17^{th} *Computer Vision Winter Workshop*
Matej Kristan, Rok Mandeljc, Luka Čehovin (eds.)
Mala Nedelja, Slovenia, February 1-3, 2012

Photogrammetric Camera Network Design for Micro Aerial Vehicles

Christof Hoppe, Andreas Wendel, Stefanie Zollmann,
Katrin Pirker, Arnold Irschara, Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{lastname}@icg.tugraz.at

Stefan Kluckner
Siemens Corporate Technology,
Graz, Austria
stefan.kluckner@siemens.com

Abstract. *Micro Aerial Vehicles (MAVs) equipped with high resolution cameras have the ability of cost efficient and autonomous image acquisition from unconventional viewpoints. To fully exploit the limited flight-time of current MAVs view planning is essential for complete and precise 3D scene sampling. We propose a novel camera network design algorithm suitable for MAVs for close range photogrammetry that exploits prior knowledge of the surrounding. Our algorithm automatically determines a set of camera positions that guarantees important constraints for image based 3D reconstruction. On synthetic experiments we demonstrate that our camera network design obtains detailed 3D reconstructions with a reduced number of images at the desired accuracy level. Comparable results are also computed on an outdoor experiment using our MAV in autonomous flight mode.*

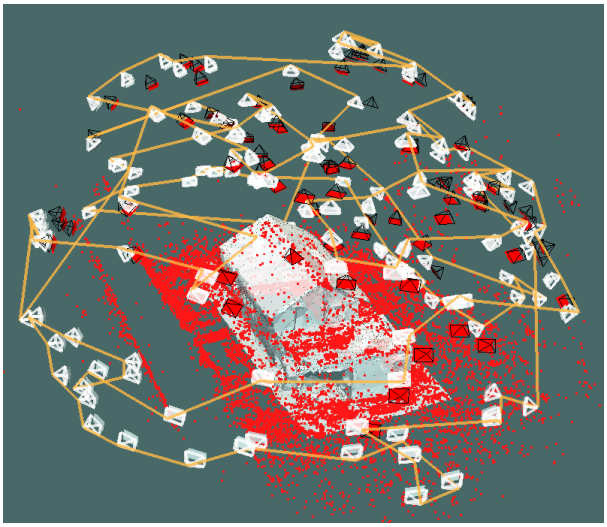


Figure 1. Computed camera network (gray) with corresponding flight path (yellow) overlayed by the result of a SfM approach (red points and red camera frustrums). Image acquisition is performed by an autonomously flying MAV.

1. Introduction

Recent Structure-from-Motion (SfM) algorithms are able to reconstruct large–scale environments from a huge number of unordered photo collections in reasonable time [5]. The vast amount of photos captured from the ground (e.g. Internet photo collections, Google Street View images) contains redundant data that allows detailed reconstructions [15]. However, often some parts of a scene are still underrepresented or missing since they cannot be captured by terrestrial image acquisition. To achive an accurate reconstruction of the entire building, images of aerial view–points are required as well.

While traditional manned airborne photogrammetric surveys are well suited for large scale Digital Surface Model (DSM) extraction of whole cities, they are usually too cost intensive to be employed for small scale reconstructions of individual buildings. For this kind of scenes, micro aerial vehicles (MAVs) equipped with consumer grade digital cameras provide a cost efficient and flexible sensor platform for rapid image acquisition [3]. Small MAVs like octo–rotor helicopter have full six degrees of freedom and can approach each point in space. They can be rapidly deployed and can operate even in densely developed areas. However, the trade-off between payload and battery power supply limits the overall MAV flight time to 10 - 20 minutes. Therefore, view planning, also denoted as Photogrammetric Network Design (PND), is an ineluctable task for complete 3D modeling of a scene where the image acquisition time is the limiting factor.

The calculation of "ideal" viewpoints for 3D reconstruction can be seen as a "chicken-and-egg"

problem. On the one hand the 3D geometry has to be known in advance to find good viewpoints. But on the other hand, we perform a 3D reconstruction to determine the underlying 3D geometry. In the field of Simultaneous Localization and Mapping (SLAM) this problem is known as Next Best View (NBV) planning and most approaches try to solve two tasks simultaneously. Hereby, a robot reconstructs its environment online and tries to find an optimal movement to refine or expand the already reconstructed area. Due to limited on-board computation power those approaches are difficult to realize on an MAV.

Finding ideal views for a 3D reconstruction is a hard task since there are a lot of competing requirements and constraints.First, a view planning algorithm should deliver a small number of views that cover the entire object to guarantee short flight and processing times. Second, redundancy is necessary to achieve an accurate reconstruction. Third, the algorithm has to satisfy constraints like overlap and viewing angle between images to facilitate vision–based similarity computations [13]. Finally, the image overlap graph should be fully connected, otherwise disjoint reconstructions are obtained.

In this paper, we focus on determining a small set of cameras that cover the whole object and allows us to compute an accurate 3D reconstruction. In practice, it is more important to guarantee a sufficient reconstruction of the scene after a flight session than to reach the smallest number of images. Our approach requires a rough surface description of the desired object given as a triangular mesh, which allows us to perform visibility tests and to find self-occlusions. For example, this mesh can be obtained from a DSM or from previous reconstructions of the object. If both are not available, a few images from a nadir view acquired by an MAV can be used to construct a rough model of the object. Based on this prior knowledge, our camera network design algorithm computes view–points that cover the whole object while taking constraints like image overlap and a maximum angle between neighboring views into account. We perform experiments on synthetic and real-world data and demonstrate that our approach creates a set of cameras that reconstructs a 3D object with high accuracy while keeping the number of images low.

In Section 2, we give an overview of research in the field of camera network design and view selection. We extensively describe our approach in Section 3, followed by our experiments. An outlook and future work is presented in Section 5.

2. Related Work

The problem of determining “good”, “ideal” or “reasonable” imaging positions is known in different areas of robotics, computer graphics, computer vision and photogrammetry.

Image based rendering uses 2D images taken from different viewpoints of a 3D object to synthesize new views. Vazquez et al. [19] generate a large number of cameras over the bounding sphere and assign each camera a so-called *viewpoint entropy* value which measures how much of a scene is seen by a certain camera. This measure takes the view angle between the camera and the surface into account but cannot be directly applied to multi-view camera network design problems. In the SLAM and SfM literature, keyframe selection is often employed to speed up processing [17] or to make larger reconstruction problems tractable [16, 7]. The objective here is often to determine and select a reduced set of views from the complete ensemble, that still achieves the desired reconstruction accuracy and completeness. In the photogrammetry community, the task of computing a view configuration for precise 3D reconstruction is known as the PND problem. Early work of Fraser [6] discusses PND in close-range photogrammetry. This work identifies that there are analytic difficulties of computing an optimal imaging geometry due to non-linearity and the multi-modality of the problem. Therefore, most existing PND methods first define network quality requirements [2] and then search for an optimal set of observations by stochastic optimization techniques [14].

In the field of robotics, the sensor placement problem is often incrementally solved by NBV planning methods [18]. These methods aim to perform scene reconstruction and sensor placement simultaneously, hence they do not need a-priori knowledge of the scene. These algorithms are iterative and require powerful processing units for real-time processing. Due to payload limitations, current MAVs often do not come with the required processing power for onboard planning. However, several methods based on NBV strategies exists that exploit prior knowledge to calculate a path for a moving robot offline. Dunn et al. [4] propose an algorithm that combines accuracy-driven NBV and path planning on given textured 3D models. Their objective function takes texture prop-

erties, uncertainty minimization and the number of triangles visible from a certain viewpoint into account. This is in contrast to the work of Wenhardt et al. [21] who propose a purely accuracy driven NBV planning. In their work, the uncertainty of 3D points is investigated by the points covariance matrix [1].

3. Methodology

Finding the minimal subset from an ensemble of camera positions that capture a complete set of 3D points can be seen as the set-cover problem, which is known to be NP-complete [10]. Adding constraints like image overlap or maximum triangulation angles can reduce the search space and the problem turns into a Constraint Satisfaction Problem [12]. We define three constraints that have to be satisfied simultaneously for an ideal camera set:

1. An upper bound of a reconstructed point's uncertainty.
2. A minimum overlap between image pairs.
3. All parts of the surface have to be covered by the images.

We have developed a strategy that selects a subset from a huge number of cameras with respect to an easy–to–compute heuristic while satisfying all three conditions. Due to the low computational complexity, our approach is able to handle large, complex models that require several hundred camera positions to be reconstructed accurately.

From an algorithmic point of view, our approach consists of three main steps: (i) camera placement, (ii) camera clustering and (iii) view selection. In the camera placement step, we initialize our search space with a discretized surface given as a triangular mesh and create cameras observing the mesh. To reduce the search space, we cluster the created cameras. We incrementally select camera positions according to our quality criterion while maintaining side constraints like image overlap and maximum angle between view points.

3.1. Camera Placement

Given a mesh M representing the object's surface, we recursively split every triangle until each edge has reached a edge length $l = rs$, where s denotes the ground sampling distance (GSD) and r is a factor that defines the maximum projection size of a triangle in the image. For each triangle T, we compute its center point T_c, its normal T_n and a camera C_i that observes T fronto-parallel with distance $d = fs$, where d depends on the groundsampling distance and the camera's focal length f.

When thinking of non-convex structures, self–occlusions may occure and the camera position might by located within a part of the object. In that case, we try to find a position along T_n whose distance is close to d. To guarantee positions that can be approached safely by the MAV, we define a margin around each obstacle. Estimated camera positions within this margin are marked as infeasible and are removed.

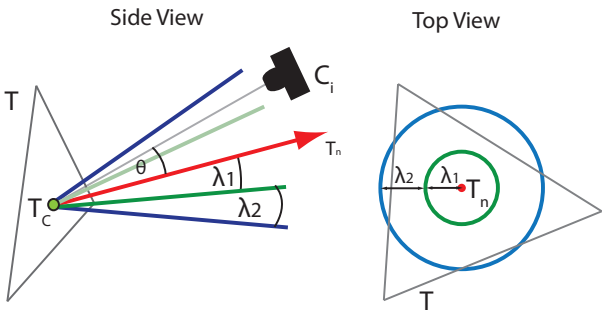


Figure 2. Camera Clustering. Given the triangles surface normal T_n, we build clusters of cameras according to the angle between camera center and T_n. In this example C_i is contained in the second cluster.

3.2. Camera Clustering

Selecting a fixed–size subset of cameras created in the previous step that minimizes the uncertainty of a point is computationally expensive, because all combination of subsets have to be enumerated. In our approach, we discretize the search space for such optimal sets by clustering cameras with respect to their intersection angles given a certain 3D point T_c.

We define $\lambda = \sphericalangle(\overrightarrow{T_n}, \overrightarrow{C_i T_c})$ as the angle between the triangle normal direction T_n and the vector connecting T_c and C_i. For each triangle T, we calculate a histogram T_H that contains cameras C_i clustered by their angle λ. In our experiments, we set the bin width of T_H to 10 degrees (see Figure 2). Cameras with λ greater than 40 degrees are not considered in the histogram. The mesh representation of our object allows raycasting to consider self occlusions of the object.

The intuition behind this is the following. As shown in [1], the depth uncertainty of a reconstructed point depends mainly on the triangulation angle.

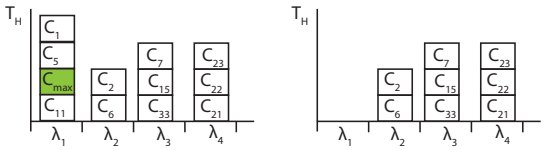


Figure 3. Sample histogram of a triangle T with four bins. a) The green box marks the camera C_{max} with highest support $S(C_i)$. b) All cameras located in the same bin as C_{max} are removed in the histogram update step.

Therefore, using cameras that observe T_c from nearly the same position do not account for a precise reconstruction. In the subsequent view selection process, we require that at least one camera of each bin in T_H is selected. This constraint reduces the cameras search space drastically, while it still guarantees a triangulation angle of at least 20 degrees which can be seen as the lower bound of the reconstruction accuracy.

This clustering allows us to define the importance of a camera for the whole reconstruction process. Typically, a camera is contained in more than a single histogram. The support of a camera C_i $S(C_i)$ is then given as the number of histograms C_i is visible in. The higher the camera's support, the more important is the camera for the overall reconstruction, because high valued cameras cover many scene points.

Algorithm 1 View selection algorithm. C is the set of all cameras C_i. R contains the cameras in the resulting PND.

1: $C_{max} = \arg\max_{C_i}\{(S(C_i))\}$
2: $R = \{C_{max}\}$
3: update $T_H(C_{max})$ for all T_H
4: $C = C \setminus C_{max}$
5: **repeat**
6: order C according $S(C_i)$
7: $C = C \setminus \{C_i | S(C_i) = 0\}$
8: **for all** C_i in C **do**
9: **for all** C_j in R **do**
10: **if** $Overlap(C_i, C_j) > th$ **then**
11: $R = R \cup \{C_i\}$
12: update $T_H(C_i)$ for all T_H
13: $C = C \setminus C_i$
14: Goto Line 5
15: **end if**
16: **end for**
17: **end for**
18: **until** Reconstructed(T) $> 95\%$

3.3. View Selection

The previous steps created a large number of feasible view points that carry a high degree of redundancy. To reduce the number of cameras, we select cameras with high support in a greedy fashion until the scene is reconstructed completely. A scene is said reconstructed *completely* by a set of cameras R, if one of the following two conditions is fulfilled for each triangle T_c:

Condition 1: At least one camera of each bin in T_H is contained in R.

Condition 2: R contains a set of cameras that allows a triangulation of T_c with an uncertainty below a threshold $\phi(T_c)$.

We measure the uncertainty of a reconstructed point by the *E-optimality criterion* [21]. Given a set of cameras L and a triangle center T_c, we calculate the covariance matrix $P(T_c, L)$ of T_c. Then, the *E-optimality criterion* is defined as

$$\Phi(T_c, L) = e_{max}(P(T_c, L)), \tag{1}$$

where e_{max} defines the maximum eigenvalue of $P(T_c, L)$ which is the largest semiaxis of the covariance ellipsoid.

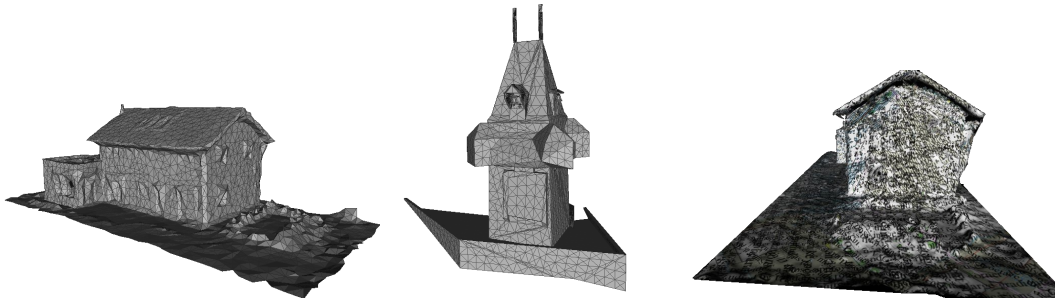
The threshold $\phi(T_c)$ is estimated for each triangle individually by a Monte Carlo simulation. We randomly choose one camera of each bin in T_H which defines the set L_t and calculate $\Phi(T_c, L_t)$. Finally, the threshold

$$\phi(T_c) = \sum_t \frac{\Phi(T_c, L_t)}{t} \tag{2}$$

is the average over all t trials, which can be interpreted as the expected reconstruction accuracy T_c.

After determining $\phi(T_c)$ for all triangles, we start the camera selection process. In the first iteration, we start with an empty set R, calculate the support $S(C_i)$ for each camera and add the camera with highest support, called C_{max}, to R.

Next, we perform a *histogram update* step on all T_H containing C_{max}. We remove all cameras from T_H that are located in the same bin as C_{max}. Therefore, the support of cameras nearby C_{max} drops and it is less likely that close-by cameras are selected in the next iteration. The histogram update is illustrated in Figure 3. If the histogram is empty, condition 1 is satisfied and the point is reconstructed. Otherwise, if



(a) (b) (c)

Figure 4. Synthetic generated views. (a) Mesh of the house. (b) Mesh of the medieval tower. (c) Rendered image from a computed view point of the house with high frequence texture.

condition 2 is satisfied, we remove all cameras from T_H. If a triangle meets one of the two conditions, it does not support any camera anymore.

In the second iteration, we recalculate the support $S(C_i)$ for each camera. To achieve an overlap with at least one camera in R, we select a camera as C_{max} that has highest support while also maintaining a certain overlap with at least one camera in R. The overlap is a precondition that the SfM algorithm is able to find feature matches between image pairs and can perform a geometric verification on the epipolar geometry. We add C_{max} to R and perform the histogram update again.

The overlap between two cameras is defined by the number of triangle centers T_c visible in both cameras and their spatial distribution in image space. The spatial distribution is important for pairwise image matching and geometric verification. So we use our coverage criterion [9] which defines a measure for the spatial distribution of feature points.

Our algorithm converges if no camera is found that meets the overlap criterion or if at least 95% of the triangle points T_c are reconstructed accurately. We tolerate that some points do not meet one of our two conditions, because in most cases the geometric configuration of a few points requires a large set of cameras to meet condition 1 or 2. Algorithm 1 summaries a pseudo-code of the proposed view selection algorithm.

4. Experiments

In this section we show that our algorithm determines a set of camera positions that guarantees that a state-of-the-art SfM algorithm can compute an overall connected reconstruction with very high accuracy. We tested the performance on two synthetic and one real world outdoor experiment.

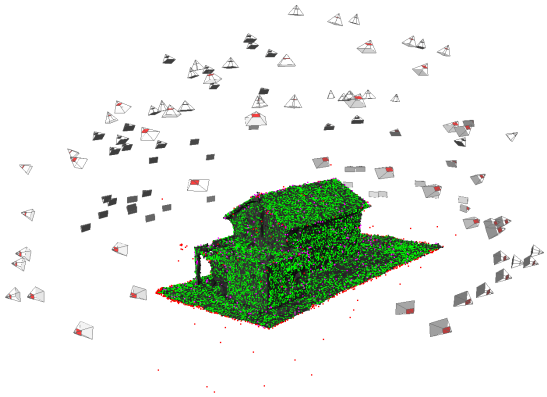


Figure 5. SfM result of the house scene. Our approach selects 106 camera positions (gray) out of 4643. The maximum error between groundtruth and estimated camera position is $0.033\,m$. The color coding of the reconstructed feature points represents the distance t to the mesh. (red: $t > 0.5\,m$, magenta: $0.0\,5m < t < 0.5\,m$, green: $t < 0.05\,m$). Best viewed in color.

In the synthetic experiment, our algorithm computes camera positions for a given mesh. We project high frequence texture on it and render the mesh from the calculated camera positions. Then these images serve as input for a SfM algorithm. We compare the accuracy of the reconstruction to the mesh that was used for generating the input images.

In the outdoor experiment, we perform view planning for the construction site of a small detached house. Since the object's geometry changes over time, the mesh used for view planning does not exactly reflects the houses geometry when taking pictures for evaluation. We show that the view points can be approached automatically by our MAV and that our calculated views cover the scene entirely.

4.1. Synthetic Scenes

We select two meshes of buildings at different complexity given in metric scale. The first mesh is a house that already has been reconstructed by the SfM approach [8] using 273 images captured by a manually controlled MAV. From the resulting semi-dense point cloud [7] we determine a surface mesh containing 7489 faces using the Poisson surface reconstruction algorithm [11]. Figure 4(a) shows the resulting mesh.

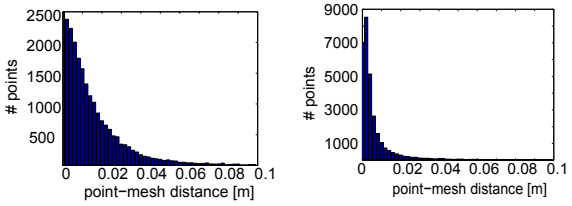
The second synthetic experiment is based on a model of a medieval tower. For both experiments, we set $r = 50$ and split triangles to a side length of $l = 0.5\,m$ resulting in 4446 faces (Figure 4(b)).

In both experiments, we use a GSD of $s = 8\,mm$ and a focal length $f = 24\,mm$ resulting in a distance parameter $d = 20\,m$. We project high frequence texture onto each triangle and render both meshes for the determined camera positions using OpenGL. Figure 4(c) shows a rendered image that serve as input of the SfM pipeline.

For the house, our algorithm computes 4643 camera positions out of 7446 faces that are reachable by the MAV poses. The remaining camera positions are removed, because they are too close to the building or to the ground. We ensure a minimum object–to–camera distance of $3\,m$ which is the expected positioning accuracy of the MAV. Our approach selects 106 out of 4643 possible camera positions. All 106 cameras are aligned by the SfM algorithm and 23,645 feature points are reconstructed. To measure the accuracy of the reconstructed sparse points we compute their normal distance to the closest triangle. 597 of 23,645 (2.5%) points have a distance larger than $0.5\,m$ and can be classified as outliers. 1235 (5.2%) points range between $0.05\,m$ and $0.5\,m$ and the remaining 92.3% are closer than $0.05\,m$. Figure 6(a) shows the histogram of the point-mesh distances.

The camera pose error $cpr = \|C_{sfm} - C_{rendered}\|^2$ measures the difference between the pose estimated by the SfM approach C_{sfm} and the rendering viewpoint $C_{rendered}$. The average cpr for all cameras on the single family house scene is $0.022\,m$ and the maximum error is $0.033\,m$. Figure 5 shows the estimated camera poses and the triangulated feature points.

For the medieval tower, 137 out of 3135 camera positions are selected. All camera positions and 30,730 feature points are reconstructed by the SfM algorithm. 535 (1.7%) 3D points are outliers,



(a) (b)

Figure 6. Histogram of distances between reconstructed points by SfM and mesh surface. The histogram of the single family house (a) show that 95% of the reconstructed points have a maximum distance of 0.05m to the surface. (b) 96% of the reconstructed points of the medieval tower have are closer than 0.05m to the object's surface.

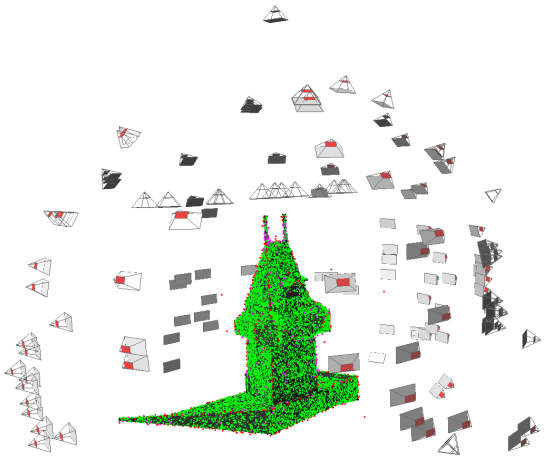


Figure 7. SfM result for the medieval tower. 137 cameras are used for reconstruction. The point colors encode the point-mesh distance t (red: $t > 0.5m$, magenta: $0.05\,m < t < 0.5\,m$, green: $t < 0.05\,m$). Best viewed in color.

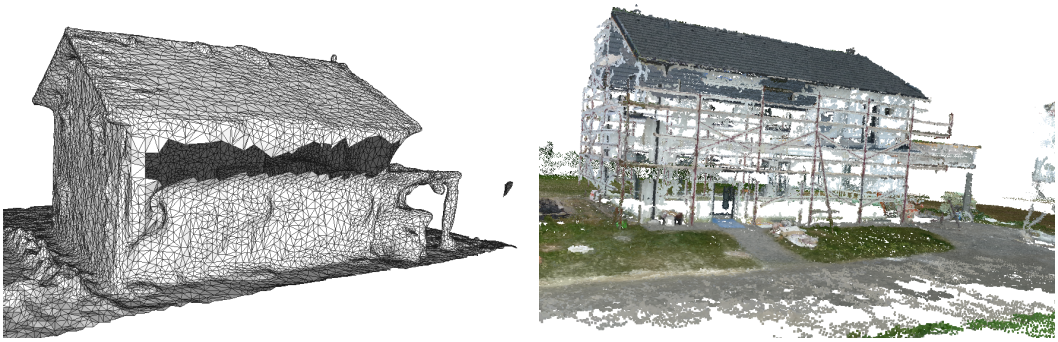
647 (2.1%) points have point-mesh distance between $0.05\,m$ and $0.5\,m$ and 29,548 (96.2%) are closer than $0.05\,m$ to the mesh. The histogram of the distances is given in Figure 6(b). Figure 7 visualizes the reconstruction result.

4.2. Real World Example

In contrast to the synthetic examples where images are rendered from noisefree camera positions, we investigate the performance of our algorithm when taking images acquired by an autonomously flying MAV[1].

The initial mesh used for view planning was reconstructed from 273 images of a house during the construction phase acquired by a manually controlled MAV. Due to randomly choosen camera positions, some parts of the building were not recovered and

[1] Asctec Falcon 8 Octo-rotor helicopter



(a) (b)

Figure 8. Difference between geometry used for view planning and resulting reconstruction. (a) The mesh used for view planning was acquired manually by an MAV which results in missing views for parts of the house. (b) In contrast, images obtained by our view planning algorithm cover the entire scene. The holes in the semi-dense reconstruction are caused by weakly textured surface. It is also visible that the geometry of the house has changed over time due to a scaffold.

therfore the mesh contains holes (Figure 8(a)). We used [20] to geo–reference our mesh. This allows as to compute camera positions in GPS coordinates. For view planning, we set the GSD to $6\,mm$ which results in a camera distance $d = 15\,m$. The camera we used for image acquisition is a Panasonic DMC-LX3 still-image camera, which has a resolution of 10Mpx and a $24\,mm$ lens.

By running our view–planning algorithm we obtain 133 camera poses that are approached autonomously by the MAV. We use a greedy path optimization approach to obtain a camera sequence with short paths in-between. We start with a fully connected graph having the camera centers as nodes, and remove the edges that intersect with the geo–referenced mesh. Every remaining edge is given a weight corresponding to the distance of the nodes. To get a smoother path, we further penalize changes in altitude and direction. Finally, given the edges and corresponding weights, we start at the point with the lowest height over ground and greedily search for the closest node. Figure 1 shows the resulting flight path.

Since the house was in construction phase, the mesh used for view planning and the real geometry when performing our experiment differ. For example, a the building was surrounded by a scaffold which was not reflected in the mesh used for view planning.

Given 133 images, the applied SfM algorithm reconstructs 20,762 feature points and 81 camera positions. The discrepancy between the number of acquired images and the images used by the SfM can be explained by several reasons: (a) the MAV is equipped with a standard GPS receiver whose accuracy ranges between 1 and 3m and (b) the MAV can approach a GPS waypoint with a precision of only $2\,m$.

In the worst case, both errors sum up and lead to a positioning error of $5\,m$. In conjunction with alignment errors of the geo–referenced mesh, some views do not observe the expected parts of the object. Furthermore, we assume textured surface which is not always present in the real world example. Furthermore, the geometry of the house changed over time which is not considered during the view planning.

Figure 1 shows planned view points (gray) and respective camera positions computed by the SfM algorithm (red). The yellow lines mark the calculated flight path. In Section 4.1 we have shown that the SfM approach delivers very accurate pose estimations, consequently the difference between C_p and C_{sfm} is mainly caused by the imprecision of the MAV. Most triangulated feature points are close to the mesh, which highlights the accuracy of the reconstruction.

To analyze if all parts of the building are observed by the 81 aligned images, we run a semi-dense reconstruction [7]. The result shown in Figure 8(b) indicates that the scene has been captured by the MAV entirely. Missing parts on the facade are caused by missing texture that prevents accurate patch-matching. This result illustrates that even a subset of the selected cameras is sufficient to capture most parts of the object but a certain accuracy can only be guranteed by the whole computed set of cameras.

5. Conclusion and Future Work

We have proposed a new photogrammetric network design algorithm that generates a small set of view points while satisfying constraints that are important for image based 3D reconstruction. In synthetic experiments, we have shown that this set guarantees highly accurate camera pose estimation as well as precise reconstruction of a 3D scene. The real world outdoor example illustrates that the computed camera positions can be approached safely in autonomous flight mode by the MAV and all visible parts of the scene can be reconstructed. The error between ground truth and reconstruction is smaller than $5cm$ for more than 92% of all reconstructed scene points.

Our approach is a first step towards a controlled and planned application of SfM for detailed outdoor reconstruction using MAVs. In future work we will take into account the positioning error of the MAV and integrate the GPS uncertainty into our approach. Furthermore, we will investigate the influence of the mesh's resolution on the view selection strategy.

Acknowledgments

This work has been supported by the Austrian Research Promotion Agency (FFG) FIT-IT projects Construct (830035) and Pegasus (825841).

References

[1] C. Beder and R. Steffen. Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *DAGM*, pages 657–666, 2006. 3

[2] S. Y. Chen, Y. F. Li, J. W. Zhang, and W. L. Wang. *Active Sensor Planning for Multiview Vision Tasks*. Springer-Verlag, 2008. 2

[3] I. Colomina, M. Blázquez, P. Molina, M. Parés, and M. Wis. Towards a new paradigm for high-resolution low-cost photogrammetry and remote sensing. In *ISPRS*, volume XXXVII, Part B1, pages 1201–1206, 2008. 1

[4] E. Dunn, J. P. van den Berg, and J.-M. Frahm. Developing visual sensing strategies through next best view planning. In *IROS*, pages 4001–4008, 2009. 2

[5] J.-M. Frahm, P. F. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building rome on a cloudless day. In *ECCV*, volume 6314, pages 368–381. Springer, 2010. 1

[6] C. S. Fraser. Network design considerations for non-topographic photogrammetry. *Photogrammetric Engineering and Remote Sensing*, 50:1115–1125, 1984. 2

[7] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 2, 6, 7

[8] A. Irschara, V. Kaufmann, M. Klopschitz, H. Bischof, and F. Leberl. Towards fully automatic photogrammetric reconstruction using digital images taken from UAVs. In *ISPRS - Advancing Remote Sensing Science*, 2010. 6

[9] A. Irschara, C. Zach, J. M. Frahm, and H. Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, pages 2599–2606, 2009. 5

[10] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, NY, 1972. 3

[11] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70, 2006. 6

[12] C. Lecoutre. *Constraint Networks: Techniques and Algorithms*. Wiley-IEEE Press, 2009. 3

[13] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65:43–72, 2005. 2

[14] G. Olague. Automated photogrammetric network design using genetic algorithms. *Photogrammetric Engineering and Remote Sensing*, 68(5):423–432, May 2002. 2

[15] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006. 1

[16] N. Snavely, S. M. Seitz, and R. S. Szeliski. Skeletal graphs for efficient structure from motion. In *CVPR*, pages 1–8, 2008. 2

[17] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Real-time monocular SLAM: Why filter? In *ICRA*, 2010. 2

[18] M. Trummer, C. Munkelt, and J. Denzler. Online next-best-view planning for accuracy optimization using an extended e-criterion. In *ICPR*, pages 1642–1645, 2010. 2

[19] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In *Vision Modeling and Visualization Conference*, pages 273–280, 2001. 2

[20] A. Wendel, A. Irschara, and H. Bischof. Automatic alignment of 3D reconstructions using a digital surface model. In *CVPR, Workshop on Aerial Video Processing*, 2011. 7

[21] S. Wenhardt, B. Deutsch, E. Angelopoulou, and H. Niemann. Active visual object reconstruction using d-, e-, and t-optimal next best views. In *CVPR*, 2007. 3, 4