# Incremental Dense Reconstruction from Sparse 3D Points with an Integrated Level-of-Detail Concept

Jan Roters and Xiaoyi Jiang

Department of Computer Science, University of Münster,
Einsteinstraße 62, 48149 Münster, Germany
{jan.roters, xjiang}@uni-muenster.de

**Abstract.** Since decades scene reconstruction from multiple images is a topic in computer vision and photogrammetry communities. Typical applications require very precise reconstructions and are not bound to a limited computation time. Techniques for those applications are based on complete sets of images to compute the scene geometry. They require a huge amount of resources and computation time before delivering results for visualization or further processing.
In the application of disaster management those approaches are not an option since the reconstructed data has to be available as soon as possible. Especially, when it comes to *Miniature Unmanned Aerial Vehicles (MUAVs)* sending aerial images to a ground station wirelessly while flying, operators can use the 3D data to explore the virtual world and to control the MUAVs.
In this paper an incremental approach for dense reconstructions from sparse datasets is presented. Instead of focussing on complete datasets and delivering results at the end of the computation process, our incremental approach delivers reasonable results while computing, for instance, to quickly visualize the virtual world or to create obstacle maps.

## 1 Introduction

Scene reconstruction from multiple images is still a hot topic in computer vision and photogrammetry community. Current approaches are focussing on accuracy while requiring a lot of computational resources and computation time and delivering results at the end of the computation process.

Since the 3D information is only available after complete computation there are some use cases which are not suitable to use these approaches, e.g. disaster management with a swarm of *Miniature Unmanned Aerial Vehicles (MUAVs)* delivering still images over the air while flying. A quick visualization would help operators to get a better view of the scene and to control the MUAVs (see Fig. 1). For that purpose, one can show the 3D points directly [1] or build up a 3D mesh from the points [2]. Another example is the creation of obstacle maps for autonomous flights of those MUAVs. Therefore, the denser the map of 3D points, the better obstacles are known and collisions can be prevented.

A lot of research has been done on sparse incremental 3D reconstruction. Whereas there are some algorithms to compute the sparse scene geometry and the camera positions at once [3], there are a lot of methods to compute this data incrementally one or a few images after another [4].

Fig. 1: Operators controlling MUAVs in a simulated environment and exploring the scene at a multi-touch wall.

Incremental approaches for dense reconstruction has not been covered a lot in literature. Current dense reconstruction approaches focus on very accurate 3D information [5] but at the cost of long computation times to further process or directly visualize the results. At the downside those dense reconstruction approaches are not designed to work incrementally and thus, they are not suitable for all applications, e.g. the previously mentioned disaster management. Incremental dense reconstruction has also been applied to live video streams [6]. Some of those approaches work with video data also delivered by MUAVs [7].

In this paper we present an approach to incremental wide baseline dense reconstruction from sparse 3D dataset computed from multiple still images. The main contributions are (1) the reconstruction of reasonable points to get a quick denser overview of the scene, (2) to get incremental supply of denser 3D data while the data is further refined incrementally in the background and furthermore, (3) the approach we present integrates a level-of-detail concept.

Our approach is presented in Section 2. In Section 3 we show some experiments and results to evaluate our incremental dense reconstruction. We conclude this work in Section 4.

## 2 Incremental Dense Reconstruction

For traditional approaches it does not matter in which order the 3D points of a scene are computed as long as the final reconstruction is correct. In general, the approaches compute dense reconstructions by searching point matches in the neighborhood around already known scene points [5]. This technique has the advantage to get a consistency between neighbored matches since they are very close to each other. Furthermore, this consistency measure also detects larger discontinuities in the depth data, e.g. at the borders of objects. As a downside, these algorithms are not designed to work incrementally, i.e. even if they may be adapted to work incrementally delivering results throughout the computation the reconstructed information is not reasonable due to low visual entropy. There is a very high density around the previously known points but the major areas between those points would not contain any information up to a later computation progress.

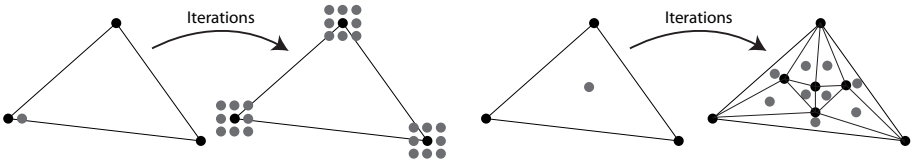



Fig. 2: Comparison of dense reconstruction methods. (a) Traditional approaches reconstruct points in the neighborhood of already known points. (b) Our incremental approach firstly reconstructs the midpoints of given triangles.

Especially in disaster management with flying MUAVs the operator does not gain much more information from a bunch of neighbored points as they may appear as one point in the virtual world which the operator explores and in which he has to control the MUAVs.

In our approach we try to create information in those major areas between known points instead of only in their neighborhood. We propose a method for dense reconstruction from a given sparse reconstruction. Instead of computing the dense reconstruction from all images at once using a lot of computation time before returning a result, our approach computes the dense reconstruction incrementally two images at a time. To get a reasonable incremental result we do not reconstruct the points in the neighborhood of known sparse points but rather the midpoints of each triangle in a given triangulation of the known points. These midpoints have the maximum distance to the three previously known points. The reconstructed point therefore gives at once more information than only neighbored points (see Fig. 2).

On the one hand the results can be visualized very quickly with good visual entropy and can be used for further computations, e.g. incremental mesh computation. One the other hand we have to deal with the problem of matching feature points without relying on the consistency of neighbored points.

In the following we outline a short overview of our algorithm. These steps describe one iteration for one image. In general, these steps have to be done for each image and each iteration.

1. Compute Delaunay triangulation from known feature points (see Sec. 2.1).
2. For each triangle
   (a) check triangle consistency (see Sec. 2.2),
   (b) match midpoint of the triangle with a second image (see Sec. 2.3),
   (c) triangulate match and verify the point with additional images (see Sec. 2.4).

The number of iterations needed for computing the dense reconstruction depends on the required density, i.e. the best level of detail (see Sec. 2.2).

### 2.1 Delaunay triangulation

A triangulation method of the sparse 2D points is used to determine reasonable points to reconstruct. The reconstruction is done for the midpoint of each triangle since it is

the point regarding to one camera that has the maximum distance to the triangle points and thus, that are the points with maximum visual entropy for our purpose.

We use the Delaunay triangulation due to its ability to maximize the minimum angle so that it is guaranteed that the minimum angle is at least as large as in any other triangulation method. In Section 2.2 we will use a filter rule to reject triangles with interior angles that are smaller than a specified threshold.

### 2.2 Triangle consistency

The feature matching is the most computational time consuming part of our approach. The only valid constraint is the epipolar constraint. But using only this constraint the search for matches on the whole epipolar line is a very computationally expensive task.

To save computational time, we have to bound the search region as much as possible. We propose to search for feature matches only in the corresponding triangle in the second image which is given by the previously known feature point matches of the first image. This boundary combined with the epipolar constraint significantly limits the search region.

This sort of boundaries has advantages and disadvantages. On the one hand, it reduces the number of points to search for a match and thus, reduces the computational time. Furthermore, as long as the correct point match is inside the triangle the uncertainties of the point matching are reduced since there may be a better match on remaining region of the epipolar line. On the other hand, the point may be outside the triangle and thus it cannot be matched correctly using this triangle constraint. This problem will be discussed further in Section 2.3.

There is no guarantee that the correct feature point is really within such a triangle. In fact, some of the triangles are more likely to contain the correct point and others are more unlikely. We propose to filter out those triangles. In our approach we use the following four filter rules.

1. **Level of detail**
   Our level-of-detail concept is covered by this filter rule. A triangle which has a smaller surface area than the specified threshold will be rejected. This threshold is given by the best level of detail.
2. **Interior angle limitation**
   A triangle that contains an interior angle less than 10 degrees will be rejected. This rule is applied to the triangles in the first image and the corresponding triangles in the second image which are deformed by another perspective.
3. **Surface area ratio**
   The ratio between the surface area of the triangle in one image has to be between $\frac{2}{3}$ and $\frac{3}{2}$. Thus, if the surface area of one triangle is larger than 1.5 times the surface area of the other triangle it will be rejected.
4. **Rotation of triangle points**
   A triangle is rejected if the rotation of the triangle points in the one image is clockwise and in the other counter clockwise or vice versa.

If a triangle is rejected the corresponding triangle in the other image is rejected as well, since they are linked through the feature matches. The given thresholds have been determined by experiments.

There are two ways to handle a rejected triangle. Firstly, the search region of a rejected triangle is extended, for instance to the whole epipolar line. Secondly, the rejected triangle will not be handled further. Since our goal is to compute the first denser reconstruction as fast as possible we have chosen the second way to handle rejected triangles.

### 2.3 Point matching

To match a midpoint we compute its feature description in the first image. Thereafter, we are searching for a match in the bounded region in the second image. Therefore, we have to compute a feature descriptor for every pixel in this boundary. Depending on the complexity of those descriptors this computation would require a lot of time.

One popular feature descriptor for dense feature matching has been presented by Tola et. al. [8] and improved by Wan et. al. [9]. Although this descriptor shows good performance in these early works we could not achieve a good matching rate in our case.

Instead of using a dense feature descriptor our approach uses Fast Retinal Keypoints (FREAK) [10] which has been designed for sparse features but shows very good performance. Furthermore, its computation process is very simple and can be implemented very efficiently on GPU.

The previously discussed problem of missing points due to matching boundaries (see Sec. 2.2) can be handled in the same way occluded points or points which lie outside the image boundary are handled. In both cases the point cannot be matched correctly. To remove possible false matches we do a thresholding operation on the feature descriptor difference.

### 2.4 Triangulation and point verification

To triangulate the matched points and therefore retrieve the reconstructed 3D point we use the normalized DLT algorithm due to its simplicity and accuracy for two view reconstructions [11].

To verify the reconstructed point we project it onto two other images which also contain the three matched points of the triangle and search for the feature point in a small region around the projection. A feature point is rejected if non of the two images is confirming the feature point at the projected position.

## 3 Experiments and Results

To evaluate our approach we have generated a ground truth dataset from the City of Sights model [12] containing seven images (see Fig. 3). We have done that by rendering the scene twice. Firstly, the model has been rendered photorealistically. Secondly, for each photorealistic rendering we extracted a depth image which represents the ground truth data. Furthermore, we stored the precise camera position and orientation for each image.

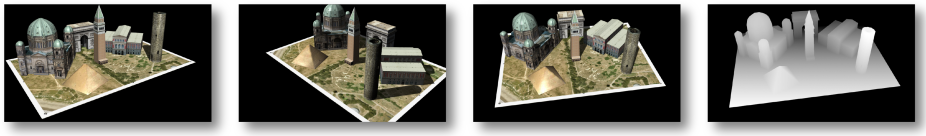
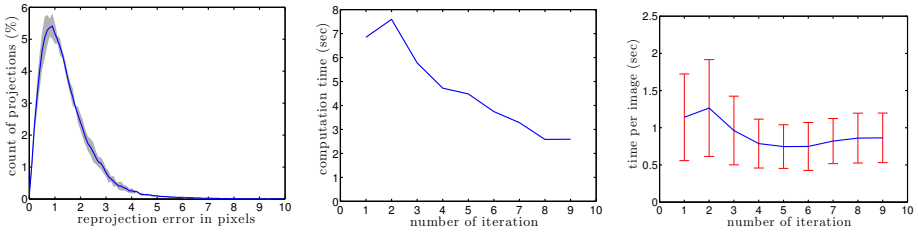


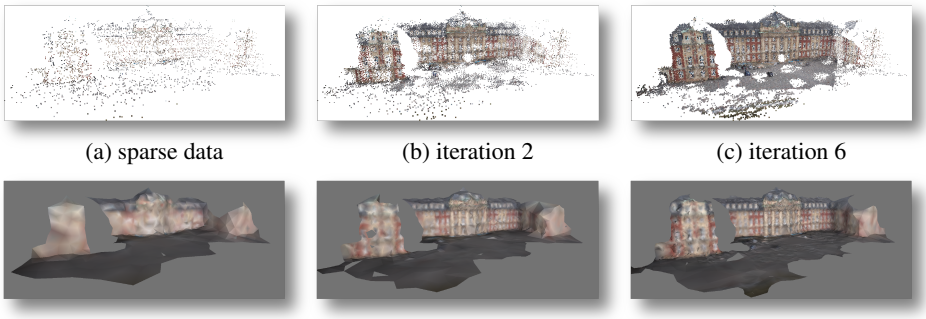
Fig. 3: Three example photorealistic renderings of the ground truth dataset. The image at the right shows a depth image corresponding to the third image.




(a) (b) (c)

Fig. 4: Evaluation: (a) Relative histogram of reprojection errors with additional standard deviation (gray area), (b) total computation time per iteration and (c) mean computation time for each image and each iteration from the ground truth dataset.

Fig. 5: Example aerial images of a testing sequence with 7 images showing the front of the castle of Münster.



(a) sparse data (b) iteration 2 (c) iteration 6

(d) sparse data (e) iteration 2 (f) iteration 6

Fig. 6: (a) - (c) Incremental dense reconstruction, (d) - (f) mesh reconstruction from the incremental dense reconstructions.

For the evaluation the black areas around the scene are ignored since there is neither correct image data nor depth data. We measure the accuracy, namely the reprojection error, i.e. the Euclidean distance between the reprojected point and correct point which is the projection of the ground truth 3D point onto the same image plane.

In comparison to the ground truth data we reach a mean accuracy of 1.499 pixels and a median accuracy of 1.220 pixels. The histogram of reprojection errors related to ground truth is shown in Figure 4(a). The standard deviation is presented by the gray area and describes the deviation of single iterations.

Further results are given by a second scene which consists of 7 real aerial images (see Fig. 5). The quality of our incremental dense reconstruction approach is shown in Figure 6(a) - (c). There, the quality of different iterations are presented. Some areas cannot be reconstructed more densely due to the triangle filters. In Figure 6(d) - (f) the process of mesh reconstruction is shown on the incrementally dense reconstructed point clouds.

Besides the accuracy we measured the computation time for each iteration. The first incremental update of the 3D points is delivered in less than 4 seconds. For the ground truth dataset the computation time for all images and one iteration is decreasing over time (see Fig. 4(b)). Whereas in iteration 2 the number of triangles has increased the number of triangles in each iteration afterwards is decreased mainly due to the level-of-detail triangle filter, i.e. more and more triangles have reached the best level of detail. Furthermore, some images have reached the best level of detail and do not need further computations. In Figure 4(c) one can see a similar result for single images, except the difference at the end of the reconstruction process. At the last iterations there is only one image left which did not have reached the best level of detail.

## 4 Conclusion

In this paper we have presented an approach to computing a dense reconstruction incrementally from wide baseline images and previously known sparse geometry. There are several applications for which our approach is applicable, especially, computation of obstacle maps and quick 3D visualization of the captured scene. While other algorithms require a lot of time to present the first result, our approach retrieves first results within a few seconds.

Furthermore, our approach reconstructs the points in a reasonable order. Instead of reconstructing neighbored points of already known scene points, the points which have the maximum distance to its neighbors are reconstructed. Thus, the major empty areas of the 3D scene gets filled earlier with information.

The feature descriptor is the most critical part in our approach since it decides whether the midpoint of a triangle could be matched correctly or not. Thus, we will concentrate on improving the used feature descriptor or on designing a new feature descriptor with better matching performance and lower computational expense.

One of the problems with this approach are the borders of objects which are unlikely to be reconstructed in general. Especially, with very wide baseline and thin objects the borders of those objects are not be reconstructed very well. On that account we

will study a hybrid approach combining the proposed method and another method for reconstructing the scene points near the borders.

## Acknowledgements

This work was developed in the project AVIGLE funded by the State of North Rhine Westphalia (NRW), Germany, and the European Union, European Regional Development Fund "Europe - Investing in your future". AVIGLE is conducted in cooperation with several industrial and academic partners. We thank all project partners for their work and contributions to the project. Furthermore, we thank Cenalo GmbH for their image acquisition.

## References

1. Roters, J., Steinicke, F., Hinrichs, K.H.: Quasi-real-time 3d reconstruction from low-altitude aerial images. In: Proceedings of the 28th Urban Data Management Symposium. (2011)
2. Vierjahn, T., Lorenz, G., Mostafawy, S., Hinrichs, K.H.: Growing cell structures learning a progressive mesh during surface reconstruction - a top-down approach. In Andujar, C., Puppo, E., eds.: EG 2012 - Short Papers, Cagliari, Sardinia, Italy, Eurographics Association (2012) 29 — 32
3. Crandall, D., Owens, A., Snavely, N., Huttenlocher, D.P.: Discrete-continuous optimization for large-scale structure from motion. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2011) 3001 – 3008
4. Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S.M., Szeliski, R.: Reconstructing rome. Computer **43** (2010) 40–47
5. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. IEEE Trans. on Pattern Analysis and Machine Intelligence **32**(8) (2010) 1362–1376
6. Newcombe, R.A., Davison, A.J.: Live dense reconstruction with a single moving camera. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. (2010) 1498 – 1505
7. Wendel, A., Maurer, M., Graber, G., Pock, T., Bischof, H.: Dense reconstruction on-the-fly. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. (2012) 1450 – 1457
8. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide baseline stereo. IEEE Trans. on Pattern Analysis and Machine Intelligence **32**(5) (2010) 815–830
9. Wan, Y., Miao, Z., Tang, Z., Wan, L., Wang, Z.: An efficient wide-baseline dense matching descriptor. IEICE Transactions **95-D**(7) (2012) 2021–2024
10. Alahi, A., Ortiz, R., Vandergheynst, P.: FREAK: Fast Retina Keypoint. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. (2012) 510 – 517
11. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2004)
12. Gruber, L., Gauglitz, S., Ventura, J., Zollmann, S., Huber, M., Schlegel, M., Klinker, G., Schmalstieg, D., Höllerer, T.: The city of sights: Design, construction, and measurement of an augmented reality stage set. In: Proc. 9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR'10), Seoul, Korea (2010) 157–163