# Depth estimation using stereo vision

▼ What is depth estimation using stereo vision

Depth estimation using stereo vision is a technique that involves determining the three-dimensional (3D) structure of a scene by analysing the disparities between corresponding points in images captured by a stereo camera setup. Stereo vision relies on the principle of triangulation, where the depth of a point in the scene is computed based on the differences in the positions of that point in the left and right images.

Here's an overview of how depth estimation using stereo vision works:

1. **Stereo Camera Setup:**
    - Two cameras are positioned side by side, mimicking the separation between human eyes. These cameras capture images of the same scene from slightly different perspectives.
2. **Image Rectification:**
    - The images from the stereo camera setup are rectified to ensure that corresponding points in the left and right images lie on the same scan line. This simplifies the depth estimation process.
3. **Feature Matching:**
    - Corresponding features (keypoints or pixels) in the rectified left and right images are identified. Matching is typically based on similarity measures, such as sum of squared differences (SSD) or normalized cross-correlation.
4. **Disparity Calculation:**
    - Disparity refers to the horizontal shift of a point between the left and right images. The disparity map is computed, indicating the pixel-wise differences in horizontal position between corresponding points.
5. **Depth Calculation:**

- Using triangulation, the depth of each point in the scene is calculated from its disparity. The baseline (distance between the two cameras) and the focal length of the cameras are critical parameters for this calculation.

6. **Depth Map:**
    - The result is a depth map, where each pixel in the image is associated with a depth value. Darker regions in the depth map correspond to closer objects, while brighter regions correspond to objects farther away.

Depth estimation using stereo vision is widely used in various applications, including robotics, autonomous vehicles, augmented reality, and 3D reconstruction. Accurate depth information is crucial for understanding the spatial layout of a scene and enabling machines to interact with the environment.

▼ Applications of depth estimation

Depth estimation has numerous applications across various domains due to its ability to provide three-dimensional information about a scene. Some key applications include:

1. **Robotics:**
    - **Navigation:** Robots can use depth information to navigate through complex environments, avoid obstacles, and plan optimal paths.
    - **Object Manipulation:** Depth perception helps in grasping and manipulating objects with precision.
2. **Autonomous Vehicles:**
    - Depth estimation is crucial for autonomous vehicles to understand the distance to other objects, pedestrians, and obstacles on the road.
3. **Augmented Reality (AR):**
    - AR applications use depth information to overlay virtual objects onto the real-world scene with accurate spatial alignment.
4. **3D Mapping and Reconstruction:**
    - Depth estimation contributes to the creation of detailed 3D maps and reconstructions of environments, useful in fields like archaeology, architecture, and urban planning.

5. **Gesture Recognition:**
    - Depth sensors, such as those used in Microsoft Kinect, enable accurate gesture recognition for human-computer interaction.
6. **Medical Imaging:**
    - Depth information is utilized in medical imaging for tasks like organ segmentation, surgical planning, and monitoring of medical procedures.
7. **Virtual Reality (VR):**
    - VR systems benefit from depth estimation to create immersive virtual environments that respond to users' movements and interactions.
8. **Security and Surveillance:**
    - Depth information aids in the analysis of video feeds for security and surveillance applications, helping to detect and track objects in a 3D space.
9. **Human-computer Interaction:**
    - Depth sensing cameras are used in various interactive applications, such as touchless interfaces and immersive gaming experiences.
10. **Industrial Automation:**
    - Depth estimation assists robots and automated systems in tasks like quality control, assembly, and inspection.
11. **Agriculture:**
    - Depth information can be used for precision agriculture, helping in tasks like plant monitoring, yield estimation, and autonomous farm equipment.
12. **Depth-based Imaging Effects:**
    - In photography and cinematography, depth information can be used to create artistic effects, such as selective focus and depth-of-field adjustments.

These applications demonstrate the versatility and significance of depth estimation in enhancing the capabilities of various technological systems and improving human-computer interactions in diverse fields.

▼ How are we going to perform our experiment

Depth estimation using stereo vision involves determining the disparity between corresponding points in left and right images captured by a stereo camera setup. Several algorithms and techniques can be employed for stereo depth estimation. Here are some common methods:

1. **Block Matching Algorithms:**
    - **Stereo Block Matching (BM):** Divides the images into blocks and searches for the most similar block in the other image. Simple and computationally efficient, but may not perform well in textured or occluded regions.
2. **Semi-Global Block Matching (SGBM):**
    - A variant of block matching that considers the global information of the image. It minimises a global energy function, taking into account consistency along multiple paths.
3. **Graph Cuts:**
    - **Graph Cuts (GC):** Formulates depth estimation as a graph-cut problem. It considers smoothness constraints and data terms to find the optimal disparity map.
4. **Belief Propagation:**
    - **Belief Propagation (BP):** Iteratively refines the initial disparity estimate based on messages exchanged between neighbouring pixels. It considers the consistency of disparities along different paths.
5. **Semi-Global Matching (SGM):**
    - An extension of SGBM that considers multiple paths along different directions. It is effective in handling occlusions and textureless regions.
6. **Local Methods:**
    - **Local Methods:** Methods based on local optimisation techniques, such as gradient-based methods or correlation-based methods.
7. **Machine Learning Approaches:**
    - **Deep Learning:** Convolutional Neural Networks (CNNs) can be trained to learn depth from stereo image pairs. Models like StereoNet and GC-Net have shown success in end-to-end depth estimation.

8. **Phase-Based Methods:**
    - **Phase-Based Methods:** Analyse the phase information of the images. Techniques like Phase Correlation and Phase-Shift Stereo use phase differences for disparity estimation.
9. **Time-of-Flight (ToF) Cameras:**
    - **Time-of-Flight (ToF) Cameras:** Use the time taken for light to travel from the camera to the object and back. These cameras provide depth information directly.
10. **Structured Light:**
    - **Structured Light:** Involves projecting a known pattern onto the scene and analysing the deformations in the pattern to estimate depth.

The choice of method depends on factors such as the scene characteristics, computational resources, and accuracy requirements. Combining different methods or using hybrid approaches is also common for robust stereo depth estimation.

▼ What is disparity

**Disparity** in the context of stereo vision refers to the apparent shift or difference in the position of an object's location between the left and right images captured by a stereo camera setup. It is a measure of how much one image has to be shifted horizontally to align with the corresponding point in the other image.

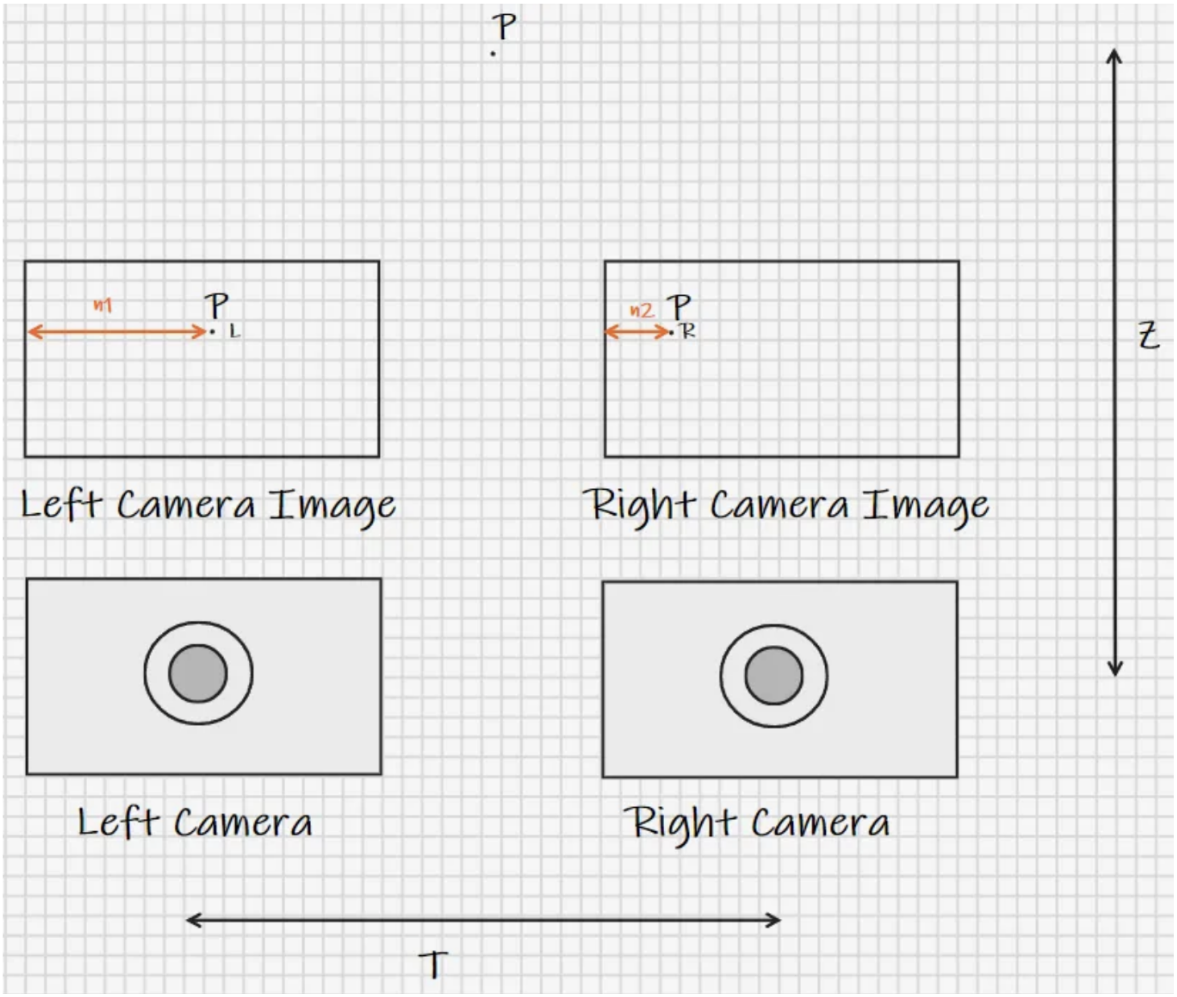
Let's break this down with an example:

1. **Stereo Camera Setup:**
    - Imagine you have a stereo camera system with two cameras, left and right, capturing the same scene simultaneously. These cameras are slightly displaced horizontally, mimicking the separation between human eyes.
2. **Corresponding Points:**
    - An object in the scene is viewed by both the left and right cameras. The projection of this object in the left image and the right image corresponds to the same 3D point in the real world.
3. **Disparity Calculation:**

- The disparity is the horizontal shift required to bring the corresponding points into alignment. It is typically measured in pixels. If the object is closer to the cameras, the disparity will be larger, and if it's farther away, the disparity will be smaller.

4. **Example:**
    - Consider a stereo pair of images where you have a distinct feature, like the corner of a book, visible in both images. In the left image, the corner might be at pixel coordinates (100, 150), and in the right image, it might be at coordinates (120, 150).
    - The disparity in this case would be the horizontal shift needed to align these points: `120 - 100 = 20` pixels. This means that the corresponding point in the right image is 20 pixels to the right of the point in the left image.
    - The relationship between depth and disparity is inversely proportional – objects that are closer will have a larger disparity, and objects that are farther away will have a smaller disparity.
5. **Depth Estimation:**
    - Knowing the baseline (horizontal separation between the cameras) and focal length, the depth of the object can be estimated from the disparity using trigonometry.

In summary, disparity is a key concept in stereo vision that quantifies the horizontal shift between corresponding points in a stereo pair of images. It forms the basis for estimating the depth information of the scene, contributing to applications such as 3D reconstruction and depth mapping.


P
n1
P
L
n2
P
R
Z
Left Camera Image
Right Camera Image
Left Camera
Right Camera
T

- P = Target point in the physical world (scene point)
- P$L$ = (x$l$,y$l$) = Point P in left camera image
- P$R$ = (x$r$,y$r$)= Point P in right camera image
- n1 = Horizontal pixel distance of point P$L$ in left camera image
- n2 = Horizontal pixel distance of point P$R$ in right camera image
- T = Baseline distance between center of left and right cameras
- f = Focal length of the camera
- d = Physical size of a pixel in camera sensor CMOS/CCD
- Z = Distance between point P and camera center

**Disparity (D) = n1-n2 =x$l$-x$r$**

▼ Relation between disparity and depth

The relationship between disparity and depth in stereo vision is based on the principles of triangulation. Triangulation is a geometric method that involves measuring the angles of a triangle to determine the distances of its sides or the lengths of its sides to determine the distances of its vertices. In the context of stereo vision:

1. **Baseline:**
    - The baseline is the horizontal separation between the two cameras in a stereo setup. It represents the distance between the optical centers of the left and right cameras.
2. **Epipolar Geometry:**
    - Corresponding points in the left and right images are connected by epipolar lines, which are lines in space along which the corresponding points must lie.
3. **Triangulation:**

- When a point in the 3D world is imaged by both cameras, the disparity between its projections on the left and right images is related to the depth of the point.

4. **Inverse Proportionality:**
    - The relationship is inversely proportional because as the depth of the point increases, the disparity decreases, and vice versa. This can be understood through the geometry of the triangles formed by the optical centers of the cameras, the point in 3D space, and its projections on the image planes.
5. **Similar Triangles:**
    - When two triangles are similar (meaning their corresponding angles are equal), the ratio of the lengths of corresponding sides is constant.
    - In stereo vision, the triangles formed by the baseline, the depth of the point, and the disparities in the left and right images are similar triangles.
    - The disparity in pixels is directly proportional to the baseline and inversely proportional to the depth.
    - Mathematically, this relationship is expressed as: $\text{Depth} \propto \text{Baseline}/\text{Disparity}$

In summary, the inverse proportionality between disparity and depth arises from the geometry of the stereo vision setup and the principles of triangulation. As objects move farther away from the cameras, their corresponding disparities decrease, providing a basis for depth estimation in stereo vision.

▼ Our implementation

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

class DepthMap:
    def __init__(self,showImages):
        # Load Images
        self.imgLeft = cv.imread('im0.png',cv.IMREAD_GRAYSCALE)
        self.imgRight = cv.imread('im1.png',cv.IMREAD_GRAYSCALE)

        if showImages:
            plt.figure()
            plt.subplot(121)
            plt.imshow(self.imgLeft)
```

```
            plt.subplot(122)
            plt.imshow(self.imgRight)
            plt.show()

    def computeDepthMapBM(self):
        nDispFactor = 12 # adjust this
        stereo = cv.StereoBM.create(numDisparities=16*nDispFactor, blockSize=21)
        disparity = stereo.compute(self.imgLeft,self.imgRight)
        plt.imshow(disparity,'gray')
        plt.show()

    def computeDepthMapSGBM(self):
        window_size = 7
        min_disp = 16
        nDispFactor = 14 # adjust this (14 is good)
        num_disp = 16*nDispFactor-min_disp

        stereo = cv.StereoSGBM_create(minDisparity=min_disp,
                                      numDisparities=num_disp,
                                      blockSize=window_size,
                                      P1=8*3*window_size**2,
                                      P2=32*3*window_size**2,
                                      disp12MaxDiff=1,
                                      uniquenessRatio=15,
                                      speckleWindowSize=0,
                                      speckleRange=2,
                                      preFilterCap=63,
                                      mode=cv.STEREO_SGBM_MODE_SGBM_3WAY)

        # Compute disparity map
        disparity = stereo.compute(self.imgLeft,self.imgRight).astype(np.float32) / 16.0

        # Display the disparity map
        plt.imshow(disparity, 'gray')
        plt.colorbar()
        plt.show()

def demoViewPics():
    # See pictures
    dp = DepthMap(showImages=True)

def demoStereoBM():
    dp = DepthMap(showImages=False)
    dp.computeDepthMapBM()

def demoStereoSGBM():
    dp = DepthMap(showImages=False)
    dp.computeDepthMapSGBM()

if __name__ == '__main__':
    # demoViewPics()
```

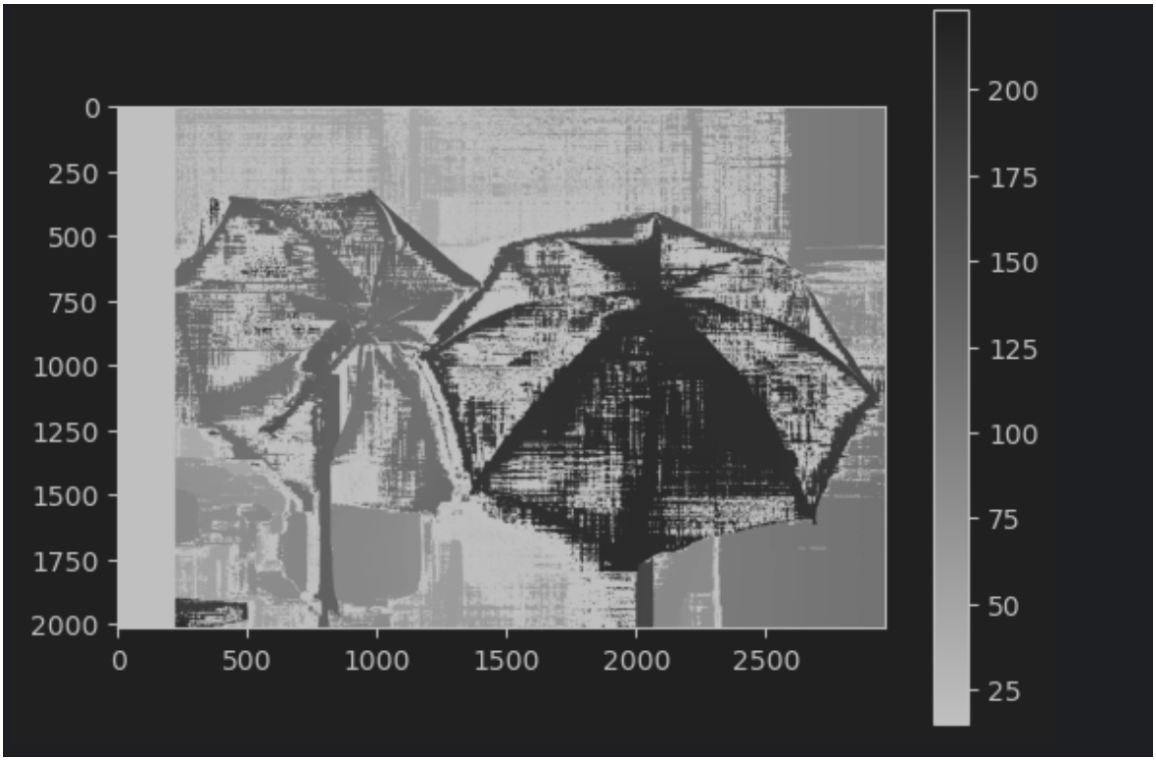
```
    # demoStereoBM()
    demoStereoSGBM()
```

▼ Our input

Left image :

Right image:

▼ Our output(depth map)



▼ Further scope for our project

- The current implementation for our project has been done using open cv and is more of a mathematical model(use of epipolar geometry) than one with neural networks and learning. This project is more inclined towards computer vision.
- For the final review rather than feeding images manually by downloading datasets from popular websites like Middlebury we can use our own camera to capture the left and right images. Apart from this we can also calculate the depth value of the objects captured in the image.
- Another addition which we can do to our current project is the use of CNN rather than open cv to perform our feature engineering. We can obtain the pixel values by using CNN to break down the image rather than doing the same with open cv.

▼ References


- https://medium.com/analytics-vidhya/distance-estimation-cf2f2fd709d8
- https://github.com/Surya-0/Depth-Estimation-using-Stereovision
- https://arxiv.org/pdf/1804.06324.pdf
- https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9675391&tag=1
- https://arxiv.org/pdf/2206.10375.pdf
- https://vision.middlebury.edu/stereo/data/scenes2014/