Team 5

**Real-time Stabilization and 3D Reconstruction of Hand Gestures and Finger Movement Traces Using LED-Equipped Gloves**

***Use Case Specification Document***

**Use Case ID: 2**

**Use Case Name: Image Processing Module  
Version No. 1**

**Project Document Revision History**

| **Version Number** | **Date** | **Revision Author** | **Description of Revision** |
| --- | --- | --- | --- |
| 1 | 10/15/24 | Soham Naik | Original Revision |
|  |  |  |  |



[**1.**](#_heading=h.gjdgxs) **Introduction 4**

[**2.**](#_heading=h.30j0zll) **Use Case Information 4**

[2.1 Actors 4](#_heading=h.1fob9te)

[2.2 Use Case Interaction 4](#_heading=h.3znysh7)

[**3.**](#_heading=h.2et92p0) **Trigger 4**

[3.1 Gesture Tracking Initiation 4](#_heading=h.26in1rg)

[3.2 Real-Time Frame Data Capture 4](#_heading=h.lnxbz9)

[**4.**](#_heading=h.tyjcwt) **Pre-condition(s) 4**

[4.1 Mobile Applications 4](#_heading=h.35nkun2)

[4.2 LED Visibility Given Ambient Lighting 4](#_heading=h.1ksv4uv)

[**5.**](#_heading=h.3dy6vkm) **Post-condition(s) 5**

[5.1 Successful Gesture Data Output 5](#_heading=h.44sinio)

[5.2 Data Exception - No Input Data 5](#_heading=h.1t3h5sf)

[5.3 Data Exception - Poor Input Quality 5](#_heading=h.2jxsxqh)

[5.4 Data Exception - Processing Error 5](#_heading=h.z337ya)

[**6.**](#_heading=h.4d34og8) **Use Case Swimlane (Activity) Diagram 6**

[**7.**](#_heading=h.3j2qqm3) **Main/Basic Flow(s) of Events (Happy Path) 8**

[7.1 Real-Time Gesture Tracking (Completed Path) 8](#_heading=h.1y810tw)

[**8.**](#_heading=h.4i7ojhp) **Alternate/Exception Flow of Events 9**

[8.1 Hardware Exception (Invalid Input Data) 9](#_heading=h.2xcytpi)

[8.2 Software/Data Exception (Invalid Data Processing) 9](#_heading=h.2s8eyo1)

[**9.**](#_heading=h.17dp8vu) **Assumptions/Business Rules including Non-Functional Requirements 9**

[9.1 Ambient Lighting 9](#_heading=h.1ci93xb)

9.2 Camera Permissions

[**10.** **Use Case Specification Review and Signoff 10**](#_heading=h.3rdcrjn)

# Introduction

This use case entails the functionality of the Image Processing Module, a key subsection of the larger Adaptive HCI system, designed to capture and process hand gestures in real-time using rolling shutter cameras and LED-equipped gloves. The user initiates gesture tracking by clicking the "Record Gestures" button, which activates the smartphone’s camera to capture hand movements in real-time. The captured frames undergo a series of pre-processing steps, including sharpening, denoising, edge detection, and thresholding, to enhance image quality.

Following pre-processing, the module extracts key features such as 2D coordinates, depth, and time from the frames using OpenCV. This data is then compiled into a 4D data structure (X, Y, Z, T), representing the LED's movement through 3D space over time. Advanced filtering techniques, including Kalman filters and spline interpolation, are applied to smooth the gesture path for improved accuracy. The final smoothed data is output as a 4D NumPy array and prepared for further analysis by the Machine Learning Module.

The use case also entails robust exception handling, addressing potential issues such as camera malfunctions or processing errors, ensuring reliable performance. As a subsection of the greater system, this module plays a critical role in delivering accurate, real-time gesture tracking for downstream applications such as gesture recognition and user interaction enhancement.

# Use Case Information

## Actors

| Actor Name | Role | Description |
| --- | --- | --- |
| Mobile Application Module | Input | Responsible for interfacing with the user, initiating gesture tracking, and managing camera access. |
| Image Processing Module (Provider) | Main | Processes captured frames, extracts key features, and performs pre-processing for gesture tracking. |
| Machine Learning Module | Output | Analyzes the processed gesture data for pattern recognition and further refinement of gesture accuracy. |
| Rolling Shutter Camera | Support | Captures frames in real-time, providing raw input for the Image Processing Module. |
| LED-Equipped Gloves | Support | Provides visible tracking points (LEDs) used by the camera to track hand movements in 3D space. |
| User | Primary Initiator | Interacts with the mobile application to initiate and stop gesture tracking by performing hand movements. |

## Use Case Interaction

The Image Processing Module use case is triggered by an initiation request from the Mobile Application Module, and it processes incoming frame data from the rolling shutter camera in real-time. As the camera captures each frame, the Image Processing Module immediately applies pre-processing and feature extraction steps. This processed data is made available to the Machine Learning Module continuously and in real-time, without waiting for all frames to be collected. The Machine Learning Module can begin analyzing and refining gesture patterns as the processed data is fed to it, ensuring a seamless flow of information across the system.

# Trigger

## Gesture Tracking Initiation

## The Image Processing Module use case is initiated by the Mobile Application Module after the user clicks the "Record Gestures" button.

## Real-Time Frame Data Capture

## The Mobile Application Module continuously feeds real-time camera frame data to the Image Processing Module.

# Pre-condition(s)

## Mobile Application

* + 1. The mobile application must be installed and configured, with access to the Camera to capture real-time frames.

## LED Visibility Given Ambient Lighting

* + 1. The LED-equipped gloves must have their lights visible and clearly detectable, ensuring they are distinguishable from ambient lighting conditions for accurate gesture tracking.

# Post-condition(s)

## Successful Gesture Data Output

* + 1. The Image Processing Module successfully processes the incoming frame data and outputs a 4D NumPy array containing the position (X, Y, Z) and time (T) values for the LED's path in 3D space, ready for further analysis by the Machine Learning Module.

## Data Exception - No Input Data

* + 1. The Image Processing Module detects that no input data (e.g., no frames captured due to camera malfunction or blocked view) is available. No gesture data is generated, and an error is logged.

## Data Exception - Poor Input Quality

* + 1. The input data is received, but due to poor ambient lighting or inability to detect the LED, the gesture tracking is incomplete or inaccurate. The system either logs an error or generates incomplete data for further analysis.

## Data Exception - Processing Error

* + 1. A software or processing error occurs during pre-processing, feature extraction, or gesture smoothing, resulting in incomplete or corrupted data. The system logs the error and no 4D data output is generated.

# Use Case Swimlane (Activity) Diagram

**6.1**

A diagram of software processing

Description automatically generated

# Main/Basic Flow(s) of Events (Happy Path)

## Real-Time Gesture Tracking (Completed Path)

## The user opens the mobile app and clicks the "Record Gestures" button.

## The Mobile App Module triggers the rolling shutter camera to capture real-time frames of the user's hand movements.

## The camera begins capturing frames using the Camera2 API and feeds them to the Image Processing Module.

## The captured frames are processed in real-time:

## The frames undergo pre-processing, which includes sharpening, denoising, edge detection, and thresholding to enhance image quality.

## Key features such as 2D position (X, Y), depth (Z), and time (T) are extracted from each frame using OpenCV.

## The Image Processing Module compiles the extracted data into a 4D array (X, Y, Z, T) representing the LED's position and movement over time.

## The gesture path is smoothed using Kalman filters, moving average filters, and spline interpolation to refine the output data.

## The final smoothed data is output as a 4D NumPy array, which is passed to the Machine Learning Module for further analysis and recognition.

# Alternate/Exception Flow of Events

## Hardware Exception (Invalid Input Data)

## The camera fails to capture frames due to a hardware malfunction or poor ambient lighting.

## The Mobile App Module detects the issue and displays an error message to the user.

## The user can retry or exit the recording session.

## Software/Data Exception (Invalid Data Processing)

## The input data is available, but the Image Processing Module fails to detect the LED or encounters an error during pre-processing or feature extraction.

## The system logs the error and prompts the user to adjust the environment (e.g., improve lighting or visibility of the LED).

## The user can retry or exit the session.

# Assumptions/Business Rules including Non-Functional Requirements

## Ambient Lighting

## The system assumes that the LED-equipped gloves will be clearly visible in the current ambient lighting conditions to ensure accurate feature extraction.

## The environment should not contain interfering light sources that can confuse the image processing algorithm.

## Camera Permissions

## The mobile app must have appropriate permissions to access the smartphone’s camera in real-time.

## The rolling shutter camera should be operational and free of obstruction.

# Use Case Specification Review and Signoff

| Review and Signoff of the Use Case Specification | | | |
| --- | --- | --- | --- |
| Name | Project Team Role | Signature | Date |
| Soham Naik | Lead Image Processing & Computer Vision Engineer | [Soham Naik](mailto:sohamn@umich.edu) | 10/16/2024 |
| Deniz Acikbas | Lead Mobile App & VR/AR Engineer | [Deniz Acikbas](mailto:dacikbas@umich.edu) | 10/16/2024 |
| Zaynab Mourtada | Lead Machine Learning Engineer | [Zaynab Mourtada](mailto:zaynabmo@umich.edu) | 10/16/2024 |
| Alan Raj | Lead 3D Visualizer Engineer | [Alan Raj](mailto:alanraj@umich.edu) | 10/16/2024 |
| Xiao Zhang | Client |  | 10/16/2024 |

## 