

Disparity Map in Stereo Vision

Last updated: September 19, 2022

Written by: baeldung (<https://www.baeldung.com/cs/author/baeldung>)

Computer Vision (<https://www.baeldung.com/cs/category/ai/computer-vision>)

Core Concepts (<https://www.baeldung.com/cs/category/core-concepts>)

1. Overview

In this tutorial, we'll talk about disparity, a fundamental concept in stereo vision. First, we'll discuss how the disparity map is used in computer vision to reconstruct the 3D structure of a scene from two images. Then we'll illustrate how to compute the disparity map.

2. Stereo Vision

We are all familiar with the perception of depth that our two eyes give us. Computers can emulate this capability with stereo vision, a computer vision technique to estimate the 3D structure of a scene from multiple images. Stereo vision is used in robot navigation applications and in the Advanced Driver Assistance Systems (https://en.wikipedia.org/wiki/Advanced_driver-assistance_system) (ADAS).

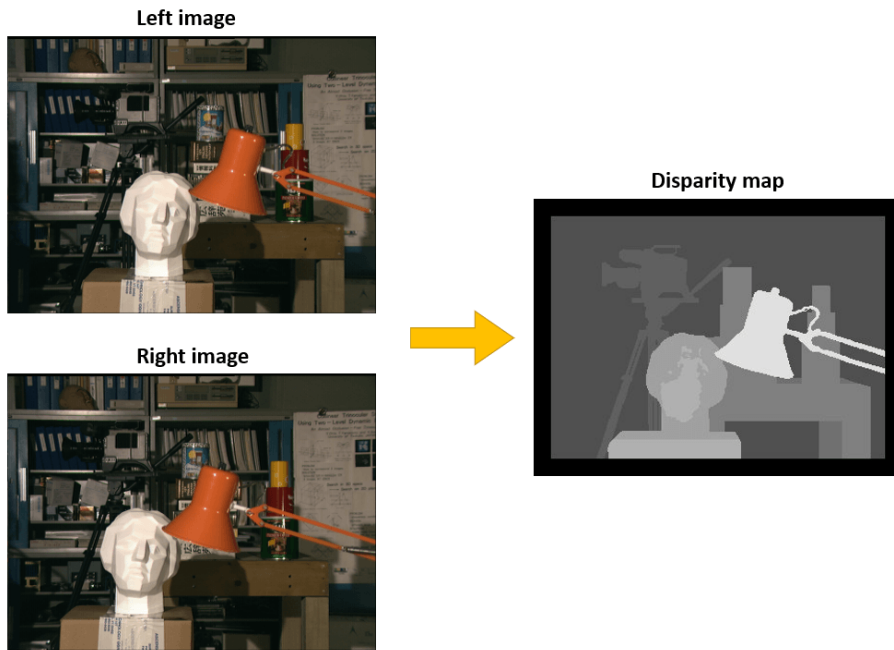
The basic idea is that two cameras take pictures of the same scene from two different positions, and we aim to estimate the depth for each pixel. This task can be done by finding matching pixels in the two stereo images and knowing the geometric arrangement of the cameras.

3. Disparity Map

The disparity is the apparent motion of objects between a pair of stereo images. To experience this, try closing one eye and then rapidly open it while closing the other one. You will note that closer objects will move significantly while objects further away will move very little. This phenomenon is called disparity.

Given a pair of stereo images, to compute the disparity map, we first match every pixel in the left image with its corresponding pixel in the right image. Then we compute the distance for each pair of matching pixels. Finally, the disparity map is obtained by representing such distance values as an intensity image.

Let's consider, as an example, the following pair of stereo images and the corresponding disparity map.



We can see that the objects in the foreground, i.e., the lamp and the statue, appear fairly shifted in the stereo images, and therefore they are marked with brighter pixels in the disparity map. Conversely, the objects in the background have low disparity since their displacement in the stereo images is very small.

As previously observed, the depth is inversely proportional to the disparity. If we know the geometric arrangement of the cameras, then the disparity map can be converted into a depth map using triangulation (<https://en.wikipedia.org/wiki/Triangulation>).

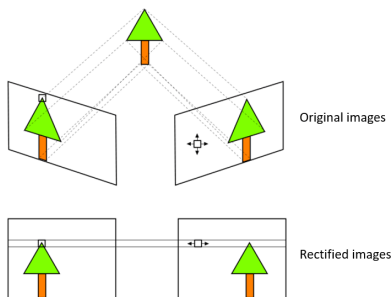
When disparity is near zero, small disparity differences produce large depth differences. However, when disparity is large, small disparity differences do not change the depth significantly. Hence, stereo vision systems have high depth resolution only for objects relatively near the camera.

4. Correspondence Problem

To compute the disparity map, we must address the so-called *correspondence problem*. This task aims at determining the pair of pixels in the stereo images that are projections of the same physical point in space.

An important simplification of the problem can be obtained by the rectification (https://en.wikipedia.org/wiki/Image_rectification) of the stereo images. After this transformation, the corresponding points will lie on the same horizontal line. This reduces the 2D stereo correspondence problem to a 1D problem.

The image rectification process is illustrated in the following figure:



A basic approach for finding corresponding pixels is the *Block Matching algorithm*. It is based on comparing a small window around a point in the first image with multiple small blocks along the same horizontal line in the second image. For each pair of windows, a loss function ([/cs/cost-vs-loss-vs-objective-function#loss-functions](https://www.baeldung.com/cs/cost-vs-loss-vs-objective-function#loss-functions)) is computed. The point (\hat{x}, y)

in the second image with minimum loss value is the best match for the point (x, y) in the first image. Hence, the disparity value at the coordinate (x, y) will be $d(x, y) = (\tilde{x} - x)$.

Two different functions are usually used for finding the matching pixels. The first is the Sum of Absolute Differences (SAD) which computes the sum of elementwise absolute differences of two windows W^L and W^R , of size $M \times N$, extracted from the two stereo images:

$$\text{SAD}(W^L, W^R) = \sum_{i=1}^N \sum_{j=1}^M |W_{ij}^L - W_{ij}^R|.$$

The second function is the Sum of Squared Differences (SSD). While the SAD computes a sum of absolute values, the SSD computes the sum of squared differences:

$$\text{SSD}(W^L, W^R) = \sum_{i=1}^N \sum_{j=1}^M (W_{ij}^L - W_{ij}^R)^2$$

In general, SAD is preferable to SSD since it is faster and more robust (/cs/robust-estimators-in-robust-statistics) to noise and outliers (/cs/ml-outlier-detection-handling#outliers).

5. Conclusion

In this article, we reviewed the concept of disparity, and we explained its importance in stereo vision. Finally, we illustrated a simple method to compute the disparity map.

Comments are closed on this article!

CATEGORIES

[ALGORITHMS\(/cs/category/algorithms\)](#)

[ARTIFICIAL INTELLIGENCE\(/cs/category/ai\)](#)

[CORE CONCEPTS\(/cs/category/core-concepts\)](#)

[DATA STRUCTURES\(/cs/category/data-structures\)](#)

[GRAPH THEORY\(/cs/category/graph-theory\)](#)

[LATEX\(/cs/category/latex\)](#)

[NETWORKING\(/cs/category/latex\)](#)

[SECURITY\(/cs/category/security\)](#)

SERIES

SITES

[BAELDUNG\(/\)](#)

[LINUX\(/linux\)](#)

[SCALA\(/scala\)](#)

[KOTLIN\(/kotlin\)](#)

ABOUT

[ABOUT BAELDUNG\(/about\)](#)
[THE FULL ARCHIVE\(/cs/full_archive\)](#)
[EDITORS\(/editors\)](#)

[TERMS OF SERVICE\(/terms-of-service\)](#) | [PRIVACY POLICY\(/privacy-policy\)](#) | [COMPANY INFO\(/company-info\)](#) | [CONTACT\(/contact\)](#)