3D Reconstruction based on Multi-View Stereo

Jonas Klötzl Regine Lendway

Figure 1. Reconstructions for the Tübingen AI Research Building (left) and for buildings of the Sand (right)

Abstract

This report provides an overview of 3D reconstruction based on Multi-View Stereo (MVS). The goal of 3D reconstruction is to model a 3D object using images as input. The type of object depends entirely on what the target is and what images are used as input. Applications range from 3D mapping and navigation to 3D printing, computational photography, video games, or cultural heritage archival. MVS constructs 3D models as point clouds based on the pixel correspondence between multiple images of an object. Recently, deep neural networks have also been used to improve the performance of MVS. In this report, however, the focus lays on two more traditional MVS reconstruction methods and their crucial steps.

1. Introduction

Inspired by the human body with its two eyes being able to perceive the environment three-dimensionally, reconstructing the world always has been a vibrant research area in the field of computer vision. Now and then the goal is to construct an algorithm that gets a collection of unstructured and highly diverse images as an input and a complete 3D reconstruction of the observed object as an output. First seminal works like the one of *Longuet-Higgins* [9] of 1981 show the first approaches to reconstructing a scene from two projections. However, new possibilities have arisen with the coming of digital cameras and higher-performing computers at the turn of the millennium. It was now possible to provide datasets like the famous Middlebury Stereo Datasets [11] with various pictures and corresponding ground truth depth values which enabled large scale empirical comparisons between different methods. As stereo setups with two views are limited to one point of view, multi-view methods were developed, resulting in building a whole 3D model of scenes and objects. The use of more pictures combined with real-world applications posed new challenges. Problems such as objects with fine surface details or crowded scenes where moving objects like persons and vehicles occur in front of the observing building or scene are investigated by *Furukawa and Ponce* [4] whose seminal work is part of this report and further explained in Section 3.1. At the same time, works of *Goesele et al.* [6] in 2007 or *Heinly et al.* [7] in 2015 show impressive results of using internet photo collections like the Yahoo 100 million image dataset [15] to reconstruct big public points of interest through multi-view stereo. Due to the increasing resolution of pictures, various methods were presented to improve the performance of MVS-algorithms. In 2012, *Bailer et al.* [1] introduced a parallelized and GPU-based depth map algorithm with normal optimization. Three years later *Galliani et al.* [5] presented a massively parallel method that enables 3D reconstruction of over ten 1.9-megapixel images using an off-the-shelf GPU in three seconds. The seminal work of *Schönberger et al.* [12], which is also covered in Section 3.2 of this report, shows how to realize a joint estimation of depth and normal for unstructured images. Apart from this, their main contribution is the pixelwise view selection for each image. Recently, Deep Learning methods show further improvement in the performance of 3D reconstruction based on multi-view stereo [16]. However, this report only covers the most seminal works which use traditional techniques.

1.1. Motivation

Over more than 30 years of research in the field of MVS are motivated by the wide range of possible applications. One area of application is the 3D mapping of surfaces. *Nakano et al.* [10] for example, use MVS to create a 3D

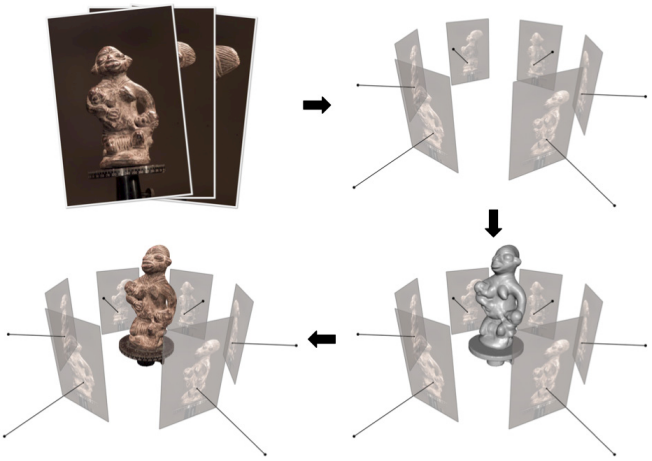

Figure 2. A multi-view stereo pipeline. Illustration from [3] Clockwise: input imagery, posed imagery, reconstructed 3D geometry, textured geometry.

model of the Nishinoshima volcano and the changes caused by eruptions. The input here is a collection of aerial photos of the volcano which are taken with an unmanned aerial vehicle. MVS also finds its application in the medical field. *Bergeles et al.* [2] implement intraoperative navigation with an advanced navigation system, wherein the intraoperative 3D structure is stably estimated from multiple stereoscopic views. Other important application areas are, for example, cultural heritage archival or the realistic modelling of objects for computer games.

2. Background

MVS reconstructions in general follow the pipeline shown in Fig. 2. The four main steps are: collecting images, computing camera parameters for each image, reconstructing the 3D object based on previous steps, and reconstructing the surface of the object. The exact implementations differ from application to application [3].

Input imagery: The first step is a useful collection of images. In the beginning of MVS research, the images used were taken under laboratory conditions. The lighting conditions and camera calibrations had to be monitored and adjustable at all times. With the improvement of hardware, great progress has been made in collecting images. The introduction of digital cameras and the overall improved computation power made it easier to capture images with better quality and resolution. These developments made it possible to move from the laboratory to small-scaled outdoor scenes, for example, the front view of a building and then even larger scenes, like entire cities [3]. Nowadays, it is even possible to use internet images as a database for the reconstruction of entire buildings [12].

Posed imagery: The second step is to determine a camera model for every input image that describes how to project a 3D point in the world into a 2D pixel coordinate in the image. The camera parameters are for example location, orientation, focal length, and pixel sensor size. Structure-from-Motion (SfM) and Visual Simultaneous Localization and Mapping (VSLAM) are two of the commonly used techniques, both depending on the correspondence cue between images and a rigid scene. VSLAM is specialized in the input of a video stream and SfM is specialized in unordered sets of overlapping images, which are used as the input for the methods in this report. For that reason, the following overview of SfM is kept short in this report: First, the 2D features in every input image need to be detected. Second, the 2D features are matched between images. These matches are then used to create 2D tracks, which are defined as the 3D coordinates of a reconstructed 3D point and the list of corresponding 2D coordinates of some images. Those tracks are then used to solve the SfM model. Lastly, the model is refined using bundle adjustment, which is not strictly part of SfM but is often required for MVS to minimize reprojection errors. The output of SfM is a sparse 3D point cloud and the reconstructed intrinsic and extrinsic camera parameters of all images [3].

Reconstruction of 3D geometry: Most methods use a sparse point cloud as the input in this step. However, voxels, meshes, depth maps, and other 3D representations are also possible [4]. The purpose of this step is to reconstruct the dense 3D shape based on the results of the previous step by matching images. For a given pixel the challenge is now to find the corresponding pixels in other images. It is necessary to have an efficient way of generating possible pixel candidates in other images and to measure how likely a given candidate is the correct match. When the camera parameters are known and the input scene is rigid the image matching problem is a 1D search. As shown in the bottom right of Fig. 2, the camera centers are known in this step. A pixel thus generates a 3D optic ray, on which the camera center and the pixel itself lie. The corresponding pixels on another image can only lie on the projection of this ray [3]. To measure how likely these two pixels match photo consistency is used, where the constraint is that matching pixels must be of a similar color [13]. As a whole, this step returns a dense 3D point cloud in the most common works, like the ones in Fig. 1 .

Reconstruction of the surface of the object: The last step is to use the depth and normal information of the fused point cloud to reconstruct the surface of the object. This step is not covered by the methods in this report but one commonly used method is Poisson surface reconstruction which creates a watertight surface from oriented point sets with a screened Poisson equation [8].

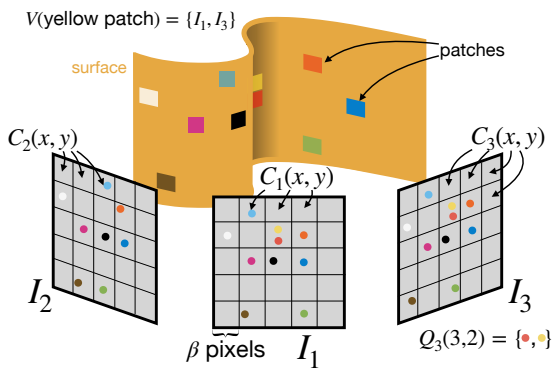

Figure 3. 3D representation of the patch model described in Section 3.1.1, adapted from *Furukawa and Ponce* [4]. The coloured dots show the image projection of the patches in the images they are visible in.

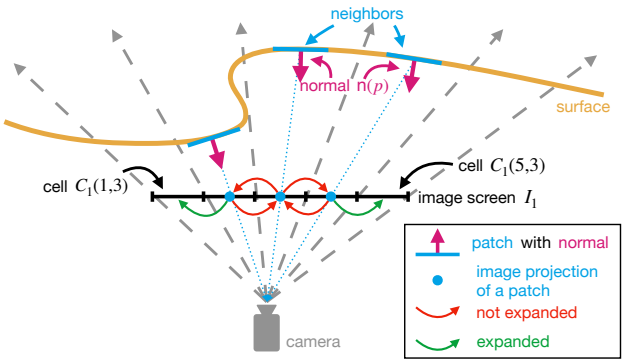

Figure 4. 2D representation of the third cell row $C_1(x, 3)$ of image I_1 in Fig. 3, showing the expansion procedure. A patch is not expanded if there already exists a patch in the image cell whose neighbour p is in the same cell. Modified illustration from *Furukawa and Ponce* [4].

3. Methods

In this section, methods of the two most seminal works in the field of Multi-View Stereo are being described. While *Furukawa and Ponce* [4] on the one hand propose the implementation of an entire MVS-Pipeline as described in Section 2, *Schönberger et al.* [12] on the other hand, focus on the reconstruction step including methods for filtering and fusion.

In the following, the focus lays on the third step in the MVS-Pipeline of the methods, the reconstruction of 3D geometry.

3.1. Furukawa and Ponce

As already mentioned in Section 2, the input of the reconstruction step of a 3D object is a sparse set of points in 3D space. Starting from these points, *Furukawa and Ponce* [4] aim to build a dense point cloud by first detecting a sufficient set of new points as described in Section 3.1.2. Then, a specific expansion procedure, which is illustrated in Section 3.1.3, determines the most suitable points to obtain a dense representation of the object.

In subsequent, the entire reconstruction step of the method of *Furukawa and Ponce* is described [4].

3.1.1 Patch Model

A 3D point is represented as a so-called patch in this method, which will be examined in the following.

In this case, a patch is defined as a local tangent plane that illustrates a specific part of a surface in an image which can be seen in Fig. 3. It is characterized by its geometrical center $\mathbf{c}(p)$, its normal vector $\mathbf{n}(p)$ pointing towards the camera, illustrated in Fig. 4, and a reference image $R(p)$, serving for comparison purposes. In addition, every patch has a set of images $V(p)$ with

$$V(p) = \{I \mid \text{Patch } p \text{ is visible in image } I\} \tag{1}$$

containing all the images in which the patch p is visible.

Furthermore, every image I_i is divided into squared image cells of $\beta \cdot \beta$ pixels. The reason for that is to receive a coarser division of each image which is beneficial for optimizing runtime (*Furukawa and Ponce* [4] chose $\beta = 2$ for all of their experiments). With $C_i(x, y)$ it is possible to directly access a specific cell in every image, see Fig. 4.

3.1.2 Identifying Cells for Expansion

Following Fig. 3, it is possible to calculate the neighboring image cells $\mathbf{C}(p)$ of a given patch p by

$$\mathbf{C}(p) = \{C_i(x', y') | p \in Q_i(x, y), |x - x'| + |y - y'| = 1\},$$

with $Q_i(x, y)$, containing the set of patches which projects into cell $C_i(x, y)$. Note that there is one case where an expansion is obsolete. Namely, if there already exists a patch p' in the image cell whose neighbor p is in the same cell $C_i(x, y) \in \mathbf{C}(p)$ as can be seen in Fig. 4.

3.1.3 Expansion Procedure

In the previous section 3.1.2, a set of neighboring image cells $\mathbf{C}(p)$ that contains different image cells $C_i(x, y)$ for which an expansion is worthwhile, is created. In the following, all the parameters of a new patch p' are initialized with the corresponding values of p or with newly calculated ones. For $\mathbf{n}(p')$, $R(p')$ and $V(p')$ the values of p can be adopted in the first step. For the center $\mathbf{c}(p')$ of the new patch p' the intercept point of the surface with the viewing ray going from the camera through the center of $C_i(x, y)$ needs to be

calculated, see blue dotted line in Fig. 4. Furthermore, a new set of images $V^*(p)$, which contains only the images of $V(p)$ whose pairwise photometric discrepancy function h is under a certain threshold α, is defined as

$$V^*(p) = \{I | I \in V(p), h(p, I, R(p)) \leq \alpha\}. \quad (2)$$

In general, the function $h(p, I_1, I_2)$ calculates the pairwise photometric discrepancy of the patch p between the two images I_1 and I_2. In this case the pairwise discrepancy between an image $I \in V(p)$ and the reference image $R(p)$ of a patch p. Given the h-function, it is now possible to define the abstract photometric discrepancy score $g(p)$ for a given patch p by

$$g(p) = \frac{1}{|V(p) \backslash R(p)|} \sum_{I \in V(p) \backslash R(p)} h(p, I, R(p)). \quad (3)$$

To optimize the parameters $\mathbf{c}(p')$ and $\mathbf{n}(p')$ it is now necessary to minimize $g^*(p)$, given as

$$g^*(p) = \frac{1}{|V^*(p) \backslash R(p)|} \sum_{I \in V^*(p) \backslash R(p)} h(p, I, R(p)), \quad (4)$$

which calculates the same photometric discrepancy score like $g(p)$ but over the reduced image set $V^*(p)$.

Next, the set $V(p')$ is extended with images found by a depth-map test which computes the corresponding depth for each image cell $C_i(x, y)$. Consequently, $V^*(p)$ needs to be updated according to the equation [2]. Note, that *Furukawa and Ponce* do not mention how this depth test is computed. The last step of the expansion procedure consists of the verification of whether $|V^*(p')| \geq \gamma$, for $\gamma \in \mathbb{N}$. If the image set $V^*(p')$ has enough pictures, the patch generation of p' was a success and is finally added to $Q_i(x, y)$ and $Q_i^*(x, y)$.

3.2. Schönberger et al. (Zheng et al.)

In contrast to the method of *Furukawa and Ponce* [4], *Schönberger et al.* [12] use photometric and geometric priors to determine a pixelwise depth and normal estimation. Building the foundation for the work of *Schönberger et al.* [12], *Zheng et al.* [17] only use photometric priors without normal estimation which leads to losses of accuracy. This is also due to the fact that *Schönberger et al.* [12] introduce a pixelwise view selection by incorporating geometric priors like triangulation, resolution and incident angles.

3.2.1 Notation

Let X^{ref} be the reference image for which the depth θ_l and normal $\mathbf{n}_l$ for every pixel l are estimated. Note that a specific pixel is not referenced by a two-dimensional tuple but rather with a one-dimensional index l. The set of M unstructured source images $X^m, m \in \{1, 2, ..., M\}$ is needed

Input: All images, depthMap (randomly initialized or from previous propagation)
Output: Updated depthMap and normals
m - image index, l - pixel index

Algorithm	**Eq.**
For $l = 1$ to L	
For $m = 1$ to M	
Compute backward message $\overleftarrow{m}(Z_{l,t}^m)$	(11)
end	
end	
For $l = 1$ to L	
For $m = 1$ to M	
Compute forward message $\overrightarrow{m}(Z_{l,t}^m)$	(10)
Compute $q(Z_l^m)$	
end	
Estimate θ_l^* and $\mathbf{n}_l^*$	(5)
For $m = 1$ to M	
Recompute forward message $\overrightarrow{m}(Z_{l,t}^m)$	(10)
end	
end	

Table 1. Algorithm to compute depth and normals for an observed object with the proposed method of *Schönberger et al.* [12], adapted from *Zheng et al.* [17].

to calculate the depth and normal of the reference image. Lastly, the M binary variables $Z_m^l \in \{0, 1\}$ define whether the pixel l of the source image X^m is selected for depth estimation of a pixel l of the reference image. $Z_m^l = 1$ means that the pixel is used for estimation, if $Z_m^l = 0$ the pixel is not used due to various issues like calibration errors or illumination aberration [17].

To understand the pseudocode of 1, the two core equations of the method of *Schönberger et al.* are examined [12], namely

$$(\hat{\theta}_l^{\text{opt}}, \hat{\mathbf{n}}_l^{\text{opt}}) = \underset{\theta_l^*, \mathbf{n}_l^*}{\text{argmin}} \frac{1}{|S|} \sum_{m \in S} \xi_l^m(\theta_l^*, \mathbf{n}_l^*) \quad (5)$$

and

$$P_l(m) = \frac{q(Z_l^m = 1) q(\alpha_l^m) q(\beta_l^m) q(\kappa_l^m)}{\sum_{m=1}^M q(Z_l^m = 1) q(\alpha_l^m) q(\beta_l^m) q(\kappa_l^m)}. \quad (6)$$

In the given context Eq. 5 calculates the optimum value of the depth θ and normal $\mathbf{n}$ for pixel l in the reference image. On the other side, Eq. 6 estimates the probability of how suitable a pixel l in the source image m is, to calculate the depth and normal for the reference image. However, the problem of these two equations is that Eq. 5 is needed to calculate Eq. 6 and vice versa. This fact motivates the use of the coordinate descent optimization method which leads to the structure of algorithm 1.

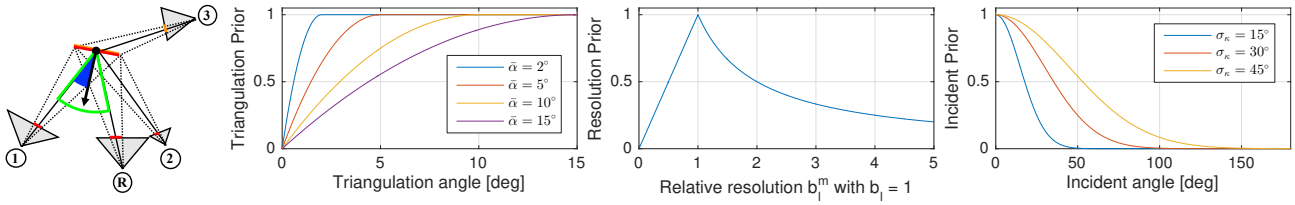

Figure 5. Leftmost graphic: red plane with reference camera (R) and source cameras (1, 2, 3); source camera 1 has high geometric priors: Triangulation (green) is good, resolution is similar to R (equal length of red lines on image screen), incident angle (blue) is below $\frac{\pi}{2}$; source camera 2 has lower geometric priors in all categories; source camera 3 cannot see the plane, lowest geometric priors around 0; on the right: graphs for geometric priors. See text for more details. Illustration from *Schönberger et al.* [12].

3.2.2 Normal and Depth Estimation with Geometric Consistency

In this subsection, Eq. 5 and the corresponding variables are explained. As stated in *Zheng et al.* [17], S is a backward sorted set of probabilities P_l^m of Eq. 6. Because $\xi_l^m(\theta_l^*, \mathbf{n}_l^*)$ is computationally expensive to calculate, empirically it suffices to take the 15 images with the highest probability, thus the first 15 values of S. Next, ξ_l^m is defined as

$$\xi_l^m = 1 - \rho_l^m + \eta \min(\psi_l^m, \psi_{\max}). \tag{7}$$

The photometric consistency term ρ_l^m compares the pixel l of the reference image with a pixel l in the source image m by using a bilaterally weighted adaption of normalized cross-correlation (NCC). The detailed equation can be viewed in Section 4.4 of *Schönberger et al.* [12]. *Zheng et al.* [17] in contrast only use normal NCC to compute the color similarity, which is, in fact, good for Gaussian noise but not for artifacts like blurred depth discontinuities.

The reprojection error ψ_l^m is defined as

$$\psi_l^m = \|x_l - \mathrm{H}_l^m \mathrm{H}_l x_l\|. \tag{8}$$

Given a reference image patch at $\mathbf{x}_l \in \mathbb{P}^2$, $\mathbf{H}_l$ calculates the transformation from the reference to the source image $\mathbf{x}_l^m = \mathbf{H}_l \mathbf{x}_l$ (see Section 4.1 in [12] for detailed specification of $\mathbf{H}_l$). Note that $\mathbf{H}_l^m$ also calculates a transformation but in the backward direction. As a result, it is possible to calculate the patch by $\mathbf{x}_l^* = \mathbf{x}_l^m \mathbf{H}_l^m = (\mathbf{x}_l \mathbf{H}_l)\mathbf{H}_l^m$ which should have a similar value like $\mathbf{x}_l$ if the geometric consistency is high. To measure the similarity the length of the vector $\mathbf{x}_l - \mathbf{x}_l^* = \mathbf{x}_l - \mathrm{H}_l^m \mathrm{H}_l \mathbf{x}_l$ is calculated as stated in Eq. 8 to obtain the difference in pixels.

Lastly, *Schönberger et al.* chose the constant regularizer to be $\eta = 0.5$ and the maximum forward-backward reprojection error to be $\psi_{\max} = 3$ px.

3.2.3 Geometric Priors for View Selection

The following subsection describes the different geometric priors $q(\alpha_l^m)$, $q(\beta_l^m)$ and $q(\kappa_l^m)$ used in Eq. 6 to enhance the view selection of pixels in source images. Comprising all the per-pixel pre-selection of pixels leads to a much more robust pixelwise view selection than the one described by *Zheng et al.* [17] who chose suitable pixels only based on color similarity.

Variational Inference: Given a restricted family of distributions, *variational inference* tries to find the optimal real posterior distribution for $q(\alpha_l^m)$, $q(\beta_l^m)$ and $q(\kappa_l^m)$ by minimizing the Kullback–Leibler (KL) divergence. Note, that this method optimizes over distributions, not over variables.

Triangulation Prior: When choosing source images regarding color similarity like *Zheng et al.* [17], the sampled images only contribute little to depth and normal estimation. Especially images with high color similarity often are taken from a nearby position to the reference image and therefore have a very small triangulation angle. After calculating the triangulation angle α_l^m between the source image m and the reference image, it is needed to build a corresponding function returning the suitability of the pixel $P(\alpha_l^m) = 1 - \frac{(\min(\overline{\alpha}, \alpha_l^m) - \overline{\alpha})^2}{\overline{\alpha}^2}$. With the *a priori* threshold $\overline{\alpha}$, it is possible to define a minimum triangulation angle. Fig. 5 shows the Triangulation Prior for different thresholds $\overline{\alpha} = 2°, 5°, 10°$ and $15°$. Note that the exact definition of α_l^m can be viewed in Section 4.2 of *Schönberger et al.* [12].

Resolution Prior: As the title of *Schönberger et al., Pixelwise View Selection for Unstructured Multi-View Stereo* [12] implies, it is crucial to take into account that the unstructured input images may have various resolutions or zoom-factors for instance. Especially, when computing the photometric consistency ρ_l^m, it is favorable to avoid over- or under-sampling by including the relative size and shape to the pixel selecting probability of Eq. 6. With $\beta_l^m = \frac{b_l}{b_l^m} \in \mathbb{R}^+$, it is possible to calculate the respective probability with $P(\beta_l^m) = \min(\beta_l^m, (\beta_l^m)^{-1})$. Note that b_l and b_l^m indicate the areas covered by the corresponding patches.

Incident Prior: The last prior deals with the fact that given a plane $(\theta, \mathbf{n}_l^m)$ it may exist a source image from whose position it is geometrically impossible to see that plane (see camera 3 in the leftmost illustration of Fig. 5). To

formalize this constraint, *Schönberger et al.* [12] introduce the incident angle κ_l^m of the pixel l of a source image m to the plane which in this case is limited to the interval $0 \leq \kappa_l^m < \pi/2$. Equally to α_l^m, κ_l^m is exactly defined in Section 4.2 of *Schönberger et al.* [12]. The appropriate probability function is then defined as $P(\kappa_l^m) = \exp(-\kappa_l^{m2}/2\sigma_\kappa^2)$. In the rightmost graph of Fig. 5, one can see that for $\sigma_\kappa = 45°$ positive incident priors are even calculated for angles with values of up to 125 degrees. This is due to the fact that κ_l^m is dependent on the constantly improving value of $\mathbf{n}_l^m$ which especially in the initial state and later may not yet have the correct value.

3.2.4 Forward Backward Algorithm

$q(Z_l^m)$ is defined as the following forward-backward algorithm:

$$q(Z_l^m) = \frac{1}{A}\vec{m}(Z_{l,t}^m)\overleftarrow{m}(Z_{l,t}^m), \tag{9}$$

with A as the normalization factor. $\vec{m}(Z_{l,t}^m)$ and $\overleftarrow{m}(Z_{l,t}^m)$ are the recursively computed terms

$$\vec{m}(Z_l^m) = P(X_l^m|Z_l^m, \theta_l, \mathbf{n}_l) \cdot \sum_{Z_{l-1}^m} \left(\vec{m}(Z_{l-1}^m)P(Z_{l,t}^m|Z_{l-1,t}^m, Z_{l,t-1}^m)\right) \tag{10}$$

and

$$\overleftarrow{m}(Z_l^m) = \sum_{Z_{l+1}^m} \Big(\overleftarrow{m}(Z_{l+1}^m)P(X_{l+1}^m|Z_{l+1}^m, \theta_{l+1}, \mathbf{n}_{l+1}) \cdot P(Z_{l,t}^m|Z_{l+1,t}^m, Z_{l,t-1}^m)\Big). \tag{11}$$

$P(X_l^m|Z_l^m, \theta_l, \mathbf{n}_l)$ is another notation for the photometric consistency term ρ_l^m used in Eq. 7 and explained in Section 3.2.2. The term, defined as $P(Z_{l,t}^m|Z_{l-1,t}^m, Z_{l,t-1}^m) = P(Z_{l,t}^m|Z_{l-1,t}^m)P(Z_{l,t}^m|Z_{l,t-1}^m)$ describes the view selection smoothness. The keynote of the probability function $P(Z_{l,t}^m|Z_{l-1,t}^m) = \begin{pmatrix} \gamma & 1-\gamma \\ 1-\gamma & \gamma \end{pmatrix}$ is to use the knowledge of the selection preferences of the previous pixels as stated in *Zheng et al.* [17]. To indicate high similarity between two neighboring pixels, γ needs to be set close to 1. However, *Schönberger et al.* [12] extend this concept by adding a temporal dependency to the pixels defining a state transition as $P(Z_{l,t}^m|Z_{l,t-1}^m) = \begin{pmatrix} \lambda_t & 1-\lambda_t \\ 1-\lambda_t & \lambda_t \end{pmatrix}$ where $\lambda_t = \frac{t}{2T} + 0.5$.

3.2.5 Computation

With all variables and probability distributions being defined, the last step is to follow algorithm 1 to obtain the overall depth and normals for an observed object.

Figure 6. Left: Sample input image of a seaside; Right: Reconstruction of the scene. Images from *Furukawa and Ponce* [4].

4. Results

For the method of *Furukawa and Ponce* [4], Fig. 6 shows the reconstruction for a seaside. Although some of the nine images in the corresponding dataset have large portions of running water in them, the method manages to successfully filter away those areas. Considering this and the fact that the work was initially presented in 2008, one has to acknowledge the striking results of the method.

On the first page of this report, Fig. 1 demonstrates the results of the method of *Schönberger et al.* [12], created with their open-source software COLMAP. For the custom reconstructions of the Tübingen AI Research Building and for buildings of the Sand an off-the-shelf DSLR camera in automatic mode is used. Note that the algorithm in general struggles to reconstruct windows or other textureless regions which can be seen in Fig. 1. For the reconstruction of the Sand, 137 images are used, for the Tübingen AI Research Building a set of 113 images, both leading to a quite impressive 3D reconstruction.

For quantitative comparison, both methods are applied on the Strecha benchmark datasets [14]. The measurement is the ratio of pixels with error less than 2cm and 10cm. On average the method of *Schönberger et al.* [12] outperforms the method of *Furukawa and Ponce* [4] by 9.325%. The biggest difference is 13.7% for the ratio of error less than 10cm on the Fountain dataset [12].

5. Summary

This report provides an overview of the most seminal works in the field of 3D reconstruction based on Multi-View Stereo. Besides, the COLMAP-Software is apllied on self taken images to demonstrate the performance of the algorithm provided by *Schönberger et al.* [12]. Covering only traditional approaches, it would be interesting to trace the improving results of future works using deep neural networks.

References

[1] Christian Bailer, Manuel Finckh, and Hendrik PA Lensch. Scale robust multi view stereo. In *European Conference on Computer Vision*, pages 398–411. Springer, 2012. 1

[2] Christos Bergeles, Philip Pratt, Robert Merrifield, Ara Darzi, and Guang-Zhong Yang. Multi-view stereo and advanced navigation for transanal endoscopic microsurgery. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 332–339, Cham, 2014. Springer International Publishing. 2

[3] Yasutaka Furukawa and Carlos Hernández. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 2

[4] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 1, 2, 3, 4, 6

[5] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 1

[6] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 1

[7] Heinly Jared, Johannes L Schonberger, Enrique Dunn, and Jan-Michael Frahm. Reconstructing the world in six days. In *CVPR*, 2015. 1

[8] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013. 2

[9] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981. 1

[10] T Nakano, I Kamiya, M Tobita, J Iwahashi, and H Nakajima. Landform monitoring in active volcano by uav and sfm-mvs technique. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(8):71, 2014. 1

[11] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002. 1

[12] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016. 1, 2, 3, 4, 5, 6

[13] Sudipta N Sinha and Marc Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 349–356. IEEE, 2005. 2

[14] Christoph Strecha, Wolfgang Von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008. 6

[15] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 1

[16] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 1

[17] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014. 4, 5, 6