Perception for Autonomous Systems

Lecture 3 - Stereo Vision (15/02/2021)

Outline/Content:

- What is Stereo Vision?
- Stereo/Epipolar Geometry
- Rectified Stereo Case
- Depth from Stereo Matches
- Correspondence Problem
    - Dense vs Sparse Correspondence
    - Local vs Global Correspondence
    - (Dis-)-Similarity Measures
- Summary

> Topics and Reading Sources:
>
> - Optional Additional Reading Material: D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” International Journal of Computer Vision, vol. 47, no. 1-3, pp. 7–42, 2002.
> - L. Nalpantidis, G. C. Sirakoulis, and A. Gasteratos, “Review of stereo vision algorithms: from software to hardware,” International Journal of Optomechatronics, vol. 2, no. 4, pp. 435–462, 2008.

What is Stereo Vision?

Stereovision (aka., stereoscopic vision or stereopsis) describes the visual perception in three dimensions.

Two sensory inputs (both eyes for humans or two cameras for machines) which capture individual images are being used in this process. These two images have overlapping areas of common visual information, but also some unique visual information. The images are then being merged together, by matching up the similarities and adding in the small differences. By doing that we obtain a three-dimensional stereo picture.

> This implies that at least two visual based sensory inputs are needed to obtain a stereo picture.

Stereo/Epipolar Geometry Article

There are two key requirements, that need to be fulfilled to calculate the 3D structure out of two images:

- **The position of cameras:**: To obtain the positions of the cameras, it is necessary to calculate the 3D points by taking one of the camera positions as the origin. Then a calibration pattern is needed to calibrate the two cameras, which allows to find the 3D points.
- **The point correspondence**: All point correspondences between the two images must be calculated for each 3D point in the scene.

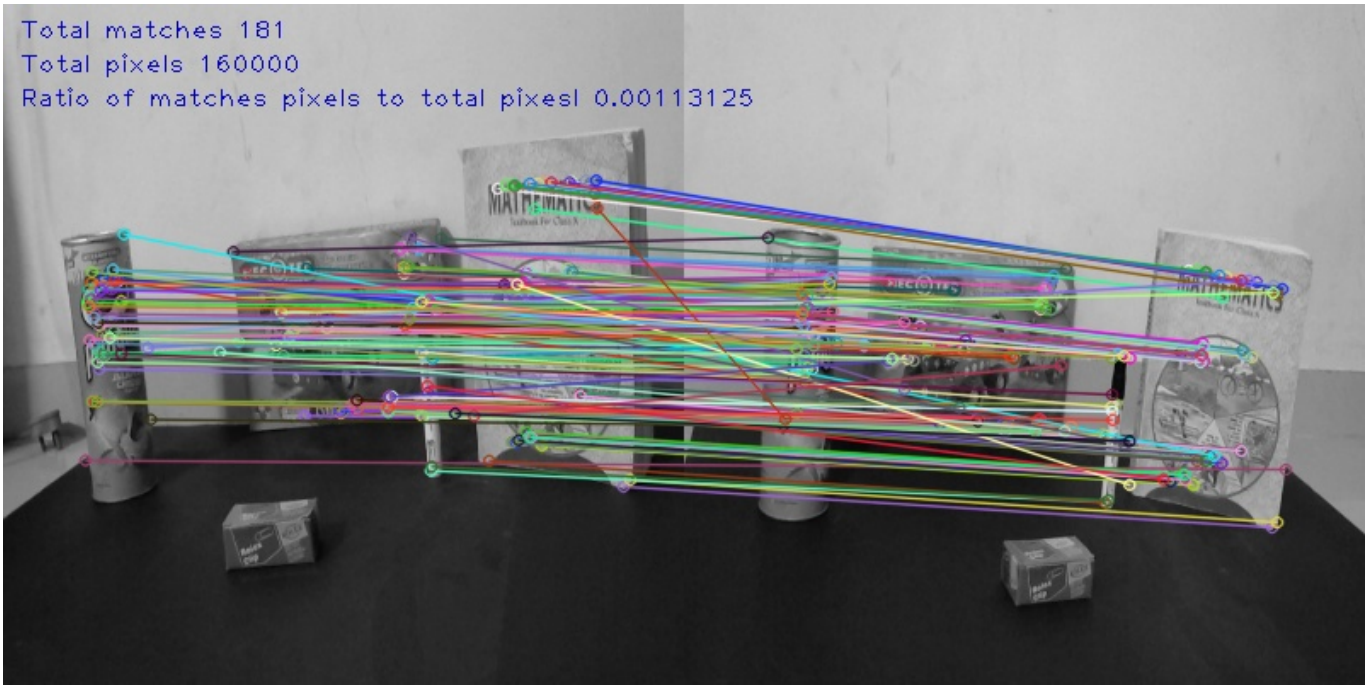
> Point correspondence is finding the corresponding pixel of one image in the other.

Point correspondence: Feature matching

Feature matching algorithms such as SIFT, SURF, or ORB are valid methods for matching features between the left and right stereo images.

However, these algorithms will most likely result in a very sparsely reconstructed 3D scene due to the poor ratio of total pixels to found pixels.

> Those approaches are also very computationally heavy, and would check each pixel, altough we only need to track pixels on our epipolar line.



Epipolar geometry to the rescue!

Epipolar geometry is crucial for stereo matching (the process of stereo vision) as it reduces the search space for point correspondence by eliminating false matches.

> Stereo matching is only feasible to do, if there's structure in the image. Therefore, stereo matching on a white wall performs very poorly.

Terminology

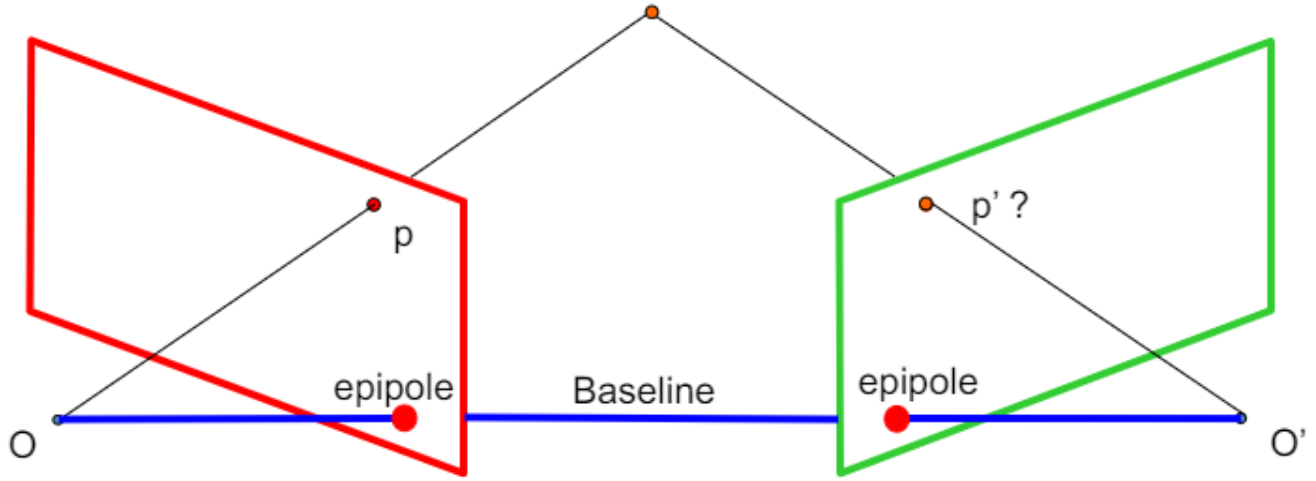
- *Baseline*: The line connecting the two camera centers.
- *Epipole*: Point of intersection of baseline with the image plane.
- *Epipolar plane*: The plane that contains the two camera centers and a 3D point in the world.
- *Epipolar line*: Intersection of the epipolar plane with each image plane.

How does it work?

We have two rigidly fixed cameras with different viewpoints capturing pictures. Now we want to know which pixels in an image pair (containing the left and right stereo images) corresponds to each other. If both cameras produce rays that pass through the red dots (projections of the 3D scene point onto the image plane), then they should intersect at that exact point, since each line passes through the scene point.

The ray from one image is reflected onto the image plane of the other image - displayed by the epipolar line that connects the epipole with the projection of the scene point in that image plane (red dot).

This way, the possible location of the right_image for example is constrained to a single line. Which reduces the search space for a pixel corresponding to a pixel of left_image.

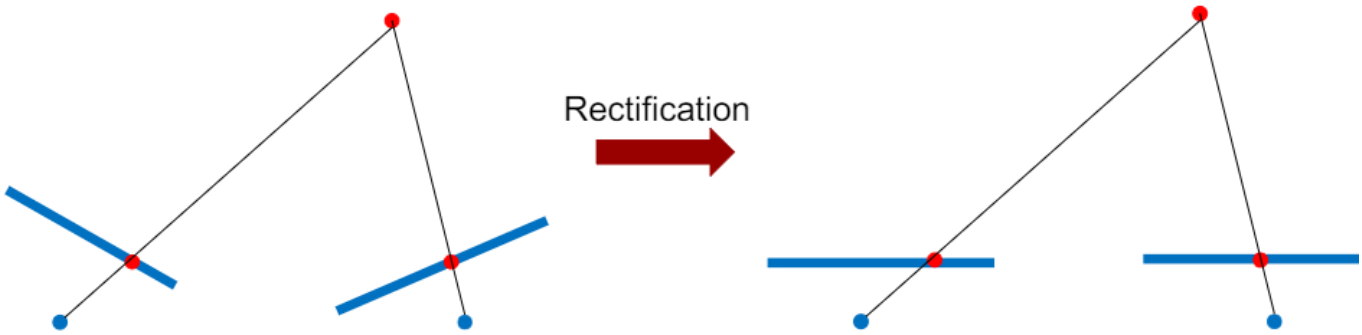


> We always spoke about finding the correspondence for one single pixel. To create a 3D structure of an image it's necessary to apply this method on each pixel of those image pairs.

Rectified Stereo Case

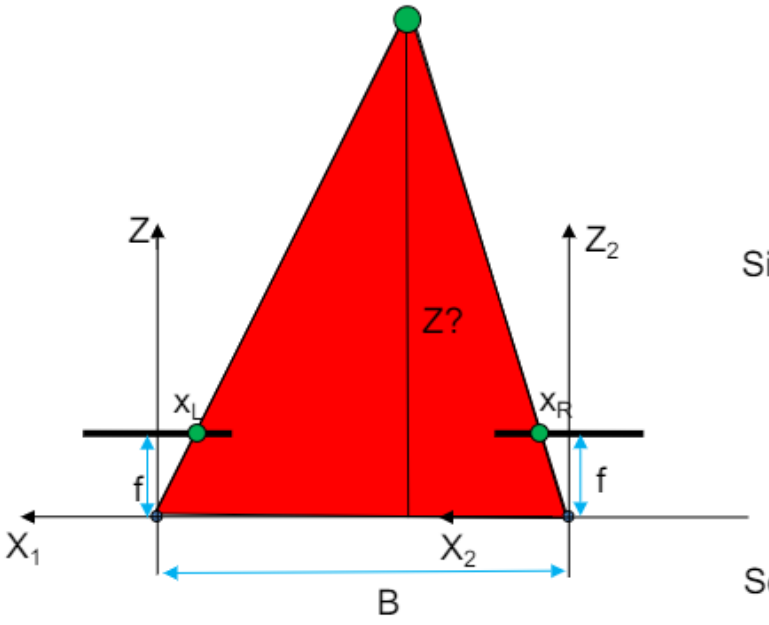
Rectification:

- The source images are projected onto a common plane parallel to the baseline B connecting the optical centers of the images
- Epipolar lines become parallel (and under certain conditions they become also horizontal)



Depth from Stereo Matches

It is important to note that Z extends from the scene point to the baseline. Therefore, we need to exploit similar triangles to formalize the following equation:



Similar triangles:

$$\frac{B+X_L-X_R}{Z-f} = \frac{B}{Z}$$

Solving for Z:

$$Z = f\frac{B}{X_L - X_R}$$

Disparity

1. The depth of the stereo image "Z" is proportional to the baseline of the point and inversely proportional to the difference of the corresponding points of the two images. This means that a small depth will result in a large difference or jump between the two images and vice versa.

2. The difference of the corresponding points of the two images is called the disparity.

3. A depth map is not the same as a disparity map. It's possible to convert from one to another as long as the focal length is known, but the difference between the two is that the disparity map is inversed and scaled, as only the pixel values are known. However, the information contained information is similar.

Correspondence Problem

Definition: The correspondence problem refers to the problem of determining which parts (clusters of pixels or individual pixels) of one image correspond to parts of another image. Most often, the differences arise from the movement of the camera, the flow of time and/or the movement of objects in the photos.

Beyond the hard constraint of epipolar geometry, there are "soft" constrains to help identify corresponding points:

- Similarity: The image pairs should be relatively similar in a sense, that they contain the same objects, colors, etc.
- Uniqueness: Objects in an image are assumed to contain unique features that are easily identifiable. These unique features should not differ in the two images, e.g. a person with a nose in the source image will most likely also have only one nose in the target image.
- Ordering: Let us assume that in a source image there is an apple, a coke and a teddy bear. We assume that this order remains the same in the target image, which is most likely the case. But there are exceptions to this rule.

- Disparity gradient is limited: The depth values should be continous for neighboring pixels. There will be jumps in depth and disparity, but they should bit limited.

> What is meant by hard constraint of epipolar geometry? This refers to the strict rule, that correpondences HAVE to be on the epipolar line. Soft constraints aren't always true, but most of the time they are.

To find matches in the image pair, we will assume:

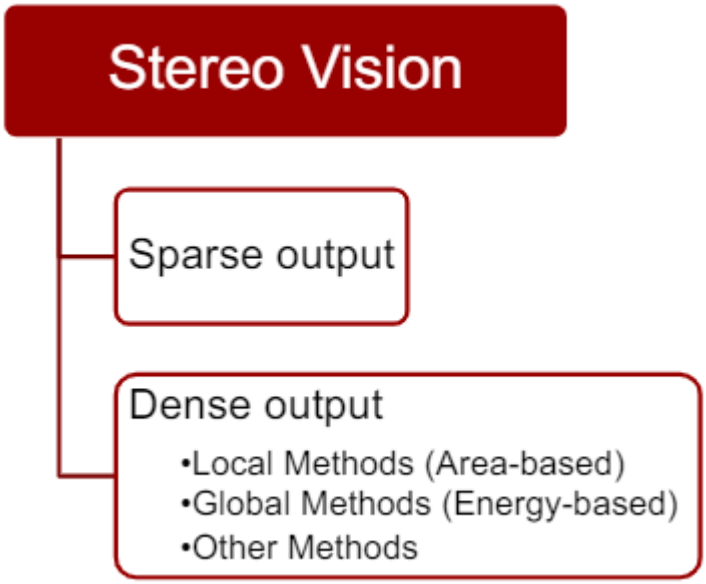
- most scene points are visibile in both images
- image regions for the matches are similar in appearance

Do we need dense or sparse stereo matching?

Before this question can be answered, we need to make sure that we understand the differences:

What's the difference between sparse or dense stereo matching? The difference lies in the number of calculations of your correspondences. With dense stereo matching, you need to calculate as many correspondences as possible, whereas with sparse stereo matching, only a small number of correspondences need to be calculated to meet the requirements of the application.

To come back to the question of which ones we need - it depends on the use case. For example, dense stereo matching is important for autonomous vehicles because you can't neglect any information to make sure it correctly assesses the situation and reacts accordingly. A face tracking system, on the other hand, needs only sparse stereo matching because it doesn't care about anything other than the face.

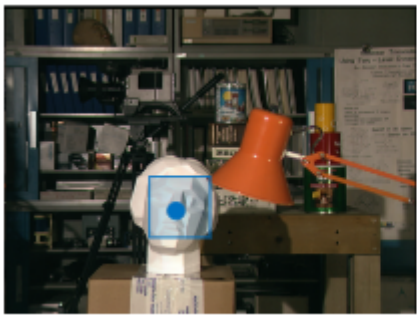
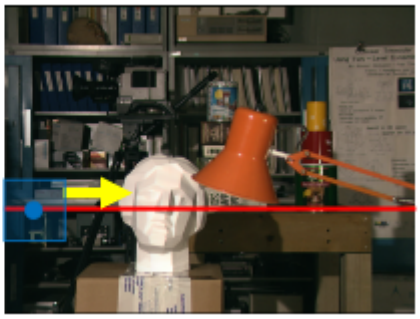


Dense Stereo Correspondence: Local Methods

Try to find correspondences for all the pixels of the reference image.

- For each epipolar line
    - For each pixel in the left image
        - Compare with every pixel on same epipolar line in right image
        - Choose the pixel that maximizes a similarity metric (or minimizes a dissimilarity metric!).

- Improvment: don't match individual pixels, but rather match windows!

> The proposed windows method slides from left to right along the epipolar line. To minimize dissimilarity while maximizing similarity.

Stereo Correpondence Metrics Different Metrics

Stereo Correspondence Metrics

- Sum of Absolute Differences (SAD)

$$SAD(x,y,d) = \sum_{x,y \in W} |I_l(x,y) - I_r(x,y-d)|$$

- Sum of Squared Differences (SSD)

$$SSD(x,y,d) = \sum_{x,y \in W} (I_l(x,y) - I_r(x,y-d))^2$$

- Normalized Cross-Correlation

$$NCC(x,y,d) = \frac{\sum_{x,y \in W} I_l(x,y) \cdot I_r(x,y-d)}{\sqrt{\sum_{x,y \in W} I_l^2(x,y) \cdot \sum_{x,y \in W} I_r^2(x,y-d)}}$$

- …many many more!!!

SAD Example:

$$\mathrm{A} = \begin{bmatrix} 2 & -10 & -2 \\ 14 & 12 & 10 \\ 4 & -2 & 2 \end{bmatrix}; \mathrm{B} = \begin{bmatrix} 6 & 10 & -2 \\ 0 & -12 & -4 \\ -5 & 2 & -2 \end{bmatrix};$$

$$\mathrm{C} = \mathrm{abs(A\text{-}B)} = \begin{bmatrix} |2-6| & |(-10)-10| & |(-2)-(-2)| \\ |14-0| & |12-(-12)| & |10-(-4)| \\ |4-(-5)| & |(-2)-2| & |2-(-2)| \end{bmatrix} = \begin{bmatrix} 4 & 20 & 0 \\ 14 & 24 & 14 \\ 9 & 4 & 4 \end{bmatrix};$$

$$\mathrm{SAD} = \mathrm{sum(C)}\ 4+20+0+14+24+14+9+4+4 = 93$$

> The closer the similarity metric calculated by the SAD is to 0, the stronger the similarity between these two images.

$$\mathrm{A} = \begin{bmatrix} 2 & -10 & -2 \\ 14 & 12 & 10 \\ 4 & -2 & 2 \end{bmatrix}; \mathrm{B} = \begin{bmatrix} 6 & 10 & -2 \\ 0 & -12 & -4 \\ -5 & 2 & -2 \end{bmatrix};$$

$$\mathrm{C} = \mathrm{A\text{-}B} \odot A - B = \begin{bmatrix} (2-6)^2 & ((-10)-10)^2 & ((-2)-(-2))^2 \\ (14-0)^2 & (12-(-12))^2 & (10-(-4))^2 \\ (4-(-5))^2 & ((-2)-2)^2 & (2-(-2))^2 \end{bmatrix}$$

$$= \begin{bmatrix} 16 & 400 & 0 \\ 196 & 576 & 196 \\ 81 & 16 & 16 \end{bmatrix};$$

$$SSD = sum(C) = 16 + 400 + 0 + 196 + 576 + 196 + 81 + 16 + 16 = 1497$$

SSD Example:

> The closer the similarity metric calculated by the SSD is to 0, the stronger the similarity between these two images. Note that SSD is generally only used due to its simplicity and relatively low computational cost - in general better results are achievable by using Normalized Cross Correlation.

Normalized Cross-Correlation Example:

$$\mathrm{A} = \begin{bmatrix} 2 & -10 & -2 \\ 14 & 12 & 10 \\ 4 & -2 & 2 \end{bmatrix}; \mathrm{B} = \begin{bmatrix} 6 & 10 & -2 \\ 0 & -12 & -4 \\ -5 & 2 & -2 \end{bmatrix};$$

$$A^2 = \begin{bmatrix} 4 & 100 & 4 \\ 196 & 144 & 100 \\ 16 & 4 & 4 \end{bmatrix}; B^2 = \begin{bmatrix} 36 & 100 & 4 \\ 0 & 144 & 16 \\ 25 & 4 & 4 \end{bmatrix};$$

$$\mathrm{nom} = \mathrm{A} \odot B = \begin{bmatrix} 2*6 & ((-10)*10) & ((-2)*(-2)) \\ 14*0 & (12*(-12)) & (10*(-4)) \\ (4*(-5)) & ((-2)*2) & (2*(-2)) \end{bmatrix}$$

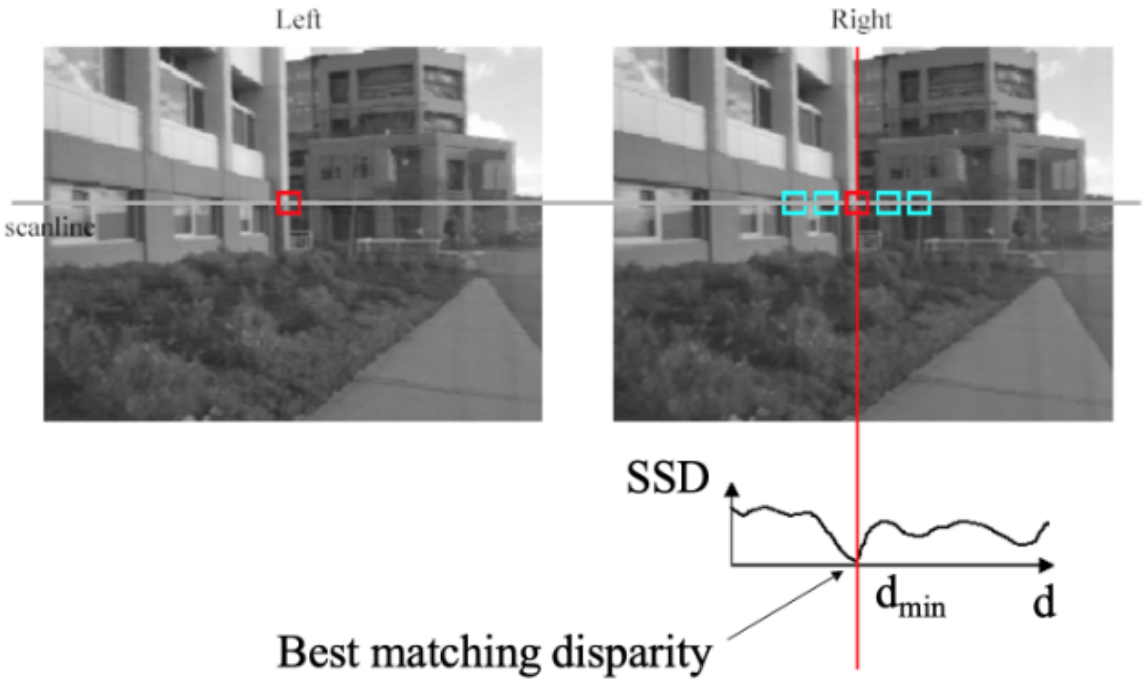
$$= \begin{bmatrix} 12 & -100 & 4 \\ 0 & -144 & -40 \\ -20 & -4 & -4 \end{bmatrix};$$

$$dom1 = A \odot A = \begin{bmatrix} 4 & 100 & 4 \\ 196 & 144 & 100 \\ 1644 & & \end{bmatrix}$$

$$dom2 = B \odot B = \begin{bmatrix} 26 & 100 & 4 \\ 0 & 144 & 16 \\ 25 & 4 & 4 \end{bmatrix}$$

$$NormalizedCross - Correlation = \frac{nom}{\sqrt{dom1} * \sqrt{dom2}} = \frac{-296}{\sqrt{572} * \sqrt{333}} = -0.678$$

Stereo Correspondence Metrics: SSD



> Visual illustration that the lowest SSD value found for a target pixel on the epipolar line indicates that this pixel is most similar to the source pixel and that there is therefore a correspondence between these pixels.

Windows Sizes

Large windows suppress noise better than smaller windows. However, if the window is too large, the image loses its fine granularity. With smaller windows, the structure is better preserved, but at the expense of a gain in noise.

Global Stereo Correspondence

Global Stereo Correspondence

- Up to this point, the disparity of each pixel was determined only by the information of the pixel itself and its neighborhood.
  - Thus, those methods are called ”local” or “area-based” methods.

- Global methods find better solutions in expense of more computations
  - Optimize jointly the disparity values of all the pixels of each scanline (e.g. Dynamic Programming)
  - Optimize jointly the disparity values of all the pixels of the image (e.g. graph cuts)

    - *In global algorithms, stereo correspondence is formulated as an energy function minimization problem, consisting of data and smoothness terms.*