

Object Speed Tracking

Software Design Document

Alex Bailey, Ben Wick, Dylan Washburne

CS 461, Fall Term

Abstract

The Object Speed Tracking system incorporates many different technologies to make it possible. This includes computer vision libraries, object tracking methods, speed algorithms, and the user interface. This document shows the intended design for the system and how it will incorporate the technology need to meet the system requirements.

Date of issue: 12/1/2016

Issuing organization: Oregon State University

Authorship: Oregon state University

Change History: First version

CONTENTS

I	Introduction	2
I-A	Purpose	2
I-B	Scope	2
I-C	Context	2
I-D	Glossary	2
II	Body	2
II-A	Identified Stakeholders	2
II-B	Interaction Viewpoint	3
II-B1	Viewpoint Description	3
II-B2	Design Concerns	3
II-B3	Design Elements	3
II-C	Information Viewpoint	3
II-C1	Viewpoint Description	3
II-C2	Design Concerns	3
II-C3	Design Elements	4
II-D	Context Viewpoint	4
II-D1	Viewpoint Description	4
II-D2	Design Concerns	5
II-D3	Design Elements	5
II-E	Algorithm Viewpoint	5
II-E1	Viewpoint Description	5
II-E2	Design Concerns	5
II-E3	Design Elements	7
II-E4	Processing Attribute	7
II-F	Design Rationale	7

I. INTRODUCTION

A. Purpose

The purpose of this document is to give the necessary and required information to effectively define our systems design and give our team guidance to execute the system throughout the implementation process.

B. Scope

This system intends to utilize a live video feed in order to determine the speed of an object. The software will be able to identify a specific type of object, such as a car, and then calculate the object's speed and display it to the user. This offers an alternative method to track speeds of moving objects.

C. Context

The Object Speed Tracking system will be a windows application that utilizes a stereoscopic camera along with a computer vision library to track and calculate the speed of moving objects. This project is intended for a class at Oregon State University. Our goal is to implement the project design. Future development plans will be based on feature needs and success of project determined by the client.

D. Glossary

Term	Definition
API (Application Program Interface)	A particular set of rules and specifications that software programs can follow to communicate with each other.
Computer vision	The ability for the computer to extract, analyze and understand from an image or video.
Object	The entity being tracked by the video feed.
Stereoscopic camera	A type of camera with two or more lens.
User Interface (UI)	The visual part of the application in which the user will interact with.

II. BODY

A. Identified Stakeholders

The main stakeholder for our product is Alex Neighbors, as the client of our project. Alex Neighbors' concerns for our product is to be able to makeup for the areas that the current object velocity tracking technology is lacking, such as weather limitation and ability to track multiple object simultaneously.

However depending on the type of object we choose for our product, there will be other groups of people who would be stakeholders. We will now discuss possible object types, the stakeholders depending on the type and their concerns. We could choose vehicles as our object. Then a law enforcement would be a stakeholder in our product as it could be used in place of current speed limit enforcement technology. Their concerns could be how many vehicles it can track at a time or the accuracy of the velocity.

We could choose people as our object. Then sports teams and sports analysts could be stakeholders for analyzing how the players are moving. Anyone group trying to study the movement of groups of people, such as someone trying to see how fast people are able to evacuate in an emergency, could be a stakeholder. Their concerns could be the speed of the calculation or speed and consistent of the frame-rate.

B. Interaction Viewpoint

1) *Viewpoint Description:* The Interaction Viewpoint describes how the different pieces of our product work together and what is sent back and forth between them.

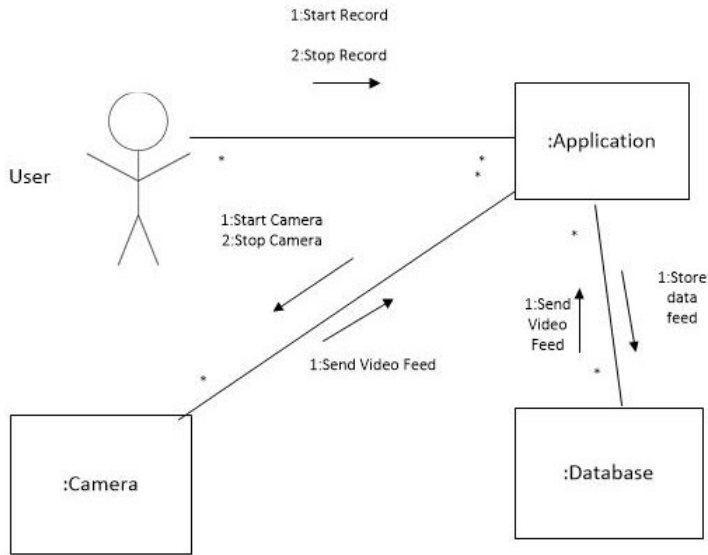


Figure 1. UML Communication Diagram

2) *Design Concerns:* The concerns for this Viewpoint are how the different components of our product will be interacting, which components will be interacting with each other, and what information will be passed between each component.

3) *Design Elements:* There are four elements in this viewpoint. The first is the user of the product. To keep the user interface simple the user will be able to start the recording and stop the recording. The second element is the application. This will be run on a computer of the user choice. The application will handle the interactions with the camera and database. It will send the start and stop signals to the camera and send the video for storage to the database. The application will also be able to receive a video stream from the database in order to recalculate the data. The third element is the camera. This will most likely be connected to the computer running the application, but not necessarily. The camera will record video when signaled and send the video feed to the application. The fourth element is the database. This will most likely be run on the same computer running the application, but not necessarily. This element will receive the video for storage from the application and will also send a video to the application for recalculating the data.

C. Information Viewpoint

1) *Viewpoint Description:* The purpose of the Information viewpoint is to describe how the data will be stored for the video radar software. The goal is to maximize efficiency and accuracy. There are two main entities within this viewpoint which include video storage and results storage.

2) *Design Concerns:* Concerns with information include storing the correct data, data management strategies and data access schemes.

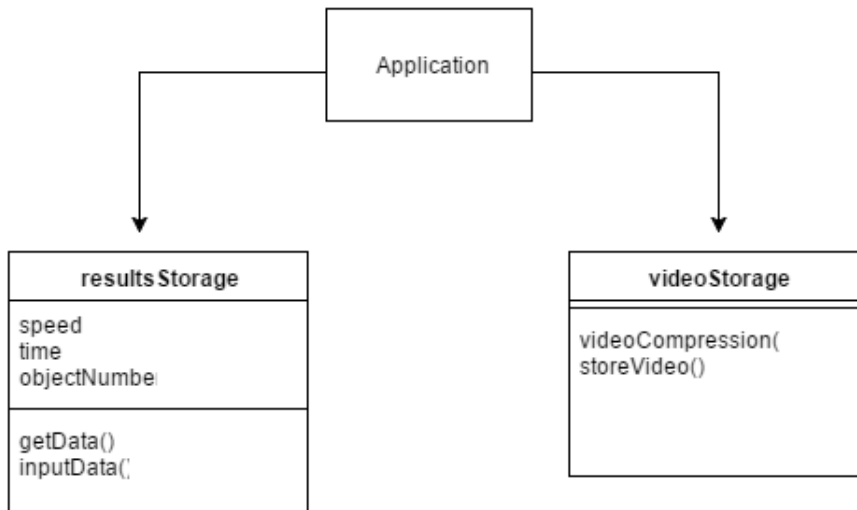


Figure 2. Flowchart diagram for information storage.

3) *Design Elements*: The user will interact directly with the application. Once setting up the system, the stereoscopic cameras will identify objects and calculate the speed at which it is moving. This information is displayed to the user on the screen as well as stored in a separate text file. The first element is the video storage. The live video feed will have be captured and stored. To do so we will implement a video capture class that is provided in the OpenCV library. This class will capture the live video feed and store the video in MP4 format into a file. The second element includes the final results being stored. The final results will be stored into a text file. The data that will be stored includes the object number for reference with the video, the speed of the object, and the time the object's speed was calculated. This will be implemented with a class called "storeResults". This class will extract the data needed from the calculation classes and the video feed and write it into a separate text file.

D. Context Viewpoint

1) *Viewpoint Description*: The Context Viewpoint describes the specific details of our project which are vital to its overall success, while sometimes not applying to any singular other category.

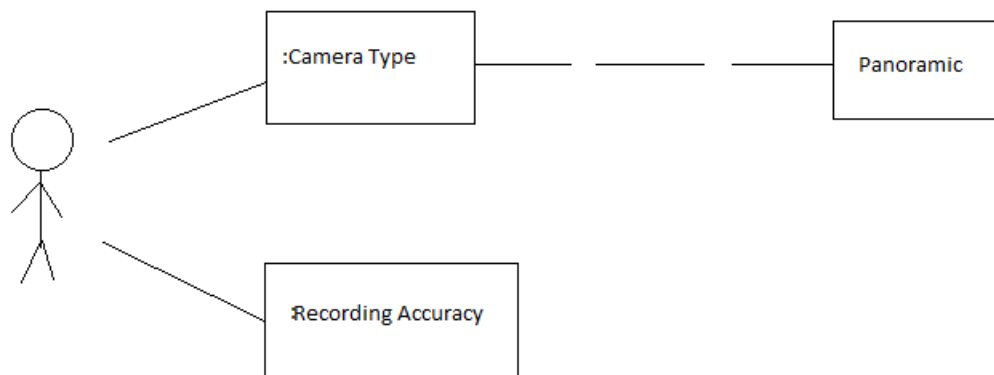


Figure 3. UML Use Case Diagram of the project's context.

2) *Design Concerns*: This Viewpoint is concerned with the specific issues which affect the performance of many other aspects of the project, which are to be adhered to throughout the building and testing of the project.

3) *Design Elements*: The first element is that the finished project will be able to return the velocity of specified objects in frame. The returned velocities will be within 90% of their actual velocities. The upcoming elements discussed are general pointers on how to achieve that outcome. The second element is that the project will utilize a bifocal camera to receive the video data. This is to allow the computer to generate information about the depth of objects in the frame. One heavily recommended consideration is to use a stereoscopic camera to fulfill this requirement, due to their nature to more easily synchronize shutter timing and create depth information. The next element is that the cameras must have ample potential range to accurately determine an object's depth. This means the cameras must be able to accurately return depth information within a zero to 100-meter range at the minimum. Were the camera able to support a zero to 150-meter range, that would be acceptable under these terms, so long as the minimum of zero is maintained and the upper bound is no less than 100 meters.

E. Algorithm Viewpoint

1) *Viewpoint Description*: This viewpoint describes method used for calculating the velocity of the desired object.

2) *Design Concerns*: The concerns of this viewpoint are determining the logic for calculating the velocity of the desired object type, determining where data would be coming from and sent to, and when data should be stored for later use and when it should be retrieved.

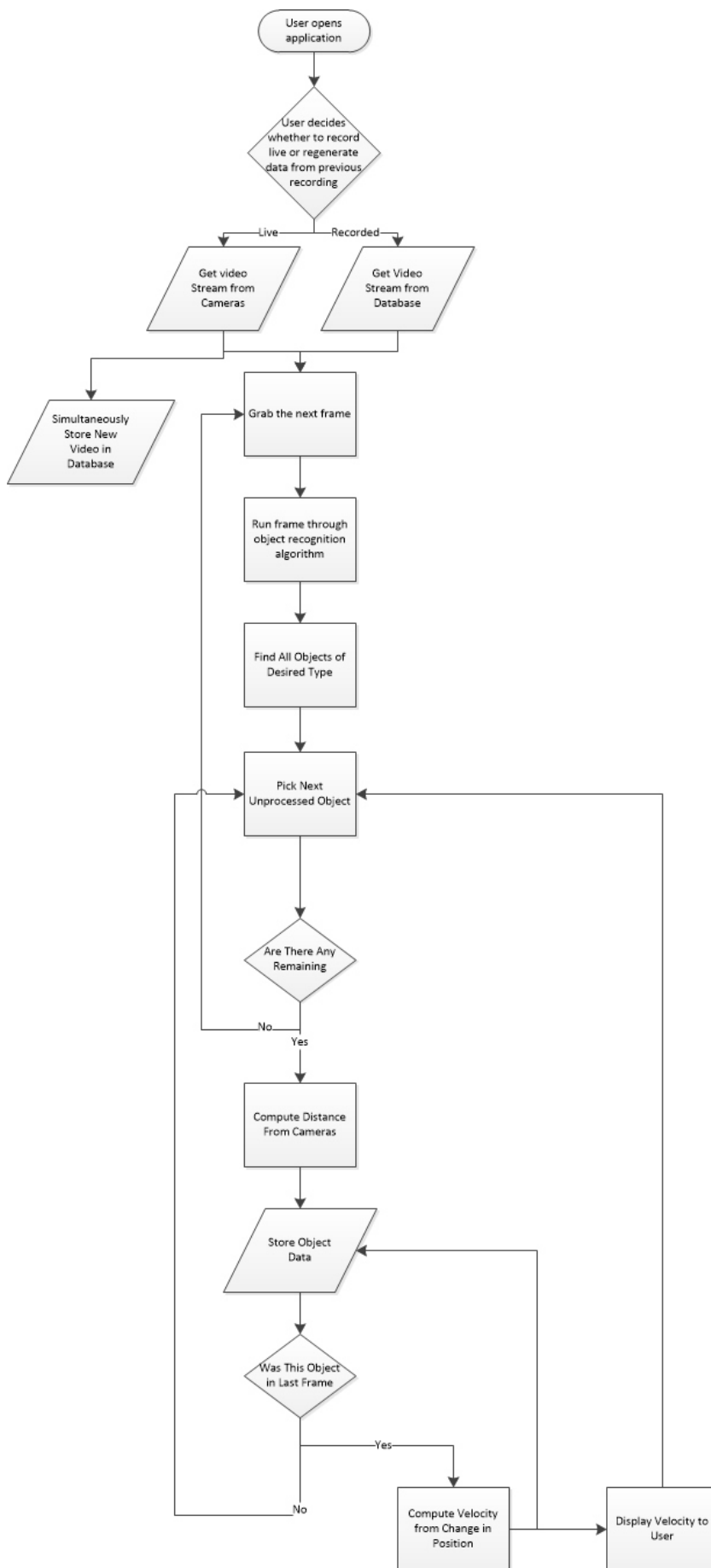


Figure 4. Flowchart Diagram of Calculating Object Velocity

3) *Design Elements:* The data elements represent the database. The element that displays velocity to user will print the information to the application window, with a way of showing the user which object is being referenced by that velocity.

4) *Processing Attribute:* The prerequisites for this flowchart is for the application to have been activated, the cameras be plugged in, the database connected to the application. The loop for finding the next object to be processed withing the frame will continue up to the minimum number of objects our product must recognize, but may or may not process the remaining objects depending on the limitation of the system. The priority of the algorithm will be to compute the velocities of the minimum number of vehicles, followed by frame-rate for displaying to velocity to the user.

F. Design Rationale

The main rational behind the creation of our product and this document was the concerns supplied by our client Alex Neighbors, making a product that can replace the current technology for object velocity tracking by making up for the pitfalls of the current technology. So we designed our product around how we could make up for the pitfalls, such as how a camera would have the ability to see and therefore track several object whereas one of the current methods, a radar gun, could not. Some design options considered for our product were should we use one or two cameras, should we store the video as the original video or with the overlay that the user will see about the objects velocity, and will the user be able to redo the calculations on a particular video file or only have the first version. The choice of having one or two cameras was something we decided on fairly early in our process as we realized that having two cameras would simplify several aspects of our product, such as determining if the object is moving away from the camera.