Stereo Reconstruction: from stereo 2D images to a 3D model

Sebastian Steinmüller
Technical University of Munich
Munich, Germany
Sebastian.steinmueller@tum.de

Ha Young Kim
Technical University of Munich
Munich, Germany
hayoung.kim@tum.de

Dan Halperin
Technical University of Munich
Munich, Germany
dan.halperin@tum.de

August 2022

Stereo reconstruction is a widely researched topic and provides a powerful tool for reconstructing 3D shapes out of 2D models. In this project, we aim to investigate a possible pipeline for creating a high-resolution 3D point cloud from a stereo setup of 2D images and compare the different matching algorithms that find the corresponding key points between them.

1 Introduction

Stereo reconstruction is an important cornerstone in the field of 3D reconstruction in computer vision. Although direct 3D methods, such as processing 3D point clouds generated by a LiDAR scanner [14], already exist and are being intensively researched, the use of relatively inexpensive cameras provides available and efficient access to the 3D domain. Starting with only two-dimensional images, it is possible to reconstruct the depth of a 3D point in the world coordinate system, with respect to the corresponding pixels in images of a scene or object, that are taken from different angles. In addition, creating a depth map from two rectified images in a parallel setup is useful for real-world scenarios, e.g., SLAM [5] for depth measurement in autonomous driving. Large and interesting works in this area include "Building Rome in a day" [17] and Microsoft-Kinect [18].

In this project, we aim to follow a selected pipeline for reconstruction, as described later in section 2, to gain hands-on experience with one of the most fundamental concepts in computer vision, to strengthen our base knowledge, in order to have a better understanding of advanced topics. Our experiments focus on the reconstruction of a given scene, which is supplied by [15] [1], and compare different matching methods such as **NSSD**, **NCC**, and **FLANN**, which are described in more detail in subsection 2.3, to determine how well they perform on the given task.

2 Related work and method

2.1 A Stereo-images setup

Stereoscopic imaging is a technique used to create or enhance the illusion that an image has a depth, by observing two similar images, that differ with a small disparity, as seen in Figure 1. Although calibration of multiple cameras is possible, for simplicity it is common to use the same camera that is moved linearly and only slightly in the scene.

Our work uses the prepared stereo configuration from the **Middlebury Stereo Datasets** [1].

2.2 Keypoints detection

3D reconstruction requires the recovery of depth data for each pixel in a given image. This is done by triangulating between the corresponding keypoints in the binocular setup of images, and then recovering the corresponding affine transformation between them in world coordinates. Several algorithms have been proposed for extracting the most interesting and unique keypoints in a given image, applied to each of the images in the setup individually. Common algorithms are either the **Harris corner detector** [7] or the **Scale Invariant Feature Transform** (SIFT) [10]. In our project, our pipeline is based on the latter SIFT algorithm, which is invariant to image scaling and rotation. However, it is only partially invariant to changes in lighting and 3D camera position.

2.3 Matching algorithms

To recover the transformation between two images, we chose the 8-point algorithm (subsection 2.4), which requires a set of matching points between the images. A common technique is to examine each pixel within a window of adjacent pixels using brute-force iterations, similar to a convolutional kernel, whose size one sets as a hyperparameter. To this end, several algorithms have been proposed to evaluate the similarity of the different windows of images.

The **Normalized Sum of Squared Differences** (NSSD) [11] computes the sum of squares between pixels in the kernels:

$$NSSD(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{m}((W_l(i,j) - \mu_l) - (W_r(i,j) - \mu_r))^2$$

where u and v are the pixel coordinates in the left image, n and m are the height and width of the windows, respectively, and W_l and W_r are the corresponding kernels from each image. μ_l and μ_r are the corresponding means of the kernels.

The **Normalized Cross Correlation** (NCC) [11] computes a correlation value between two kernels:

$$NCC(u,v) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} W_l(i,j) \cdot W_r(i,j)}{\left(\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} W_l(i,j)^2)}\right) \cdot \left(\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} W_r(i,j)^2)}\right)}$$

The **Fast Library for Approximate Nearest Neighbors** (FLANN) [16] is a library for approximate nearest neighbor search in high-dimensional spaces and implemented in OpenCV. Unlike the first two algorithms, this algorithm works directly with the keypoints themselves.

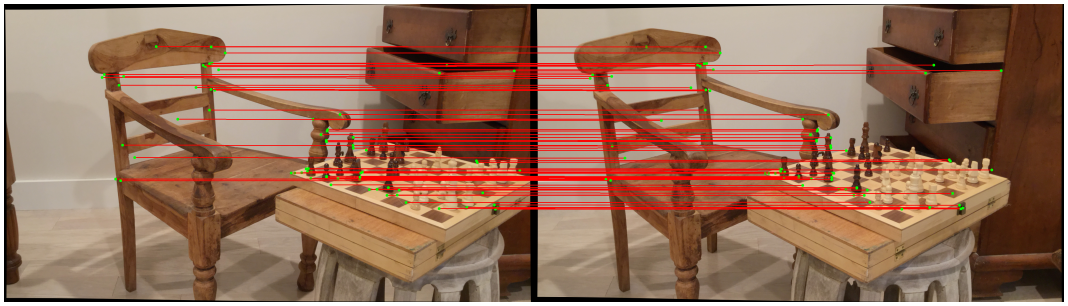


Figure 1: Matching corresponding key points between binocular images with the NSSD algorithm, over 500 points, that were extracted with SIFT [10]. For more information, see Table 1.

2.4 The 8 points algorithm

This algorithm is fundamental in the recovery of 3D data [2]. Given at least 8 pairs of matching points from two different point clouds, one could estimate the Essential matrix [9] $E = \hat{T}R$, such that for each pair of matched points $x_l \in I_l$, $x_r \in I_r$, the epipolar constraint:

$$Rx_l + t = x_r \rightarrow \hat{T}Rx_l + \hat{T}t = \hat{T}x_r \rightarrow x_r^T\hat{T}Rx_l = x_r^T\hat{T}x_r \rightarrow x_r^T\hat{T}Rx_l = 0 \rightarrow x_r^T E x_l = 0$$

must hold, with no regard to the depth coefficient, and in a calibrated cameras' setup [4].

Then, one could extract R, which is the rotation between the two points clouds and $\hat{T}$, that is the skew-symmetric representation of the entries of the translation matrix t.

For better approximation, we used the **Least Median of Squares** (LMedS) [12] algorithm to filter the outliers, much like RANSAC [6]. Here, LMedS is used for faster computation.

2.5 Rectification and epipolar geometry

Since the depths of the points in world coordinates are still unknown, the projection of each key point $x \in I_l$ onto I_r is denoted by a line of possible placements, which we call the **epipolar line**. The baseline between two optical centers intersects with the image planes at the **epipoles**. To perform triangulation and recover the depth of each pixel, the images must be **rectified**, which is done using the recovered rotation and translation between the images. Then the images are transformed to the same plane and are parallel in the world coordinates, resulting in each epipole being estimated at infinity and the epipolar lines appearing parallel and with the same y horizontal coordinate value in the world.

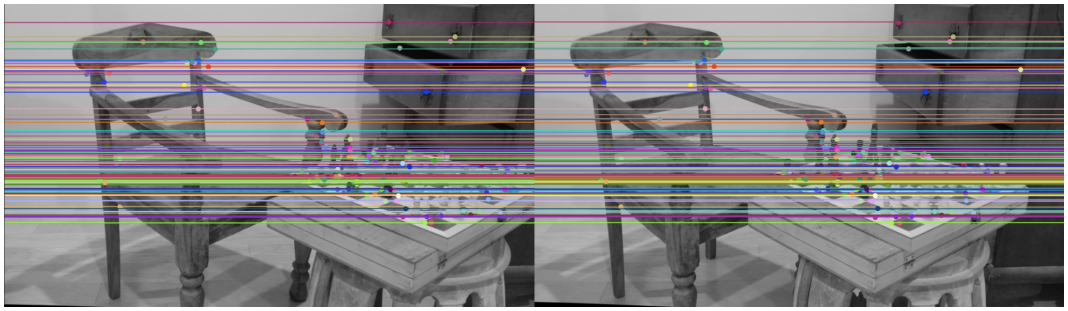


Figure 2: Rectified images. The epipolar lines of matching points seem parallel, with the same y coordinate.

2.6 disparity

After the rectification, for most pixels in the left image, there is a corresponding pixel in the right image in the same horizontal line. The difference in the x coordinate of the corresponding pixels is known as **disparity** [13]. This value is proportional to the distance of the objects from the camera, i.e. pixels that are closer to the camera in 3D will have a smaller disparity value than objects that are further away. The disparity can be defined by the following equation:

$$d = \frac{bf}{z}$$

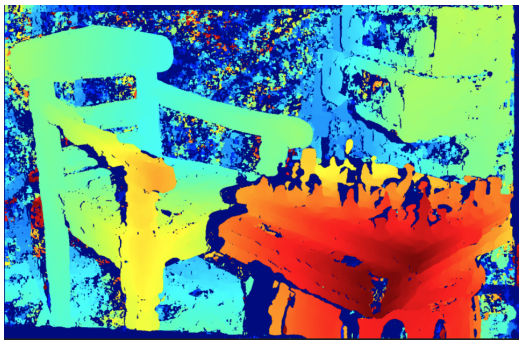
where z is the distance of the object from the camera (recovered at earlier stages), b is the baseline between the points and f denotes the focal length of the camera.

2.7 Reconstruction to 3D

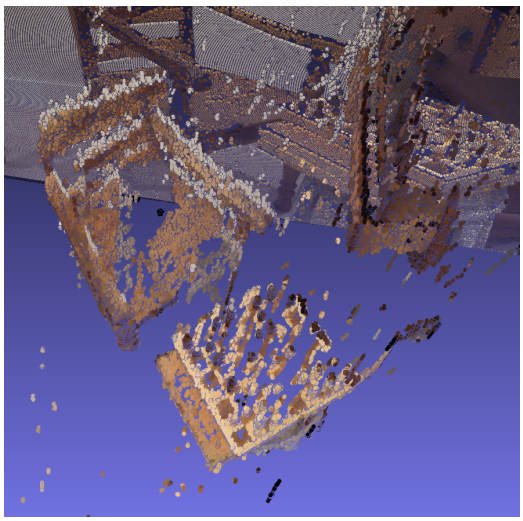
Using the disparity and rectified images, one can reconstruct the 3D coordinates by using the **disparity to depth** matrix Q. The Q is nothing but a combination of the intrinsic and the baseline:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c'_x)/T_x) \end{bmatrix} \cdot \begin{bmatrix} u \\ v \\ disparity(x,y) \\ 1 \end{bmatrix}$$

Where T_x is the length of the baseline between the origins of the two cameras, f is the focal length, and c_x, c_y are the coordinates of the principal point (the center of the image) of the right camera, while c'_x is the x coordinate of the left camera. The y coordinate is not needed because the images are rectified, so $c_y = c'_y$.



(a) Disparity map. Hotter colors represent a smaller change between corresponding pixels, hence the object they represent is closer to the camera.



(b) Image reconstruction as a point cloud, top view.

Figure 3: A disparity map and a 3D reconstructed image.

Then, we could calculate the projection of these pixels onto the 3D world coordinates. For each pixel (u, v), the projection function f is:

$$f(u,v) = \begin{bmatrix} \frac{X}{W} \\ \frac{Y}{W} \\ \frac{Z}{W} \end{bmatrix}$$

3 Experiments and evaluation

Given the pipeline presented in section 2, our goal in this project is to compare the three different matching algorithms presented in subsection 2.3. To this end, we have collected data for two measurements. The first is the number of well-matched keypoints (with a high match score) out of all candidates, extracted by SIFT [10]. The second is the Hausdorff distance [8], a distance-based scoring metric for finding the closeness of two point clouds, which, in comparison to the Chamfer Distance metric, takes the max distance between the two.

Let $A, B \subset \mathbb{R}^n$. Then the one-sided Hausdorff distance between A and B is defined as

$$d(A \to B) = \sup_{x \in A} \inf_{y \in B} |x - y|.$$

Hence:

$$d_H(A, B) = \max(d(A \to B), d(B \to A))$$

We use this metric to compare the reconstruction from the disparity we recovered to the reconstruction from a 3D model with a ground truth point cloud. This ground truth point cloud is obtained using disparity from non-rectified images and Q matrix constructed only with camera intrinsics. The Q matrix can be used for both, since it is derived directly from the intrinsic and calibration data. Our results are displayed in Table 1.

In implementing the pipeline, we relied heavily on OpenCV [3], which makes it easy to use and read the code. However, we implemented most of it ourselves before recognizing, that OpenCV implementations exist.

4 Analysis

Considering Table 1, it can be seen, that for lower amounts of initial keypoints the hausdorff distance is comparably high, while it levels out, when this number is increased. It needs to be mentioned, that

Metric	**Key points / Candidates**	NSSD	NCC	FLANN	GT
NSSD	7/250	-	0.768	0.403	0.22
NCC	73/250	0.768	-	0.366	0.92
FLANN	55/250	0.403	0.366	-	0.554
NSSD	36/500	-	0.123	0.194	0.159
NCC	165/500	0.123	-	0.201	0.169
FLANN	135/500	0.194	0.201	-	0.208
NSSD	118/1000	-	0.141	0.141	0.143
NCC	318/1000	0.141	-	0.134	0.183
FLANN	245/1000	0.141	0.134	-	0.15

Table 1: Evaluation of the Hausdorff distance metric for the **Chess data** [8] between all 4 points clouds, that correspond to the different matching methods (subsection 2.3). **Lower is better**. In addition, for each matching method we note the number of keypoints that were considered as real matches, out of all found candidates.

Metric	**Key points / Candidates**	NSSD	NCC	FLANN	GT
NSSD	68/250	-	0.192	0.223	0.237
NCC	103/250	0.192	-	0.225	0.207
FLANN	98/250	0.223	0.225	-	0.256
NSSD	119/500	-	0.204	0.196	0.216
NCC	206/500	0.204	-	0.206	0.234
FLANN	203/500	0.196	0.206	-	0.225
NSSD	264/1000	-	0.207	0.184	0.207
NCC	438/1000	0.207	-	0.182	0.207
FLANN	403/1000	0.184	0.182	-	0.238

Table 2: Evaluation of the Hausdorff distance metric for the **Artroom data** [8] between all 4 points clouds, that correspond to the different matching methods (subsection 2.3). **Lower is better**. In addition, for each matching method we note the number of keypoints that were considered as real matches, out of all found candidates.

the reconstruction based on 250 keypoints doesn't have a good enough quality to be able to recognize the scene. Therefore the outlier performance doesn't have a high significance and the results for larger amounts of keypoints are of more interest for this data. All three matching methods lead to a similar result and besides NSSD, that performed the best in all cases, it can't be determined, if NCC or FLANN performed better than each other.

To validate this data, we also performed a second evaluation based on the artroom1 data from the Middlebury Stereo dataset [1]. Here similar results could be achieved with the exception, that NSSD was not exclusively the best performing method, being outperformed by NCC for 250 keypoints (0.237 compared to 0.207 respectively). Also the reconstructed point clouds for 250 keypoints have been valid for this data.

The performance of NSSD is especially remarkable, because in almost all tests it achieved one of the top performances with at the same time significantly less matched keypoints.

Furthermore, it needs to be noted, that in general the Hausdorff distance between the individual clouds decreases with an increasing number of used initial keypoints. This could not be confirmed by our second test. The reason might be, that for the artroom1 dataset the matching methods lead to a much larger amount of matched keypoints, especially for a smaller amount of initial keypoints. For example, NSSD matched 7 keypoints out of 250 for the chess dataset and 68 keypoints for the artroom1 dataset.

5 Conclusion

In this project, we followed a well-known pipeline to reconstruct a 3D model from 2 images in a stereo setup from the Middlebury stereo dataset. In doing so, we implemented the different steps and visualized them to better understand the big concepts behind it. To this end, we implemented 3 different matching algorithms to compare and determine which algorithm is better suited for the task at hand. We have seen, that NSSD performed best, even outperforming the OpenCV implementation of FLANN, while also the overall performance can differ greatly based on the chosen dataset. In continuation of this project, an expansion towards multi-view stereo, based on multiple images is possible, as well as an approach towards a more extensive analysis of point matching algorithms.

References

[1] Middlebury stereo datasets. https://vision.middlebury.edu/stereo/data/. 2021 mobile datasets - 24 datasets obtained with a mobile device on a robot arm.

[2] In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] R. Deriche, Z. Zhang, Q. T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In Jan-Olof Eklundh, editor, *Computer Vision — ECCV '94*, pages 567–576, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[5] Hugh F. Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam) : Part i the essential algorithms. 2006.

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981.

[7] Christopher G. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[8] Daniel Kraft. Computing the hausdorff distance of two sets from their distance functions. 30(01):19–49, mar 2020.

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

[10] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[11] Badrul Mohamad, Shahrul Yaakob, Rafikha Aliana A. Raof, A. Nazren, and Mohd Wafi Nasrudin. Template matching using sum of squared difference and normalized cross correlation. pages 100–104, 12 2015.

[12] Wei Mou, Han Wang, G. Seet, and Lubing Zhou. Robust homography estimation based on nonlinear least squares optimization. volume 2014, pages 372–377, 12 2013.

[13] M Mozammel, Hoque Chowdhury, Md Al-Amin, and Md Bhuiyan. A new approach for disparity map determination. *DAFFODIL INTERNATIONAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY*, 4, 01 2009.

[14] Santiago Royo and Maria Ballesta-Garcia. An overview of lidar imaging systems for autonomous vehicles. *Applied Sciences*, 9:4093, 09 2019.

[15] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001.

[16] Arul Suju and Hancy Jose. Flann: Fast approximate nearest neighbour search algorithm for elucidating human-wildlife conflicts in forest areas. pages 1–6, 03 2017.

[17] R. Szeliski, B. Curless, S. M. Seitz, N. Snavely, Y. Furukawa, and S. Agarwal. Reconstructing rome. *Computer*, 43(06):40–47, jun 2010.

[18] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia - IEEEMM*, 19:4–10, 02 2012.