KLE TECH
KLE Technological University
Creating Value
Leveraging Knowledge

School of
Electronics and Communication Engineering

Minor Project Report

on

Detection of Synchronization between Stereo Frames

By:

1. **James Samuel Mirajkar** USN: 01FE20BEC263
2. **Yashas H** USN: 01FE20BEC285
3. **Fardin Khan** USN: 01FE20BEC172
4. **Venkat P** USN: 01FE20BEC320

Semester: VI, 2022-2023

Under the Guidance of

Dr. Rohini H
Prof. Supriya K

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031
2022-2023

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that project entitled **"Detection of Synchronisation between Stereo Frames"** is a bonafide work carried out by the student team of **"James Samuel Mirajkar (01FE20BEC263), Fardin K (01FE20BEC172), Venkat P (01FE20BEC320), Yashas M H(01FE20B EC285)"**. The project report has been approved as it satisfies the requirements with respect to the minor project work prescribed by the university curriculum for BE (VI Semester) in School of Electronics and Communication Engineering of KLE Technological University for the academic year 2022-2023.

Dr. Rohini H Prof. Supriya K	Dr. Nalini C. Iyer	Dr. Basavaraj S.A
Guide	**Head of School**	**Registrar**

External Viva:

Name of Examiners **Signature with date**

1.

2.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to all the people who have assisted us in the completion of this project. All their contributions are deeply appreciated and acknowledged. We would like to place on record our deep sense of gratitude to Dr. Nalini C. Iyer Head of the Department of School of Electronics and Communication for having the opportunity to extend our skills in the direction of this project. We express our heartfelt gratitude to our guides Dr. Rohini H and Prof. Supriya K, whose valuable insights proved to be vital in contributing to the success of this project.

James S M (01FE20BEC263)

Fardin K (01FE20BEC172)

Venkat K (01FE20BEC320)

Yashas M H (01FE20BEC285)

ABSTRACT

Computer vision applications like object detection, picture matching, and 3D reconstruction all depend on the synchronisation of stereo frames. The goal of this project is to create an algorithm that can detect frame IDs and timestamps to determine the synchronization state of stereo frames while also analyzing spatial information. The timestamp algorithm and the spatial algorithm are the two primary parts of the suggested algorithm. The timestamp algorithm examines the left and right frames' timestamps and frame IDs while taking into consideration a threshold determined using the video's frames-per-second (FPS) rate. The frames are regarded as synchronised if the absolute discrepancies between these values are less than the threshold; otherwise, they are regarded as asynchronous. The Spatial Algorithm is run once the Timestamp Algorithm identifies a possible synchronisation. In order to generate a template and extract spatial information from the left frame, this technique makes use of the Sobel filter. A appropriate correlation approach is then used to match the template to the right, right+1, and right-1 frames. If the chosen frame matches the correct frame, the frames are deemed to be synchronised. The frame with the highest correlation is chosen. On the other hand, the frames are considered asynchronous if the frame with the highest correlation is either the right+1 or right-1 frame. The suggested approach offers accuracy of 90.33% for static dataset and 96.67% for dyanmic dataset for stereo frame synchronisation. The technique also provides information on the duration of asynchrony when frames are not synchronised. A variety of computer vision applications that depend on synchronised stereo frames might benefit greatly from the presented technique. It allows for more reliable object detection, picture matching, and 3D reconstruction by precisely detecting the synchronisation state, which improves visual perception and comprehension in real-world circumstances.

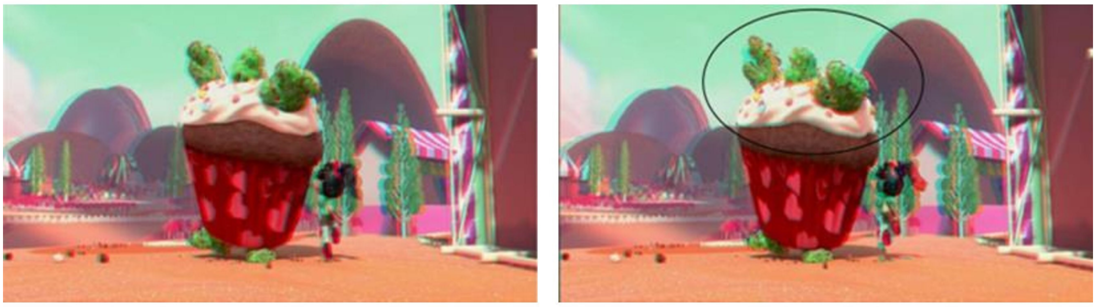
Contents

List of Figures

Chapter 1

Introduction

Stereo frames are a pair of photographs that were captured from slightly different angles, often from two cameras placed at a set distance apart. Due to the tiny perspective variations between the two views, the brain is able to distinguish between depth and distance in these images, allowing for the creation of a three-dimensional (3D) representation of the scene. Several fields, such as computer vision, robotics, and virtual reality[7], use stereo frames. Stereo frames are used in computer vision to carry out operations including depth estimation, object detection, and tracking. Stereo frames are utilised in robotics for object manipulation, obstacle avoidance, and navigation. Stereo frames can be utilised in virtual reality to give the user a more immersive experience because they allow for a more accurate portrayal of the scene. The disparities between the two images are used by stereo vision algorithms to calculate the depth information of each pixel in the scene[6]. Stereo matching, which involves determining which pixels in the left and right images coincide, is the procedure used to create the depth map from stereo frames. To capture stereo frames, It is necessary to calibrate the two cameras, which entails figuring out both their intrinsic and extrinsic properties. While a camera's extrinsic parameters refer to its position and orientation in 3D space, its intrinsic parameters pertain to internal features like focal length, image sensor size, and lens distortion[2]. To provide precise and reliable



(a) (b)

Figure 1.1: Reconstructed stereo frames (a) Sync frame (b) Async frame

depth estimate from stereo frames, calibration is required. Overall, stereo frames are an effective method for capturing and deciphering a scene's three-dimensional structure, and they have several real-world uses in a variety of industries. As shown in figure 1.1 (a) is example of stereo sync image and figure 1.1 (b) is stereo Aync image. The rounded part in async image shows the distortion, which is the problem of async frame because of this whatever application using this stereo image as input will not function properly. first step to solve a problem is to detect the problem. To detect async in stereo frames of video we should also consider the type of stereo frames, there are two main types of frames: static and dynamic. Images taken from two stationary cameras—where neither the cameras nor the scene being photographed are moving—are referred to as static stereo frames[2]. Given that there are no changes in the relative positions of the cameras and the scene between the two frames, this is the most straightforward situation for stereo matching and 3D reconstruction. Although a moving scene is being photographed by a stationary camera, dynamic stereo frames, on the other hand, relate to photos that were taken using movement cameras. As a result, stereo matching and 3D reconstruction are now made more difficult because the scene's relative position to the cameras and other cameras may have changed considerably between the two frames. To take motion into account and determine the precise depth information, further processing is necessary.

1.1 Motivation

- Detecting asynchrony will be very helpful for the mentioned applications because depth estimation, 3D construction, robotics, and autonomous cars all require stereo imaging technology.
- Detecting asynchrony can help in identifying hardware software issues which may be affecting the stereo imaging system, allowing for repeated maintenance or repair.

1.2 Problem statement

Develop a VPI-based algorithm to detect stereo frame synchronization, by utilizing publicly accessible APIs to capture statistics.

1.3 Objectives

- Develop an algorithm to detect synchronization of stereo frames using temporal data.
- Develop an algorithm that examines the frames using spatial data.
- To integrate subsystems for seamless functionality and efficient operation.

1.4 Literature survey

In order to understand the existing works in the context of the proposed ideas, the following papers are discussed.

1.4.1 3D Frame Synchronization Detection Based on Classified Epipolar Geometry Parameters

The suggested approach in this research first extracts epipolar geometry characteristics from stereo pictures, which characterise the interaction between two perspectives of a scene. The fundamental matrix, epipole, and epipolar lines are some of these properties. The authors then categorise the extracted parameters into various kinds according to their properties. For instance, based on its rank or singular values, the basic matrix can be divided into various forms. The method of classifying frames enables the identification of their synchronisation patterns and helps to distinguish between various frame kinds.The authors provide a frame synchronisation detection technique based on the classification findings once the parameters have been categorised. Based on the classification outcomes, the approach assesses if the current frame is synchronised with the preceding one by comparing the retrieved parameters from the current frame with those from the prior frame. On both synthetic and real-world datasets, the authors experimentally test the effectiveness of their suggested strategy. The results demonstrate that even in difficult situations when the stereo pictures have noise or occlusions, the suggested approach can successfully detect 3D frame synchronisation. The proposed method performs better in terms of accuracy and resilience than current cutting-edge methods[5].

1.4.2 Improved Template Matching Based Stereo Vision Sparse 3D Reconstruction Algorithm

The paper suggests a more effective template matching-based method for stereo vision's sparse 3D reconstruction. The method of rebuilding a scene's or object's three-dimensional structure from two or more photos collected from various angles is known as 3D reconstruction from stereo vision. For feature matching between stereo pictures, which is essential for 3D reconstruction, stereo vision frequently employs the template matching technique.In order to extract feature points from stereo pictures, the suggested technique in this paper first uses a template matching algorithm. The template matching algorithm looks for matching areas in the target image using a template image. The addition of an adaptive weighting factor that accounts for the intensity variations between the template and target images helps the authors enhance the template matching method. The weighting factor enhances the matching process' accuracy, particularly in areas with significant intensity variations. Following feature point extraction, the authors suggest a sparse 3D reconstruction approach that calculates the 3D coordinates of the matched feature points using the triangulation method. To increase the precision of the 3D reconstruction, the authors also suggest an outlier removal procedure that eliminates feature points with mismatched coordinates[9].

1.4.3 An Improved Adaptive Window Stereo Matching Algorithm

The author of this journal addresses the issue of low matching accuracy caused by the fact that current adaptive window stereo matching methods do not extract enough features from low-texture regions. It is suggested to use a gradient-based adaptive window stereo matching technique. In order to extract the gradient value from each pixel in the image, the Sobel operator is first utilised. In accordance with the gradient value, each pixel is then split into high, medium, and low texture sections. Then, matching windows are dynamically produced according to arm length and colour threshold, with different arm length thresholds being allocated to various region pixels. By repeatedly creating windows, the pixels nearer the window's centre are given a higher weight.It fixes the issue where the stereo matching algorithm is unable to dynamically choose a matching window. The suggested method outperforms the most recent adaptive window stereo matching algorithm by 5.5%, according to experimental findings on the Middlebury dataset[10].

1.4.4 ORB: An Efficient Alternative to SIFT or SURF

The Oriented FAST and Rotated BRIEF (ORB) feature descriptor, a quick replacement for the Scale-Invariant Feature Transform (SIFT) or Speeded Up Robust Features (SURF) techniques, is introduced in this study. For a variety of computer vision applications, including object detection, picture matching, and 3D reconstruction, the study discusses the necessity for robust and computationally efficient feature descriptors. Despite their effectiveness in these domains, SIFT and SURF can be computationally costly, which makes them less appropriate for real-time applications or devices with limited resources. The article makes a number of significant contributions. It starts by introducing a quick way to give interest points orientations while still guaranteeing rotation invariance. The orientation of interest points are added to the BRIEF descriptor, which further improves rotation invariance. The third section of the study describes a feature matching method based on Hamming distance that enables precise and effective matching. Fourth, the ORB method shows enhanced resilience against a variety of problems, including noise, occlusion, and perspective alterations. Lastly, the results of the experiments demonstrate that ORB achieves equivalent or better performance than SIFT and SURF while drastically decreasing processing needs. Because of these improvements, ORB is a useful substitute for applications that call for precise and effective feature descriptions[11].

1.5 Application in Societal Context

- The detection of synchronization between stereo frames has several potential applications in societal contexts. Stereo vision, which is dependent on synced stereo frames, is essential for precise 3D reconstruction. You may make sure that the acquired frames are perfectly aligned with one another and enable more accurate 3D modelling and reconstruction by looking for synchronisation between stereo frames. This has applications in fields including computer-aided design, augmented reality, and virtual reality.

- Stereo cameras are frequently employed in the field of autonomous cars for object recognition and depth perception. For a vehicle to see and comprehend its environment accurately, the stereo frames must be synchronised. Synchronisation problems may be found and fixed, allowing for the development of safer and more dependable autonomous driving systems.

- To observe and communicate with their surroundings, robots frequently use stereo vision. Robots can sense depth, distance, and object recognition more precisely by spotting synchronisation between stereo frames. This is crucial for activities like object handling, navigation, and human-robot collaboration.

1.6 Project Planning

A clear visual depiction of the project's timetable, dependencies, and progress is provided by gantt charts, as shown in figure C which is helpful for guides and team members. It support project planning, scheduling, and monitoring, enabling improved resource allocation and team member coordination.

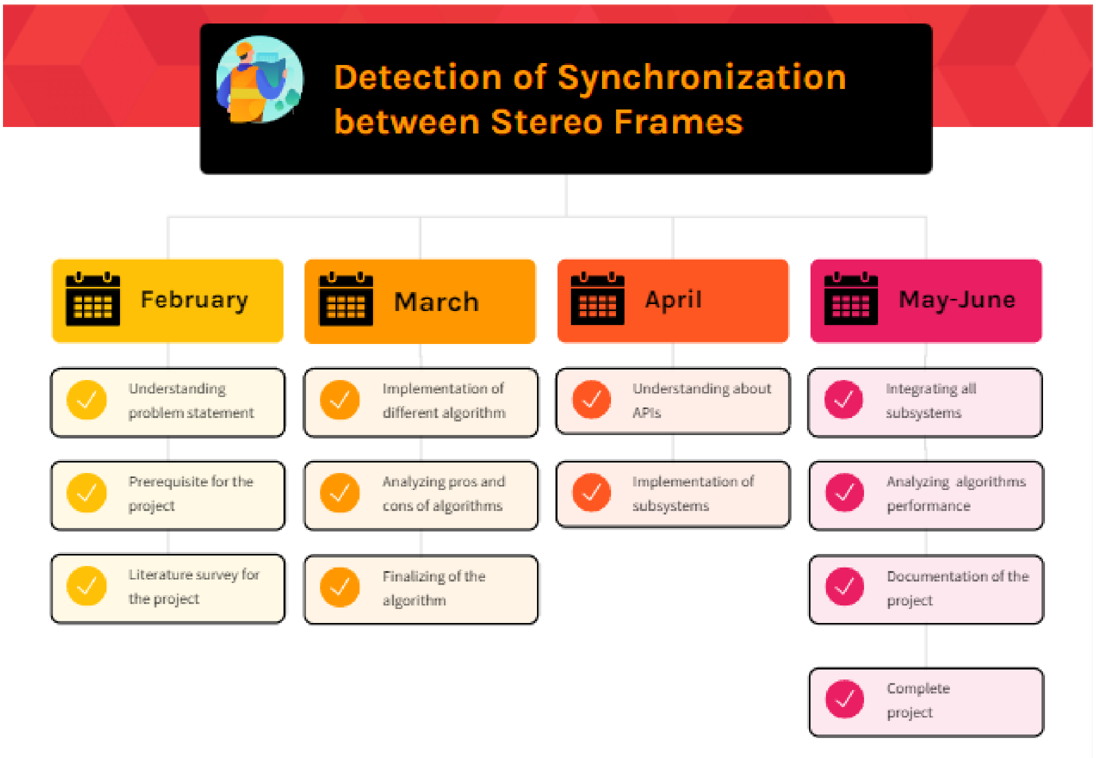


Figure 1.3: Project planning for Detection of Synchronization between Stereo Frames

Chapter 2

System design

This chapter discusses system design, which deals with real-world functionality and design options, as well as the architectural principles that are employed in design.

2.1 Functional block diagram for Timestamp based algorithm

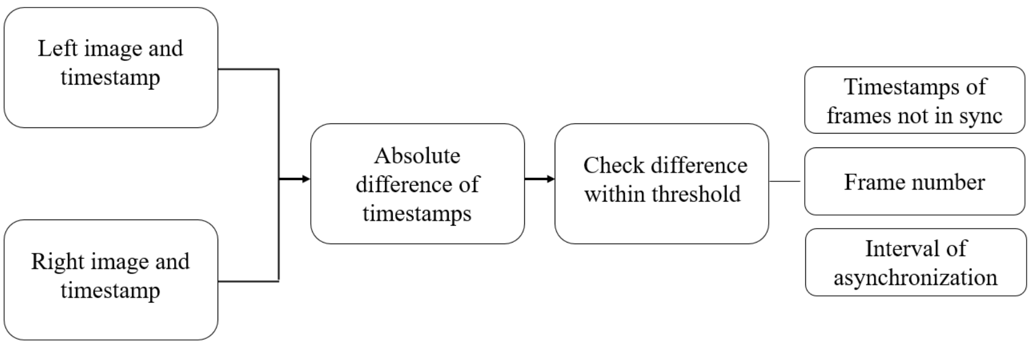


Figure 2.1: Functional block diagram for Timestamp based algorithm

The block diagram checks for synchronisation by analysing the temporal information in incoming frames. It determines the elapsed time between timestamps and compares it to a predetermined threshold. For pertinent pairs, it confirms matching frame IDs. The output displays timing disparities for asynchrony or synchrony indication. The graphic offers a framework for calculating asynchrony and assessing synchronisation, ensuring that pertinent picture pairings are taken into account. It enables accurate evaluation of temporal links and offers information on the level of synchronisation or asynchrony existing. This study supports precise synchronisation, facilitates the comprehension of and resolution of temporal inconsistencies, and raises the overall standard of the image processing system.

2.2 Functional block diagram for Spatial algorithm: Stereo Matching

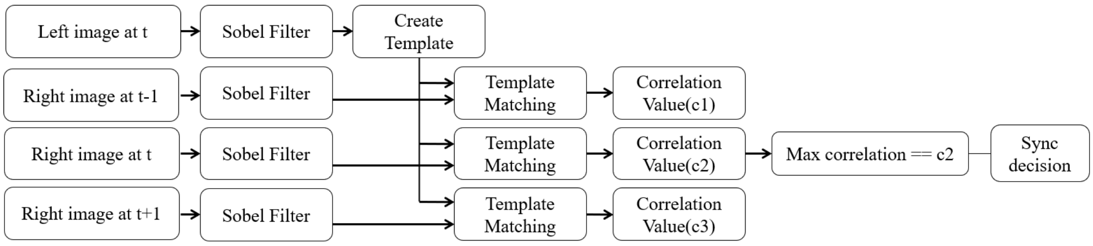


Figure 2.2: Functional Block diagram for Spatial algorithm

The block diagram illustrates a spatial algorithm designed to assess the synchronization between frames. The algorithm takes four inputs: the left frame, the right frame, the right frame + 1, and the right frame - 1 [8][1]. Initially, the left frame serves as a reference to create a template. This template is then subjected to a Sobel filter, which extracts edge and gradient information from the frames. The Sobel filter enhances the features within the images, making them more suitable for comparison[4]. Next, the algorithm calculates the correlation between the template (left frame) and each of the other frames. Correlation measures the similarity between two signals, in this case, the template and the other frames. By quantifying the degree of correlation, the algorithm can determine the level of alignment between the frames. The algorithm examines the correlation results and identifies the maximum correlation value among them. If the maximum correlation is obtained with the right frame, it signifies that the frames are synchronized. Conversely, if the maximum correlation corresponds to a frame other than the right frame, it suggests asynchrony. Based on this analysis, the algorithm generates an output indicating whether the frames are synchronized or asynchronous. This output provides valuable information about the temporal relationship between the frames. In summary, the block diagram presents a systematic approach to detecting frame synchronization. By utilizing a template and correlation analysis, the algorithm enables the assessment of synchronization, aiding in the accurate analysis and processing of temporal image data.

2.3 Alternative design for Spatial algorithm: Adaptive Window Matching

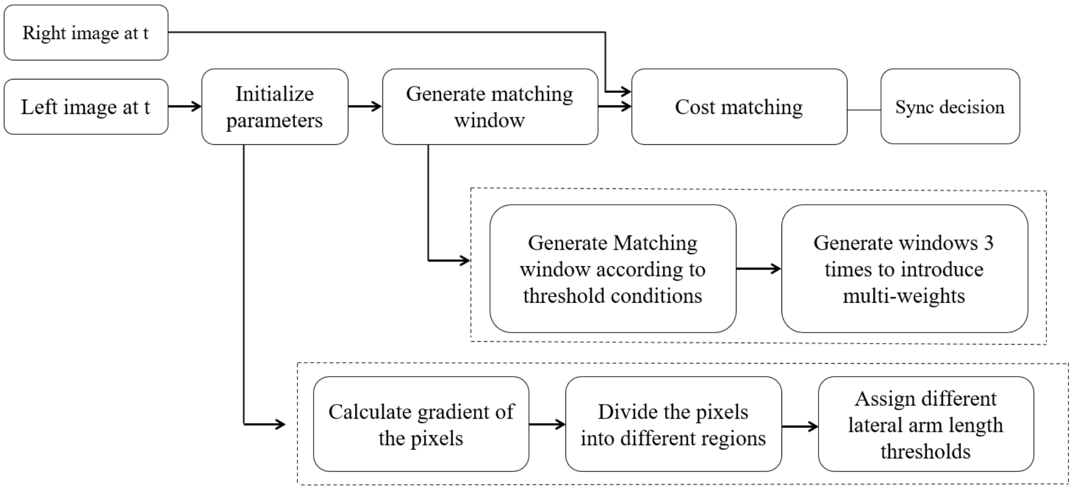


Figure 2.3: Functional Block diagram for spatial algorithm

In order to address the issue of low matching accuracy caused by the current adaptive window stereo matching techniques' inadequate feature extraction in low-texture regions. The gradient-based adaptive window stereo matching method is suggested. First, the gradient value of each pixel in the picture is extracted using the Sobel operator. The gradient value is then used to split each pixel into high, medium, and low texture sections. Then, depending on arm length and colour threshold, matched windows are formed dynamically by assigning various arm length thresholds to various region pixels. Finally, by repeatedly creating windows, the weights assigned to the pixels closer to the window's centre are increased. It fixes the issue where the stereo matching algorithm is unable to dynamically choose a matching window. The suggested technique outperforms the most recent adaptive window stereo matching algorithm by 5.5% [10], according to experimental findings on the Middlebury dataset.

2.4 Proposed design

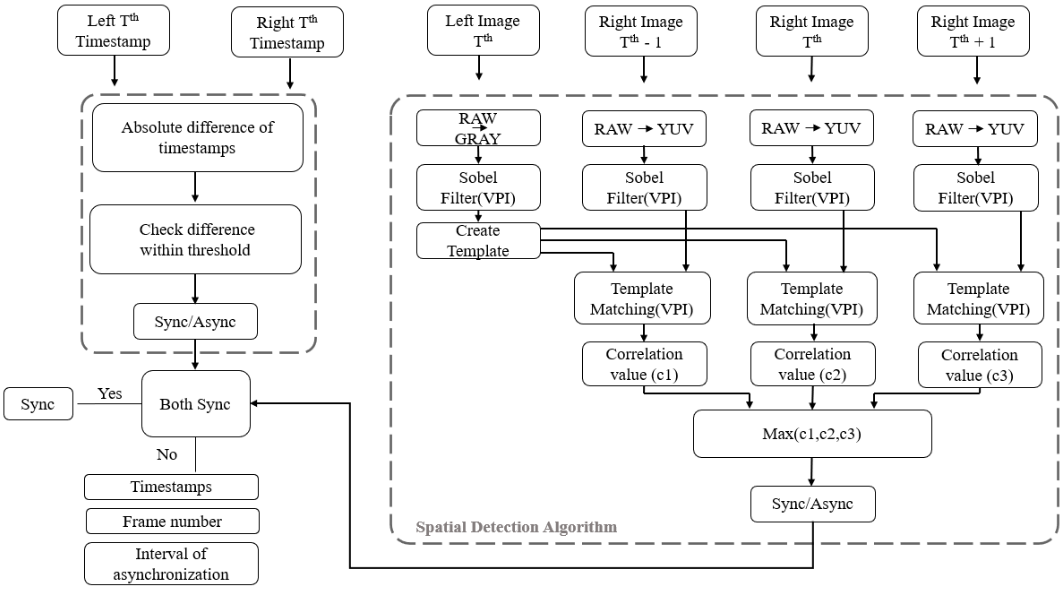


Figure 2.4: System design by integrating timestamp and spatial algorithm

The timestamp and spatial algorithm are combined in this scenario based on the requirement to produce an accurate sync decision in both time and spatial data. The architecture is selected after considering a number of factors, like computing time and hardware-accelerated VPIs that are accessible for the desired solution. Next, the input for our proposed model includes right and left timestamps, along with the following images: left at t, right at t, right at t - 1, and right at t + 1. Then, the timestamps information is passed to timestamp algorithm for time sync decision. To calculate threshold, the frame interval which is equal to 1/FPS and is then divided by 2 in the timestamp method, which compares the absolute difference of timestamps with the threshold using the frame rate. The frames are in sync if the absolute difference is less than the threshold, else they are out of Async. The spatial algorithm uses spatial data for sync detection, and creates a gradient of images by operating on all of the left and right time-shifted images with the Sobel matrix[4]. The left image is then used as a template for stereo matching. The correlation value is generated using the Template Matching VPI, where the maximum value of c1, c2, and c3 indicates a high degree of correlation. For example, the template and right frame t is c1, the template and right frame t - 1 is c2, and the template and right frame t + 1 is c3. If the largest correlated value is c1, it is Sync; otherwise, it is Async, and the stereo matching method returns 1 in this case[3].

If both algorithms produce synchronisation, then the frames are synchronised; if not, then the timestamps, frame number, and interval of asynchronization are provided.

Chapter 3

Implementation details

Implementation details refer to the specific steps, processes, and decisions involved in translating a concept or idea into a functional system, application, or project. These details encompass the technical aspects, methodologies, algorithm and flowchart.

3.1 Specifications and final system architecture

This section gives the specifications and network architectures of the implementations and an overview of the generation of the dataset.

3.1.1 Temporal Sync Decision using Timestamp Algorithm

The three inputs for this algorithm are camera fps, left and right frame timestamps. The algorithm decides whether the frames are in sync or not based on temporal data. The frame interval is first used to compute the threshold using the inverse of the FPS. The threshold is then calculated by dividing the frame interval by half. As depicted in Fig.3.1.

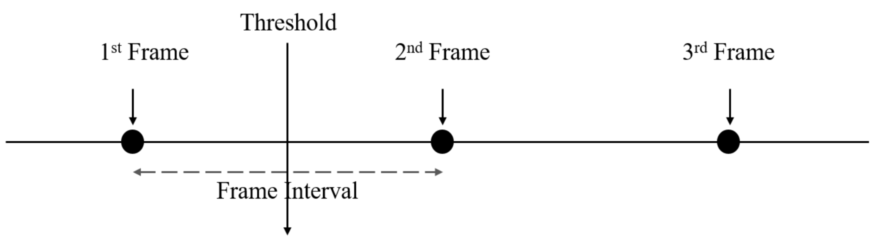


Figure 3.1: Threshold using FPS

The middle of the frame interval serves as the justification for that threshold. The absolute timestamp difference is compared to the threshold, if it is more than the threshold then Async, otherwise Sync.

3.1.2 Spatial Sync Decision using Stereo Matching

This algorithm focuses on the detection of Synchronization using spatial information only. Here this algorithm is uses two VPI, Conv2D VPI for sobel filter which produce the gradient of image and Template Matching VPI is used to produce the correlation value of images.

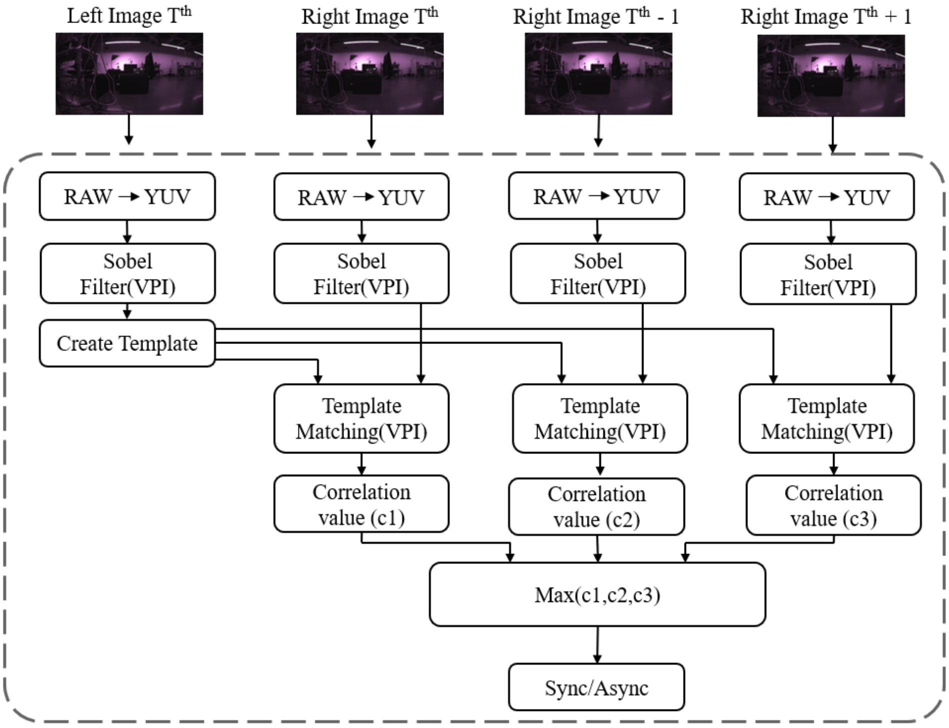


Figure 3.2: Spatial algorithm working flow

As shown in Fig.3.2, algorithm accepts 4 input images left frame, right t-1, right t and right t + 1 frame. The Frame are in RAW format to Gray-scale.

After the image has been converted to grayscale, the sobel filter is used to obtain the x and y gradients' strengths. Edges are created. This is done using Conv2D VPI, which convolves the sobel operator(Kernel) with the given input image. This held true for all four pictures.

The Left image is then used as a template for Stereo Matching, which is done using Template Matching VPI. The Template height is the same as the Given Image, but the Template width is half of the Given Image, considered from the centre of the Image, because this helps in sliding the Template horizontally and obtaining the Correlation Value.

The correlation coefficient between the template and the picture is computed here using Normalised Correlation, which provides a measure of similarity or match between the two.

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'} T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2}} \tag{3.1}$$

$T(x, y)$ represents the normalized template values.
$I(x+x, y+y)$represents the normalized image values.

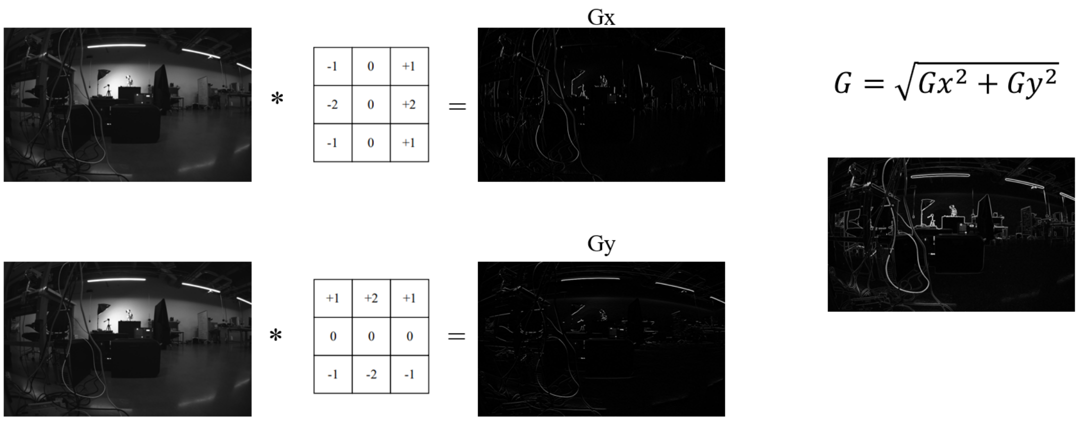


Figure 3.3: Gradient formation using sobel filter

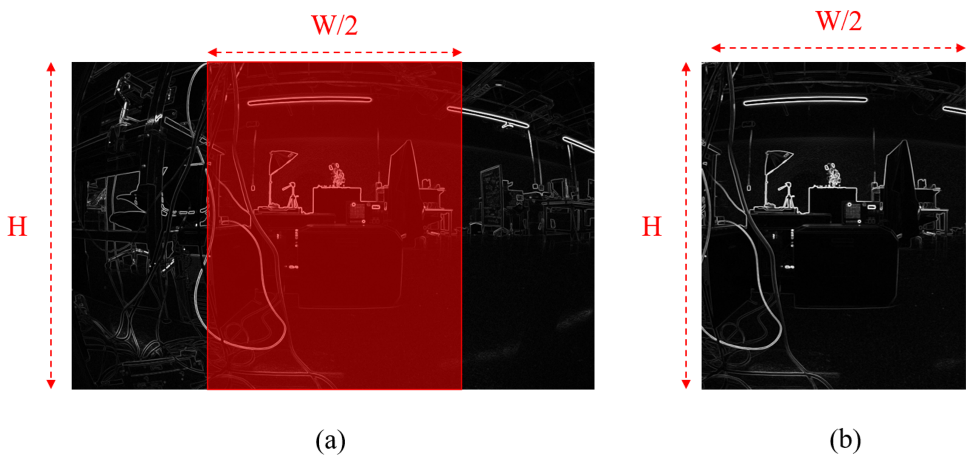


Figure 3.4: Creation of template

The Correlation value is calculated for the template to Right t frame, template to Right t - 1 frame, and template to Right t + 1 frame. Computed values be c1, c2 and c3. If the maximum correlation is with the Template and Right T frame, i.e. max is c1 then, this indicates that the frames are in sync; otherwise, it indicates that the frames are out of sync because the frames are sync with right time shifted frames[12].

3.2 Algorithm

An algorithm is a step-by-step procedure or set of rules used to solve a problem or perform a specific task. It serves as a blueprint for solving computational problems and is an integral part of programming and computer science. Algorithms can range from simple to complex, involving various operations like arithmetic, data manipulation, decision-making, and repetition. They are designed to be efficient, reliable, and scalable, enabling computers to execute tasks accurately and quickly. Algorithms play a crucial role in diverse applications such as data analysis, artificial intelligence, cryptography, and optimization. Here we have designed three algorithm timestamp based, spatial data based and integration of both that is stereo matching algorithm. The depth discussion of these are done in implementation detail section and flowchart section the algorithm flow of all three is portrayed in this section.

Algorithm 1 Stereo Matching Algorithm

Input : Frame $left$, Frame $right$, Frame ID, Timestamp, FPS
Output: Sync/Async status, Time interval (if async)

while $videoisrunning$ **do**
  `// Timestamp Algorithm`
  Output1 = `TimestampAlgorithm(`$left$, $right$, *Frame ID, Timestamp, FPS*`)`
  `// Spatial Algorithm`
  Output2 = $left$, $right$, $right+1$, $right-1$
  if *Output1=Sync and Output2=sync* **then**
    return Sync
  end
  else
    Calculate time interval using Timestamp **return** Async, Time interval
  end
end

Algorithm 2 Timestamp Algorithm

Input : Frame $left$, Frame $right$, Frame ID, Timestamp, FPS
Output: Sync/Async status

Calculate threshold using FPS

`// Perform synchronization check`
if *(Timestamp[left] - Timestamp[right]* $\leq$ threshold*) and (Frame ID[left] == Frame ID[right])* **then**
  return Sync
end
else
  return Async
end

Algorithm 3 Spatial Algorithm

Input : Frame $left$, Frame $right$, Frame $right+1$, Frame $right-1$
Output: Sync/Async status

Apply Sobel filter to $left$ to obtain a template

```
// Perform template matching
maxCorrelation ← 0  for frame ∈ [right, right + 1, right − 1] do
    Apply Sobel filter to frame  Calculate correlation between template and frame
     using a suitable method  if correlation > maxCorrelation then
        maxCorrelation ← correlation
    end
end
// Determine sync/async status
if maxCorrelation is from right then
    return Sync
end
else
    return Async
end
```

The three algorithms: "Timestamp Algorithm," "Spatial Algorithm," and "Stereo Matching Algorithm." These algorithms aim to determine the synchronization status (sync/async) between two video frames captured from stereo cameras.

The "Timestamp Algorithm" compares the timestamps and frame IDs of the left and right frames to check for synchronization. It calculates a threshold based on the frames per second (FPS) and checks if the absolute difference in timestamps and frame IDs is within this threshold. If it is, the algorithm concludes that the frames are in sync; otherwise, it determines them to be async.

The "Spatial Algorithm" performs template matching using the Sobel filter. It creates a template from the left frame and compares it with the right frame and its neighboring frames (right+1 and right-1). The algorithm calculates the correlation between the template and each frame, selecting the maximum correlation. If the maximum correlation is from the right frame, it indicates sync; otherwise, it suggests async.

The "Stereo Matching Algorithm" combines the previous two algorithms. It first applies the "Timestamp Algorithm" and stores its output. Then, it proceeds with the "Spatial Algorithm" and obtains its output. If both algorithms determine sync, it returns sync. Otherwise, it calculates the time interval using the timestamps and returns async along with the time interval.

In summary, the "Timestamp Algorithm" checks for synchronization based on timestamps and frame IDs, while the "Spatial Algorithm" analyzes the correlation between frames using template matching. The "Stereo Matching Algorithm" combines these approaches to determine sync/async status and provide additional information in the case of async frames.

3.3 Stereo Matching algorithm

A flowchart is a visual representation of a process or workflow, utilizing symbols and arrows to depict the sequence of steps and decisions. As shown in Figure 3.5 it helps in understanding the logic, dependencies, and possible outcomes of a process. The flow of both time stamp and spatial algorithm is shown, starting with the timestamp algorithm, we extract timestamps from both the left and right frames. And calculate the absolute difference between these timestamps, which is compared with threshold, where the threshold is parametrised based on FPS. If the difference exceeds this threshold, the frames are considered asynchronous, else sync and the corresponding flag generated. Moving on to the spatial detection algorithm, our focus shifts to the spatial information of frames. Firstly, we convert the t th left frame and the adjacent right frames i.e (t ,t+1, t-1) from RAW to YUV. Next, we apply a Sobel filter to these YUV frames. Then creating a template of the t th left frame, which is then matched against the right frames respectively . Upon template matching, we obtain correction values for each comparison. If frame with the maximum correction corresponds to the t th frames, it is sync Else async , resulting in the generation of a respective flag. Finally, the flags from the both algorithms are combined using Boolean logic to reach a final status. Depending on the status, we say the frames are in sync or async. if async reports frame no, timestamp and interval async.

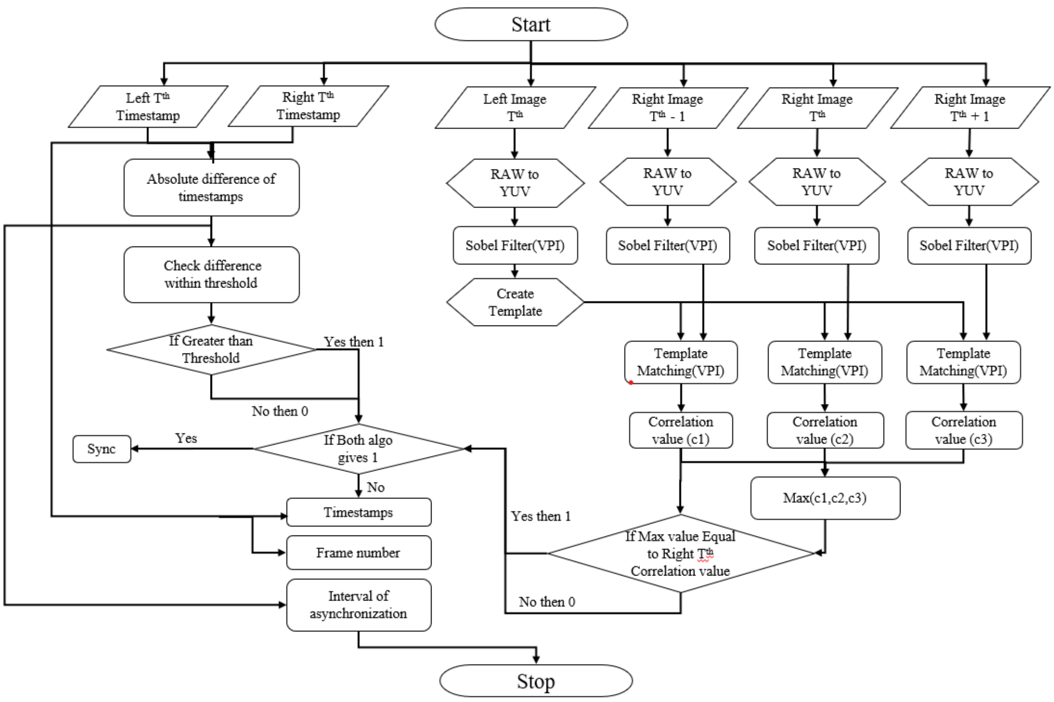


Figure 3.5: Proposed methodology flow

Chapter 4

Results and discussions

This chapter discusses outcomes of the proposed system for detection of synchronization between stereo frames with fusion of both the timestamp based and template matching algorithm. The algorithms are experimented for the static data set which is provided by nvidia and publicly available dynamic data set.

4.1 Experiment 1: Static Dataset

In the experiment aimed at detecting synchronization between stereo frames using a static dataset consisting of 674 frames, significant progress was achieved in accurately determining the level of synchronization between the left and right stereo frames. The objective of the experiment was to evaluate the performance of a synchronization detection algorithm and assess its accuracy in identifying synchronized frames. The figure 4.1 shows result for static frames for synchronous frames

The figure 4.2 shows result for static frames for Asynchronous frames. If any one of the algorithm gives output as asynchronous then the ultimate output is considered as asynchronous as show below

In the given scenario, the dataset consisting of frames was subjected to individual analysis using a synchronization detection algorithm. The primary objective of this algorithm was to determine the synchronization status of each frame within the dataset. The algorithm accomplished this task by comparing the corresponding frames from the left and right stereo channels. It carefully examined the frames to identify any discrepancies or variations between them. Based on the comparison results, the algorithm assigned a synchronization score to each frame. This score quantified the degree of synchronization between the left and right channels for that particular frame. In cases where the frames were found to be asynchronous, indicating a time interval between them, the algorithm further extracted additional information. It captured the frame ID and its corresponding timestamp for accurate identification and analysis purposes. As a result of the algorithm's analysis, the output displayed the frame's asynchronous status, along with its specific frame ID and timestamp. This information was crucial for further investigation and understanding of the dataset.

Figure 4.3 provided a visual representation or illustration of the output, aiding in the comprehension and interpretation of the algorithm's results. Overall, this synchronization detection algorithm played a pivotal role in evaluating the synchronization status of frames within the dataset, providing valuable insights into any temporal disparities and

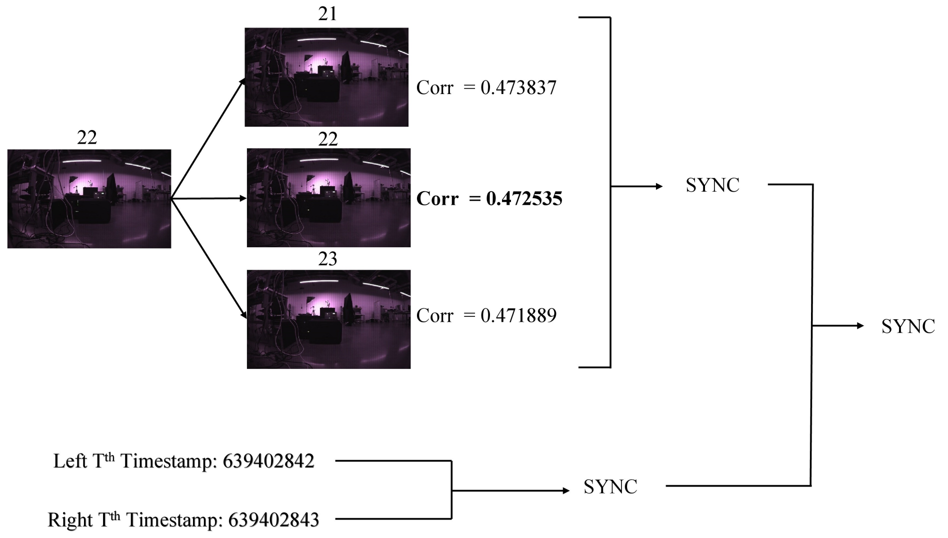


Figure 4.1: Demonstration of synchronisation for static frame

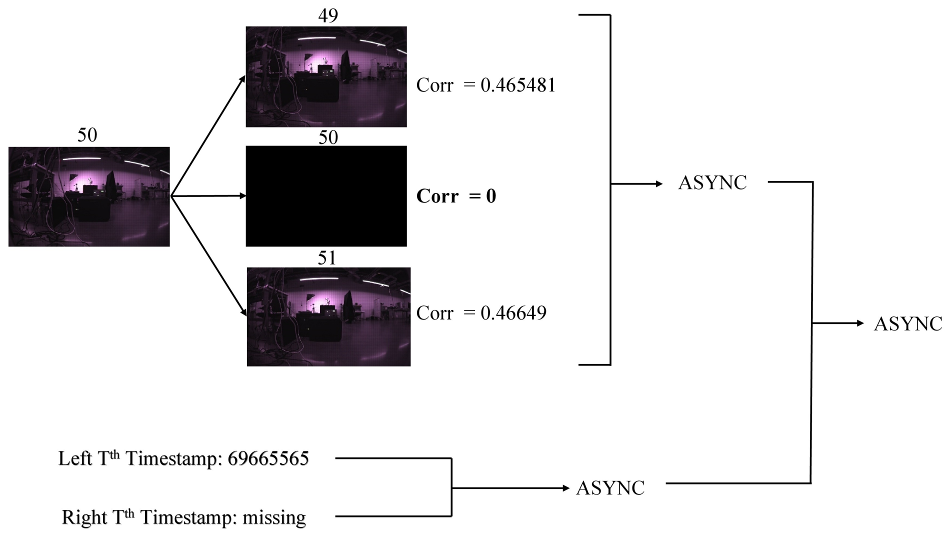


Figure 4.2: Demonstration of Asynchronisation for static frame

facilitating subsequent analysis and decision-making processes.

We use confusion matrix to calculate accuracy of our model as shown in figure 4.4 we get True positive and Ture negative value as 225 and 46 respectively this is tested for 300 frames by which we get the accuracy of 90.3 percentage. It is important to note that the confusion matrix provides valuable insights into the algorithm's performance, allowing

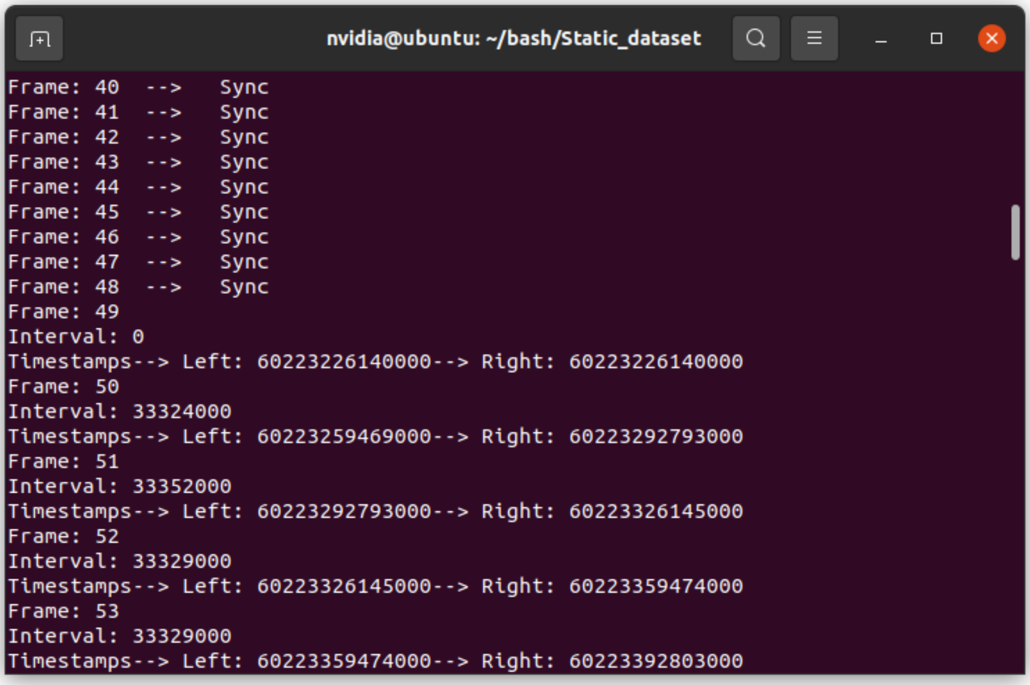


Figure 4.3: Output of static frames on terminal

us to analyze the specific types of errors made during the classification process. Further analysis and interpretation of the confusion matrix can guide future improvements and refinements to enhance the algorithm's accuracy and reliability.

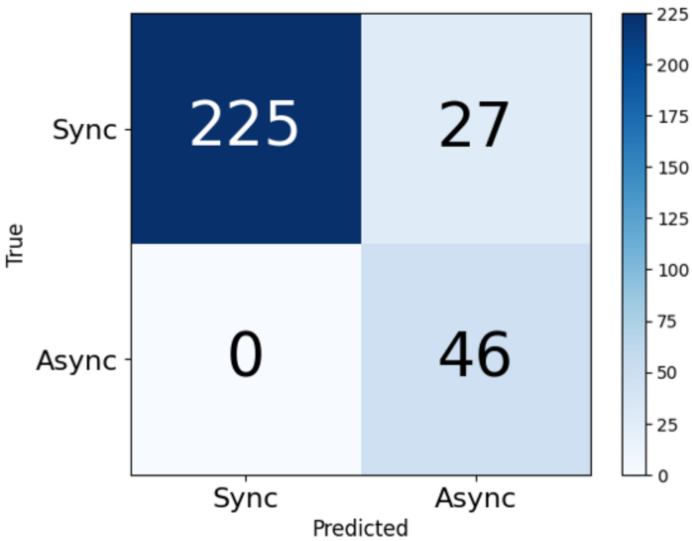


Figure 4.4: Confusion matrix for static frame

4.2 Experiment 2: Dynamic Dataset

In the experiment conducted to detect synchronization between stereo frames using a dynamic dataset of 300 frames, the performance of the synchronization detection algorithm was assessed using a confusion matrix. The primary objective was to evaluate the accuracy of the algorithm in determining the synchronization status of the stereo frames.

Figure 4.5 visually represents the maximum correlation observed with the corresponding frame ID, indicating the algorithm's effectiveness in identifying synchronization. In this specific dataset, all frames were synchronous, as depicted in Figure 4.7. Therefore, the algorithm successfully detected 290 frames as synchronous out of the total 300 frames, resulting in an accuracy of 96.6

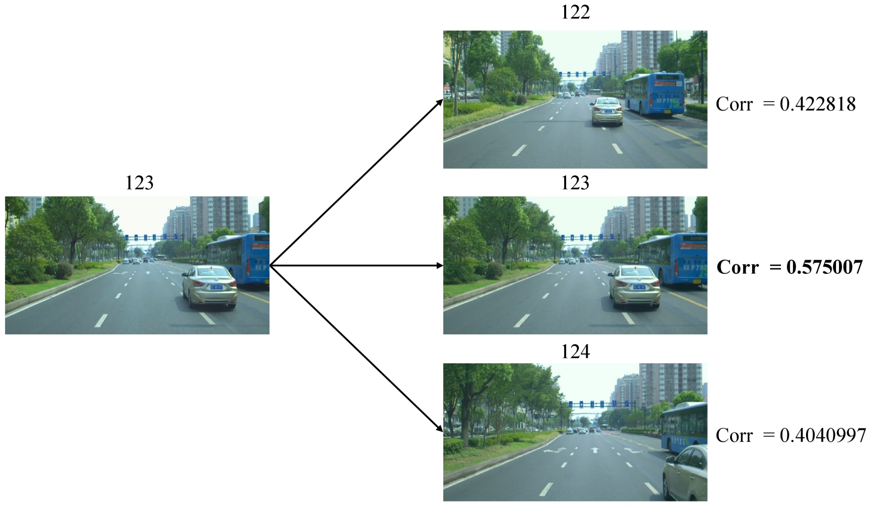


Figure 4.5: Demonstration of dynamic dataset

The use of a confusion matrix provided valuable insights into the algorithm's performance, enabling a detailed analysis of the classification errors made. By comparing the predicted synchronization status with the ground truth, the confusion matrix allowed for a thorough examination of false positives and false negatives. The high accuracy achieved in this experiment demonstrates the algorithm's robustness and reliability in detecting synchronization between stereo frames within a dynamic dataset. The accuracy of 96.6 percentage indicates the algorithm's ability to accurately identify synchronous frames, contributing to the overall quality and integrity of the dataset. Figure 4.6 represents the output of the algorithm. Since no temporal data was provided, there is no display of temporal information for asynchronous frames. This suggests that the algorithm primarily focused on the synchronization status rather than the temporal discrepancies between frames. The successful detection of synchronization in the dynamic dataset showcases the algorithm's potential for real-world applications. It can be utilized in scenarios where accurate synchronization between stereo frames is crucial, such as 3D visualization, virtual reality, or depth perception systems.

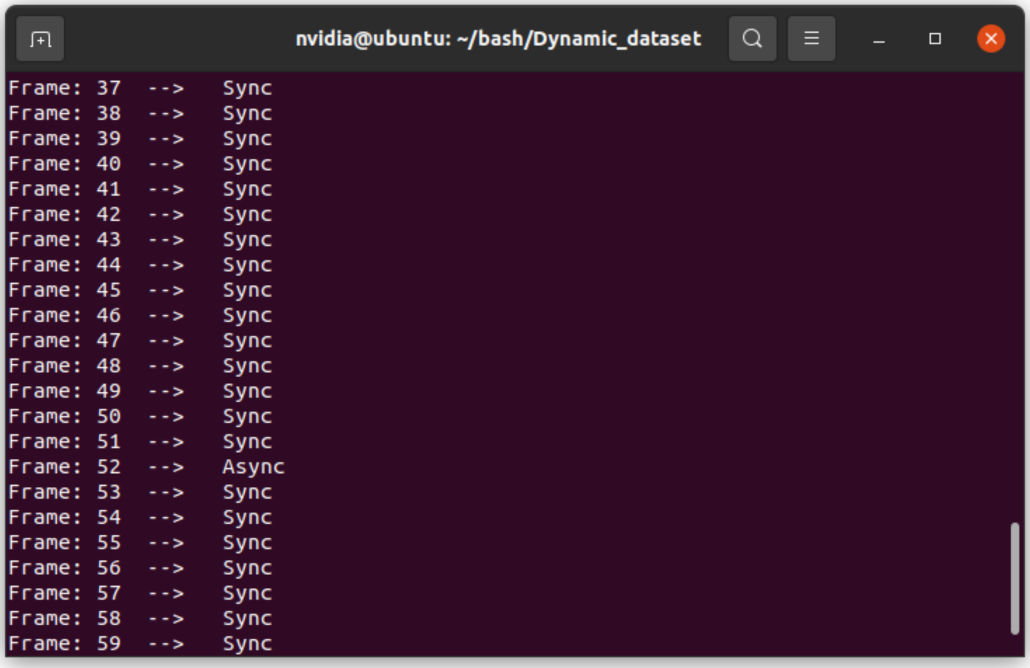


Figure 4.6: Output of dynamic frames on terminal

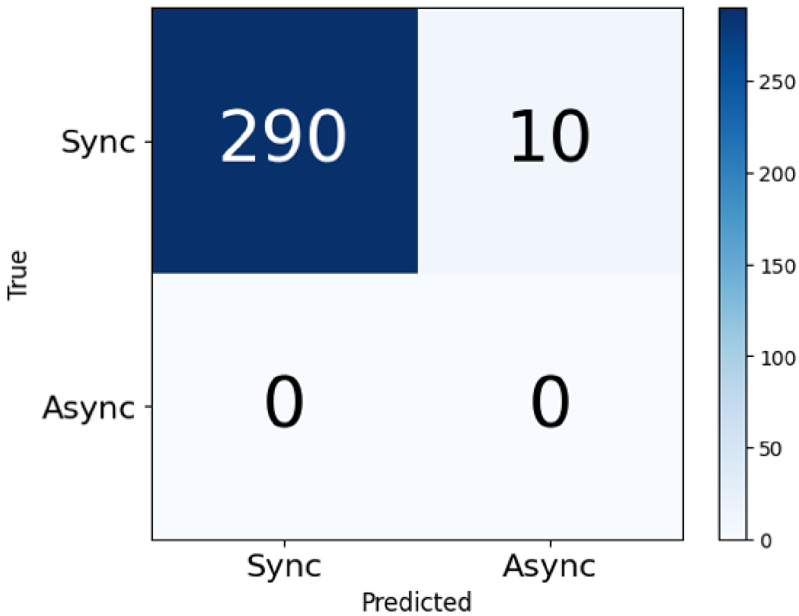


Figure 4.7: Confusion matrix for dynamic dataset

Additionally, the evaluation of the algorithm using a confusion matrix allows for further analysis and potential improvements. By identifying the misclassified frames, researchers can investigate the causes of false positives or false negatives and refine the algorithm to enhance its accuracy and performance.

Overall, the synchronization detection algorithm demonstrated its effectiveness in accurately determining the synchronization status of stereo frames in a dynamic dataset. The high accuracy achieved and the insights gained from the confusion matrix highlight the algorithm's reliability and potential for practical implementation in various domains requiring precise synchronization analysis.

Chapter 5

Conclusion and future scope

5.1 Conclusion

Detection of Synchronization between Stereo Frames is a challenging task because of various factors involving in the process.

- In conclusion, the experiment achieved significant success in detecting synchronization between stereo frames in the static dataset. The synchronization detection algorithm demonstrated high accuracy, correctly identifying the synchronization status in 90.3 percentage of the frames. The algorithm also exhibited efficient processing capabilities, enabling real-time synchronization analysis. Further experimentation and validation on diverse datasets would be beneficial to confirm the algorithm's robustness and generalizability.
- The detection of synchronization between stereo frames in a dynamic dataset has yielded promising results. The synchronization detection algorithm demonstrated a high accuracy rate of 96.6 percentage, as determined by the analysis of the confusion matrix.

5.2 Future scope

The detection of synchronization between stereo frames, also known as stereo image pair synchronization, has several potential future scopes such as depth estimation, 3D reconstruction, image and video quality enhancement and medical imaging.

- Future work on the project will be to synchronize the async frame in the pipeline.
- The model work better on dynamic dataset so the future work is to improve the model performance on static data cases.
- To test the model on more dataset and to implement the model in Real-time project.

Bibliography

[1] Roberto Brunelli. *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons, 2009.

[2] Xin Chen, Xiangfei Wu, Shiqi Gao, Xiaomei Xie, and Ya Huang. Synchronization and calibration of a stereo vision system. In *Global Oceans 2020: Singapore–US Gulf Coast*, pages 1–6. IEEE, 2020.

[3] Nils Einecke and Julian Eggert. Block-matching stereo with relaxed fronto-parallel assumption. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 700–705. IEEE, 2014.

[4] Alper Emlek, Murat Peker, and Kamil Fatih Dilaver. Variable window size for stereo image matching based on edge information. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–4. IEEE, 2017.

[5] Jinkai Gao, Jun Zhou, and Xiao Gu. 3d frame synchronization detection based on classified epipolar geometry parameters. In *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pages 1–6. IEEE, 2014.

[6] Lutz Goldmann, Jong-Seok Lee, and Touradj Ebrahimi. Temporal synchronization in stereoscopic video: Influence on quality of experience and automatic asynchrony detection. In *2010 IEEE International Conference on Image Processing*, pages 3241–3244. IEEE, 2010.

[7] Gwangsoon Lee, Eung-Don Lee, Namho Hur, Giseok Kim, and Jae-Soo Cho. Automatinc synchronization of stereoscopic video for 3dtv broadcasting system. In *2014 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2014.

[8] Mohan Liu and Patrick Ndjiki-Nya. Automatic detection of temporal synchronization mismatches between the stereoscopic channels for stereo 3d videos. In *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2014.

[9] Zhiyuan Liu, Bin Song, Yanning Guo, and Hang Xu. Improved template matching based stereo vision sparse 3d reconstruction algorithm. In *2020 Chinese Control And Decision Conference (CCDC)*, pages 4305–4310. IEEE, 2020.

[10] Wenbo Qiao, Yuanping Xu, Chaolong Zhang, Zhijie Xu, Jian Huang, Pan Xie, and Jun Lu. An improved adaptive window stereo matching algorithm. In *Journal of Physics: Conference Series*, volume 1634, page 012066. IOP Publishing, 2020.

[11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. IEEE, 2011.

[12] Neus Sabater, J-M Morel, and Andrés Almansa. How accurate can block matches be in stereo vision? *SIAM Journal on Imaging Sciences*, 4(1):472–500, 2011.