University of Amsterdam

Honours Thesis Extension

Triangulation for Depth Estimation

August 20, 2022

Student:
Niels Sombekke
12685739

Lecturer:
Dr. Arnoud Visser

1 Introduction

The perception of depth is essential for the understanding and creation of a 3D world. It can be used in a wide variety of applications such as 3D scene reconstruction and AR but is especially useful in robotics, where information of three dimensions can be used for perception, navigation and planning [1]. Depth can be retrieved by a range of different sensors such as an active range sensor or a camera. While the first can provide highly precise depth information, it is generally more expensive. The camera however has a relative cheap production cost and recent developments in deep learning have made it a viable option to consider for retrieving depth. The task of measuring depth from either a monocular image or stereo images is called depth estimation and is seen as a computer vision task. In this paper I will primarily focus on the theory behind stereo vision, providing a mathematical background of the triangulation process and use this to show the differences and similarities between a metric depth, disparity and inverse depth map.

2 Stereo Vision

Stereopsis is the perception of depth using binocular vision and one of the depth cues used by humans and other animals. The different lateral positions of the eyes result in two slightly different images which get projected to the retinas [2]. These differences are primarily in the relative horizontal position of objects in the scene and can be referred to as horizontal disparities. These disparities are processed in the visual cortex which eventually yields the ability to perceive depth. Other depth cues include size of objects, texture, linear perspective and motion parallax.

This capability of stereopsis can be emulated by computational systems in the form of stereo vision. By finding correspondences between points that are seen in both images and using a known baseline separation between cameras it is possible to compute the 3D location of these points.

2.1 Linear Camera Model

Background knowledge is needed on the workings of a single camera before explaining the process of acquiring depth using two cameras and a triangulation method, for this the linear camera model is used. This model is also known as the pinhole camera model, it can be used to describe the mathematical relationship between the coordinates of a point in 3d space and its projection onto the image plane. The aperture of the camera is as a point hence the name pinhole.

In figure 1 two models of the pinhole camera are displayed from the side. The left model shows the projection plane behind the pinhole while the right model shows it in front of it. The right model is also called the virtual pinhole camera as it cannot be implemented in practice, but provides a theoretical camera which is simpler to analyse because it produces an unrotated image. This model will be used for the remainder of this section.

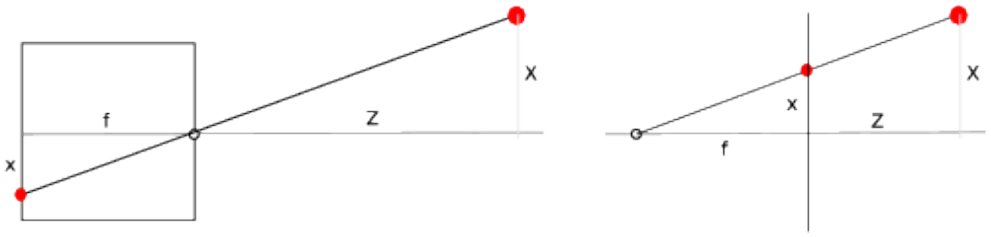

Figure 1: Physical model of pinhole camera on the left, virtual model where the projection plane is in front of pinhole on the right, by [3]

The model used puts the projection plane at $Z = f$ where Z is along the axis perpendicular to the walls and f is the focal distance. Since this creates two similar triangles it follows that:

$$\frac{f}{Z} = \frac{x}{X}, \qquad \frac{f}{Z} = \frac{y}{Y}$$

$$x = f\frac{X}{Z}, \qquad y = f\frac{Y}{Z}$$

These equations can be used to describe the relation between the 3d coordinates and its image coordinates on the image plane.

To map these image plane coordinates to the camera's image sensor, the pixel densities should be taken into account. Pixels may be rectangular so the density (pixel/mm) m is different for the x and y direction. This mapping can be described as follows:

$$u = m_x x = m_x f\frac{X}{Z}, \qquad v = m_y y = m_y f\frac{Y}{Z}$$

Both $m_x f$ and $m_y f$ can be combined to get the focal lengths in pixels in the x and y directions:

$$u = f_x\frac{X}{Z}, \qquad v = f_y\frac{Y}{Z}$$

The principle point is the point where the optical axis intersects the image plane, it can also be referred as the image center. Pixel coordinates are usually not given with respect to a frame that is centered at the pixel coordinates, most often the top-left corner of the image sensor is treated as its origin. This should be accounted for by adding the pixel coordinates (o_x, o_y) of the principle point to our equations:

$$u = f_x\frac{X}{Z} + o_x, \qquad v = f_y\frac{Y}{Z} + o_y$$

Using homogeneous coordinates the non-linear equations can be represented in linear matrix form, which is more convenient. The $3x4$ matrix in the middle is called the internal camera matrix K:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Rewriting the equations also benables the computation from 2D to 3D shows that it is not possible for a point as Z can't be isolated, which emphasises that a stereo camera model is required:

$$X = \frac{Z}{f_x}(u - o_x), \qquad Y = \frac{Z}{f_y}(v - o_y), \qquad Z > 0$$

2.2 Simple Stereo System

In order to reconstruct 3D more information is needed, a simple way of doing this is by using two identical cameras where one of the cameras is displaced along the horizontal direction. This displacement can be seen in figure 5, it is called the horizontal baseline and is indicated with B. This setup with a left camera and a right camera is called a simple stereo system and is a form of binocular vision. It can also be simulated using a single camera which takes two images with a horizontal displacement of B.

First, a matching scene point must be found in both images. This process of finding matching scene points is called stereo matching and will be covered later. In figure 5 these points are represented by x for the left camera and x' for the right camera. They can also be written in the pixel format introduced earlier:

$$(u_l, v_l), \qquad (u_r, v_r)$$

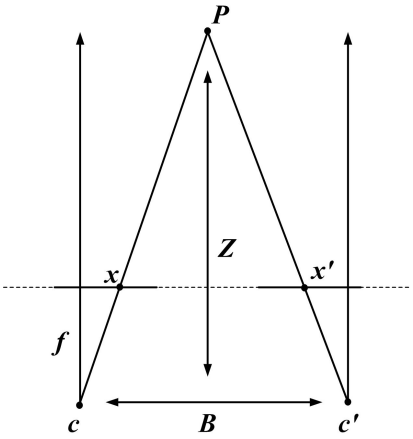

Figure 2: Simple Stereo System, by [4]

The point where both camera rays, which go through their respective pixel point, intersect is where the scene point is located corresponding to the two image points.

Using the linear camera model equations as seen in the previous section equations can be obtained for both pixel points, notice the minus B in the right camera which corrects for the horizontal displacement:

$$u_l = f_x\frac{X}{Z} + o_x, \quad v_l = f_y\frac{Y}{Z} + o_y \qquad u_r = f_x\frac{X-B}{Z} + o_x, \quad v_r = f_y\frac{Y}{Z} + o_y$$

$$(u_l, v_l) = (f_x\frac{X}{Z} + o_x, f_y\frac{Y}{Z} + o_y) \qquad (u_r, v_r) = (f_x\frac{X-B}{Z} + o_x, f_y\frac{Y}{Z} + o_y)$$

These equations can be solved for X, Y and Z:

$$u_l = f_x\frac{X}{Z} + o_x \qquad u_r = f_x\frac{X-B}{Z} + o_x$$

$$X = Z(\frac{u_l - o_x}{f_x}) \quad \xrightarrow{\text{Insert}} \quad u_r = f_x(\frac{X}{Z} - \frac{B}{Z}) + o_x$$

$$u_r = f_x(\frac{Z\frac{u_l - o_x}{f_x}}{Z} - \frac{B}{Z}) + o_x$$

$$u_r = f_x(\frac{u_l - o_x}{f_x} - \frac{B}{Z}) + o_x$$

$$u_r = u_l - o_x - \frac{Bf_x}{Z} + o_x$$

$$u_r = u_l - \frac{Bf_x}{Z}$$

$$u_r - u_l = -\frac{Bf_x}{Z}$$

$$(u_l - u_r) = \frac{Bf_x}{Z}$$

$$Z = \frac{Bf_x}{(u_l - u_r)}$$

$$Z = \frac{Bf_x}{(u_l - u_r)} \quad \xrightarrow{\text{Insert}} \quad X = Z(\frac{u_l - o_x}{f_x})$$

$$X = \frac{Bf_x}{(u_l - u_r)} \frac{(u_l - o_x)}{f_x}$$

$$X = \frac{Bf_x(u_l - o_x)}{f_x(u_l - u_r)}$$

$$X = \frac{B(u_l - o_x)}{(u_l - u_r)}$$

$$v_l = f_y \frac{Y}{Z} + o_y \quad \xrightarrow{\text{Similar to X}} \quad Y = Z(\frac{v_l - o_y}{f_y})$$

$$Z = \frac{Bf_x}{(u_l - u_r)} \quad \xrightarrow{\text{Insert}} \quad Y = Z(\frac{v_l - o_y}{f_y})$$

$$Y = \frac{Bf_x}{(u_l - u_r)} \frac{(v_l - o_y)}{f_y}$$

$$Y = \frac{Bf_x(v_l - o_y)}{f_y(u_l - u_r)}$$

The final equations for X, Y and Z thus are:

$$X = \frac{B(u_l - o_x)}{(u_l - u_r)}, \qquad Y = \frac{Bf_x(v_l - o_y)}{f_y(u_l - u_r)}, \qquad Z = \frac{Bf_x}{(u_l - u_r)}$$

In all three denominators $(u_l - u_r)$ is found, this is the difference of the u coordinate of the same scene point in the two images also known as the disparity. The equation for depth Z and figure 3 show that depth Z is inversely proportional to disparity. Which means that the disparity will be large when a scene point is very close to both cameras and will shrink as it moves away from the cameras. When the scene point approaches infinite depth the disparity will go to zero, which means that there is no difference of the position of the point in the images.

Similarly, the disparity is proportional to the baseline B. As you increase the baseline the difference and thus the disparity between the two images will also increase.

Finally, by rearranging the depth equation Z it can be showed that disparity is proportional to inverse depth:

$$Z = \frac{Bf_x}{(u_l - u_r)} \quad \xrightarrow{\text{Rearrange}} \quad (u_l - u_r) = \frac{Bf_x}{Z}$$

An inverse depth map is able to represent features with a depth value of infinite, as these values will become zero which leads to fewer problems [5].

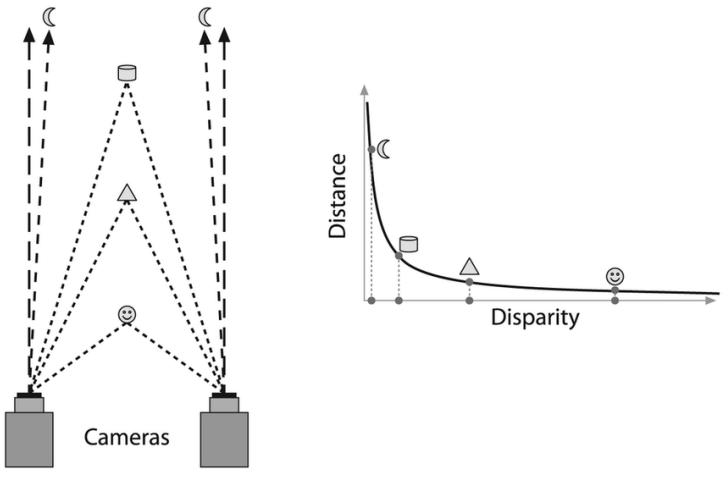

Figure 3: Depth and disparity are inversely related, so precise depth measurement is restricted to nearby objects, by [6]

Figure 4: Disparity map (right) obtained with a mobile device on a robot arm, using the technique of [7]

Figure 3 shows the extraction of a disparity map from two images, if the camera calibration is known a metric depth map can be generated using these disparities. This example also shows that for some areas no disparity can be extracted, which is indicated with the black color. This is called the correspondence problem, a fundamental problem in computer vision. [8]

2.2.1 Stereo matching

Finding the correspondences between two stereo images is called stereo matching and is required for extracting the disparity values. Traditional approaches are based on finding matching features or templates, where two corresponding points are found using similarity metrics. Newer stereo matching methods based on deep learning have achieved performance far exceeding the traditional approaches. [9]
The traditional matching approaches use the fact that $v_l = v_u$, indicating that there is no vertical disparity. As a result, it can be concluded that corresponding scene points lie on the same horizontal scan line. By taking a small template window from the left image and sliding it along its horizontal scan line on the right image a correspondence can be found. A small template window will give good localization but is sensitive to noise, a larger window has poor localization but is less sensitive as it obtains more robust matches. A solution for this is using an adaptive window size approach where the window size with the best similarity measure for that specific point is used. [11] Commonly used similarity metrics are: Sum of Absolute Differences, Sum of Squared Differences and Normalized Cross-Correlation. [12]

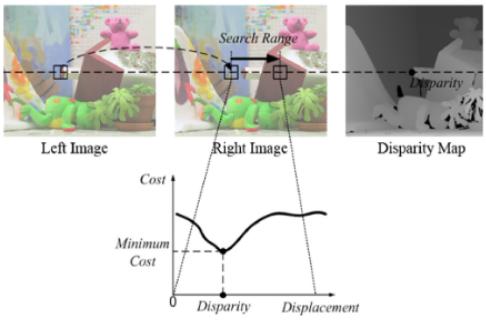

Figure 5: Template matching, by [10]

3 Conclusion

This paper has shown the geometric and mathematical background behind stereo vision and the maps it can produce using a simple stereo system and the triangulation process. The extracted equations show that disparity is inversely proportional to depth Z, making the disparity map proportional to the inverse depth map. A disparity map visualizes the horizontal displacement of a 3D scene point's projections between the left and right image, metric depth can be obtained by using the equation for Z but this is only possible when camera calibration is known. Stereo matching is needed to find corresponding points between the two images, these correspondences are used to extract the disparities. Traditional stereo matching algorithms use a horizontal sliding window approach, the window size can be adjusted to increase or reduce noice and localization quality. Newer algorithms using deep learning have a greatly increased accuracy and thus are now the preferred approach.

References

[1] N. Sombekke, "Monocular depth estimation for light-weight real-time obstacle avoidance," 2022.

[2] R. Patterson and W. L. Martin, "Human stereopsis," *Human factors*, vol. 34, no. 6, pp. 669–692, 1992.

[3] R. van den Boomgaard, "Image processing and computer vision." [Online]. Available: https://staff.fnwi.uva.nl/r.vandenboomgaard/ComputerVision/

[4] E.-K. Lee, S.-U. Yoon, and Y.-S. Ho, "Generation of multiple depth images from a single depth map using multi-baseline information," in *International Workshop on Advanced Image Technology (IWAIT 2008)*, vol. 88, 2008, pp. 1–6.

[5] J. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular slam." Robotics: Science and Systems, 2006.

[6] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[7] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German conference on pattern recognition.* Springer, 2014, pp. 31–42.

[8] A. S. Ogale and Y. Aloimonos, "Shape and the stereo correspondence problem," *International Journal of Computer Vision*, vol. 65, no. 3, pp. 147–162, 2005.

[9] M. S. Hamid, N. Abd Manap, R. A. Hamzah, and A. F. Kadmin, "Stereo matching algorithm based on deep learning: A survey," *Journal of King Saud University-Computer and Information Sciences*, 2020.

[10] Y.-H. Seo, J.-S. Yoo, and D.-W. Kim, "A new parallel hardware architecture for high-performance stereo matching calculation," *Integration, the VLSI Journal*, vol. 51, pp. 81–91, 2015.

[11] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE transactions on pattern analysis and machine intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[12] M. Hisham, S. N. Yaakob, R. Raof, A. A. Nazren, and N. Wafi, "Template matching using sum of squared difference and normalized cross correlation," in *2015 IEEE student conference on research and development (SCOReD).* IEEE, 2015, pp. 100–104.