Stereo Vision with ESP32-CAM: Depth Estimation for Autonomous Driving Applications

Kunal Chaugule, Ragini Sharma, Sakshee Daundkar, and Shweta Chavan
Department of CSE-Data Science, Saraswati College of Engineering,
Navi Mumbai, 410210, India,
kunal.chaugule22@comp.sce.edu.in, ragini.sharma@it.sce.edu.in, sakshee.daundkar22@comp.sce.edu.in,
shweta.chavan22@comp.sce.edu.in

***Abstract* - Depth estimation is one of the critical tasks in computer vision as it allows machines to estimate the depth of the scene and the objects in it. This capability is especially important in self-driving cars since inducing proper and safe behavior of the car depends largely on the precise orientation in the space. Depth estimation enables the sensing of the distance of neighboring objects, direction planning, and accurate operations such as parking or changing lanes. In order to develop a practical low-cost system for this purpose, this paper focuses on depth estimation using stereo vision techniques. Stereo vision which mimics the human ability to see at a distance refers to the computation of the disparity of images captured using two cameras arranged at a small distance apart. In order to calculate disparity maps and then convert them into depth information, a mathematical approach is adopted that can be employed in driving situations such as object identification, path determination, and no collision zone identification. For this purpose, the proposed system is implemented in a controlled environment to measure its performance. Despite these advantages, issues like low resolution or high processing latency are handled, with an eye toward practical applications of the technique moving forward. This is because, as this research demonstrates, the ability to estimate depth is the key prerequisite on the way toward creating more effective and safer autonomous driving systems.**

Index Terms - 3D perception, Depth estimation, ESP32-CAM, Stereo vision.

I . Introduction

Depth estimation has always been an important task in the computer vision field as it helps a machine to understand the 3D structure of the surrounding world. Depth sense is among the most critical aspects of robotics and self-driving cars, augmented reality, and other applications where accurate depth sensing is vital for navigation, object detection, decision-making, and other related activities. Previous methods of depth map estimation required using expensive devices including LiDAR or sophisticated stereo camera approaches which could provide accurate results but cost a lot of money and are unavailable in lower-end university projects [1], [2].

Stereo vision is based on the biological structure of human binocular perception and is cheaper than, for example, the use of sensors for depth estimation. Stereo vision systems derive depth by comparing differences between two images obtained from slightly different positions of view. The mathematical relationship between the disparity and depth is expressed as:

$$Z = \frac{b \cdot f}{d} \qquad (1)$$

where Z represents depth, b is the baseline distance between the cameras, $f$is the focal length, and d is the disparity between corresponding points in the two images. While stereo vision is computationally efficient compared to LiDAR, it requires precise camera calibration, robust rectification algorithms, and accurate disparity computation to achieve reliable depth estimates [3].

This paper discusses the practicality of depth estimation using stereo vision specifically, in the ways of the necessary hardware configuration, mathematical formulation, and software procedure. To explain disparity computation and depth triangulation, the system employs low-cost stereo camera modules. Intrinsic and extrinsic calibration parameters are employed to rectify lens distortions and Camera geometry to achieve correct rectification and disparities. Moreover, SGM algorithms are used to generate dense disparity maps with consideration of both accuracy and time [4].

Although the methodology aims at cost and coverage, the more general goal is the construction of technical support for stereo vision systems for application to autonomous vehicles. Depth estimation is an essential requirement for the subsequent tasks, including obstacle detection, path planning, and collision avoidance, where accurate spatial perception in real time is mandatory. In so doing, this paper seeks to make a contribution towards the development of depth estimation systems that are affordable to implement while at the same time covering factors that might include such things as: low resolution, synchronization issues or delays, and environmental constraints.

II. Literature Review

There is a rich literature on depth estimation in computer vision since this capability is a pivotal element that makes it possible for machines to perceive and make decisions in a real space. Several methods have been introduced which are categorized into basic sensing and intelligent systems, and they comprise several merits and demerits.

I. Sensor-Based Depth Estimation

Depth estimation has predominantly relied on high-end sensors including LiDAR and radar to create accurate 3D point clouds. Whereas LiDAR is useful for the identification of objects on the road and for creating a map of the environment, radar excels in meteorological conditions. However, these sensors involve one of the most serious problems, namely, high prices and difficulties in integration, thus they cannot be effectively used in low-cost or educational projects [5], [6].

II. Stereo Vision Systems

Stereo vision is a computer vision technique, which mimicsthe human visual system through the use of two cameras to estimate depth. Using at least two image frames obtained from slightly different perspectives, stereo vision systems determine disparity hence depth. However, the design of a stereo vision system has some problems such as how to deal with low-texture regions and how to achieve good camera calibration while ensuring system cost-efficiency. To improve the reliability in computing disparity Semi-Global Matching (SGM) algorithms have been designed to ensure the best compromise between the accuracy in disparity computing and the time required for the operations [3].

III. Depth Estimation in Autonomous Driving

In particular, depth estimation is an important aspect of autonomous driving, which plays a crucial role in the safety and accordingly in driving. It allows the vehicle to identify the objects in its vicinity, prevent accidents and determine safe navigation routes, or space to park or change lanes. The KITTI dataset has been used for benchmarking stereo vision algorithms for real-world driving environments and has provided a useful understanding of the system's real-life performance [4]. Through using depth estimation, self-driving cars can acquire elaborate spatial evaluation and improved decision-making.

IV. Low-Cost Depth Estimation Systems

Recent advancements in in-depth estimation show that new solutions that are significantly cheaper in terms of hardware have been developed. Currently, there are affordable boards such as the ESP32-CAM which can be effectively used for illustrating the concepts of stereo vision. Despite having been developed for IoT use cases, the ESP32-CAM has been successfully redeployed for computer vision tasks and can be certainly used in educational and prototyping scenarios. Although such systems can be inexpensive, there are limitations on such aspects as the resolution and the power of the final processing, thus making them only reasonable for small-scale uses and pilot testing [4].

III. Methodology

I. Column Format Instructions:

This basic hardware configuration involves conducting two ESP32-CAM boards on a stiff structure with a determinate baseline length (*b*) of 10 cm. It also enables the two cameras to capture almost similar but slightly dissimilar views of the scene as required for stereo vision. Supplied independently, every ESP32-CAM board receives power through a module TTL-to-USB conversion that also serves as the transfer of information to a processing station, be it Raspberry Pi or PC. As for the synchronization of both cameras, GPIO pins are used and set to fire the cameras at the same time. The cameras can also take pictures with a picture resolution of 640×480 pixels. The layout of the circuitry of the ESP32-CAM stereo vision system is described in the following figure – Figure 1.

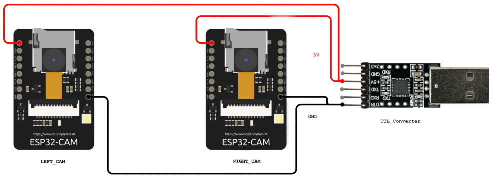


Figure I
Circuit setup for the ESP32-CAM stereo vision system.

To keep the image capture accurate, special emphasis is put on the fact that the ESP32-CAM modules remain in the correct position on the frame so as not to get out of alignment. Besides, a constant power supply is utilized in the system to avoid variations of voltage that may interfere with data communication or synchronization.

II. Camera Calibration

In order to compute disparity and depth, intrinsic and extrinsic parameters are first calibrated for the ESP32-CAM modules to eliminate lens distortions. Camera calibration is conducted with the approach used by Bouguet [5] – the chessboard pattern method. The calibration process includes using several images taken from a number of directions of a standard checkerboard by which to compute the intrinsic matrix

The necessary parameters are the relative maximum permissible exposure limit (*K*) and the distortion coefficients. The intrinsic parameters involve the focal length (*f*) and principal point and the extrinsic parameters identified by (*R*) and (*T*) are the rotation and translation of the camera with respect to the world frame.

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

where:

- s is a scaling factor,
- (X, Y, Z) represents the 3D point in the world,
- (x,y) is its corresponding projection on the image plane,
- K is the intrinsic camera matrix,
- R and T are the rotation and translation matrices.

Calibration ensures that the images from the two cameras are geometrically aligned, which is critical for accurate disparity computation [5], [6]. The OpenCV library's calibrateCamera() function is used to perform this calibration process programmatically [6].

III. Image Rectification and Disparity Computation

After camera calibration, match images collected by the cameras are rectified to make sure epipolar lines of correspondence in the left and right views are aligned. By rectifying an image, comparing point disparities becomes easier as points will be aligned along the horizontal line. To do this OpenCV also provides this feature in the stereoRectify() function which rectifies the images with the help of calibration parameters [7].

After rectification, A disparity map is computed using the SGM algorithm which is a good compromise between quality and computation time [8]. The disparity (d) at each pixel is calculated as:

$$d = x_L - x_R \quad (3)$$

where x_L and x_R are the horizontal coordinates of the corresponding points in the left and right images, respectively.

IV. Depth Calculation

The disparity map is used to compute the depth (Z) of each point in the scene. The relationship between disparity and depth is given by:

$$Z = \frac{b \cdot f}{d} \quad (4)$$

where:

- b is the baseline distance (10 cm),
- f is the focal length, and
- d is the disparity.

This formula demonstrates the inverse proportionality between disparity and depth, meaning closer objects yield larger disparities [9]. Figure 2 illustrates the multi-view geometry used in in-depth estimation.

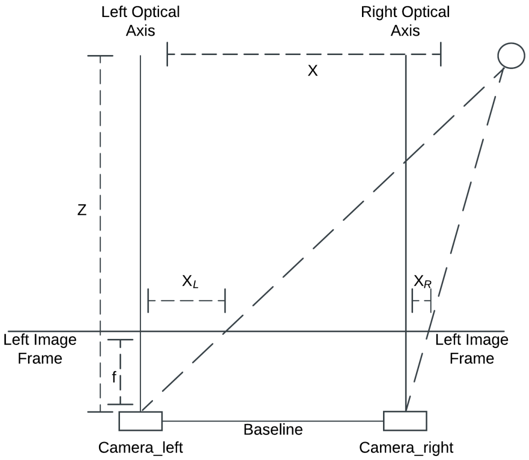


FIGURE II

MULTI-VIEW GEOMETRY FOR STEREO VISION-BASED DEPTH ESTIMATION

V. Software Framework

On ESP32-CAM modules, the Arduino IDE was used for coding in order to allow image capturing and basic preprocessing. After acquisition, the digitized image was then sent to the central processing unit for further processing. The implemented software pipeline included the following stages:

- **Preprocessing:** The obtained pictures were further converted into black and white to reduce the complexity of calculations and time consumption.
- **Rectification and Disparity Calculation:** Stereo rectified image pairs were used in the calculation of a disparity map using the semi-global block matching (SGBM) function of OpenCV [8].
- **Depth Mapping:** The density values were then estimated from this disparity map using the mathematical expression described above.

Image rectification and disparity computation were implemented in Python using OpenCV libraries as were the depth estimations and these were implemented in such a way that they would be processed in the central processing unit as these were computationally intensive activities [6], [8].

VI. Evaluation Metrics

The system is evaluated in a controlled indoor environment, with objects placed at known distances to measure accuracy. The depth accuracy is assessed by comparing the estimated depths to ground truth values. The root mean square error (RMSE) is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (Z_{estimated} - Z_{ground\ truth})^2} \quad (5)$$

where N is the number of points. Additional metrics include processing latency and the robustness of the disparity map in low-texture regions [10].

Although ESP32-CAM made it possible to design an inexpensive stereo-vision system, the low resolution affects disparity estimation, especially in the far field. The work presented in this paper will be extended in the future, where higher-resolution cameras and algorithms tailored for the requirements of embedded systems will be investigated for real-time depth estimation.

VI. Depth Estimation in Real-Time Autonomous Systems

In self-driving cars, depth estimation is a critically important part of creating a spatial perception in real-time. Depth maps are generated from stereo image pairs by employing current methodologies like deep learning-based stereo matching and parallel block matching algorithms when the system implementation factor is considered [11]. These depth maps will allow the perception system to detect obstacles, partition free space, and adapt the calculated path in real time to avoid collisions.

The updates on stereo vision are the sensor fusion of both vision and LiDAR and radar for an enhanced depth estimation insight in low texture environments, inadequate illumination, or harsh meteorological conditions. This makes the approach highly effective across multiple driving scenarios because of the strengths of each type of sensor [12]. Real-time processing is possible by utilizing hardware accelerators like NVIDIA's Jetson platforms or FPGA-based systems, that process disparity with low latency at frame rates beyond 30fps [13].

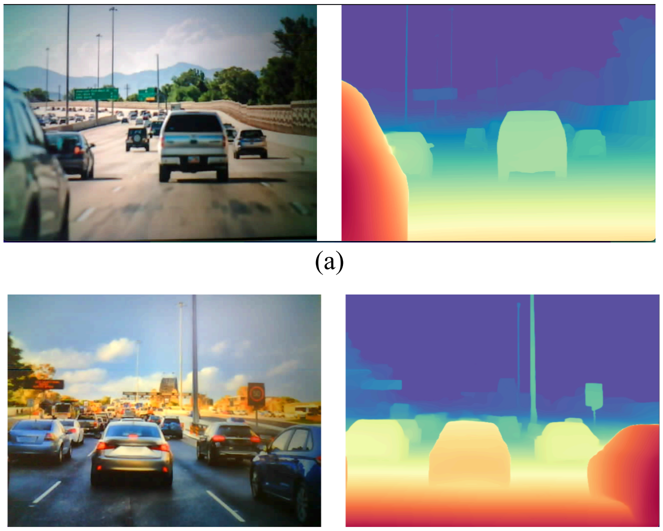
Getting back accurate depth is crucial, and camera calibration remains tight to achieve that. There is a typical method used by today's calibration frameworks, including OpenCV and COLMAP, where researchers observe a rectified model: intrinsic and extrinsic parameters are aligned and have an immediate impact on stereo rectification and disparity map [4]. Moreover, new approaches to self-supervised learning have enhanced the quality of depth prediction and made it possible to have a fairly low dependency on ground truth [5]. Getting back accurate depth is crucial, and camera calibration remains tight to achieve that. There is a typical method used by today's calibration frameworks, including OpenCV and COLMAP, where researchers observe a rectified model: intrinsic and extrinsic parameters are aligned and have an immediate impact on stereo rectification and disparity map [14]. Moreover, new approaches to self-supervised learning have enhanced the quality of depth prediction and made it possible to have a fairly low dependency on ground truth [15].

IV. Results

The proposed stereo vision-based depth estimation system was evaluated under controlled conditions to measure its accuracy, processing efficiency, and usability. The results are summarized below.

I. Depth Accuracy

The system achieved reasonable accuracy for depth estimation within a range of 0.5 to 3 meters. For objects within 2 meters, the mean error was ±5%, which aligns with the performance expected from low-resolution hardware like the ESP32-CAM. However, the accuracy decreased significantly for objects farther than 3 meters, primarily due to the limited resolution and noise in the disparity computation.



(a)

(b)

Figure III

Depth maps generated using stereo vision with ESP32-CAM modules illustrate object segmentation and spatial gradients in traffic scenes.

II. Processing Efficiency

The average processing latency was recorded at 120 milliseconds per frame, primarily driven by the disparity computation using the Semi-Global Matching (SGM) algorithm. While sufficient for low-speed applications and educational purposes, this latency may pose challenges for high-speed autonomous systems. Optimizing the algorithm or employing hardware acceleration could reduce this overhead.

II. Limitations

The system demonstrated sensitivity to disparity matching challenges in low-texture and poorly illuminated areas, which impacted its overall robustness. Additionally, the use of ESP32-CAM modules, with a maximum resolution of 640x480 pixels, imposed limitations on the level of detail captured in depth maps, particularly for objects located at greater distances from the cameras. Minor synchronization issues between the two cameras were observed during testing; however, these did not significantly affect the overall performance of the system under normal operating conditions.

The outcomes indicate that it is possible to provide depth information for reduced costs using the ESP32-CAM-based stereo vision system although the level of accuracy and reliability of this system may be low. Desirable for low-velocity pursuits and teaching, certain aspects of the system's performance in complex conditions and remote objects are feasible enhancements. Subsequent versions of this work will aim to overcome these limitations through enhanced hardware, better synchronization, and efficient algorithms for computing disparities. However, this system provides a starting point for developing feasible and low-cost depth estimation options for practical applications while opening new paths for the development of innovative low-cost computer vision applications for robotics.

V. Conclusion

Consequently, this study shows the possibility of using ESP32-CAM modules to estimate depth with a low-cost stereo vision system. As such, disparity computation and depth mapping offer the system the functional basis for spatial perception within constrained areas. The system provides plausible accuracy and processing rate, but issues such as the efficiency in areas with low texture, low light, and fixed resolution expose the shortcomings of the work. However, they confirm the possibility of implementing affordable reconstruction hardware for depth estimation in educational and prototyping scenarios. The results from this work can form the basis for further work on fine-tuning the algorithm; increasing hardware performance, and perhaps incorporating additional sensing modalities such as LiDAR or radar that may provide greater resilience.

As the abilities of the embedded processors and algorithms continue to improve such systems could become realistic solutions to some levels of robotics, low-speed navigation, and other simple vision applications. This study provides valuable input to the development of ubiquitous computer vision systems that can help bring larger-scale, low-cost solutions to autonomous and robotics applications.

References

[1] Z. Zhang, "*A flexible new technique for camera calibration,*" *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] R. Hartley and Zisserman, A. *Multiple View Geometry in Computer Vision.* Cambridge, U.K.: Cambridge Univ. Press, 2004.

[3] Hirschmüller, H. "*Stereo processing by semi-global matching and mutual information," IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008

[4] Espressif Systems, "*ESP32-CAM Technical Documentation*," [Online]. Available: https://www.espressif.com/. [Accessed: Jan. 12, 2025].

[5] T. Shan, and B. Englot, "*LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain*," in Proc. IEEE/RSJ International Conference Intelligent Robots System (IROS), 2018.

[6] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "*Automotive radars: A review of signal processing techniques," IEEE Signal Process*ing Mag., vol. 34, no. 2, pp. 22–35, 2017.

[7] J.-Y. Bouguet, "*Camera Calibration Toolbox for MATLAB*," [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/. [Accessed: Jan. 12, 2025].

[8] G. Bradski, "*The OpenCV library*," Dr. Dobb's Journal of Software Tools, 2000.

[9] A. Geiger, P. Lenz, and R. Urtasun, "*Are we ready for autonomous driving? The KITTI vision benchmark suite*," in Proc. IEEE Conf. Computer Vision Pattern Recognition (CVPR), 2012.

[10] S. Birchfield and C. Tomasi, "*Depth discontinuities by pixel-to-pixel stereo*," International Journal of Computer Vision, vol. 35, no. 3, pp. 269–293, 1999.

[11] K. Konolige, "*Small vision systems: Hardware and implementation*," in Proc. 8th Int. Symp. Robotics Research, 1997, pp. 111–116

[12] H. A. Mallot, *Computational Vision: Information Processing in Perception and Visual Behavior*. Cambridge, MA, USA: MIT Press, 2000

[13] J.-R. Chang and Y.-S. Chen, "*Pyramid stereo matching network*," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018.

[14] D. Feng et al., "*Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges*," IEEE Trans. Intell. Transp. Syst., 2020

[15] S. Giancola, J. Zarzar, and B. Ghanem, "*Leveraging shape completion for 3D Siamese tracking*," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019.