Chapter

Efficient Depth Estimation Using Sparse Stereo-Vision with Other Perception Techniques

Satyarth Praveen

Abstract

The stereo vision system is one of the popular computer vision techniques. The idea here is to use the parallax error to our advantage. A single scene is recorded from two different viewing angles, and depth is estimated from the measure of parallax error. This technique is more than a century old and has proven useful in many applications. This field has made a lot of researchers and mathematicians to devise novel algorithms for the accurate output of the stereo systems. This system is particularly useful in the field of robotics. It provides them with the 3D understanding of the scene by giving them estimated object depths. This chapter, along with a complete overview of the stereo system, talks about the efficient estimation of the depth of the object. It stresses on the fact that if coupled with other perception techniques, stereo depth estimation can be made a lot more efficient than the current techniques. The idea revolves around the fact that stereo depth estimation is not necessary for all the pixels of the image. This fact opens room for more complex and accurate depth estimation techniques for the fewer regions of interest in the image scene. Further details about this idea are discussed in the subtopics that follow.

Keywords: stereo vision, computer vision, disparity, depth estimation, camera, feature extraction

1. Introduction

As researchers and innovators, we have often tried to take hints and ideas from nature and convert them into beautiful versions of technology that can be used for the betterment and advancement of the human race. The human eyes inspire yet another artificial visual system, the stereo vision. The idea is to use the parallax error from two different viewing angles of the same object to estimate the distance of the object from the camera. The parallax error is inversely proportional to the depth and brings it down to a single trivial equation, whereas the estimation of the parallax error, known as the disparity between the pixels in the image frames, is a much engaging nontrivial task to handle. Depth estimation is possible only for the overlapping fields of view between the two views as shown in **Figure 1**. The multi-view system is a much better, reliable, and robust setup for depth estimation of the objects in the image compared to a monocular view. Details regarding this are discussed in the following subsections of the chapter.

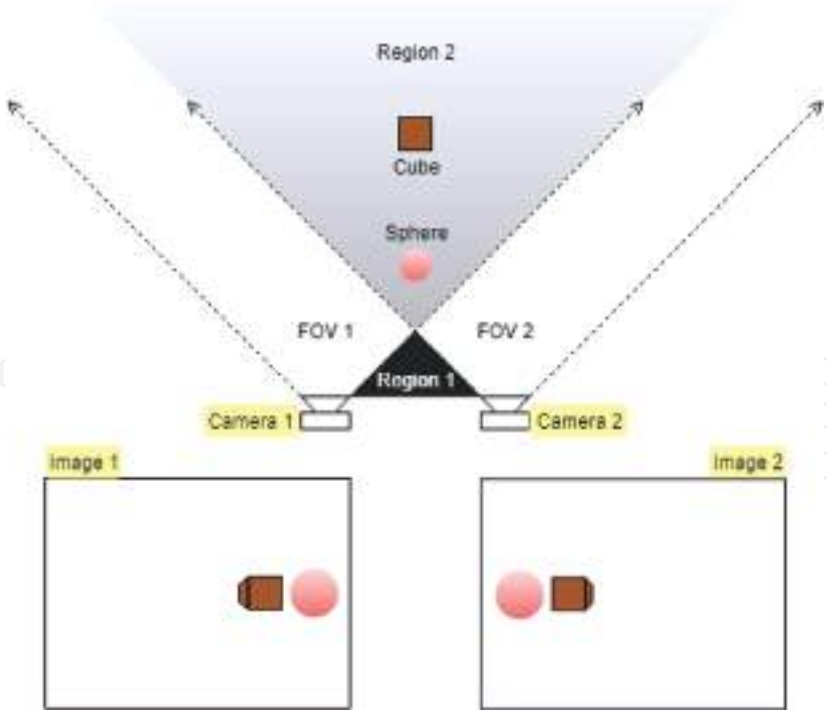

Figure 1.
The stereo setup.

This instrument was first described to us in 1838 by Charles Whitestone to view relief pictures. He called it the stereoscope. A lot of other inventors and visionaries later used this concept to develop their versions of stereoscopes. It even led to the establishment of the London Stereoscopic Company in 1854. The concept of depth estimation using multiple views was used even for the estimation of the distance of the far away astronomical objects in the early times. The depth is also directly proportional to the distance between the two cameras of the stereo vision system, also called the baseline. Hence the estimation of such vast distances demanded us to use the longest possible baseline length that we could use. So the data was recorded from Earth being on either side of the sun, making the baseline length to be the same as the diameter of the Earth's orbit around the sun, and then the depth of the astronomical objects is measured. This method was called the stellar parallax or trigonometric parallax [1].

Considering other applications, robotic applications demand plenty of stereo vision systems for close object depth estimations. Be it humanoids, robots for folding clothes or picking objects, or even autonomous vehicles, stereo vision systems solve many complexities. On top of that, if the use case is for unidirectional short-range applications, good stereo systems can even eradicate the need for lidars or radars and hence aid toward much cost-cutting.

This chapter presents a new idea while using the existing techniques for depth estimation. The motivation is to make the depth estimation procedure a lot lighter and faster. In simple words, the intension is to avoid the calculation of depth for the pixels that are not required. It is most usable when coupled with other perception techniques like object-detection and semantic-segmentation. These perception steps help us rule out the unrequired pixels for which depth estimation can be avoided. The implications and findings of this are discussed later.

Future sections of the chapter are primarily segregated as the Background and the Proposed Approach. The Background is arranged as follows: the overview of the architecture; camera calibration; stereo matching problem, i.e., disparity; and depth estimation. Further, the proposed approach contains the algorithm, the results, and possible future works.

2. Background

2.1 The overview of the stereo architecture

This architecture presents a simple overview of how the stereo system works. As shown in **Figure 2**, cameras with similar properties are calibrated individually for their intrinsic calibration parameters (Subtopic 2.2.1). The two cameras are then mounted on a rigid stereo rig and calibrated together as a single system to get the extrinsic calibration parameters (Subtopic 2.2.2). The images collected from the two cameras are then undistorted to remove the camera distortion effects. From the extrinsic calibration parameters, we know the rotation and translation of one camera w.r.t. the other (right camera w.r.t. the left camera); we use this information to align the two images from the stereo system along the epipolar line (Subtopic 2.2.2). The image pair is then used for disparity estimation (Topic 2.3), the most nontrivial part of the process. The concept proposed in this chapter targets this substep of the process. Perfect pixel matching is a hard problem in itself. So, achieving a real-time performance on images makes the problem nothing but more complex. Once we have a pixel-to-pixel correspondence between the two images, i.e., the disparity for each pixel, we can directly compute the depth for each of them using a single formula. The following topics discuss the steps as mentioned above in greater detail.

2.2 Camera calibration

Camera calibration is a fundamental step in computer vision applications. There are two aspects of camera calibration, namely, intrinsic calibration and extrinsic calibration. Some of the experts whose algorithms are used for camera calibration are Zhang [2], Scaramuzza [3], Jean-Yves Bouguet [4], and Tsai [5].

2.2.1 Intrinsic camera calibration

Intrinsic calibration, Step 2 in **Figure 2**, provides us with the internal properties of the camera, such as focal length in pixels, optical center in pixels, shear constant, aspect ratio, and distortion coefficients.

- The *optical center* is the position in the image that coincides with the principal axis of the camera setup.

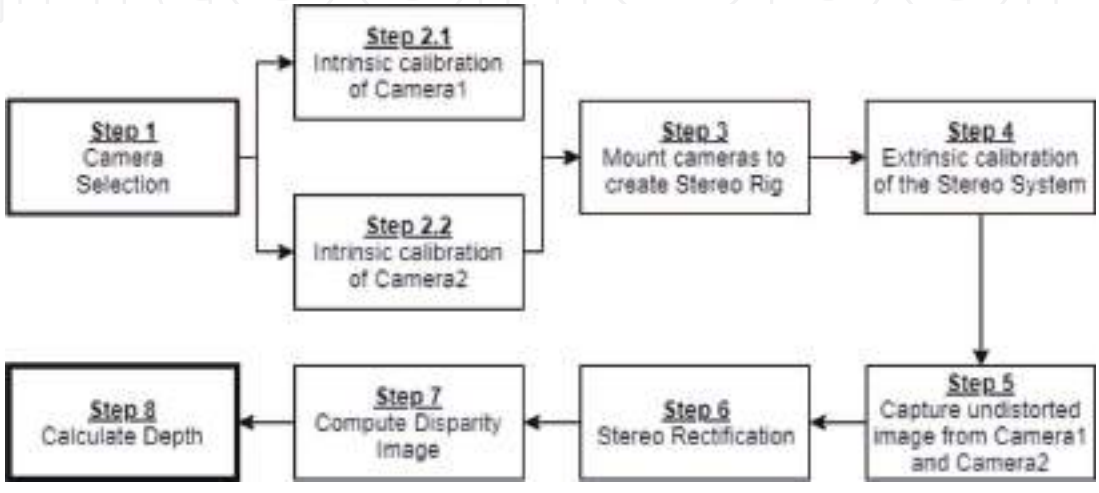

Figure 2.
The architecture overview.

- *Shear* is the slant orientation of the image recorded. This disorientation may occur during the digitized process of grabbing the image frame from the sensors. Based on today's technical advancements and complex systems, it is safe to assume that the recorded image has zero or very close to zero shears.

- *Aspect ratio* defines the shape of the pixels of the image sensor. For example, the National Television System Committee (NTSC) TV system defines non-square image pixels with an aspect ratio of 10:11. However, in most of the general cases, it is safe to assume that pixels are square and hence the aspect ratio is 1.

- *Distortion coefficients* are used to undistort the recorded image from the camera. The camera image is prone to pick up some distortions based on the built of the lenses and the camera system or based on the position of the object and the camera. The former is called optical distortion, and the latter is called perspective distortion. Distortion coefficients are used to undistort the optical distortions only. Undistorting the images ensures that the output image is not affected by any of the manufacturing defects in the camera, at least in the ideal case. There are three kinds of optical distortions:

    - *Barrel distortion*: the lines seem to be curving inward as they move away from the camera center.

    - *Pincushion distortion*: the lines seem to be curving outward as they move away from the camera center.

    - *Mustache distortion*: this is a mix of the two distortions and the toughest one to handle.

2.2.2 Extrinsic camera calibration

While intrinsic calibration provides us with intrinsic camera properties, extrinsic calibration provides us with external details like the effective movement w.r.t., a reference point in the three-dimensional world coordinate system. These constants incorporate the movement of the camera frame in six degrees of freedom. Considering the axes shown in **Figure 3**, if the image plane lies in the X-Y plane and the camera is oriented along the Z-axis, the six degrees of freedom are translation along the X-axis, translation along the Y-axis, translation along the Z-axis, rotation along the X-axis (pitch), rotation along the Y-axis (yaw), and rotation along the Z-axis (roll).

Extrinsic calibration, Step 4 in **Figure 2**, is particularly crucial in the stereo camera setup because it gives the exact baseline distance between the two camera centers. The approximate baseline is decided initially before setting up the camera units. This decision is necessary and different depending on the application of the stereo system. As the baseline length is directly proportional to the detected object depth, a more extended baseline would increase the range of the system to measure more considerable distances, while a shorter baseline would allow only short-range depth estimation. The downside to a larger baseline is the smaller overlap between the views of the two cameras. So although the system would have a greater range, it will only be for a smaller section of the view, whereas a stereo system with a smaller baseline would have a much larger overlapping view and hence would provide short-range distance estimation for a more extensive section of the view. Neither of the two systems can replace one another. Hence, keeping this significant difference in mind while choosing the correct baseline is essential.

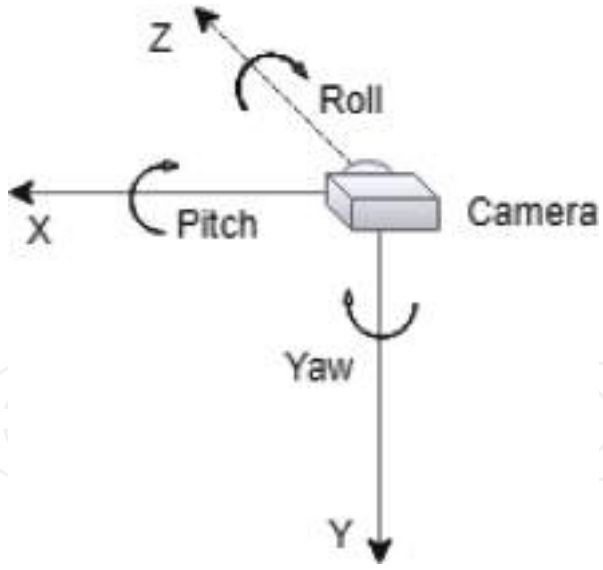

Figure 3.
Camera axes.

In the stereo camera system, one camera is the reference frame, and the other camera is calibrated w.r.t. the first camera. Hence, after the extrinsic calibration, if the two cameras are arranged along the X-axis, the baseline length information is returned as the translation along the same axis. Along with the rotation and translation, stereo calibration also updates the focal length of the overall stereo camera system. This focal length is common to both the cameras in the stereo system and is different from that of the individual focal lengths of the two cameras. The reason is that the two cameras now need to look at the joint portion of the scene; hence, choosing similar cameras, Step 1 in **Figure 2**, if not identical, can be an essential factor for a good stereo system. Dissimilar cameras significantly affect the image quality when using a common focal length. The camera with a more considerable difference between the old focal length and the new focal length gives a highly pixelated image. This difference in the image quality of the two cameras reflects in the later stage of disparity estimation. It makes the process of finding the corresponding pixels in the two images much harder; hence, it might lead to wrong disparity estimation or unnecessary noise.

Another use of the extrinsic parameters is image rectification, Step 6 in **Figure 2**. Computing disparity is not impossible without this step, but the problem statement becomes a lot easier if we rectify the output images of the stereo pair. Also, unrectified images are more prone to incorrect disparity estimation. In this step, we warp the output image of the second camera using the extrinsic parameters w.r.t. the reference camera. This warping ensures that the pixels belonging to the same objects in the two cameras lie along the same scan line in both images. So instead of the larger search space, i.e., the complete image, the search for disparity estimation can be restricted to a single row of the image. This scan line is called the epipolar line, and the plane that intersects with this epipolar line and the object point in 3D world coordinate is called the epipolar plane (see **Figure 4**). This process dramatically reduces the computations required by the disparity algorithm.

2.3 Disparity/stereo matching

This section talks about the most nontrivial aspect of the entire process of depth estimation using stereo, i.e., computing the disparity map from the stereo image pair. If considering the raw image pair from the stereo, the entire image is the search space to find the corresponding matching pixel. Although we might be able

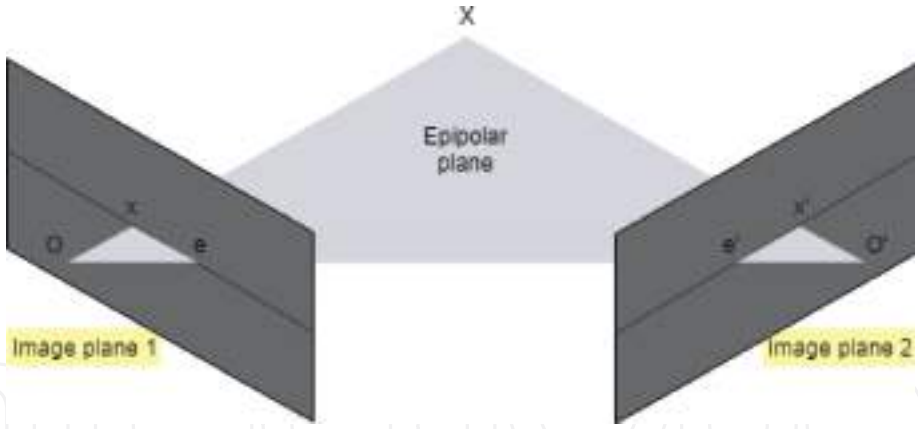

Figure 4.
The epipolar plane. X is the object point in the world coordinates, x and x′ are the corresponding pixels in the two image planes, e and e′ are the epipoles of the two image planes, and O and O′ are the corresponding camera centers.

to streamline the search space a little bit based on common sense, that will still not be comparable to searching a single row of the image. In an ideal case, the most robust system would be the one that can overlook all the image distortions, artifacts, and occlusion cases and give us a pixel-to-pixel disparity estimation by finding its perfect match in the corresponding image. [6–8] are some of the datasets that provide us with the ground truth disparity images along with the stereo image pair (see **Figure 5**). Researchers came up with different novel ideas and techniques involving custom calibration methods, high-end camera units, sensors, and better disparity estimation techniques to estimate sub-pixel disparities for highly accurate ground truth [9, 10]. While these methods are suitable to generate ground truths, real-time systems demand inexpensive solutions. Hence, in most of the cases, the applications do not require extremely accurate calibration but rely on fairly good camera calibration, inexpensive image rectification, and simple matching algorithms to get good enough disparity maps.

One of the significant elements of the stereo matching algorithms is the cost function that is used to evaluate the similarity. Some of the significant cost functions are:

The sum of squared difference (SSD)

$$C_{SSD}(d) = \sum_{(u,v) \in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2 \quad (1)$$

Figure 5.
Middlebury stereo dataset. Scene (left), ground truth disparity (right).

The sum of absolute difference (SAD)

$$C_{SAD}(d) = \sum_{(u,v) \in W_m(x,y)} |I_L(u,v) - I_R(u-d,v)| \tag{2}$$

Normalized cross-correlation (NCC)

$$\text{Normalized pixel} : \hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\left|\left|I - \bar{I}\right|\right|_{W_m(x,y)}} \tag{3}$$

$$C_{NC}(d) = \sum_{(u,v) \in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v) \tag{4}$$

In Eqs. (1)–(4), below is the legend for the symbols used:

I_L – *Left image or first camera image*

I_R – *Right image or second camera image*

W_m – *Matching window*

d – *Pixel disparity*

$I(u,v)$ – *Image pixel intensity at location u, v*

Although these cost functions are decent choices for similarity measure, they are considerably affected by factors such as illumination differences and viewing angles. To minimize the effect these factors have on the output, the pixel patches used for similarity check can be normalized before using SSD or SAD similarity values. Some other approaches that help make the algorithm independent of such factors are rank transform and census transform. These transformations eliminate the sensitivity toward absolute intensity and outliers.

Despite handling these sensitive cases, it takes a lot to estimate a dense disparity output. Obtaining a "dense" disparity map with restricted computations is the major challenge when designing algorithms. The dominant factors affecting the similarity measure of the corresponding pixels are as follows:

- Photometric constraints (Lambertian/non-Lambertian surfaces)

Lambertian surfaces follow the property of Lambertian reflectance, i.e., they look the same to the observer irrespective of the viewing angle. An ideal "matte" surface is an excellent example of a Lambertian surface. If the surface in the scene does not follow this property, it might appear to be different regarding illuminance and brightness in the two camera views. This characteristic can lead to incorrect stereo matching and hence wrong disparity values.

- Noise in the two images

Noise can be present in the images as a result of low-quality electronic devices or shooting the images at higher ISO settings. Higher ISO settings make the sensor more sensitive to the light entering the camera. This setting can magnify the effect of unwanted light entering the camera sensor and is nothing but noise. This noise is most certainly different for the two cameras and hence again making disparity estimation harder.

- Pixels containing multiple surfaces

This issue occurs mainly for an object lying far away in the scene. Since the baseline is directly proportional to the distance of objects, stereo systems with smaller

baseline face this issue even at average distances, whereas systems with larger baseline face it at a greater distance. It's something similar along the lines of Johnson's criteria [11] that we are a little helpless for this kind of problem. Hence it is crucial to choose the stereo baseline suitable to one's use case.

- Occluded pixels

These are those pixels of the 3D scene that are visible in one frame and not visible in the other (see **Figure 6**). It is practically impossible to find the disparity of these pixels as no match exists for that pixel in the corresponding image. The disparities for these pixels are only estimated with the help of smart interpolation techniques or reasonable approximations.

- The surface texture of the 3D object

This property of the object is another factor leading to confused or false disparity estimation. Surfaces such as a blank wall, road, or sky have no useful texture, and hence it is impossible to compute their disparity based on simple block matching techniques. These kinds of use cases require the intelligence of global methods that consider the information presented in the entire image instead of just a single scan line (discussed later in the chapter).

- The uniqueness of the object in the scene

If the object in the scene is not unique, there is a good chance that the disparity computed is incorrect because the algorithm is vulnerable to matching with the wrong corresponding pixel. A broader view of the matching patch can help here up to a certain extent, but that comes with the additional cost of required computations.

- Synchronized image capture from the two cameras

The images captured from the two cameras must be taken at the same time, especially in the moving environment scenarios. In the case of continuous scene recording, the output from the two cameras can be synchronized at the software level, or the two cameras can be hardware-triggered for the synchronized output image. While hardware trigger gives perfectly synchronized output, software level synchronization is a lot more easily achieved with decently accurate synchronization.

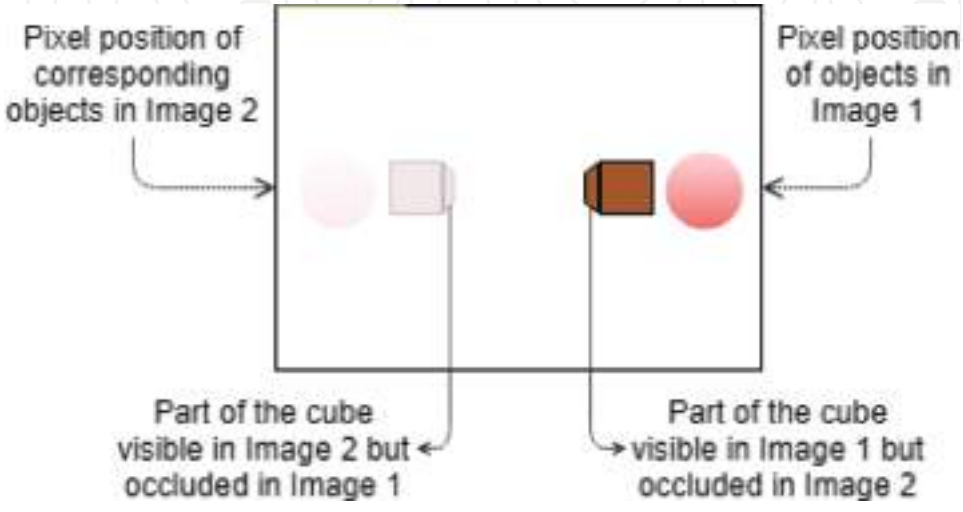

Figure 6.
Occlusion.

A few of these unfavorable aftereffects can be handled with post-processing of the disparity maps, but they can aid us only to a certain extent. A dense disparity map at real time is still a nontrivial task. Some of the cases to be kept in mind when working on a post-processing algorithm are as follows:

- Removal of spurious stereo matches

The median filter is an easy way to tackle this problem. However, it might fail in the case of a little larger spurious disparity speckles. Speckle filtering can be done using other approaches, such as the removal of tiny blobs that are inconsistent with the background. This approach gives decent results. Though this removes most of the incorrect disparity values, it leaves the disparity maps with a lot of holes or blank values.

- Filling of holes in the disparity map

Many factors lead to blank values in the disparity map. These holes are caused mainly due to occlusion or the removal of false disparity values. Occlusion can be detected using the left–right disparity consistency check, i.e., two disparity maps, each w.r.t. the first and second camera image can be obtained, and the disparity values of the corresponding pixels must be the same; the pixels that are left out are ideally the occluded pixels. These holes can be filled by surface fitting or distributing neighboring disparity estimates.

- Sub-pixel estimation

Most of the algorithms give integer disparity values. However, such discrete values give discontinuous disparity maps and lead to a lot of information loss, particularly at more considerable distances. Some of the common ways to handle this are gradient descent and curve fitting.

Having seen the cost functions and the challenges in computing the disparity, we can now go on to the algorithms used for its computation. Starting from a broader classification of the approaches, they are talked about in the following subtopics for the most common techniques of disparity estimation.

2.3.1 Local stereo matching methods

Local methods tend to look at only a small patch of the image, i.e., only a small group of pixels around the selected pixel is considered. This local approach lacks the overall understanding of the scene but is very efficient and less computationally expensive compared to global methods. The issue with not having the complete understanding of the whole image leads to more erroneous disparity maps as it is susceptible to the local ambiguities of the region such as occluded pixels or uniform-textured surfaces. This noise is taken care of up to a certain extent by some post-processing methods. The post-processing steps have also received significant attention from the experts as it helps keep the process inexpensive. Area-based methods, feature-based methods, as well as methods based on a gradient optimization lie in this category.

2.3.2 Global stereo matching methods

Global methods have almost always beaten the local methods concerning output quality but incur large computations. These algorithms are immune to local

peculiarities and can sometimes handle difficult regions that would be hard to handle using local methods. Dynamic programming and nearest neighbor methods lie in this category. Global methods are rarely used because of their high computational demands. Researchers mostly incline toward the local stereo matching methods because of its vast range of possible applications with a real-time stereo output.

Block matching is among the simplest and most popular disparity estimation algorithms. It involves the comparison of a block of pixels surrounding the pixel under study. This comparison between the two patches is made using one or a group of cost functions that are not restricted to the ones mentioned above. SSD and SAD perform pretty well and hence are the first choices in many algorithms (see **Figure** 7 for the disparity output of the stereo block matching algorithm).

Some modifications to this basic approach that exists in the current literature are variations in the shape, size, and count of the pixel blocks used for each pixel of interest. Other areas of modification include the cost function and preprocessing and post-processing of the disparity map. [12–14] are some examples of the approaches mentioned above. Although most of these modifications show improvement in the accuracy and quality of the obtained disparity map, they all come with an added computational expense. Hence, like most of the algorithmic choices, even the stereo matching algorithms boil down to the direct trade-off between computation and accuracy. So it is particularly important to choose the algorithms based on the specific applications and the use case that governs their usability. With these limitations in place, the time has presented us with excellent technical advances, and hence many researchers are now devising solutions with the power of GPUs in mind. Parallelizing the above algorithms makes them compatible to run on GPUs and overcome most of the speed limitations. Though the number of computations being done is almost the same, their parallel execution takes a lot less time compared to their serial execution. This advancement opens doors for the execution of more complex algorithms much faster and hence allows better quality outputs in real time.

2.4 Depth estimation

2.4.1 Conventional method

Once we already have the disparity map for a pair of stereo images, getting the pixel-wise distance from it is the easy part. This information can be obtained using a linear formula (see Eq. (5)):

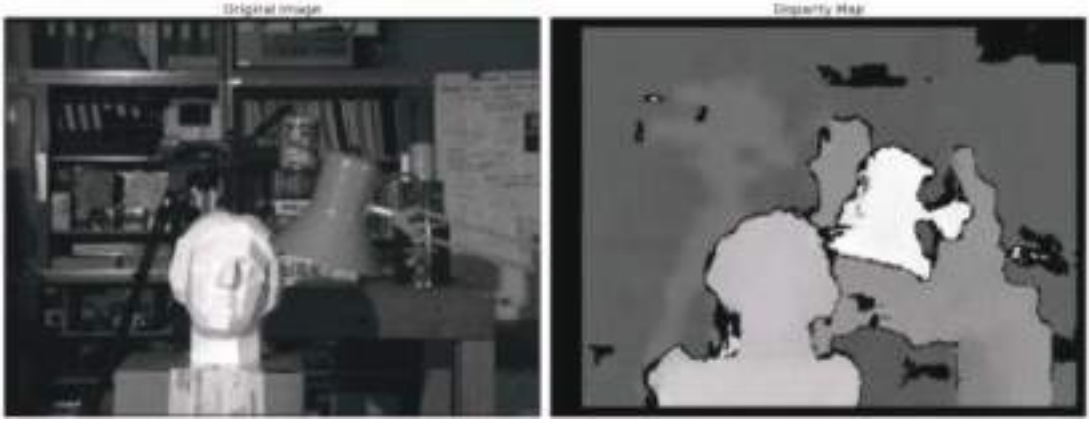

Figure 7.
Disparity output using stereo block matching algorithm.

$$z = \frac{f \times B}{d} \tag{5}$$

As discussed earlier, the formula for depth incorporates its inversely proportional relation to the disparity as well as the directly proportional relation to the baseline. Focal length and baseline are stereo camera constants that are obtained from the stereo calibration.

In Eq. (5) and **Figure 8**, below is the legend for the symbols used:

z – *Depth of the object point from the stereo unit in meters*

f – *Effective focal length of the stereo unit in pixels*

B – *Baseline distance between the two camera units in meters*

d – *Pixel disparity*

O – *Object point in the world frame*

C_1, C_2 – *Camera* 1 *and Camera* 2

I_1, I_2 – *Corresponding image from Camera* 1 *and Camera* 2

Eq. (5) can be better understood using the following simple depth proof:

As we can see from the diagram in **Figure 8**, the camera plane is parallel to the image plane:

$$\therefore \Delta OPC_1 \sim \Delta C_1MI_1 \tag{6}$$

$$and\ \Delta OPC_2 \sim \Delta C_2NI_2 \tag{7}$$

from Eq. (6), we know that

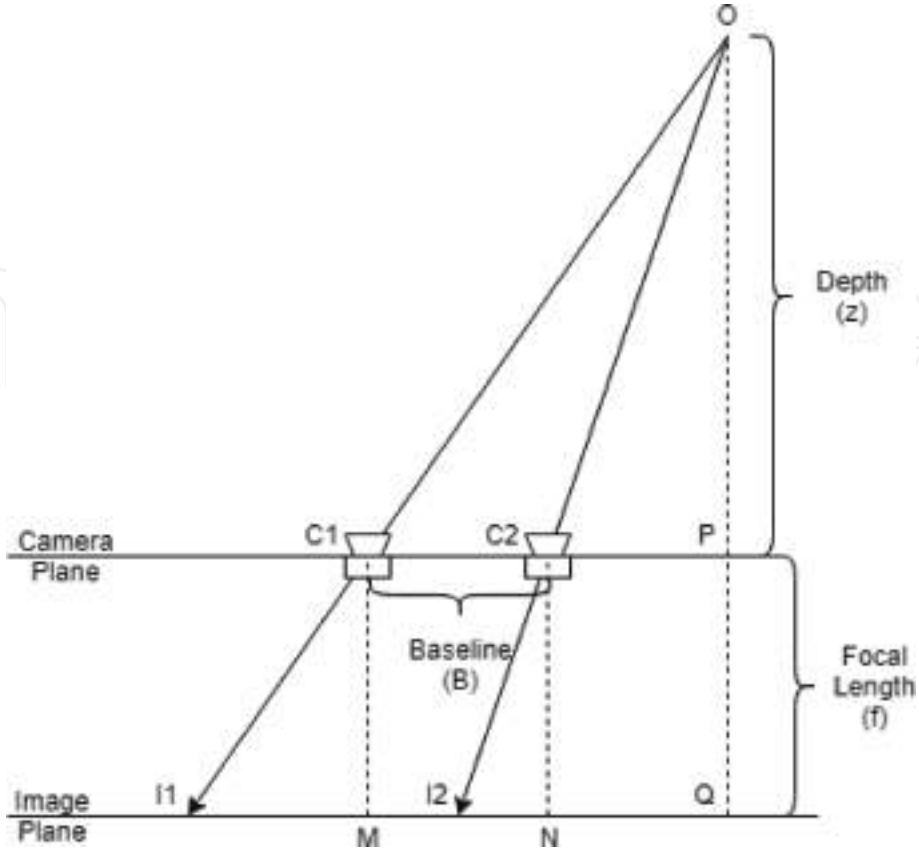

Figure 8.
The stereo vision geometry.

$$\frac{z}{f} = \frac{C_1P}{I_1M} \tag{8}$$

and from Eq. (7),

$$\frac{z}{f} = \frac{C_2P}{I_2N} \tag{9}$$

Since baseline is the distance between the two cameras in a stereo unit,

$$\therefore B = C_1P - C_2P \tag{10}$$

from Eqs. (8) and (9), we can rewrite Eq. (10) as

$$B = \frac{z}{f} \times (I_1M - I_2N) \tag{11}$$

From the definition, it is evident that $(I_1M - I_2N)$ is nothing but disparity. Therefore, from Eq. (11) we arrive at the original equation of depth, i.e.,

$$z = \frac{f \times B}{d} \tag{12}$$

The proof for the above equation implies that the depth from the stereo unit is only dependent on the stereo focal length, the baseline length, and the disparity between the corresponding pixels in the image pair. For this exact reason, depth estimation using stereo is more robust and better suited. It is independent of any orientation or poses of the stereo unit w.r.t. the scene in the 3D world coordinates. The depth of an object shown by the stereo unit is not affected by any movement of the unit at the same distance from the object. This characteristic does not hold when the depth is being estimated using the monocular camera using methods other than deep learning. Calculating depth using a monocular camera is highly dependent on the exact pose of the unit w.r.t. the scene in the 3D world coordinates. The pose constants that work for depth estimation in one pose of the camera are most certainly guaranteed not to work when the camera is repositioned to some other pose at the same depth from the object.

2.4.2 Deep learning method

All the methods discussed above are ultimately static methods that work on the base ground of traditional computer vision. Deep learning, gaining popularity in the recent years, has shown promising results in almost all fields that it has been applied to. Sticking to the trend, the researchers and experts used it to estimate depths and disparity as well, and as expected, the results are encouraging enough for all enthusiasts for further motivated research.

Exploiting the limits of deep learning, it has also shown motivating results for depth on monocular images as well. This idea is particularly interesting because in this approach the learning model can be trained without the need for disparity map or depth information [15–17]. The output from one camera is treated as the ground truth for the other camera's input image. The logic is, to give as output, a disparity map which when used to shift the pixels of the first camera image gives us an image that is equivalent to the second camera image. This disparity output is then used to compute depth using the simple depth formula (**Figures 9–11**).

Figure 9.
Disparity output using deep learning methods [15].

3. Proposed approach

Many researchers have been working and brainstorming on the issue of sparse disparity maps. A lot of the real-time non-deep learning disparity methods fail to generate dense disparity maps. And deep learning methods have been lagging behind in this case because they are slower and lack accuracy. The performance comparison mainly assumes embedded hardware and not high-end compute machines. However, this approach aims to eradicate their need for certain use cases. The focus of this approach is to quest on the fact if sparse disparity map is really an issue. Sticking to the motivation of this chapter, sparse disparity maps are more than enough to give meaningful information if combined with other smart perception techniques. For example, if methods like object-detection [18, 19] or semantic-segmentation [20–23] give an output of identified object pixels in the image, sparse stereo output can be used to estimate the depth of the entire identified pixel group with the help of only a few major feature pixels. As a researcher, it is essential to acknowledge the fact that "one solution fits all" is not always the best approach for performance-centric problems. Moreover, because the dense disparity maps take up much computational power, we drop the aim of doing so.

If the obtained output is a sparse disparity, high credibility is a nonnegotiable requirement. While many hacks are used to filter the nonsensical disparity values, they are ultimately heuristics and not smart techniques that have any understanding of the scene. There is always the possibility that sometimes the good disparity values are filtered out. Since the current approach works on mostly the most critical feature points, the credibility for their disparity is the maximum in the selected region of pixels. More so, multiple distance functions reaffirm the calculated disparity. Higher confidence in the output disparity can be obtained by making use of higher level structural information of the objects in the scene. The structural buildup of the scene is lost information that is mostly neglected in the non-deep learning approaches. The approach proposed in this chapter intends to use this information to our advantage.

3.1 Algorithm

The conventional techniques start with a window scan of the entire image and look for the best disparity values. Mostly a post-processing step follows which

deletes the spurious disparity values from the final output. In this proposed approach, a post-processing step is not required, and the correspondence algorithm runs for a much smaller number of pixels compared to the entire image. Following the above statement, this approach starts with finding the most prominent features in the two input images. It is critical because it ensures three things—the selection of discrete pixels that ensure high disparity confidence, the removal of any post-processing step, and a drastic reduction in the input size for disparity estimation. The first point takes care of high credibility, whereas the other two points ensure a significant performance boost.

Once we have the features of the two input images, we use a combination of multiple techniques. Since it is a conventional image processing technique with the requirement of not being computationally heavy, there is only so much information that each method can carry. A combination of the same has the potential to overcome this flaw.

The first technique, i.e., the **feature matching technique**, is the most dynamic part of the algorithm. It requires modification for every different type of feature selection. As the feature of interest for this chapter is line segments, the discussion restricts to the same. Here the features not only are matched by pixel values but depend on the feature properties as well. For example, the slope is an essential property for a line. It helps to identify the similarity in the structure of the compared scene. However, this has the naïve loophole that it can match with any similar-looking line. Hence, it is not possible to entirely rely on this distance estimation technique.

The second technique is the typical **window matching technique** (see **Figure 12**). However, the difference is the size and shape of the window decided for each individual feature. The line segment detected in that area governs the shape and size of each window. The window must cover entirely the smaller of the two lines (detected lines in the two input images). For a little context, a few pixels pad the feature line within the window (see **Figure 12**). This one difference from the typical window matching makes much difference because each feature has a unique size which indicates that each window captures a significant image feature in its entirety and not just clueless parts of it. Irrespective of the added advantage, this method still has all the flaws of the box matching technique, the significant difference being the difference in the illumination of the two camera views. This difference can lead to erroneous disparity values. The next distance estimation technique handles this flaw.

Figure 10.
Left image overlapped with the detected feature lines [7].

Figure 11.
Right image overlapped with the detected feature lines [7].

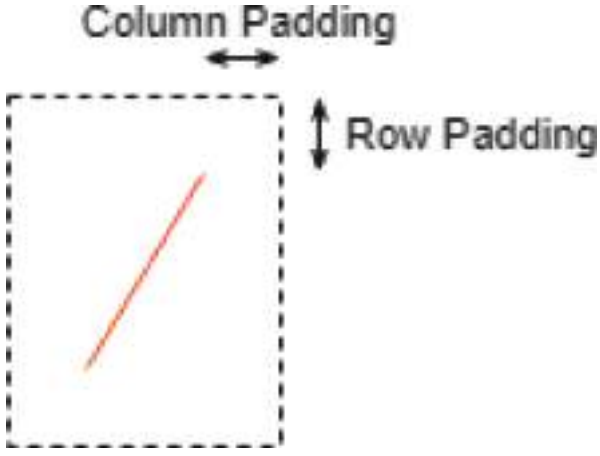

Figure 12.
Window matching technique: the window covers the entire feature segment along with some pixel padding.

The third technique is the **census feature matching technique** (see **Figure 13**) which makes the pixel matching intensity independent. It captures the relationship between the intensity values in a selected neighborhood and does not rely on exact intensity values. Although this step makes the previous distance estimation seem redundant, it helps in the cases where the relation between pixels intensities is the same for multiple positions of the search space. On top of that, unlike the window matching technique, the census features require a single point of interest for each window and hence cannot have a non-square window size for the image features. **Figure 13** shows the use of census features for this approach.

While the above metrics help find an accurate match of the corresponding pixels, it is necessary to identify the pixels that do not have a corresponding matching pixel. It is mainly the case with occluded pixels and is a significant factor to take care of to ensure high accuracy. The steps that ensure this necessity are feature matching and disparity aggregation (discussed later) steps. In the feature matching step, a corresponding match is searched only for features that are fundamentally and structurally the same. Failure to find such candidates leads to dropping the particular feature. After this initial screening, disparity aggregation does the final screening. Here if the disparity values obtained from the different metrics go out of a range, they are rejected. This thresholding can be relied upon because the estimated ranges are in the depth space.

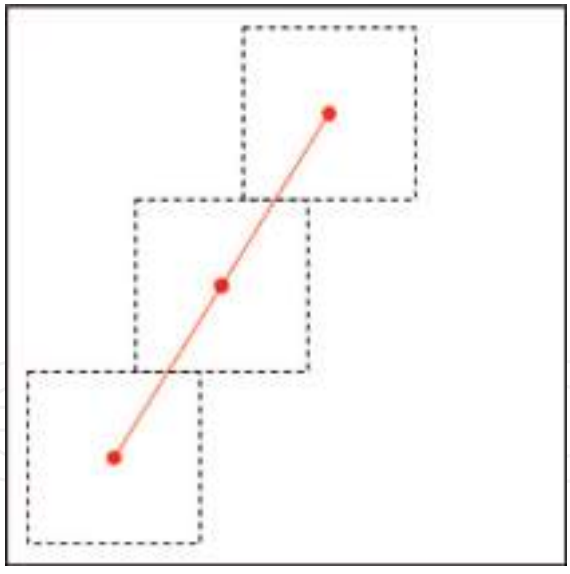

Figure 13.
Census feature matching technique—redline is the feature segment, the red dots are the pixels of interest on the feature segment, the dashed squares are the census feature kernels for the pixels of interest.

Next is the **disparity aggregation** step that combines the disparity values obtained from all of the above metrics. The main characteristic encapsulated here is the fact that this aggregation step can reject outlier disparity values as well. The upper and lower bound of the disparity values can be obtained from Eq. (12). Extending the same we can get the disparity error range in the pixel space (Eq. (13)).

$$Disparity\ Error\ Range = -fB\frac{(x+y)}{(z-x)(z+y)} \tag{13}$$

In Eq. (13), below is the legend for the symbols used:

$z - Depth\ of\ the\ object\ point\ from\ the\ stereo\ unit\ in\ meters$

$f - Effective\ focal\ length\ of\ the\ stereo\ unit\ in\ pixels$

$B - Baseline\ distance\ between\ the\ two\ camera\ units\ in\ meters$

$x - Arbitrary\ margin\ distance\ in\ meters\ taken\ infront\ of\ the\ object$

$y - Arbitrary\ margin\ distance\ in\ meters\ taken\ behind\ the\ object$

3.2 Results

Figure 14 shows the final result of the above approach. The disparity values of the image features are color-coded based on the disparity values. Visually the output looks much inferior to the standard disparity estimation techniques, but the motivation of this chapter has been different since the beginning. This approach is capable of performing better than the typical approaches because in a combined pipeline, i.e., its combination with other smart perception techniques, it's capable of performing much better because it mostly avoids the false disparity values.

3.3 Future work

Although the proposed method is promising in some instances, it will not always perform better for obvious reasons. In case there is a requirement to estimate the disparity of some pixel that does not lie in the feature pool, this algorithm is bound

Figure 14.
Final feature disparity map—red denotes closer pixels, blue denotes farther pixels.

to fail. In such cases, custom image descriptors, where closest feature points can define the pixel of interest, can be used to overcome the above flaw. Getting this right is a challenge because the disparity between the two stereo images makes it a nontrivial problem to select the correct features to describe the pixel of interest. Since the introduced disparity can lead to some difference in the background of the two input images, hence not all features can be used to describe the pixel of interest.

Another critical factor that can help in improving the performance of the output is better identification of the edges of the detected objects. Something that I would like to call "dislocated kernels" might help improve the accuracies. The idea is – not to be restricted by the fact that the pixel of interest needs to lie at the center of the kernel.

All of the above ideas and approaches work behind a single motivation of attaining the maximum credibility of the computed output. Along the same lines, if the current approach can be optimized enough, we might have enough room for the conventional yet effective stereo consistency check. Since even this check is to be performed on the feature elements of the image, the number of input pixels is meager and can lead to very high confidence overall.

4. Conclusion

Throughout this chapter, we talked about the fundamental details with a slight background of the stereo vision system and about how sparse disparity maps can be highly credible. We discussed the primary use cases and applications along with the conceptual working of this system. As discussed earlier, there are a lot of complex challenges to be taken care of when using stereos for any application. The solutions to these challenges are no magic bullet and require some digging in to figure out the solution that works best for the chosen application. Many experts have launched ready-made stereo vision systems with a reasonable amount of accuracy to save researchers from the efforts of setting up a good stereo system themselves. These are fit to be used for almost all personal projects, and some of them are even suitable for extensive projects. Examples of some of these products are ZED Stereo, Microsoft Kinect, Bumblebee, and many more. Multiple solutions have come into existence, to speed up the process of depth estimation. While some of these devices use custom-built hardware for faster computation, others use a variety of cameras, e.g., infrared cameras, to make the process of stereo matching much more comfortable. The approach proposed in this chapter guides toward making use of even the sparse disparity maps with greater confidence.

This chapter was an attempt to cover most of the fundamental concepts that govern the working of the stereo vision systems and give an alternative for fast depth estimation techniques. The intention was to give enthusiastic readers enough information about the topic by the end of the chapter, to make them capable of digging deeper into advances sections of the module.

Acknowledgements

I want to use this space to thank first and foremost my ex-employer, the Hi-Tech Robotic Systemz, and my mentor at the company, Gaurav Singh, for introducing me and giving me enough opportunities in this domain that helped me grow my knowledge in this field.

Next, I would like to thank my friends Karan Sanwal, Nalin Goel, Smriti Singh, Megha Mishra, Subhash Gupta, Shilpa Panwar, and Priyanka Tete for their constant support in reviewing the chapter and providing me with valuable insights for the modification of this content.

Last but not least, I would like to thank my parents and my sister for their motivating supports toward writing this chapter.

Had it not been the constant backing of all these people, I might not be able to imagine writing this chapter. So a heartily thank you to all of them.

Author details

Satyarth Praveen
Master of Engineering in Robotics, University of Maryland, College Park, Maryland, United States of America

*Address all correspondence to: satyarth@terpmail.umd.edu

References

[1] The ABC's of Distances [Internet]. 2018. Available from: http://www.astro.ucla.edu/~wright/distance.html [Accessed: 07-09-2018]

[2] Zhang Z. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000;**22**: 1330-1334

[3] Scaramuzza D, Martinelli A, Siegwart R. A toolbox for easily calibrating omnidirectional cameras. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on 9 October 2006; IEEE; pp. 5695-5701

[4] Camera Calibration Toolbox for Matlab [Internet]. 2015. Available from: http://www.vision.caltech.edu/bouguetj/calib_doc/ [Accessed: 14-10-2015]

[5] Tsai Camera Calibration [Internet]. 2003. Available from: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/ [Accessed: 05-11-2003]

[6] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision. 2002;**47**(1–3):7-42

[7] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The kitti vision benchmark suite. Computer Vision and Pattern Recognition (CVPR). In: 2012 IEEE Conference on 16 June 2012; IEEE; pp. 3354-3361

[8] Silberman N, Hoiem D, Kohli P, Fergus R. Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision. Berlin, Heidelberg: Springer; 2012. pp. 746-760

[9] Scharstein D, Szeliski R. High-accuracy stereo depth maps using structured light. In: Computer Vision and Pattern Recognition, 2003; Proceedings 2003 IEEE Computer Society Conference on 18 June 2003; IEEE; Vol. 1. pp. I-I

[10] Scharstein D, Hirschmüller H, Kitajima Y, Krathwohl G, Nešić N, Wang X, et al. High-resolution stereo datasets with subpixel-accurate ground truth. In: German Conference on Pattern Recognition. Cham: Springer; 2014. pp. 31-42

[11] Sjaardema TA, Smith CS, Birch GC. History and evolution of the Johnson criteria. SANDIA Report, SAND2015-6368. 2015

[12] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: Computer Vision and Pattern Recognition CVPR 2005; IEEE Computer Society Conference on 20 June 2005; IEEE; 2005. Vol. 2. pp. 807-814

[13] Hirschmuller H. Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2008;**30**(2):328-341

[14] Spangenberg R, Langner T, Adfeldt S, Rojas R. Large scale semi-global matching on the CPU. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE. 2014. pp. 195-201

[15] Godard C, Mac Aodha O, Brostow GJ. Unsupervised monocular depth estimation with left-right consistency. CVPR. 2017;**2**(6):7

[16] Zbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research. 2016;**17**(1–32):2

[17] Mayer N, Ilg E, Hausser P, Fischer P, Cremers D, Dosovitskiy A, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 4040-4048

[18] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. pp. 779-788

[19] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: Single Shot Multibox Detector. In: European Conference on Computer Vision. Cham: Springer; 2016. pp. 21-37

[20] Paszke A, Chaurasia A, Kim S, Culurciello E. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147. 2016

[21] Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix [Internet]. 2013. Available from: http://ksimek.github.io/2013/08/13/intrinsic/ [Accessed: 13-08-2013]

[22] Bhatti A. Current Advancements in Stereo Vision. Rijeka: InTech; 2012. https://scholar.google.com/scholar?oi=gsb95&q=current%20advances%20in%20stereo%20vision%20rijeka&lookup=0&hl=en

[23] Hartley R, Zisserman A. Multiple View Geometry in Computer Vision. Cambridge University Press; 2003. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=multiple+view+geometry+in+computer+vision&btnG=&oq=multiple+view+