Splash: Stream Processing Language for Autonomous Driving *

Soonhyun Noh and Seongsoo Hong

Artificial intelligence based on deep learning techniques became the essential part of an autonomous vehicle due to its superior performance in scene recognition missions. The AI technology applied in vehicle development is often referred to as Automotive AI. To fully support Automotive AI, it is necessary to technically address the extra requirements that arise differently from those of conventional vehicles. Two of the most important requirements of Automotive AI are (1) real-time stream processing and (2) reliability. To support real-time stream processing, it is necessary to guarantee timing constraints in stream processing. To offer sufficient reliability, it should be able to ensure the successful fulfillment of normal operations even in various exceptional situations that occur during driving vehicles.

Currently, most companies and research institutes that develop autonomous vehicles rely on application developers to meet the above requirements. Since application developers often resort to a time consuming and iterative tuning process, this approach leads to inefficiencies in the development process and can lead to fatal injuries due to unintended circumstances that are overlooked during the tuning process.

In this paper, we propose the stream processing language named Splash to overcome the limitation of the traditional development methodology. Splash has five advantages. First, Splash visually expresses the flow of sensor stream data processing so that developers can easily grasp the complex interworking of a given AI program. Second, Splash let developers explicitly annotate the timing constraints of stream processing. Third, Splash allows developers to define exceptions and specify the handling of each exception. Fourth, Splash can describe complex synchronization issues of sensor fusion algorithms more perceptibly. Finally, Splash supports integration between domains that are developed with different programming paradigms such as data-driven and time-driven.

To support the stream processing required in autonomous driving, Splash allows developers to specify the three basic operational semantics which consists of (1) data processing semantics, (2) path control semantics and (3) triggering semantics. Data processing semantics are for developers to freely program stream operations as they intend. Path control semantics are for developers to control the flow of stream data on the data flow graph that represents stream data processing. Triggering semantics are semantics to determine the initiation points and methods of the stream processing.

Splash also offers timing semantics for specifying the time constraints for real-time stream processing. Splash supports four timing constraints: (1) deadline, (2) input/output minimum rate constraint, (3) freshness constraint and (4) correlation constraint. Deadline is the time limit from when the processing of a given stream data element starts to when the result is produced. The input/output minimum rate constraint is the minimum number of stream data elements to be input and output per unit time. The freshness constraint is the time at which a given stream data element is valid after the data element is generated. The correlation constraint is defined as the maximum input time difference between two or more stream data elements that must be processed together.

Splash introduces six language constructs to represent basic operational semantics and timing semantics: (1) factory, (2) processing cell, (3) fusion operator, (4) select operator, (5) port and (6) pipe. The factory is the largest building block of stream processing and contains a data flow graph that combines Splash language constructs internally. The node of the data flow graph is the processing cell, fusion operator, selection operator and factory. The edge of the data flow graph is the pipe. The processing cell is the smallest execution unit of stream processing. It takes a data element as input, performs an operation defined by the developer, and outputs the result. The fusion operator is an operator that merges multiple stream data into one stream data. The selection operator is an operator that selects the delivery path for an input stream data. The port is a component of a processing cell, fusion operator, selection operator or factory where data can enter or leave. The pipe is a delivery path for data and connects two ports.

Splash also supports exception handling. We define an exception as a situation in which the abovementioned semantics are not followed. There are three kinds of exceptions: (1) timing violation exception, (2) invalid data exception and (3) absent data exception. A timing violation exception is a violation of a specified timing constraint. Invalid data exception refers to a case where the value of input data is out of the valid range. Absent data exception is a case where there is no input data element to process. Developers can write an exception handler for each exception.

We have verified the Splash programming using adaptive cruise control (ACC) and lane keeping assist system (LKAS) algorithms. We expressed the algorithms using the Splash language constructs and specified timing constraints and exception handling. We then generated source code for the algorithms and demonstrated the successful operation of the algorithms on the runtime based on Linux and DDS.

*This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. R7117-16-0164, Development of wide area driving environment awareness and cooperative driving technology which are based on V2X wireless communication)

Soonhyun Noh is with the Electrical and Computer Engineering Department, Seoul National University, Seoul, Korea.

Seongsoo Hong is with the Electrical and Computer Engineering Department, Seoul National University, Seoul, Korea. (corresponding author to provide phone: 82-2-880-8357; fax: 82-2-871-5974; e-mail: sshong@redwood.snu.ac.kr).