3D reconstruction from 2D images

P. Brandauer, S. Jasonarson, R. Jöchl & P. Mayer

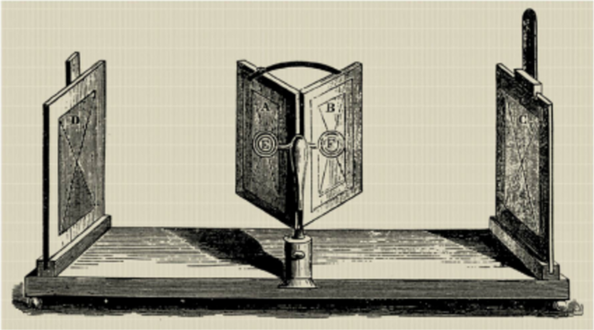
Master Seminar 1

February 22, 2019

Outline

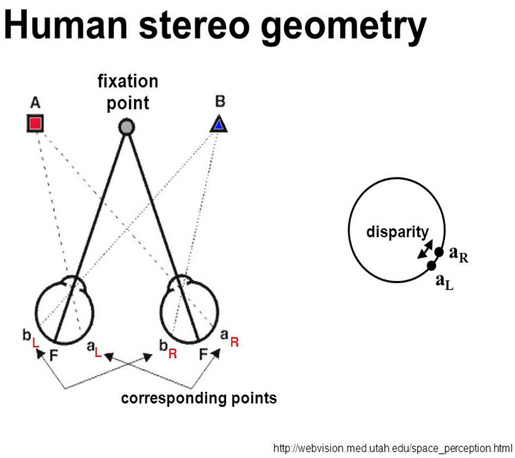
Introduction

First developments of getting 3D information from 2D images was done by Charles Wheatstone in 1838 [Wheatstone, 1838]. This developed into stereoscopy.



Stereoscopic invention with mirrors [Moratto, 2019a]

Simple model of the human stereoscopic perception



Simple model of the eye [Kalloniatis and Luu, 2019]

Stereoscopic Image - example

Example of an image taken with 2 cameras [Stereotron, 2019]

Modern Hardware to capture 3D information

- Fujifilm finepix real 3D W3 [Fujifilm, 2019]
- FPV3DCAM Blackbird 2 (for flying drones, etc.) [FPV, 2019]
- Panasonic AG-3DP1 production video camera [Panasonic, 2019]
- Photo from an Xbox Kinect with infrared dots [BZippo, 2019]

Algorithm overview

- The principles: [Scharstein et al., 2001]
  1. Calibrate cameras
  2. Find epipolar line.
  3. Rectify the pair of images - Reduce matching to a 1D search problem.
  4. Matching Cost computation: Calculate disparity for each pixel of the rectified pair.
  5. Cost aggregation: Connects the matching costs within a certain neighbourhood.
  6. Estimate disparity map: Minimizing a matching cost globally, semi-globally or locally.
  7. Disparity refinement - Smoothing.

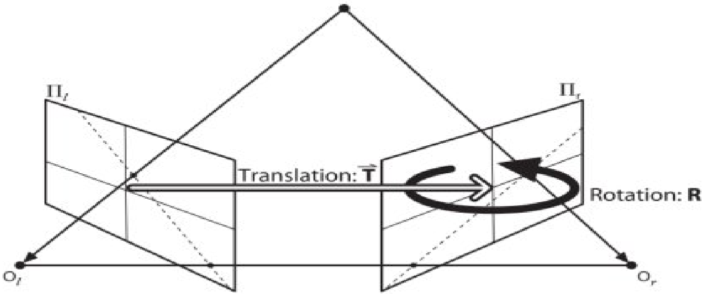
Calibration of Cameras

Everything about gathering 3D data from 2 cameras is easier if the 2 cameras have the same properties:

- focal length
- aperture
- same horizontal axis
- same distance from photographed object

It means less work needs to be done on the image rectification side.

Image plane transformation



Example transformation of an image plane [OpenCV, 2019].

- Each image plane can be transformed to another image plane by translation and rotation
- The transformation can be described by the Essential matrix E
- The Fundamental matrix additionally takes camera parameters into account

[OpenCV, 2019]

Epipolar Geometry

- Describes the geometric relationship of two (or more) images of the same object
- Mainly used to extract 3D information of 2D images by correspondence analysis e.g. in Machine Learning or Robotics

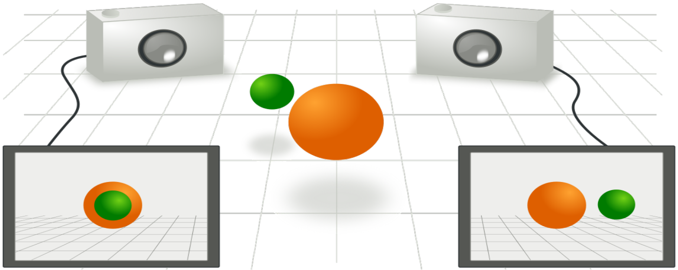
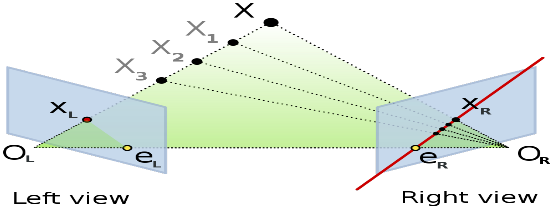


Illustration of an use case of Epipolar Geometry. Two cameras take an image of an object constellation from different perspectives [Nordmann, 2008].



Example of Epipolar Geometry with two image planes [Nordmann, 2007]. O denotes the respective image center and e the respective epipole of each image plane.

- Assumption: Relative position of the image planes is known
- An infinite set of points $\{X_i | i \in \mathbb{N}\}$ is projected onto one point X_L in the left image plane
- The mapping of each of these points onto the right image plane is a possible mapping of X (epipolar line of X_R)

[OpenCV, 2019]

Image Rectification

- Goal: make epipolar lines horizontal to simplify matching
- I.e. warp image planes in a way that there is only a horizontal displacement
- Can be done with the Essential matrix if the cameras are calibrated, i.e. the cameras have the same parameters
- Can be done in general with the Fundamental matrix
- Properties of the transformed images:
  - All epipolar lines are parallel to the horizontal axis
  - Corresponding points have identical vertical coordinates

[Fusiello et al., 2000]

Camera Calibration

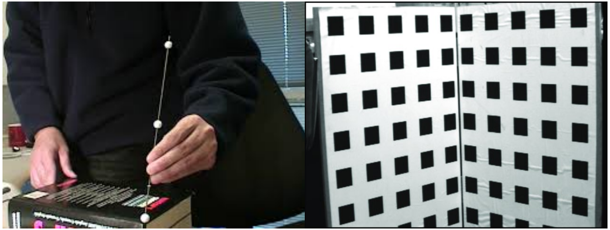
- Goal: Find transformation which turns the images of the camera into that of a ideal pinhole camera
- Requires lens and image sensor parameters which determine which incoming light beam is projected to which pixel in the image plane
- Parts of the transformation:
  - Camera matrix (matrix with the camera parameters)
  - Distortion vector
- Types of calibration:
  - Self-calibration (without calibration object)
  - Camera resectioning (with calibration objects)

[Mathworks, 2018]

Camera Resectioning

- Determination of the camera parameters by real world objects with known 3D coordinates
- Common calibration objects:
  - Spatial rectangular patterns (3D)
  - Planar checkerboard pattern (2D)
  - Bars with at least three collinear spherical objects (1D)

[Mathworks, 2018]



Bar with three collinear spherical objects as 1D calibration object (left) [Zhang, 2001] and spatial rectangular pattern (right) [Lioulemes, 2013].

Stereo Matching

- Goal: Find the displacement of objects in both images (disparity)

Shows an image and the corresponding disparity image [mid, 2019].

Matching Cost Computation

- Calculate the cost for each pixel for every possible disparity

Rectified images with one location of the epipolar-line [Moratto, 2019b].

Matching Cost Computation

- Simplified example, use absolute pixel-difference as cost function based on [Hirschmuller, 2005][Moratto, 2019b].

Image I1

8	10	8	60	62	9	7	8

Image I2

8	10	58	64	6	10	6	10

- Calculation of the cost for all possible disparities:

Disparity / Pixel	1	2	3	4	5	6	7	8
0								
1								
2								
3								

Matching Cost Computation

- Simplified example, use absolute pixel-difference as cost function based on [Hirschmuller, 2005][Moratto, 2019b].

Image I1

8	10	8	60	62	9	7	8

Image I2

8	10	58	64	6	10	6	10

- Calculation of the cost for all possible disparities:

Disparity / Pixel	1	2	3	4	5	6	7	8
0	0	0	50	4	56	1	1	2
1								
2								
3								

Matching Cost Computation

- Simplified example, use absolute pixel-difference as cost function based on [Hirschmuller, 2005][Moratto, 2019b].

Image I1

8	10	8	60	62	9	7	8

Image I2

8	10	58	64	6	10	6	10

- Calculation of the cost for all possible disparities:

Disparity / Pixel	1	2	3	4	5	6	7	8
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2								
3								

Matching Cost Computation

- Simplified example, use absolute pixel-difference as cost function based on [Hirschmuller, 2005][Moratto, 2019b].

Image I1

8	10	8	60	62	9	7	8

Image I2

8	10	58	64	6	10	6	10

- Calculation of the cost for all possible disparities:

Disparity / Pixel	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3								

Matching Cost Computation

- Simplified example, use absolute pixel-difference as cost function based on [Hirschmuller, 2005][Moratto, 2019b].

Image I1

8	10	8	60	62	9	7	8

Image I2

8	10	58	64	6	10	6	10

- Calculation of the cost for all possible disparities:

Disparity / Pixel	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3	X	X	X	52	52	49	57	2

Cost Aggregation

- Find a path trough the matrix with minimal cost.

Disparity / Pixel	1	2	3	4	5	6	7	8
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3	X	X	X	52	52	49	57	2

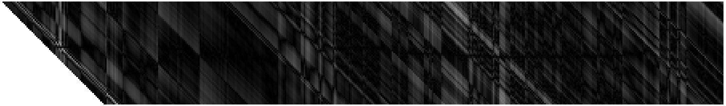


Figure: Visualisation of a cost matrix (black = min. cost, white = max. cost) [Moratto, 2019b].

Cost Aggregation

- Naive way, select for each pixel the disparity with the lowest cost.

Disparity / Pixel	1	2	3	4	5	6	7	8
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3	X	X	X	52	52	49	57	2

- Could result in zigzag jumps that do not represent the reality.

Cost Aggregation

Disparity / Pixel	$1/C_r$	$2/C_r$	3	4	5	6	7	8
0	0/0	0/0	50	4	56	1	1	2
1	X	2/3	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3	X	X	X	52	52	49	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(1,0) = 0$

$C_r(2,0) = 0 + 0 = 0$

$C_r(2,1) = 2 + 0 + P1 = 3$

Cost Aggregation

Disparity / Pixel	$1/C_r$	$2/C_r$	$3/C_r$	4	5	6	7	8
0	0/0	0/0	50/50	4	56	1	1	2
1	X	2/3	2/3	2	2	3	3	2
2	X	X	0/4	50	4	55	1	2
3	X	X	X	52	52	49	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(3,0) = 50 + 0 = 50$

$C_r(3,1) = 2 + 0 + P1 = 3$

$C_r(3,2) = 0 + 3 + P1 = 4$

Cost Aggregation

Disparity / Pixel	1/C_r	2/C_r	3/C_r	4/C_r	5	6	7	8
0	0/0	0/0	50/50	4/8	56	1	1	2
1	X	2/3	2/3	2/5	2	3	3	2
2	X	X	0/4	50/54	4	55	1	2
3	X	X	X	52/57	52	49	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(4,0) = 4 + 3 + P1 = 8$

$C_r(4,1) = 2 + 3 = 5$

$C_r(4,2) = 50 + 4 = 54$

$C_r(4,3) = 52 + 4 + P1 = 57$

Cost Aggregation

Disparity / Pixel	$1/C_r$	$2/C_r$	$3/C_r$	$4/C_r$	$5/C_r$	6	7	8
0	0/0	0/0	50/50	4/8	56/62	1	1	2
1	X	2/3	2/3	2/5	2/7	3	3	2
2	X	X	0/4	50/54	4/10	55	1	2
3	X	X	X	52/57	52/57	49	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(5,0) = 56 + 5 + P1 = 62$

$C_r(5,1) = 2 + 5 = 7$

$C_r(5,2) = 4 + 5 + P1 = 10$

$C_r(5,3) = 52 + 5 + P2 = 63$

Cost Aggregation

Disparity / Pixel	$1/C_r$	$2/C_r$	$3/C_r$	$4/C_r$	$5/C_r$	$6/C_r$	7	8
0	0/0	0/0	50/50	4/8	56/62	1/9	1	2
1	X	2/3	2/3	2/5	2/7	3/10	3	2
2	X	X	0/4	50/54	4/10	55/63	1	2
3	X	X	X	52/57	52/57	49/57	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(6,0) = 1 + 7 + P1 = 9$

$C_r(6,1) = 3 + 7 = 10$

$C_r(6,2) = 55 + 7 + P1 = 63$

$C_r(6,4) = 49 + 7 + P1 = 57$

Cost Aggregation

Disparity / Pixel	1	2	3	4	5	6	7	8
0	0	0	50	4	56	1	1	2
1	X	2	2	2	2	3	3	2
2	X	X	0	50	4	55	1	2
3	X	X	X	52	52	49	57	2

- Proposed method from [Hirschmuller, 2005], introduce two penalties $P1$ and $P2$ where $P1 < P2$.

$$C_r(p,d) = C(p,d) + min \begin{cases} C_r(p-1,d) \\ C_r(p-1,d+1) + P1 \\ C_r(p-1,d-1) + P1 \\ min_i C_r(p-1,i) + P2 \end{cases}$$

Example $P1 = 1,\ P2 = 6$

$C_r(6,0) = 1 + 7 + P1 = 9$

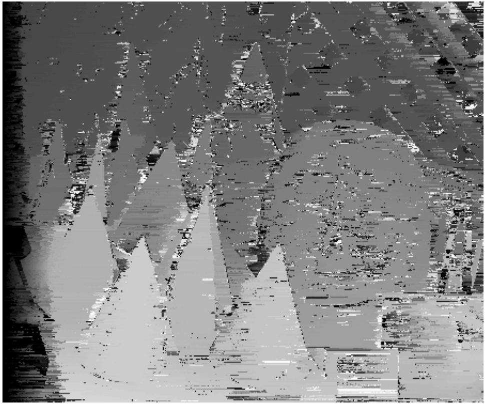
$C_r(6,1) = 3 + 7 = 10$

$C_r(6,2) = 55 + 7 + P1 = 63$

$C_r(6,4) = 49 + 7 + P1 = 57$

Cost Aggregation

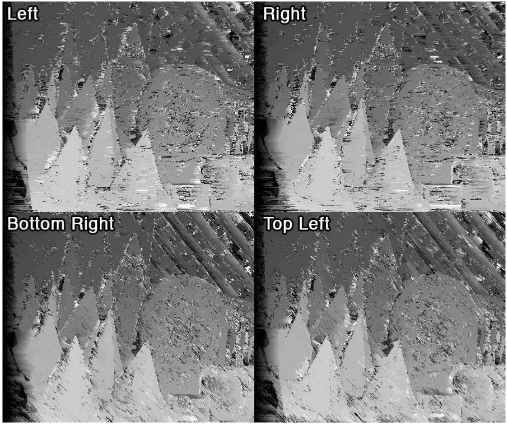
- Streaky artefacts, because considering each line independently.



Result after doing the presented method for each line [Moratto, 2019b].

Cost Aggregation

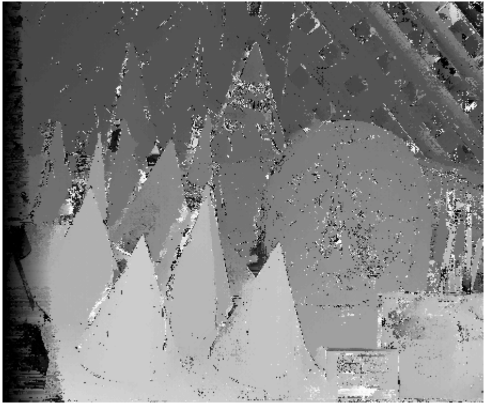
- Proposed solution form [Hirschmuller, 2005] is to do this method for different angles.



Result after doing the presented method for different angles [Moratto, 2019b].

Cost Aggregation

- Then taking the minimum for each pixel of all different angles.

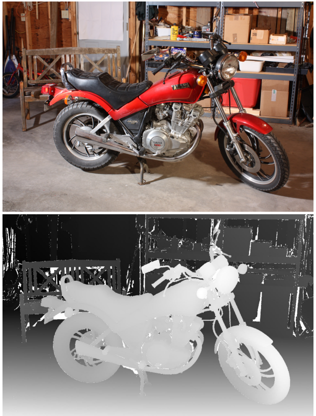


Resulting disparity image [Moratto, 2019b].

Benchmark and Evaluation

- Stereo datasets with accurate ground-truth disparities with a standardised camera model and calibration metadata:
  - *Middlebury Stereo Datasets:* Static two-view indoor scene. [Scharstein et al., 2014]
  - *KITTI Stereo 2015 Evaluation:* Dynamic scenes real-world urban driving scenarios with temporal displacement. [Menze et al., 2018]
  - *Robust Vision Challenge:* Combine indoor and outdoor scenes of the mentioned datasets and provides a public leaderboards for evaluation.
- Now: Instead of Stereo-(Two-View)-Datasets a strong drift towards Multi-View datasets.

Example - Middlebury Stereo Dataset



(a) Left image with corresponding ground-truth disparity.

(b) Right image with corresponding ground-truth disparity.

Example - KITTI Stereo 2015

KITTI images showing urban-driving scenario.

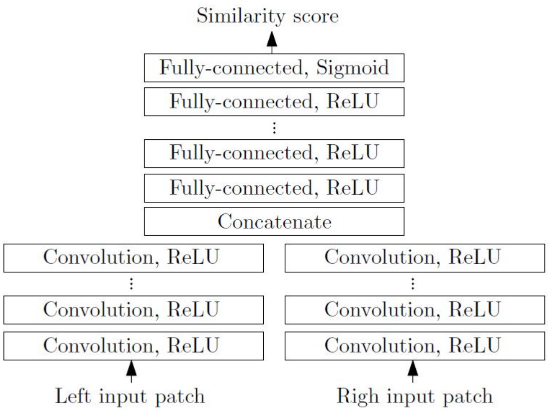
State of the Art Algorithms

- Algorithm propositions:
  - Stereo Matching by Training a CNN to Compare Image Patches. [Zbontar and LeCun, 2015]
  - Sparse Stereo Disparity Map Densification using Hierarchical Image Segmentation. [Drouyer et al., 2017]
  - 3D cost aggregation with multiple minimum spanning trees for stereo matching. [Li et al., 2017]
  - ...

Stereo matching with a CNN (1/2)

- *Paper:* J. Zbontar and Y. LeCun "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches", 2015. [Zbontar and LeCun, 2015]
- *Approach:*
  - Matching problem by a supervised learning approach using convolutional neural networks.
  - Similarity is measured on the feature vectors instead of the raw image intensity values.
  - Compute similarity score only for parts of images (i.e.: patches) and repeat recomputed for every disparity under consideration.
- *Improvements:* Instead of Fully-connected Layers, apply convolutional layers 1x1 kernels to estimate whole disparity map.
- *NOTE:* Disparity refinement still needs to be done!

Stereo matching with a CNN (2/2)

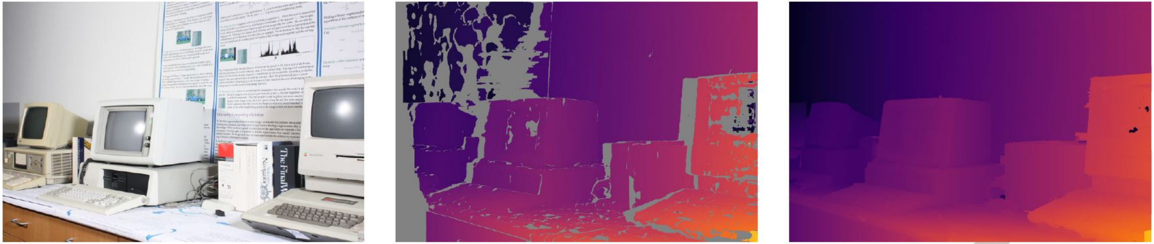


Architecture of the CNN. The inputs are two image patches (only parts of an image) and the output is a single real number between 0 and 1, which is interpreted as a measure of similarity between the input images. [Zbontar and LeCun, 2015]

Sparse Stereo Disparity Map Densification (1/2)

- *Paper:* S. Drouyer et al., "Sparse Stereo Disparity Map Densification using Hierarchical Image Segmentation", 2017. [Drouyer et al., 2017]
- *Assumptions:*
  - Some, mostly local methods produce sparse disparity maps.
  - Global methods tend to produce dense disparity maps.
  - General methods in literature: Diffusion, interpolation or edge matching.
- *New Approach:* Top Down Segmented Regression (TDSR)
  1. Segment the image into regions.
  2. For each region, calculate disparities using a regression model.
  3. If the result is not satisfying, the process is repeated on a finder region's segmentation.
- Can be used as Post-Processing step for any stereo-matching algorithm.

Sparse Stereo Disparity Map Densification (2/2)



TDSR algorithm performed on an image (left) from the Middlebury dataset. The middle image shows the sparse disparity map. The right figure shows the disparity map after Densification. [Drouyer et al., 2017]

Thank you for your attention!

If there are questions, feel free to ask.

Bibliography I

[mid, 2019] (2019).
The middlebury computer vision pages.

[BZippo, 2019] BZippo (2019).
kinect-in-infrared.
https://bbzippo.wordpress.com/2010/11/28/kinect-in-infrared/.

[Drouyer et al., 2017] Drouyer, S., Beucher, S., Bilodeau, M., Moreaud, M., and Sorbier, L. (2017).
Sparse stereo disparity map densification using hierarchical image segmentation.
In *13th International Symposium, ISMM 2017*, Mathematical Morphology and Its Applications to Signal and Image Processing, pages 172–184, Fontainebleau, France.

[FPV, 2019] FPV (2019).
Blackbird 2 3d fpv camera.
http://fpv3dcam.com/products-fpv3dcam/the-blackbird2-3d-fpv-camera.html.

Bibliography II

[Fujifilm, 2019] Fujifilm (2019).
finepix real3d w3.
http://www.fujifilm.com/products/3d/camera/finepix_real3dw3/.

[Fusiello et al., 2000] Fusiello, A., Trucco, E., and Allesandro, V. (2000).
A compact algorithm for rectification of stereo pairs.
Machine Vision and Applications, 12:16 – 22.

[Hirschmuller, 2005] Hirschmuller, H. (2005).
Accurate and efficient stereo processing by semi-global matching and mutual information.
In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814 vol. 2.

[Kalloniatis and Luu, 2019] Kalloniatis, M. and Luu, C. (2019).
Space perception.
https://webvision.med.utah.edu/book/part-viii-gabac-receptors/space-perception/.

Bibliography III

[Li et al., 2017] Li, L., Yu, X., Zhang, S., Zhao, X., and Zhang, L. (2017).
3d cost aggregation with multiple minimum spanning trees for stereo matching.
Appl. Opt., 56(12):3411–3420.

[Lioulemes, 2013] Lioulemes, A. (2013).
Robot visual homing.
Master's thesis, University of Ioannina.

[Mathworks, 2018] Mathworks (2018).
What is camera calibration?
https://de.mathworks.com/help/vision/ug/camera-calibration.html.

[Menze et al., 2018] Menze, M., Heipke, C., and Geiger, A. (2018).
Object scene flow.
ISPRS Journal of Photogrammetry and Remote Sensing (JPRS).

Bibliography IV

[Moratto, 2019a] Moratto, Z. (2019a).
Charles wheatstone-mirror stereoscope.
https://www.researchgate.net/figure/Charles-Wheatstone-mirror-stereoscope-XIX-century-1-Public-Domain_fig1_289952571.

[Moratto, 2019b] Moratto, Z. (2019b).
Semi-global matching.

[Nordmann, 2007] Nordmann, A. (2007).
Epipolar geometrie.
https://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg.

[Nordmann, 2008] Nordmann, A. (2008).
Aufnahme mit zwei kameras.
https://commons.wikimedia.org/w/index.php?curid=3550774.

Bibliography V

[OpenCV, 2019] OpenCV (2019).
Epipolar geometry.
https://docs.opencv.org/3.4/da/de9/tutorial_py_epipolar_geometry.html.
Accessed: 2019-01-23.

[Panasonic, 2019] Panasonic (2019).
Panasonic ag-3dp1.
https://www.panasonic.com/in/business/broadcast/camera/3d/ag-3dp1.html.

[Scharstein et al., 2014] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. (2014).
High-resolution stereo datasets with subpixel-accurate ground truth.
In *Pattern Recognition*, pages 31–42. Springer International Publishing.

[Scharstein et al., 2001] Scharstein, D., Szeliski, R., and Zabih, R. (2001).
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.

Bibliography VI

[Stereotron, 2019] Stereotron (2019).
Festung königstein / sachsen / cross-view / stereoscopy / hdr / raw.
https://www.flickr.com/photos/stereotron/31460315734.

[Wheatstone, 1838] Wheatstone, C. (1838).
Contributions to the physiology of vision.
Royal Society of London, 128:371 – 394.

[Zbontar and LeCun, 2015] Zbontar, J. and LeCun, Y. (2015).
Stereo matching by training a convolutional neural network to compare image patches.
CoRR, abs/1510.05970.

[Zhang, 2001] Zhang, Z. (2001).
Camera calibration with one-dimensional objects.
Technical report, Microsoft Research.