WCET Calculations Report

June 5, 2025

Abstract

This document represents our calculations and considerations for the UTFusion pipeline. The document also serves as a guide for future framework updates and while we do not include exact measurements from real-world data, we will explore the effectiveness of out pipeline in a parametric fashion and provide future readers with necessary information to calculate worst-case scenario latency.

1 Introduction

Previously we introduced the pipeline as UTFusion which served as an intermediate module between sensor data and decision making module in the UTCar project. We also divided the whole system into submodules that will be explained in further detail in this document. The overall architecture aims to deliver real-time and accurate data to decision making module to operate in time and meet the necessary deadlines that are required for a reliable autonomous driving system.

The document consists of following sections: in Section 2 we examine the considerations and limitations that make this analysis applicable, in Section 3 we give a brief explanation of the pipeline in which the project is going to operate, and in the Section 4 we will give parametric solutions to calculate WCET of the pipeline according to use case.

2 Considerations and Limitations

As we were designing the pipeline, we did not have access to real-world data from sensor periods and WCET of various modules in the pipeline as the exact model of sensors were not finalized and the modules simply did not exist and therefore couldn't be analyzed. Therefore, we decided to take a parametric approach and designed a framework that could work with many types of modules and sensors, given that following guidelines were met:

A. In our calculations, we assumed that the Fusion module always has a longer runtime compared to sensor data delivery system. Our calculations may or may not be valid under any other assumptions.

B. We designed our system for a non-preemptive hardware with no cache. In case of including cache in the system, WCET serves as an upperbound but further calculations may result in a lower WCET. In case of using a preemptive system we do not guarantee any upper or lower bound for latency.

C. We do not take responsibility for Depth Estimation module, or any other modules that are designed by another group, we simply provide calculations assuming the correctness of WCET given by the people who designed the module.

D. We expect all the objects on the road to be visible to both camera and radar sensors and have a minimum distance from eachother to be considered separate objects. This minimum distance will be discussed in the next phase.

E. We expect the radar data to be more accurate compared to stereo camera, which is the case in most of the sensors.

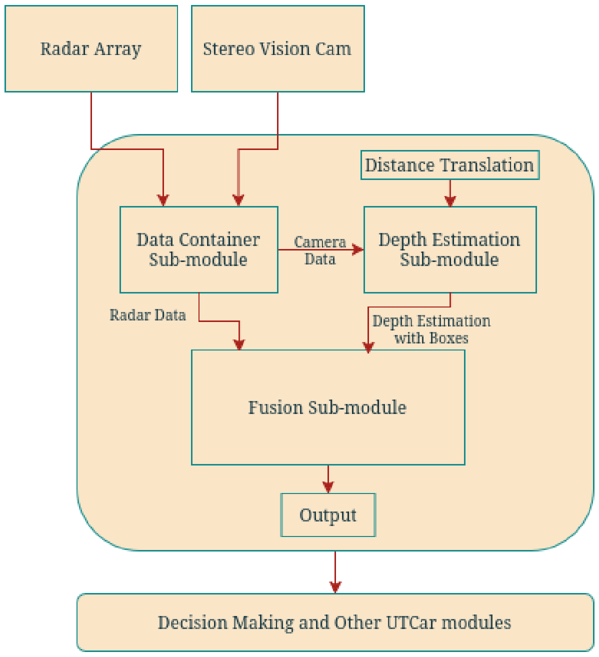
Radar Array
Stereo Vision Cam
Distance Translation
Data Container Sub-module
Camera Data
Depth Estimation Sub-module
Radar Data
Depth Estimation with Boxes
Fusion Sub-module
Output
Decision Making and Other UTCar modules

Figure 1: Overall UTFusion Architecture

3 Overall Pipeline

As shown in Fig. 1, the overall architecture of UTFusion consists of several submodules including Data Container, Fusion, Sensors, Distance Translation, and Depth Estimation. The data periodically arrives from radar and stereo vision camera and independantly gets written into data container. Depending on the arrival time, the data is consumed by the submodules either by polling or signaling. Fusion submodule waits for the Depth Estimation submodule to finish its work (for detailed information, refer to Depth Estimation using Stereo Cameras CPS project) and then fuses the radar array data with boxes acquired from Depth Estimation submodule. The output is then saved into a buffer and process runs periodically. It's worth mentioning that the Sensor-Data Container part of the system and the Depth Estimation and Fusion modules can and should run in different event loops and the data transfer is lock free.

3.1 Data Container

Data container consists of two buffers, each containing two sections to store data from camera and radar sensors. Each of these buffers are circular buffers with a fixed length and all the data gets timestamped upon arrival. An additional pointer is used to point to one of the buffers, and this pointer shows the readable buffer. Each time one of the sensors outputs a data that is not valid due to timestamp difference with latest data written by the other sensor, pointer is flipped and therefore we assure that at each given time after the first valid data pair, the container can provide valid data to other submodules. This architecture also minimizes the critical section of the I/O operations to flipping the pointer.

3.2 Distance Translation

Radar array and Stereo camera are both located in the front section of the vehicle, but sensors have a small difference in height and vertical distance from objects. This leads to slight error in distance estimation of the object and may have a huge impact on the detected angle of the object with respect to vehicle heading. Additionally, radar array is not placed as a straight line, but in a curve which needs to be translated into a two dimensional image. Accumulation of these errors in the fusion process alongside with an expected distance estimation error of both stereo camera and radar sensors can lead to disastrous results given that many RL models (which are strong candidates for decision making in autonomous driving) are sensitive to small changes in input. Therefore, we propose a Distance Translation submodule that applies a transformation on the data and converts all distances to calculated from a single reference point. This translation is calculated beforehand and therefore is a static part of the system.

3.3 Depth Estimation

We analyze the Depth Estimation submodule as a black box (although we have sufficient knowledge from the overall architecture of the submodule for time estimations). This module gets the camera data and distance translations as input and generates bounding boxes of visible objects inside the image. Afterwards, distance of each bounding box from the vehicle is calculated using only stereo camera data and given to the Fusion module as the input.

3.4 Fusion

In the Fusion section, we get bounding boxes and determine which radar sensors from the array correspond to each pixel of the image depending on the FOV of the sensors. Closest object in each FOV is selected using the data from Depth Estimation bounding box distances and these distances are corrected using radar data and closest bounding box distance in each FOV is scaled using a coefficient derived from radar outputs. The output of this module is stored in a buffer and the whole process is repeated periodically.

4 Worst Case Execution Time

4.1 Standard Response Time

There is no single standard for Autonomous Driving vehicles response time and it varies depending on the decision making model and various scenarios. The average response time of a human driver is measured between 1.5-2 seconds and many autonomous driving vehicles respond under 100ms. Since the required deadline is ambiguous, we do not guarantee a single response time and instead we calculate the worst case response time and response distance in the framework. A deadline can be configured in the specific use case. We also aim to maintain 30 FPS output from the whole pipeline but this time constraint heavily depends on the decision making module and Depth Estimation submodule.

4.2 UTFusion Response Time Calculations

We define the worst case scenario is follow:

- The vehicle must travel on full speed Vmax.
- A previously undetected object appears on the road in a distance D.
- All of the modules in the pipeline experience their WCET and integration happens to be in WCET too.

And we expect our pipeline to respond just in time and stop the vehicle before collision happens. If the conditions are not met, we propose lowering Vmax until the worst response time does not result in collision.

According to the architecture in Fig. 1 and our previously defined calculations in the project proposal, worst case scenario can be calculated as:

$$MaxDelay = DM + DR + UTFusion + ST$$

$$PipeLen = DM + DR + UTFusion$$

$$UTFusion = F + DE + BD$$

$$BD = 2A + I/O$$

$$A = A1 + A2$$

in which DM is the WCET of Decision Making module, DR is latency between decision making and reaction, ST is the stop time of the vehicle from Vmax, F is the Fusion module, DE is Depth Estimation WCET, BD is buffered data age and A1 and A2 are the maximum age of data when arrived from a sensor. More detailed explanation of the reason behind BD calculation can be found in the project proposal.

Furthermore we can calculate the maximum difference traveled by the vehicle before stopping and following expression must hold true:

$$PipeLen * Vmax + (Vmax)^2/2\mu g \leq D$$

in which g is the gravitational acceleration and μ is the coefficient of friction.

in our case, VL531X radar module operates in 50Hz frequency measuring 135cm distance under strong ambient light in short distance mode and IMX219-83 stereo vision camera operates in 60Hz frequency according to the official documentations. The actual radar distance however, was measured as 400cm in the lab environment and we prefer to use this measurement over the worst case scenario since the project will run in controlled lab enviornment only. Therefore, we can replace D with 400cm and A with 37ms. Also we measured Vmax of UTCar in 5m/s and the ST in 50cm.

According the following data, the overall pipeline should complete one full cycle in 150ms to meet the deadline and is guaranteed to detect objects within the 4m distance. In this 150ms, 37ms is the data latency, therefore the following inequality most be satisfied:

$$DM + DR + F + DE < 112\,\mathrm{ms}$$

I/O time is excluded since it is negligible compared to other parts of system because of the Data Container unit's unique architecture. We do not take responsibility regarding to DM, DR or DE, but will calculate the $WCET(F)$ once the project is finished and leave the rest of the calculations to the lab members.

The only remaining constraint that we are going to cover is the latency in which the data given by sensors is going to be considered valid. As the fusion algorithm greatly affects the maximum timestamp difference, we could not find a standard time difference to implement and came up with a empirical limit that we believe will be sufficient for our case. The method is described below but can change in the future.

We calculate maximum valid timestamp difference by accounting two factors: speed of the traveling vehicle and timestamps of the data. We expect the stereo camera depth estimation to be accurate in a range of at least 5cm, therefore any images that are taken by the camera or observations by radar that happened withing this distance is still reasonably valid. As mentioned before the vehicle's maximum speed is 5m/s, and therefore a maximum timestamp difference of 10ms is still valid. We formulated this calculation as follows:

$$max(radarError, cameraError)/Vmax \geq MaxTimestampDiff$$