The Intro to Computer Vision labs will be run in Google Colaboratory, a Jupyter notebook environment that runs entirely in the cloud, you don't need to download anything. To run these labs, you must have a Google account.

Step 1: click on the assignment invite link -> **Accept this assignment**. Refresh page -> individual repo for the specific assignment is created automatically

Project 2: 2 weeks

https://classroom.github.com/a/47z2DZ4B

Step 2: Navigate to http://colab.research.google.com/github -> Click the **Include Private Repos** checkbox -> **select the correct repo** (SistemeDeVedereArtificiala/assignment_name-student_name) -> Click on the jupyter notebook of the current assignment

Step 3: [GitHub sign-in window] In the popup window, sign-in to your Github account and authorize Colab to read the private files.

Step 4: [in colab] **File** -> **Save a copy to GitHub**. Select the correct repository for the SPECIFIC assignment -> Click the **Include Colab Link** -> Click **OK**

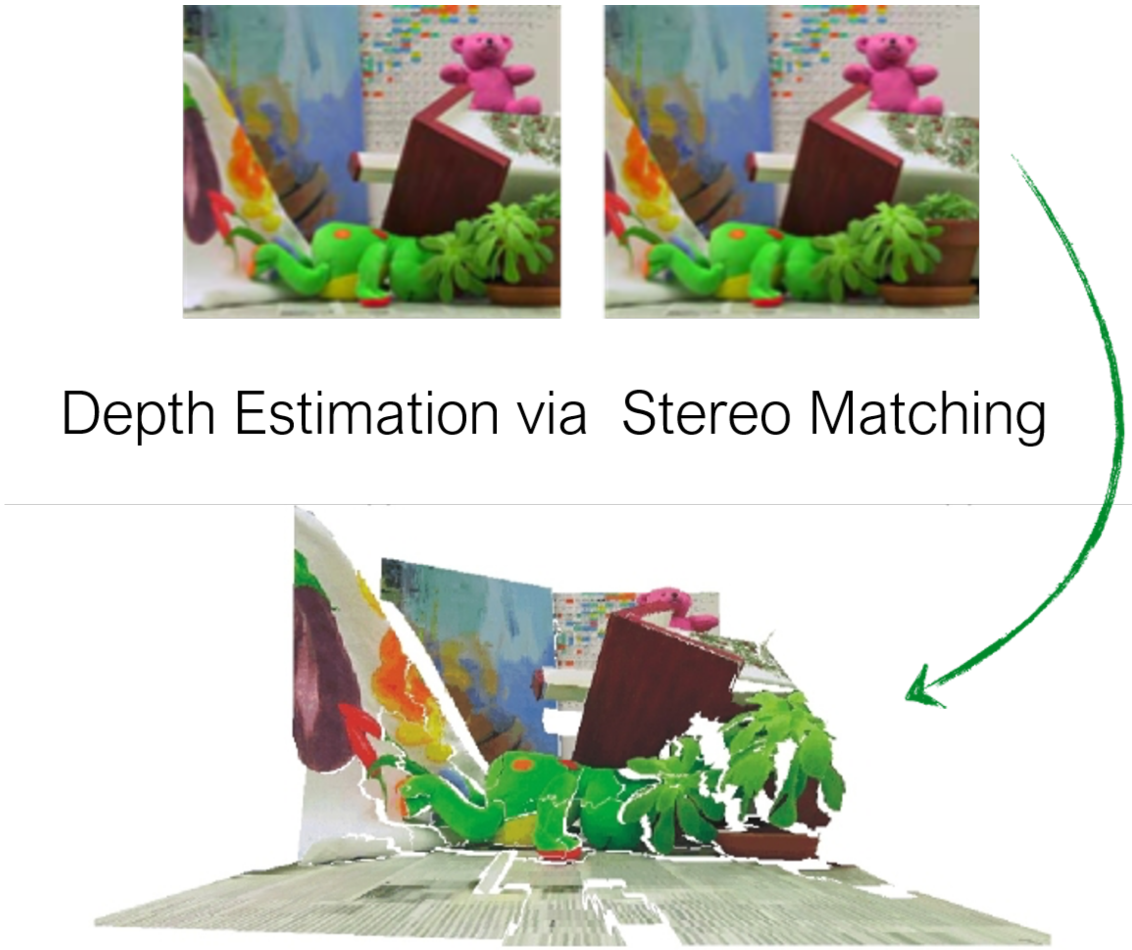
Step 5: [in colab] Navigate to the **Runtime** tab --> **Change runtime type**, under **Hardware accelerator** select **GPU/TPU** (tensor processing unit) according to your needs.

Read the suggestions and accomplish all tasks marked with **#TODO**.

!!! At the end of each laboratory **REPEAT step 4 in order to SAVE** the answers to your private repository (individual for each assignment)

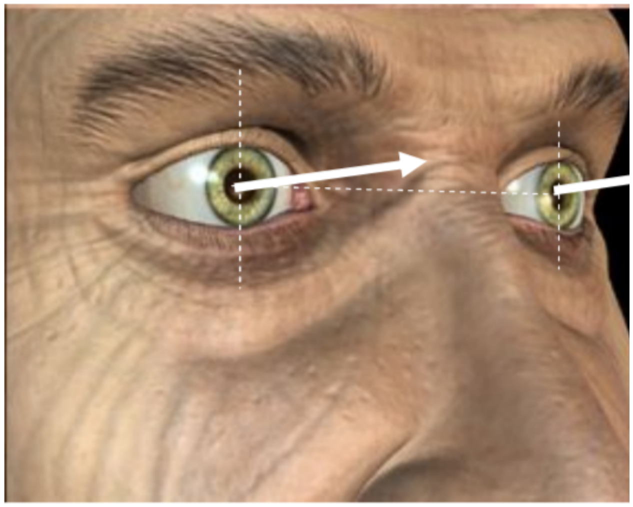
Project 4: Obstacle detection based on disparity maps

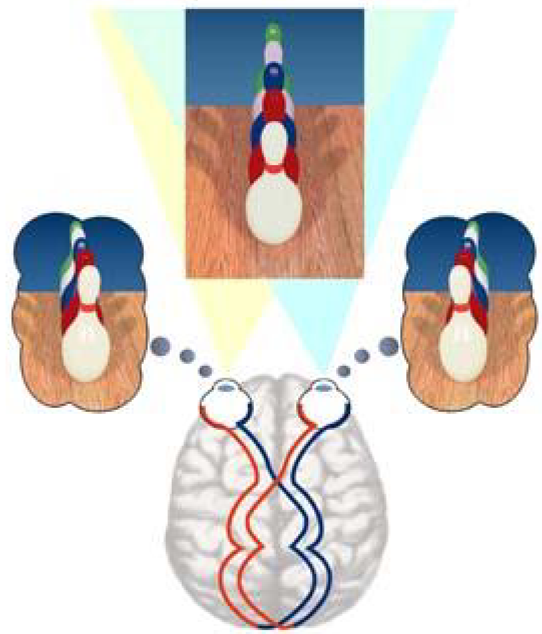
Week 1: Depth estimation and 3D reconstruction based on disparity maps



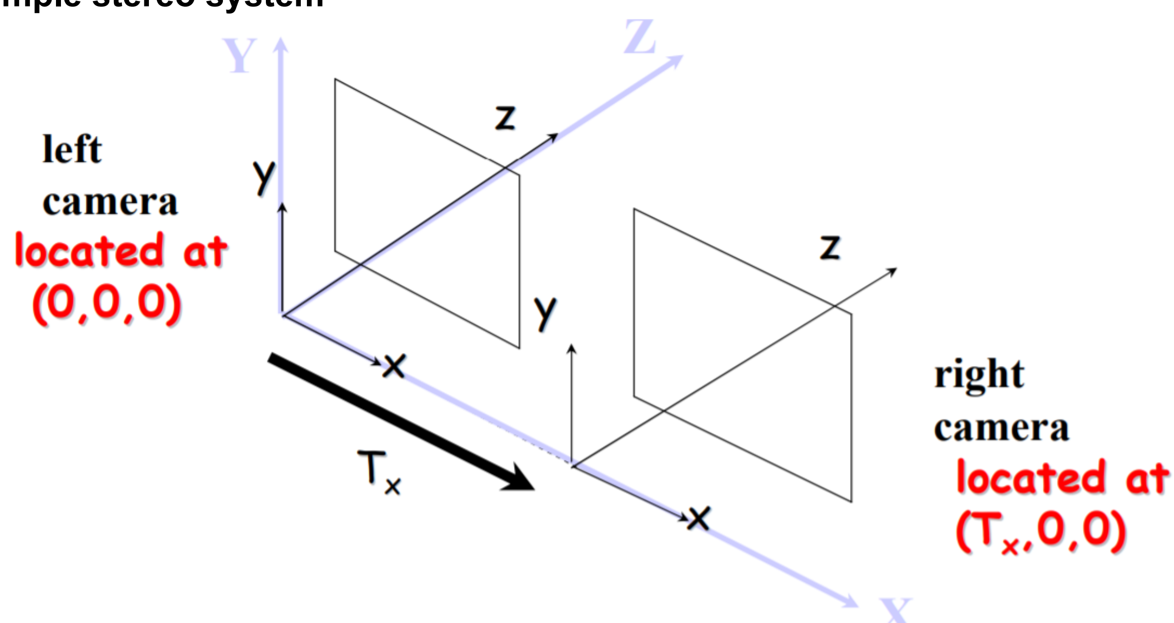
Stereo vision

Stereo vision is the process of extracting 3D information from multiple 2D views of a scene. Stereo vision is used in applications such as advanced driver assistance systems (ADAS) and robot navigation where stereo vision is used to estimate the actual distance or range of objects of interest from the camera.
The 3D information can be obtained from a pair of images, also known as a stereo pair, by estimating the relative depth of points in the scene. These estimates are represented in a stereo disparity map, which is constructed by matching corresponding points in the stereo pair



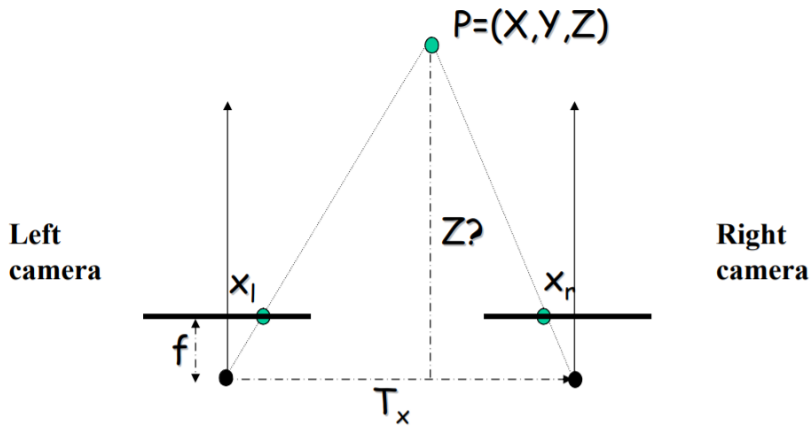


A simple stereo system



Right camera is simply shifted by **Tx** units along the **X** axis. Otherwise, the cameras are identical (same orientation / focal lengths).

Top Down View (XZ plane)



Translated by a distance T_x along X axis
(T_x is also called the stereo "baseline")

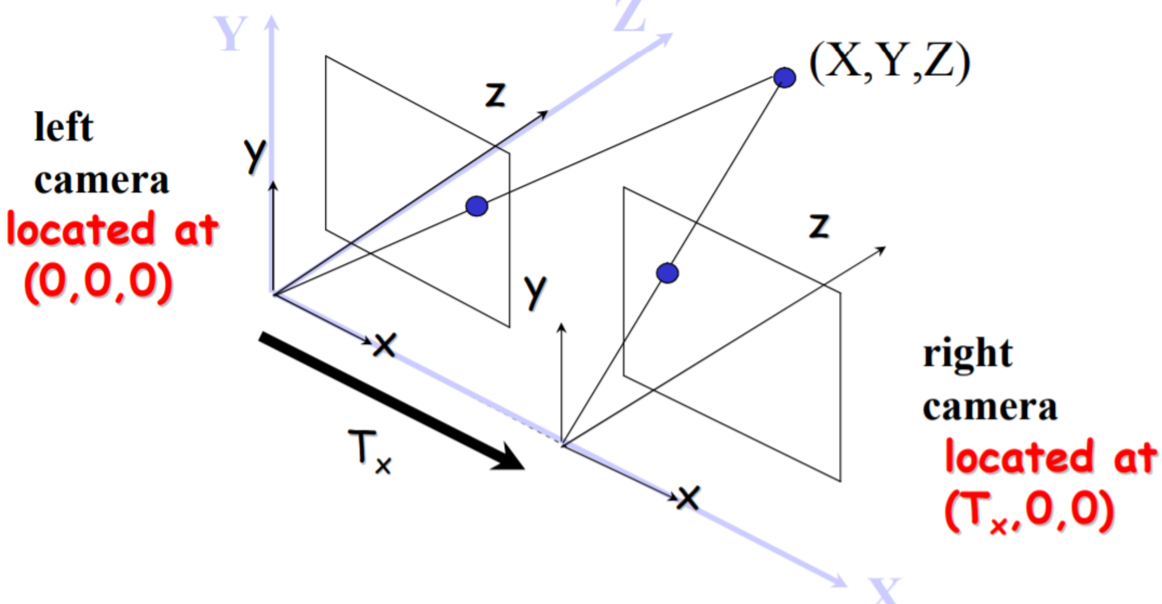
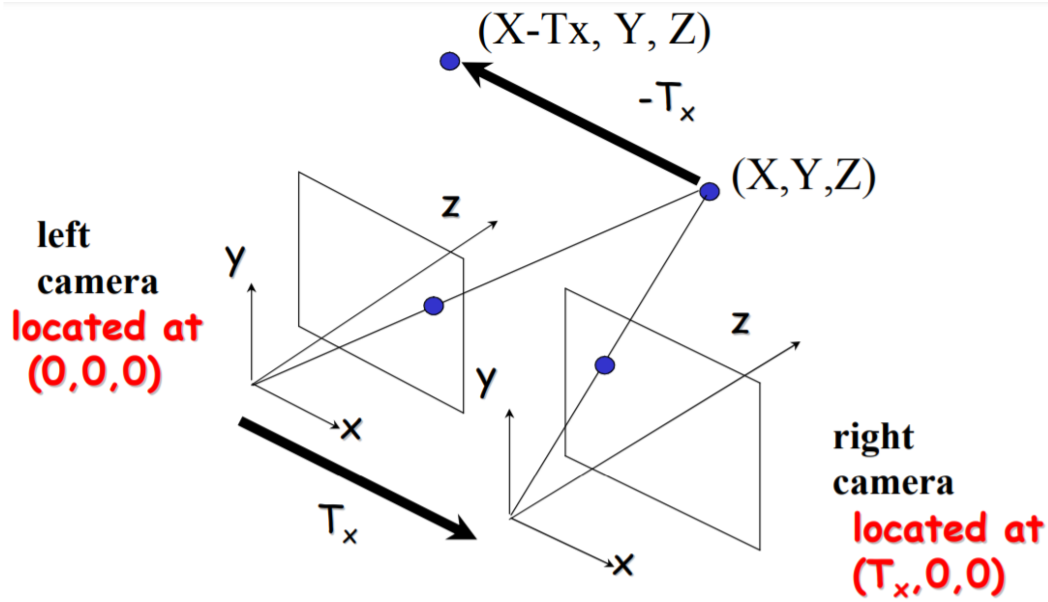


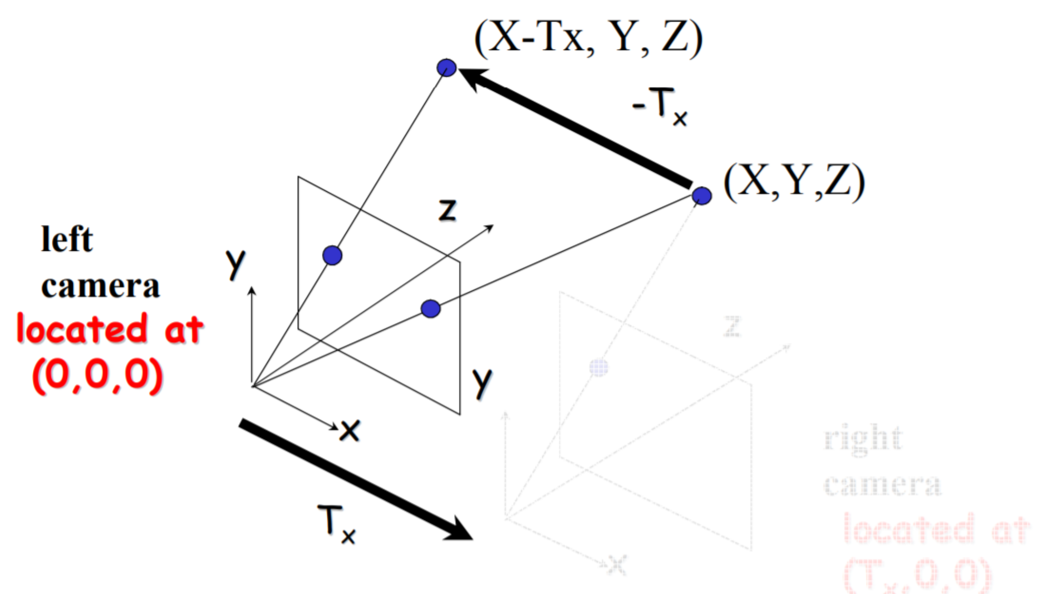
Image coords of point (X,Y,Z) in Left Camera:

$$x_l = f\frac{X}{Z},\ y_l = f\frac{Y}{Z}$$

Q: What are image coords of that same point in the Right Camera?



Translating the camera to the right by Tx is equivalent to leaving the camera stationary and translating the world to the left by Tx.



$$x_r = f\frac{X - T_x}{Z},\ y_r = f\frac{Y}{Z}$$

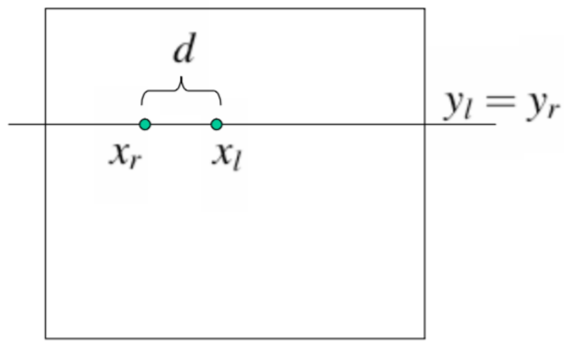
Stereo Disparity

Left camera

$$x_l = f\frac{X}{Z} \qquad y_l = f\frac{Y}{Z}$$

Right camera

$$x_r = f\frac{X - T_x}{Z} \qquad y_r = f\frac{Y}{Z}$$



Stereo Disparity

$$d = x_l - x_r = f\frac{X}{Z} - \left(f\frac{X}{Z} - f\frac{T_x}{Z}\right)$$

$$d = \frac{f\,T_x}{Z}$$

depth

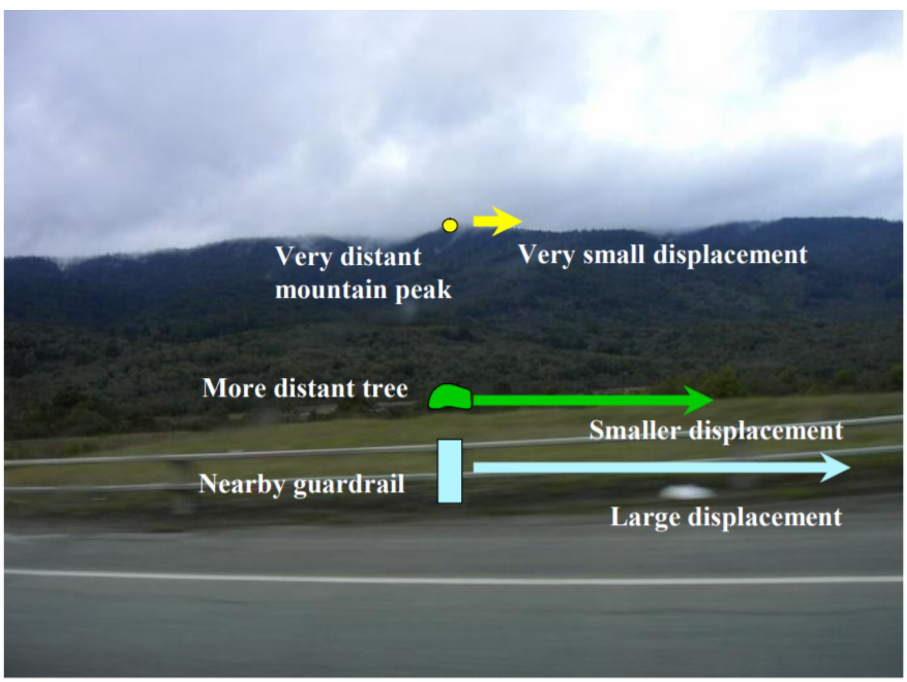
baseline

$$Z = \frac{f\,T_x}{d}$$

disparity

Important equation!

Note: Depth and stereo disparity are inversely proportional => this is why near objects appear to move more than far away ones when the camera translates sideways.



Disparity estimation

Disparity estimation algorithms fall into two broad categories: **local methods** and **global methods**. **Local methods** evaluate one pixel at a time, considering only neighboring pixels. **Global methods** consider information that is available in the whole image. **Local**

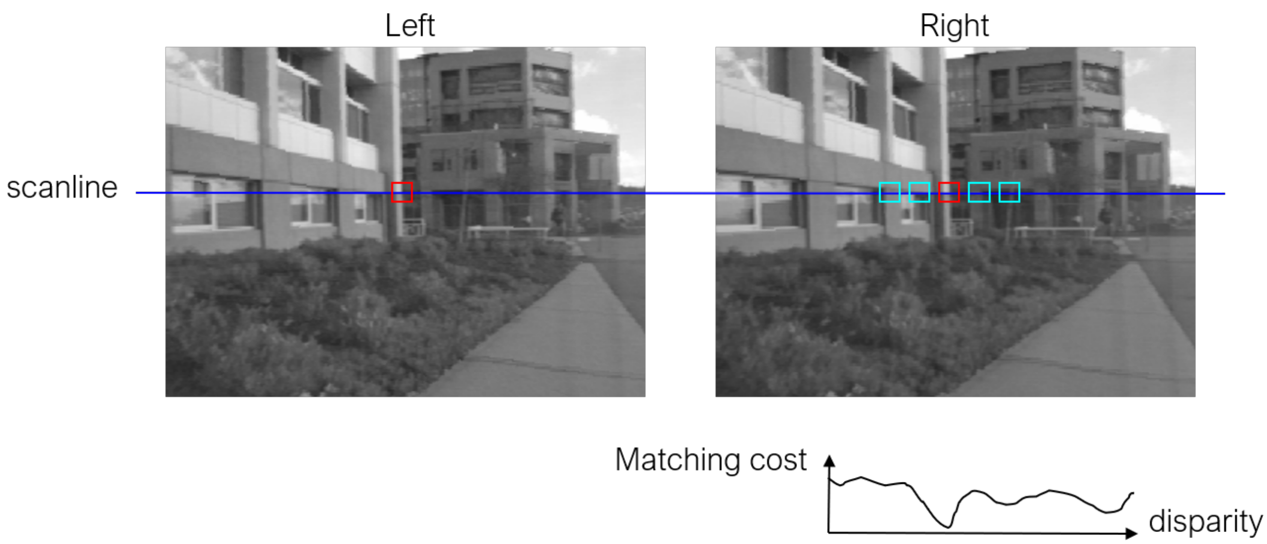
methods are poor at detecting sudden depth variation and occlusions, and hence **global methods** are preferred.

The Block-Matching algorithm

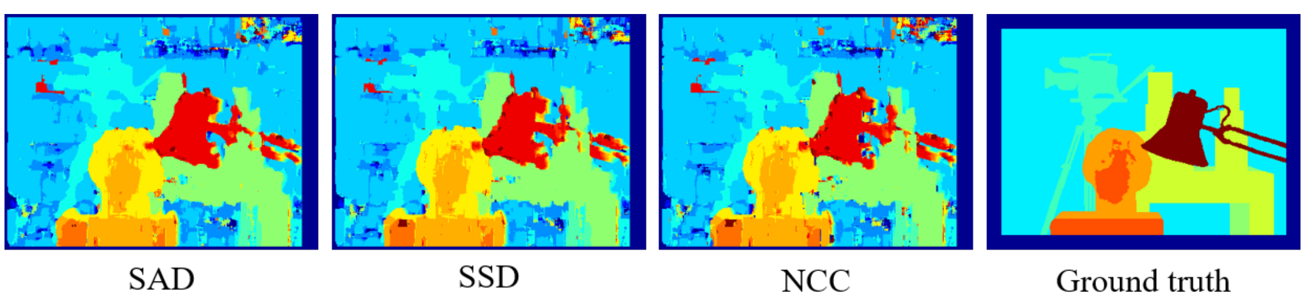
Estimates disparity at a point by comparing a small region about that point with congruent regions extracted from the other image

Three classes of metrics used for the comparison:

- Correlation (NCC - normalized Cross Correlation)
- Intensity difference (SAD - Sum of Absolute Differences, SSD - Sum of Squared Differences)
- Rank (rank transform, census transform)



Slide a window along the epipolar line and compare the contents of that window with the reference window in the left image.



Results using different metrics

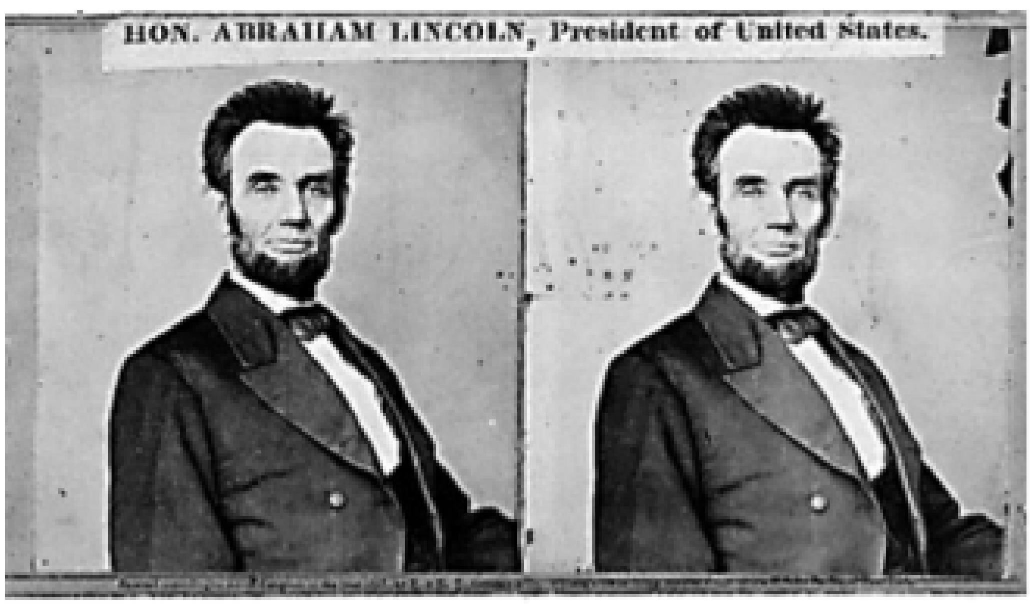
W = 3

W = 20

The effect of window size

- smaller window: more detail(+), more noise(-)
- larger windows: smooter disparity maps(+), less detail(-), fails near boundaries(-)

Generally, stereo block matching will fail for *textureless regions*, *repeated patterns* and *specularities*.



textureless regions

repeated patterns

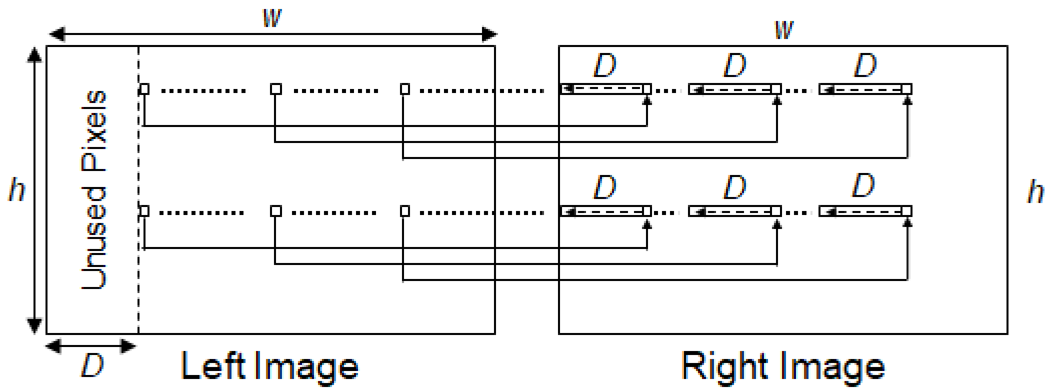
specularities

The Semi-Global Block Matching algorithm

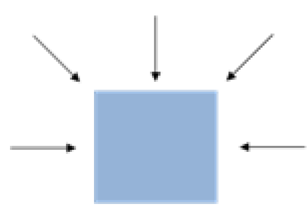
Semi-global matching uses information from neighboring pixels in multiple directions to calculate the disparity of a pixel. Analysis in multiple directions results in a lot of computation. Instead of using the whole image, the disparity of a pixel can be calculated by considering a smaller block of pixels for ease of computation. Thus, the Semi-Global Block Matching (SGBM) algorithm uses block-based cost matching that is smoothed by path-wise information from multiple directions.

Using the block-based approach, this algorithm estimates an approximate disparity of a pixel in the left image from the same pixel in the right image. Before going into the algorithm and implementation details, two important parameters need to be understood: **Disparity Levels** and **Number of Directions**.

- **Disparity Levels**: Disparity levels is a parameter used to define the search space for matching.
- **Number of Directions**: To optimize the cost function, the input image is considered from multiple directions. In general, accuracy of disparity result improves with increase in number of directions.



As shown in figure above, the algorithm searches for each pixel in the Left Image from among D pixels in the Right Image. The D values generated are D disparity levels for a pixel in Left Image. The first D columns of Left Image are unused because the corresponding pixels in Right Image are not available for comparison. In the figure, w represents the width of the image and h is the height of the image. For a given image resolution, increasing the disparity level reduces the minimum distance to detect depth. Increasing the disparity level also increases the computation load of the algorithm. At a given disparity level, increasing the image resolution increases the minimum distance to detect depth. Increasing the image resolution also increases the accuracy of depth estimation. The number of disparity levels are proportional to the input image resolution for detection of objects at the same depth. This example supports disparity levels from 8 to 128 (both values inclusive). The explanation of the algorithm refers to 64 disparity levels. The models provided in this example can accept input images of any resolution.



5 Directions

This example analyzes five directions: left-to-right (A1), top-left-to-bottom-right (A2), top-to-bottom (A3), top-right-to-bottom-left (A4), and right-to-left (A5).

Further study at:

1. [Depth Map from Stereo Images]
2. Chapter 11 - Stereo correspondence from Algorithms and Applications, 1nd Edition, Richard Szeliski
3. Chapter 12 - Depth estimation from Algorithms and Applications, 2nd Edition, Richard Szeliski