LTIMindtree

WORD DOCUMENT

SRS DOCUMENT

MULTI-SOURCE LIVE VIDEO STREAMING SYSTEM

Document Version / Details: Ver. 2.1/ 02-Jan-2023

1. INTRODUCTION

The Multi-Source Live Video Streaming System is a sophisticated platform engineered to efficiently handle numerous live video feeds originating from different locations or sources. Its primary objective is to orchestrate these various streams in such a way that they remain synchronized and seamlessly integrated to ensure a consistent and immersive viewing experience. This system employs advanced technologies and protocols to minimize the delay between the actual event occurrence and its transmission to viewers, resulting in low-latency streaming.

This kind of system is especially vital for events where multiple angles or perspectives need to be presented concurrently, such as sporting events or concerts. It aims to provide audiences with a captivating and engaging live experience, effectively bringing together diverse video sources into a cohesive, synchronized, and captivating viewing stream.

2. PURPOSE

The real-time system is designed to efficiently handle live video streams from diverse sources, including cameras and drones, during a live event. The primary objectives are:

Multi-Source Integration: Collect and integrate video feeds from various sources, ensuring compatibility and seamless aggregation.

Low-Latency Processing: Implement algorithms for real-time video processing with minimal latency, enabling swift analysis and seamless live streaming.

Synchronization: Establish synchronization mechanisms to align video frames from different sources, ensuring a coherent and unified viewing experience.

3. SCOPE

The real-time system for handling live video streams during a live event encompasses a comprehensive set of functionalities and features to ensure a seamless and immersive viewing experience. The scope of the system includes: Support integration with various video sources, including cameras and drones, to capture diverse perspectives of the live event. Implement protocols for establishing a reliable connection with each video source and managing the incoming streams. Develop a mechanism for efficiently sharing video frames among different processing modules in real-time. Utilize optimized data transfer protocols to minimize latency and ensure a smooth flow of video frames between modules.

Optimize video processing algorithms and network configurations to achieve low-latency streaming, minimizing the delay between the live event and the viewer's experience.

Implement adaptive streaming techniques to dynamically adjust video quality based on available bandwidth without compromising latency.

4. OVERVIEW

The challenge is to design a real-time system tailored for the management of live video streams originating from diverse sources like cameras and drones during live events. The key objectives are to facilitate efficient sharing of video frames among different processing modules, maintain synchronization across multiple sources, and ensure low-latency streaming. The system aims to deliver a seamless live viewing experience by addressing the following components.

Incorporate a mechanism to seamlessly integrate video feeds from various sources, including cameras and drones, ensuring compatibility and ease of aggregation.

Implement an optimized framework for sharing video frames in real-time among different processing modules. The focus is on minimizing latency and enhancing the flow of video frames. Address synchronization challenges arising from variations in frame rates, network latencies, and diverse device specifications. The goal is to achieve a cohesive and synchronized viewing experience for the audience.

5.SPECIFIC REQUIREMENTS

The specific requirements are:

Task 1: Inputting the data and reading the data

1.1: Convert the video frames to text file.

1.2: Acquire and store raw video frames from different sources using shared memory, using semaphores synchronize the process.

Task 2: Video Frame Processing Pipeline

2.1: Create multiple threads for image stabilization, color correction and resolution adjustments.

2.2: Using semaphores for multithreading application.

Task 3: Identifying the object and tracking

3.1: Creating a process for object detection using SDL library.

Task 4: Encoding and compressing

4.1: Creating threads one for encoding and another for compressing the video.

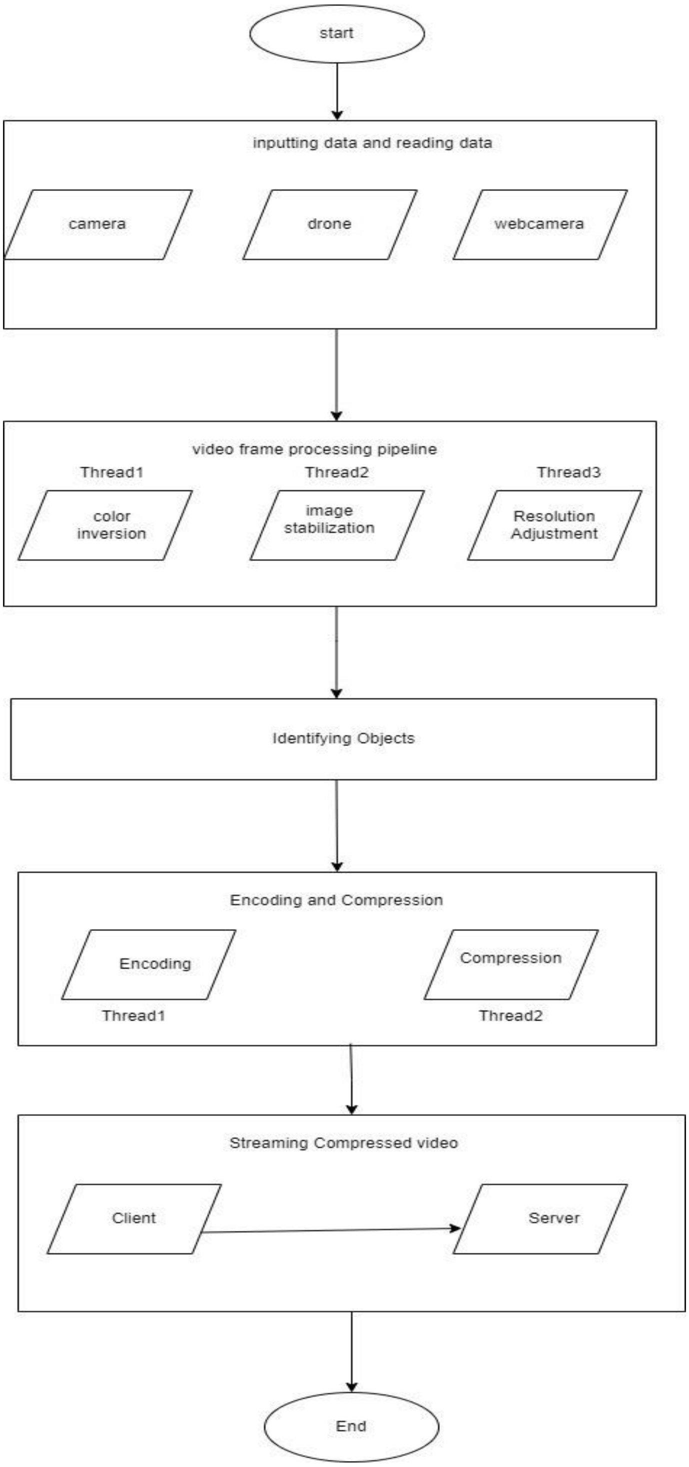
4.2: Synchronization between two threads, using semaphores.

Task 5: Streaming the compressed data

5.1: Creating a process which uses UDP method in Sockets to transfer the raw video frames to different users.

5.2: Synchronization in socket process using semaphores.

FLOWCHART


start
inputting data and reading data
camera
drone
webcamera
video frame processing pipeline
Thread1
Thread2
Thread3
color inversion
image stabilization
Resolution Adjustment
Identifying Objects
Encoding and Compression
Encoding
Compression
Thread1
Thread2
Streaming Compressed video
Client
Server
End

LTIMindtree

Let’s get to the future, faster. Together.