large amounts of repetitive texture in urban scenes. A point-cloud model is produced, which is subsequently converted to a dense 3D model using a modified version of the ProFORMA carving technique [9].

Kang and Szeliski [5] describe a system with a similar set-up to our approach for the PC. Their system is capable of performing 3D reconstructions of indoor environments from multiple panoramas.

---

\*e-mail: qp202@cam.ac.uk
\^e-mail: arth@icg.tugraz.at
\dagger e-mail: er258@cam.ac.uk
\ddot{\text{§}}e-mail: reitmayr@icg.tugraz.at
However, the computational complexity involved is likely to be too expensive for a mobile phone implementation due to optical flow computations to calculate spline-based warps between the panoramas.

Effectively, we are reducing the redundancy in the input data as early as possible. The capture of a panorama from rotating a mobile device creates a large field-of-view image containing almost the same information as many images taken from positions close by (see Fig. 1). Additionally, capturing images instead of a continuous video stream is more robust and easier to execute as individual images can be replaced with new images. The wide field-of-view allows us to reduce the number of images to keep feature matching and optimisation efforts to a minimum. These properties make the approach attractive for single users equipped with mobile devices.

2 System Details

The user records several wide field of view images using a mobile phone from different positions. These images are roughly aligned so that subsequent stages can use this assumption to reduce computational cost. Features are detected in these images and matched to create correspondences and a sparse 3D point cloud is reconstructed. This point cloud is then used to create a dense polygonal representation textured with data from the original images.

The aim is to have a complete rapid reconstruction pipeline running on the Nokia N900, enabling wide field-of-view image collection and 3D reconstruction to be performed on the device itself. Algorithms were first implemented on the PC (2.5GHz Intel dual core), before being ported to the N900 (600MHz ARM).

2.1 Wide Field of View Image Creation

Panoramic images are created online on the mobile phone whilst panning the camera around the user’s location. We use a version of the PanoMT system [14] developed at TU Graz, compiled for the N900. The user uses the mobile phone to collect wide field-of-view images of the same building/outdoor scene from different locations. The program maps keyframes from the live video onto a cylindrical map. The tracking and map building works at frame-rate on the phone, whilst saving the cylindrical image at the end takes approximately 1 second.

2.1.1 Cylindrical Camera Model

When generating wide field-of-view images, the perspective projected image from the camera is mapped onto a cylinder. Let \( \mathbf{x}_w = (x_w, y_w, z_w, 1)^T \) be the (homogeneous) 3D coordinate of a point in the world frame and \( \mathbf{E}_c \) an element of \( SE(3) \) [13] representing the pose of camera \( c \). The 3D coordinate of the point in the coordinate frame of camera \( c \), \( \mathbf{x}_c = (x_c, y_c, z_c, 1)^T \) is expressed as:

\[
\mathbf{x}_c = \mathbf{E}_c \mathbf{x}_w
\]

The point of intersection, \( \mathbf{x}_i \), of the ray cast by \( \mathbf{x}_c \) and a cylinder of radius 1 with its axis aligned with the y-axis of camera \( c \) is given by:

\[
\begin{align*}
\mathbf{x}_i &= \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \frac{1}{\sqrt{x_c^2 + z_c^2}} \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \\
\end{align*}
\]

(2)

If the centre of the cylindrical image of size \( S_x \times S_y \) is taken to be where the positive \( z \)-axis intersects the cylinder, and \( V \) is and vertical field of view (in radians), the function taking a point on the cylinder in 3D to cylindrical image pixel coordinates \( (X,Y)^T \) (origin in top left) is:

\[
\theta = (\text{atan}2(x_i, z_i)) \pmod{2\pi}
\]

(3)

\[
\begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} S_y \left( 0.5 + \frac{\theta}{2\pi} \right) \\ S_x \left( 0.5 - \frac{V}{2\pi} \right) \end{pmatrix} \pmod{S_y} \pmod{S_x}
\]

(4)

2.2 Generating correspondences

2.2.1 Image Alignment

The panoramic images are centered around the first keyframe used to initialize the capture and thus, the cylindrical images may not be aligned to each other. To make subsequent feature matching more robust and efficient, we calculate a coarse alignment between multiple cylindrical images. Alignment is described only as rotation around the cylinder axis, assuming a level horizon and similar standing height. Super-subsampled images of resolution 36x4 pixels are created by averaging rectangular regions in the cylindrical images. The sum squared difference (SSD) of intensity over 36 horizontal shifts is then calculated, giving a penalty score of half the intensity range when at least one of the two pixels being compared is completely black (unfilled regions of the partial panoramas are coloured black). The alignment rotation is the shift which generates the lowest SSD. This procedure is used to align all images to the first image through iteratively aligning each new image to the previous one.

2.2.2 Feature matching

SURF Features [3] are matched across aligned images only within a constrained width (20% of the image width) to reduce the search space and take advantage of the fact that the images are aligned. Multiple hypothesis matches are generated, whereby all matches within a certain distance in SURF space are recorded. The distance is quite large to enable features distorted by the effects of projecting onto a cylindrical camera to still be potential matches. Multiple hypotheses matching is very important for many urban environments where there is repetitive texture in the scene (such as window corners), as due to the user moving from the original spot, there is no guarantee that the actual repetitive feature in a different image will have the smallest SURF distance. We use an efficient implementation of SURF designed for mobile platforms [1], which can extract 4000 features from a 2304x464 cylindrical image in 8 seconds on the N900.

2.3 Epipolar Geometry Recovery

When the first two images are collected, no information is known about the 3D locations of features identified in the images. We attempt to recover the epipolar geometry of the cameras by using 2D feature correspondences and the five-point pose algorithm [8]. Hypotheses for RANSAC are generated using a subset of the multiple hypotheses matches, and are limited to "unique matches." A unique match is defined as a match whose nearest SURF neighbour is less than 40% the distance of its second nearest neighbour. The multiple hypothesis matches are then used in each RANSAC iteration to calculate the number of epipolar inliers for each hypothesis. Once an essential matrix with a high number of inliers is obtained from RANSAC, the matrix is decomposed into 4 possible solutions for the rotation and translation of the cameras (translation up to scale). The correct solution is chosen by taking a correspondence and triangulating its 3D position using each of the 4 poses, then choosing the solution which corresponds to the point being in the positive direction of the ray for both cameras.

2.4 Bundle Adjustment

Having obtained the camera pose and a set of 2D feature correspondences for two frames, it would now be possible to simply find the closest point of approach of rays cast by the 2D observations to convert the 2D features to 3D landmarks (a 3D point with information about where and in which images it was observed). This approach is, however, limited to two images, and does not provide a way of reducing the re-projection error. Thus instead, we use a triangulation scheme which expresses the maximum likelihood 3D position of a landmark in terms of camera pose and 2D observations.
only, and in a manner which works for more than 2 frames. This enables us to calculate the derivatives of how the maximum likelihood 3D position of the feature changes with respect to camera motion, enabling us to perform bundle adjustment, the minimisation of reprojection error of landmarks with respect to both camera poses and landmark positions. Using this formulation of triangulation, however, actually means that landmarks are not explicitly represented but are a function of camera pose, allowing 3D landmark positions to be factorised out of the bundle adjustment, greatly reducing the computational complexity. This approach directly uses the sparseness inherent in the structure from motion problem, instead of applying the Schur decomposition to achieve similar results.

2.5 Pose Estimation using Three Point Pose

Pose estimation between the first two images (when only 2D-2D correspondences are known) is conducted using the five point pose algorithm, as discussed in Section 2.3. After the initial pair, points are triangulated and new images are added to the reconstruction using 2D-3D correspondences and the three point pose algorithm [4]. Once we obtain camera poses of additional frames using the three point pose algorithm, we bundle adjust the results to obtain an accurate 3D point cloud.

2.6 Surface Recovery

The model output from bundle adjustment is a 3D point cloud, which whilst capturing the geometry of the observed features, is only a sparse representation of the scene. For many augmented reality applications, and indeed a useful visual representation of the scene for the user, a dense 3D model is required. We perform a 3D Delaunay Triangulation of the point cloud to obtain the convex hull of the points partitioned into tetrahedra. A modified version of the probabilistic space carving used in [9] is used to obtain the surface model, with changes implemented to allow the system to perform inside-out space carving.

Space partitioning is performed through a Delaunay tetrahedralisation using QHull [2]. A Delaunay tetrahedralisation is the extension of a 2D Delaunay triangulation to 3D. The convex hull of the point cloud is partitioned into tetrahedra with the condition that the circumsphere of each tetrahedron contains no vertices of any other tetrahedra. This process places tetrahedra over concavities in the scene, and so a further step is required to remove these extra tetrahedra.

The method of tetrahedra carving to recover the surface mesh is based on [9], modified to work for our set-up. Tetrahedra are carved based on landmark visibility, with the probability of a tetrahedron existing being reduced if it occludes a landmark.

It is not necessary to test all tetrahedra, as surface tetrahedra act as a barrier, shielding tetrahedra below from influencing the surface mesh. In [9], tetrahedra are carved in a recursive manner, starting from those tetrahedra on the convex hull. If a tetrahedron is carved away, then its neighbours are tested for removal. This method works for the outside-in case, where the camera is moved around the outside of an object. When modelling scenes, however, it is often the case that the camera is within the convex hull of the points in the environment, in which case starting the carving process with convex hull tetrahedra may not carve away any tetrahedra at all (as occluding tetrahedra actually only exist within the convex hull). Therefore we generalise the carving process such that it works for both outside-in and inside-out cases.

The first step of the algorithm is to determine which tetrahedra contain camera points. This can be simply done by forming plane equations for each face of each tetrahedron. If the a point corresponding to the camera position lies on the “internal” side of each plane formed from each face of a tetrahedron, then the point is contained inside the volume of the tetrahedron. Tetrahedra which contain a camera point should be removed, and the recursive carving algorithm started at tetrahedra which neighbour the removed tetrahedron. If any cameras are outside the convex hull of the object, then the point lies within the infinite tetrahedron, and thus carving should be performed on its neighbours, which correspond to tetrahedra on the convex hull.

Once recursive carving is complete, the surface mesh can be extracted by marking all triangular faces of tetrahedra with no neighbours. The cylindrical image is mapped onto a cube in order to allow efficient and perspective correct texture mapping of the surface mesh.

3 Results

Figure 2 shows results from various stages of the reconstruction process for a cathedral. Wide field-of-view images are matched to each other to obtain correspondences to build the point cloud. The point cloud is then converted into a dense surface model and texture mapped. Figures 3 and 4 show the reconstruction results for two different scenes, with scenes being rendered from novel camera views in the vicinity of the original cameras. In figure 3 it can be seen that there are tetrahedra high up in the air, which is a result of not having the correct camera angles (due to being limited to being close to the ground) to carve away those tetrahedra. This causes the geometry of the scene to be incorrect, but still fit for purpose as a coarse model for augmented reality applications. Areas of interest still exhibit the correct approximate geometry which would be good enough to track from, for example using the approach described by Rosten et al. [10]. The whole reconstruction process currently takes about 5 seconds for 3 wide field of view images on a desktop PC, and approximately 90 seconds on the N900.

4 Conclusion

The presented method is able to efficiently create coarse reconstructions of environments using only a few images. The reconstructions are only approximations but are available within a few seconds of recording each image, thus allowing instant use in augmented reality applications.

Current limitations are mainly the robustness of matching the images to each other. Given repetitive structure typically found in urban environments, the pure feature-based approach can lead to an overwhelming set of outliers. Furthermore having only a few
images to work from makes the system more prone to self occlusions in the environment. However, we consider these limitations as worthwhile trade-offs for a rapid system response.

5 Future Work

The design decisions made for this system were all with a view to enabling capturing wide field-of-view images and rapid 3D reconstruction on a mobile phone. Phones provide a simple to use interface in a portable device, and recent advances in mobile processors have enabled considerably more processing power in these devices. In our next steps, we plan to further optimise algorithms on the mobile phone to reduce runtime, add an additional matching step once the poses are known to extract more correspondences for a more detailed model, as well as fuse the system with additional mobile phone sensor information such as GPS and compass.

Acknowledgements

This work was partially funded through the Christian Doppler Laboratory for Handheld Augmented Reality and the project HYDROSYS (EU FP7/DGINFSO grant 224416).

References