TECHNISCHE UNIVERSITÄT CHEMNITZ

Informatik

Dr. Uhlman

Bachelor Thesis

Image-Based Photometric Reconstruction of Geometrically Simple Objects for Mixed Reality Visualization

Xiangyu Tong

Chemnitz, den 5. Januar 2021

Betreuer: Dr. Uhlman

Tong, Xiangyu
Image-Based Photometric Reconstruction of Geometrically Simple Objects for Mixed Reality Visualization
Bachelor Thesis, Informatik
Technische Universität Chemnitz, Januar 2021

Thanksgiving

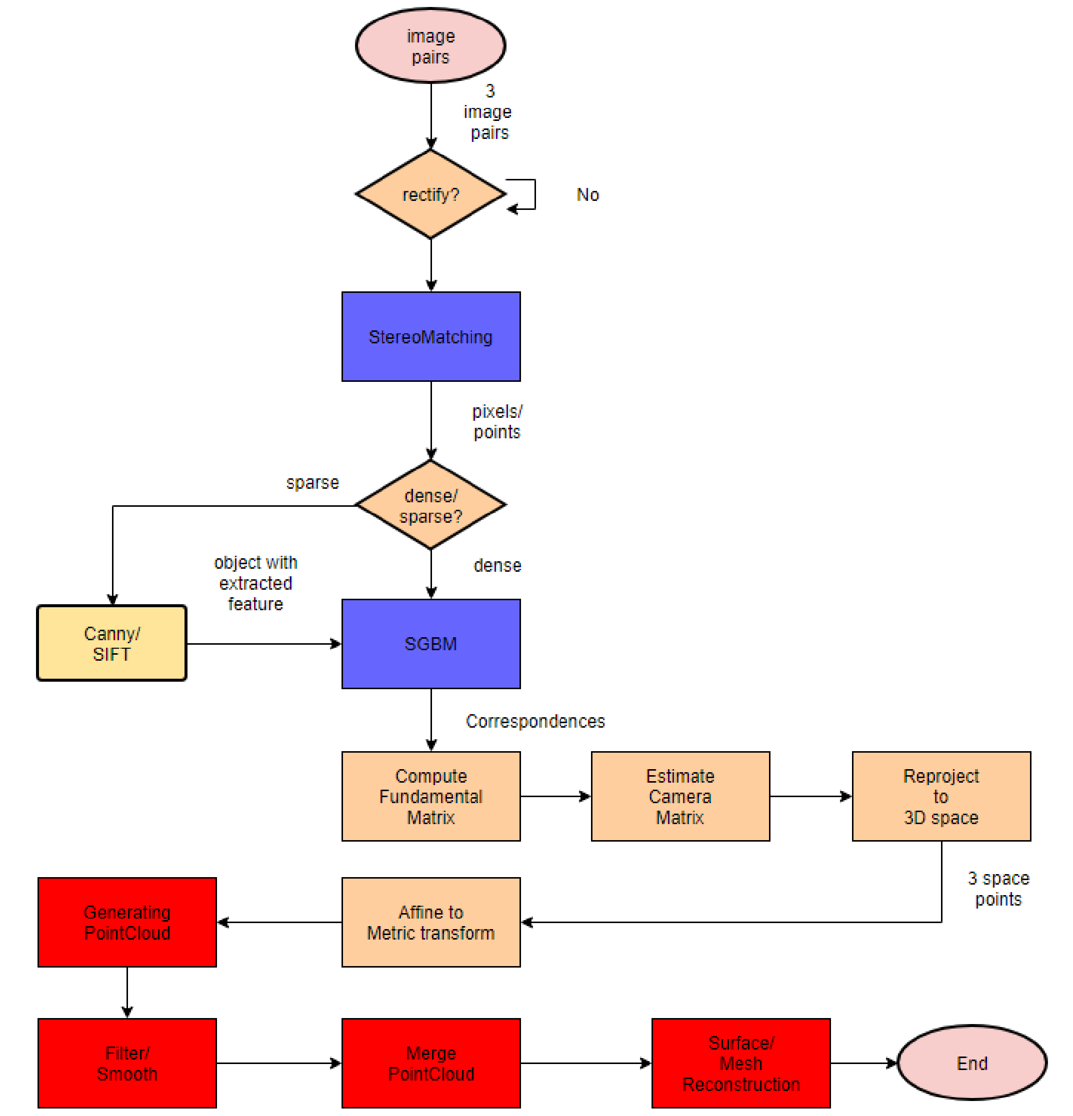
First of all, I want to thank my mentor Dr. Uhlman, who gave me a lot of valuable opinions on my bachelor thesis, so that I have a goal and direction in developing and writing this report.

Abstract

This thesis presents a complete working pipeline of image-based 3D reconstruction. The key components of the pipeline include stereo matching, generating point cloud, surface reconstruction.

Inhaltsverzeichnis

1 Overview of Working Pipeline



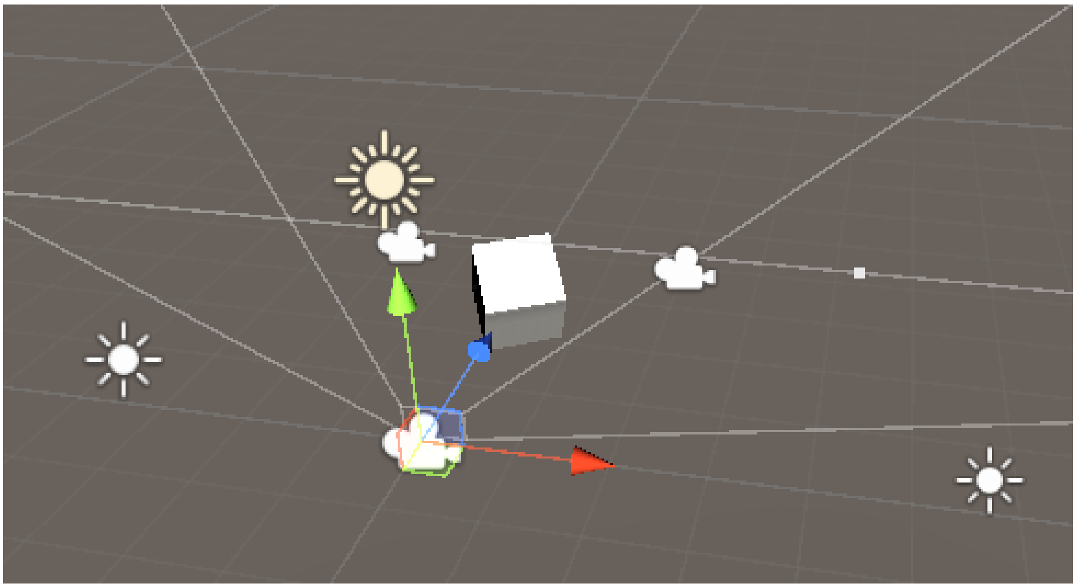
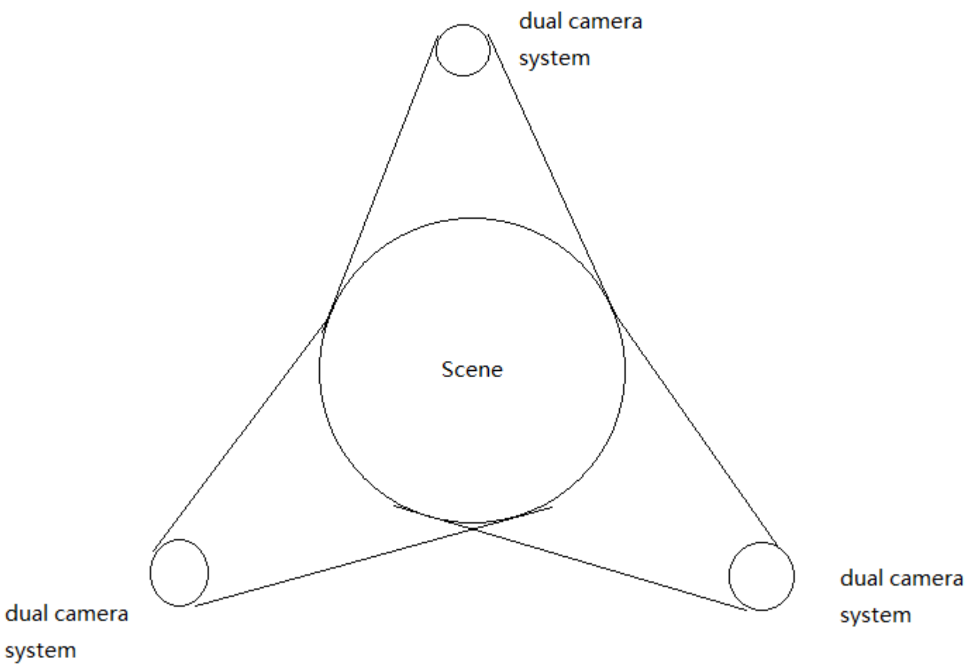


Abbildung 1.1: 3 dual camera systems of each two 120 degree



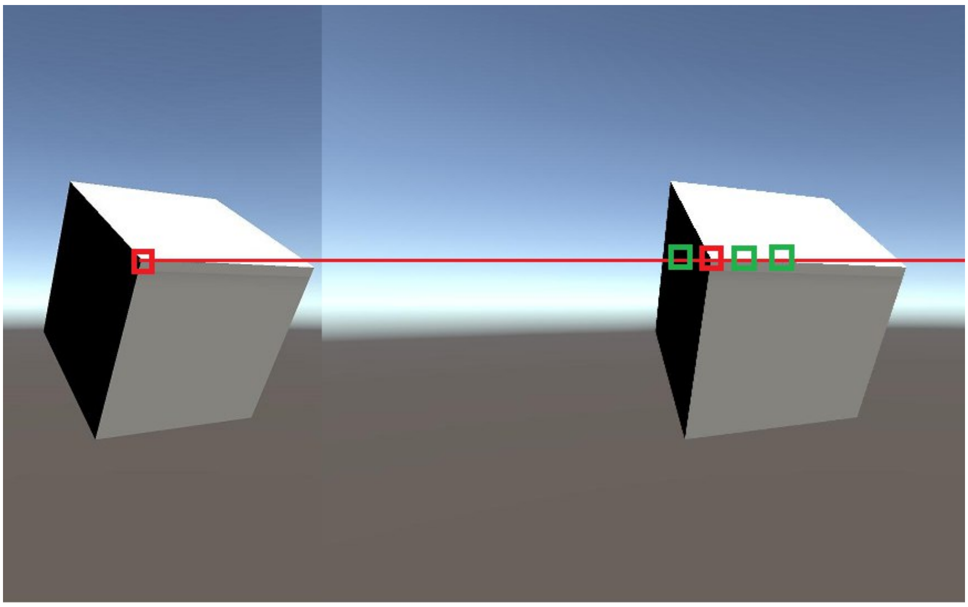
2 Photogrammetric Reconstruction

2.1 Correlation Based Algorithms: Dense Set

By using SBM we can find the pair correspondences x and x' in two images which are projections of the same 3D point X. The main idea of correlation based SBM is that, covering image into many windows and comparing the similarity of windows(blocks) between two images. Each window is a vector in an m^2 vector space. Due to epipolar constraint the searching space based on raster scan order is reduced to one dimension. The horizontal shift between correspondences is called disparity.

Similarity Measure

- Sum of Absolute Differences: $C_{SAD}(d) = \sum_{(i,j)\in W} \mid \hat{I}_L(i,j) - \hat{I}_L(i-d,j) \mid$
- Sum of Squared Differences: $C_{SSD}(d) = \sum_{(i,j)\in W} (\hat{I}_L(i,j) - \hat{I}_L(i-d,j))^2$
- Normalized Correlation: $C_{NC} = \sum_{(i,j)\in W} \hat{I}_L(i,j)\hat{I}_L(i-d,j)$



As SBM works not well in textureless areas, repeated patterns, and specularities we need improve Stereo Block Maching to make disparity values change slowly. Hence we use following 2 criteria for a good correspondences:

1. Match Quality: similarity measure between windows
2. Smoothness: adjacent pixels should usually shift about the same amount of unit

Then we define an energy function:

$$E(d) = E_d(d) + \lambda E_s(d) \tag{2.1}$$

where $E_d(d)$ is SSD distance between widows centered at (i, j) and $(i + d(i, j), j)$

$$E_d(d) = \sum_{(i,j)\in W} C(i, j, d(i, j)) \tag{2.2}$$

while $E_s(d)$ measures extent to which d is not piecewise smooth

$$E_s(d) = \sum_{(p,q)\in \varepsilon} V(d_p.d_q) \tag{2.3}$$

$$V(d_p, d_q) = \begin{cases} 0 & d_p = d_q, \\ 1 & d_p \neq d_q \ . \end{cases}$$

This optimization is in general NP-hard. However, there are still several ways to get an approximate solution typically. Here we using dynamic programming approximations. For that we define $DP(x, y, d)$ as the minimum cost of solution such that $d(x, y) = d$

$$DP(x, y, d) = C(x, y, d) + \underset{d'}{min}\{DP(x - 1, y, d') + \lambda \mid d - d' \mid\} \tag{2.4}$$

Energy functions of this form $E(d) = E_d(d) + \lambda E_s(d)$ can also be minimized using graph cuts.[3] But here we don't talk about it anymore.

In particular this energy function form is applied in SGBM(Semi-Global Block Matching) Algorithm from OpenCV. [4]

$$E(D) = \sum_{p}(C(P, D_p)) + \sum_{q \in N_p} P_1 I[\mid D_p - D_q \mid = 1] + \sum_{q \in N_p} P_2 I[\mid D_p - D_q \mid > 1] \quad (2.5)$$

where p and q are pixels, N_p are the neighbor pixels

General step of SGBM

1. Preprocessing
2. Cost Calculation
3. Dynamic Programming
4. Postprocessing

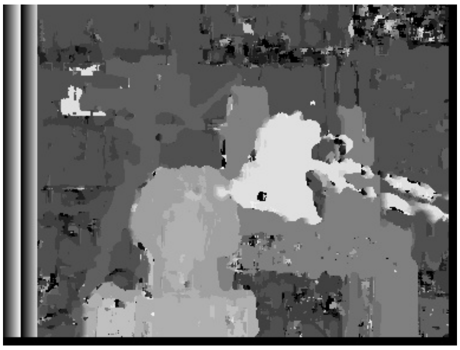


Abbildung 2.1: SAD

Abbildung 2.2: SGBM

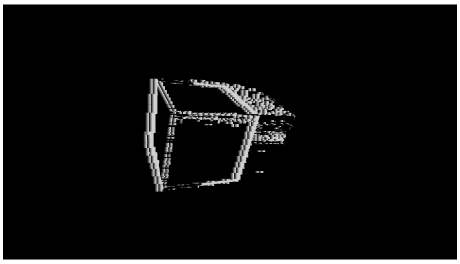


Abbildung 2.3: SAD

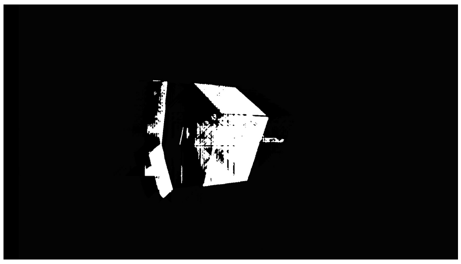


Abbildung 2.4: SGBM

2.2 Feature Based algorithms: Sparse Set

Since corners, edges and lines are commonly used as feature, the feature detection algorithms are mostly considered in our case.

2.2.1 SIFT Algorithm

2.2.2 Canny Edge Detector

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986.

- Noise Reduction with a 5×5 Gaussian filter
- Finding Intensity Gradient of the Image
- Non-maximum Suppression
- Hysteresis Thresholding

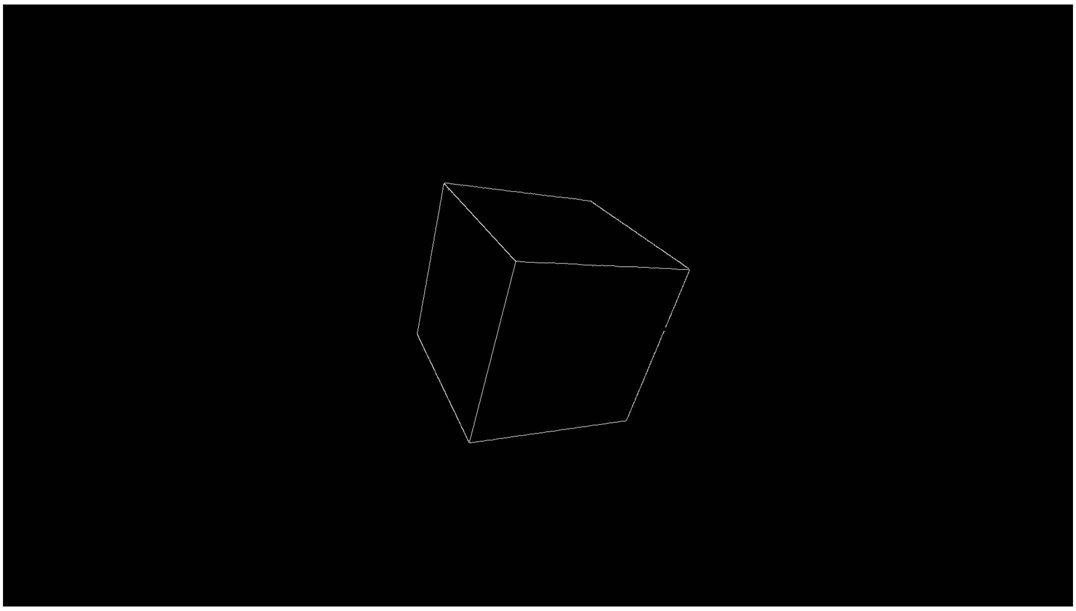


Abbildung 2.5: Edge detection using Canny

2.3 Reproject Image Points into 3D Space

This chapter describe how can we reproject image correspondences into 3D space. Suppose we get a set of image correspondences $x \leftrightarrow x'$ from **Chapter.1**. We expect that those correspondences come from a set of 3D points X_i, which are our goal and not known yet. Additionally, the calibration of cameras are unknown, means we also have no idea about the camera matrices or intrinsic matrices. For this step of our working pipeline, The task is to find the camera matrices P and P', as well as the corresponding X_i in 3D Space such that

$$x_i = PX_i \quad x'_i = P'X_i \quad for\ all\ i. \tag{2.6}$$

2.3.1 Compute Fundamental Matrix

Lemma 2.3.1. *The fundamental matrix satisfies the condition that for any pair of corre- sponding points $x \leftrightarrow x'$ in the two images*

$$x'Fx = 0 \tag{2.7}$$

This lemma shows a way characterizing the fundamental matrix without reference to the camera matrices, i.e. only in terms of corresponding pair. Besides, according to the works from Hartley and Zisserman, we have a general form for computing F

$$F = K'^{-T}[t]_\times RK^{-1} \tag{2.8}$$

In our case, since the motion of cameras from dual camera system is a pure translation with no rotation and no change in the internal parameters. We may assume that the world coordinate frame is aligned with first camera. Therefore, $P = K[I \mid 0]$ and $P' = K'[I \mid t]$. Indeed,

$$K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , t = \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix} \tag{2.9}$$

Then from (2,3)

$$\begin{aligned} F &= K'^{-T}[t]_\times RK^{-1} \\ &= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 1 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \tag{2.10}$$

2.3.2 Compute the camera matrices

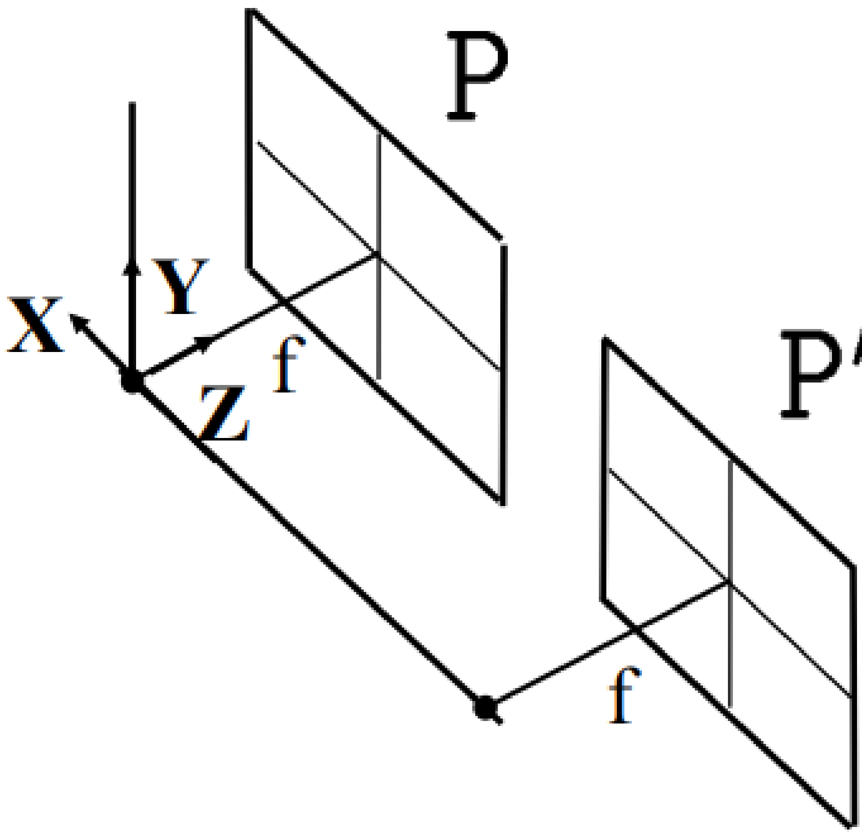


Abbildung 2.6: dual parallel camera system

As suggested by Hartley and Zisserman [1], *The camera matrices corresponding to a fundamental matrix* F *may be chosen as* $P = [I \mid 0]$ *and* $P' = [[e']_\times F \mid e']$

The epipole e is the null-space vector of F, i.e $Fe = 0$
The camera translation is parallel to the x-axis, since

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = 0 \tag{2.11}$$

so that

$$e = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{2.12}$$

2.3.3 Parallel Cameras

Even more, if the image point x is normalized as $x = (x, y, 1)^T$, then from $x = PX = K[I \mid 0]X$, the space point's coordinates are $(X, Y, Z)^T = zK^{-1}x$, where Z is the depth of the point X frome the camera center measured along the principal axis of the first camera. Since $x' = P'X = K[I \mid t]$ we have the mapping from an image point x to an image point x'

$$x' = x + Kt/Z \tag{2.13}$$

In particular for Parallel Cameras, from(2,4) we have

$$Z = f\frac{B}{d} \tag{2.14}$$

where B is the stereo baseline $B = t_x$ and d is the disparity $d = x' - x$
Subsequently

$$\begin{aligned} X &= x * Z/f \\ Y &= y * Z/f \end{aligned} \tag{2.15}$$

2.3.4 Affine To Metric Reconstruction

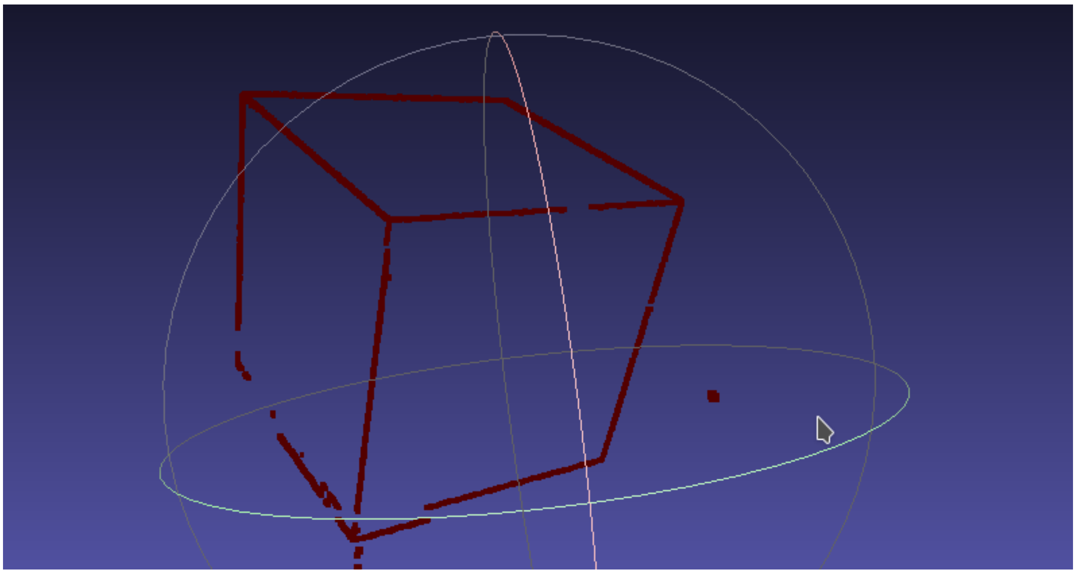


Abbildung 2.7: Reconstruction with affine ambiguity

Considering the motion of the cameras in our case, we obtain an affine reconstruction from above method. However, we still need to progress our step to metric reconstruction. Since the key to metric reconstruction is the identification of the absolute conic ω_∞, the next step will be to apply the affine transform to the affine reconstruction so that the identified absolute conic in the standard Euclidean frame. The resulting reconstruction is then related to the true reconstruction by a projective transformation which fixes the absolute conic.
In practice we consider the image of the absolute conic in one of the images. Subsequently we will show how the image of the absolute conic ω can be used to define the homography H which transforms the affine reconstruction to a metric reconstruction.

Lemma 2.3.2. *Suppose that the image of the absolute conic is known in some image to be ω, and one has an affine reconstruction in which the corresponding camera matrix is given by $P = [M \mid m]$. Then the affine reconstruction may be transformed to a metric reconstruction by applying a 3D transformation of the form*

$$H = \begin{bmatrix} A & 0 \\ 0^{(3)} & 1 \end{bmatrix} \tag{2.16}$$

where A is obtained by Cholesky factorization from the equation $AA^T = (M^T \omega M)^{-1}$

Based on Hartley and Zisserman's result, we also have

Let v_1 and v_2 be the vanishing points of two lines in an image, and let ω be the image of the absolute conic in the image. If θ is the angle between the two line directions, then

$$\cos\theta = \frac{v_1^T \omega v_2}{\sqrt{v_1^T \omega v_1}\sqrt{v_2^T \omega v_2}} \tag{2.17}$$

Indeed, this manner to metric reconstruction depends on identifying the image of the absolute conic. Following are 2 ways of doing this, which are mainly applied in our case

1. **Constraints arising from scene orthogonality** Vanishing point pairs v_1 and v_2 , which results from orthogonal scene lines, places a single linear constraint on ω:

$$v_1^T \omega v_2 = 0$$

2. **Constraints arising from the same cameras in all images** One of the properties of the absolute conic is that its projection into an image relies not on the position or orientation of the camera, but only the calibration matrix. Since in our case, all cameras have the same calibration matrix one has that $\omega = \omega'$, which means that the image of the absolute conic is the same in both images.Due to the absolute conic lies on the plane at infinity, the image may be transferred from one view to the other through the infinite homography. Hence,

$$\omega' = H_\infty^{-T} \omega H_\infty^{-1} \tag{2.18}$$

The prerequisite of forming this equation is that we already obtain an affine reconstruction.

(10.3) implies a set of linear equations for the entries of ω. Generally it places four constraints on ω, while ω has five degrees of freedom.

2.4 Point Cloud Processing

2.4.1 Point Cloud Filter

2.4.2 Point SET Registration

2.4.3 Merge 3 Point Clouds

2.5 Generating triangle meshes

The technology of triangulation for a set of discrete points is significant in computer graphics.

Definition 2.5.1 (Triangulation)**.** A **triangulation** of a finite point set $P \subset \mathbb{R}^2$ is a collection S of d-simplices, such that

(1) $conv(P) = \bigcup_{T \in S} T$; (Union Property)

(2) Any pair of these simplices intersects in a common face; (Intersection Property)

Especially, due to the exceptional geometric properties, the Delaunay triangulation is recongnized as the best triangulation.

Definition 2.5.2. A triangulation of a finite point set $P \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the circumcircle of every triangle is empty, that is, there is no point from P in its interior.

There are 3 common algorithms which can implement Delaunay triangulation. Here in this chapter we only talk about Delaunay algorithm based on region growing space [5] Brassel and Reif proposed the main idea of the algorithm in 1979. The specifig steps are given as follows:

(1) For a arbitrary point of point set, find its nearest point and forms basic Delaunay-edge

(2) witch Delaunay criteria search the third point which is located in the right position of basic edge

(3) create Delaunay triangle and connect the points of basic edge with the third point to form new basic Delaunay-edges

(4) Iteration with step 2 and 3 until all the basic edges have been proceed

2.5.1 Greedy Projection Triangulation

In this chapter we use the method in PCL library called triangulation greedy algorithm, which is a projection of the original point cloud fast triangulation algorithm. The algorithm assumes a smooth surface, a uniform density of point cloud, and smoothing the surface can not be fixed in the holes while triangulation. This method is based on Delaunay algorithm on region growing space:

- A point V and its vicinity are projected onto a target plane through the normal, denoted as the point set $\{S\}$
- From $\{S\}$ forms all the edges between each two points and linearly sorted in ascend order of distance
- Add the shortedt edge at each stage and remove it from the memory. If the edge satisfies the criteria of Delaunay-Triangulation, then it is added to the triangulation, otherwiese, it is removed.
- obtaining a triangular mesh model according to the topological connection relations of points in the plane

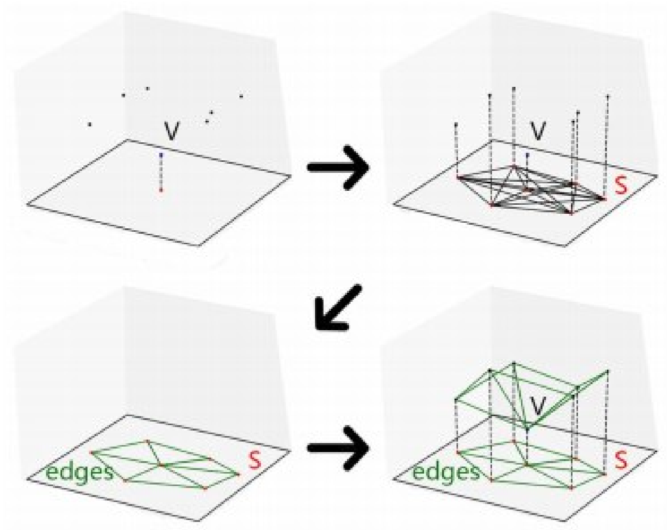


Abbildung 2.8: Greedy projection triangulation [5]

2.5.2 Poisson Reconstruction

3 Issues and possible solutions

4 Conclusion

5 Notation

Literaturverzeichnis

[1] Khalid, Sayood(2006) et al., Introduction to Data Compression.

[2] Thomas H. Cormen, Charles E. Leiserson (2009), introduction to Algorithms, Third Edition, MIT Press.

[3] A detailed and very accessible overview of Huffman codes is provided in Huffman Codes, by S. Pigeon [36], in Lossless Compression Handbook.

[4] Details about nonbinary Huffman codes and a much more theoretical and rigorous description of variable-length codes can be found in The Theory of Information and Coding, volume 3 of Encyclopedia of Mathematic and Its Application, by R.J. McEliece.

[5] Brassel, Reif(1979) A Procedure to Generate Thiessen Polygons. Georaphical Analysis, 11, 289-303.