

# SIMERO Software System Design and Implementation

Thorsten Gecks

AG Eingebettete Systeme und Robotik (RESY),  
Fachbereich Informatik, Universität Kaiserslautern,  
<http://resy.informatik.uni-kl.de/>

1. Motivation and Introduction
2. Basic Design Decisions
3. Major System Components
4. Robot Controller Software Part
5. Results



# Motivation and Problem Description

---

## Safe Human-Robot-Interaction

- Avoid accidents injuring human operators
- Avoid collisions with material
- Cooperation of Humans and Robots
- Provide real-time reactions

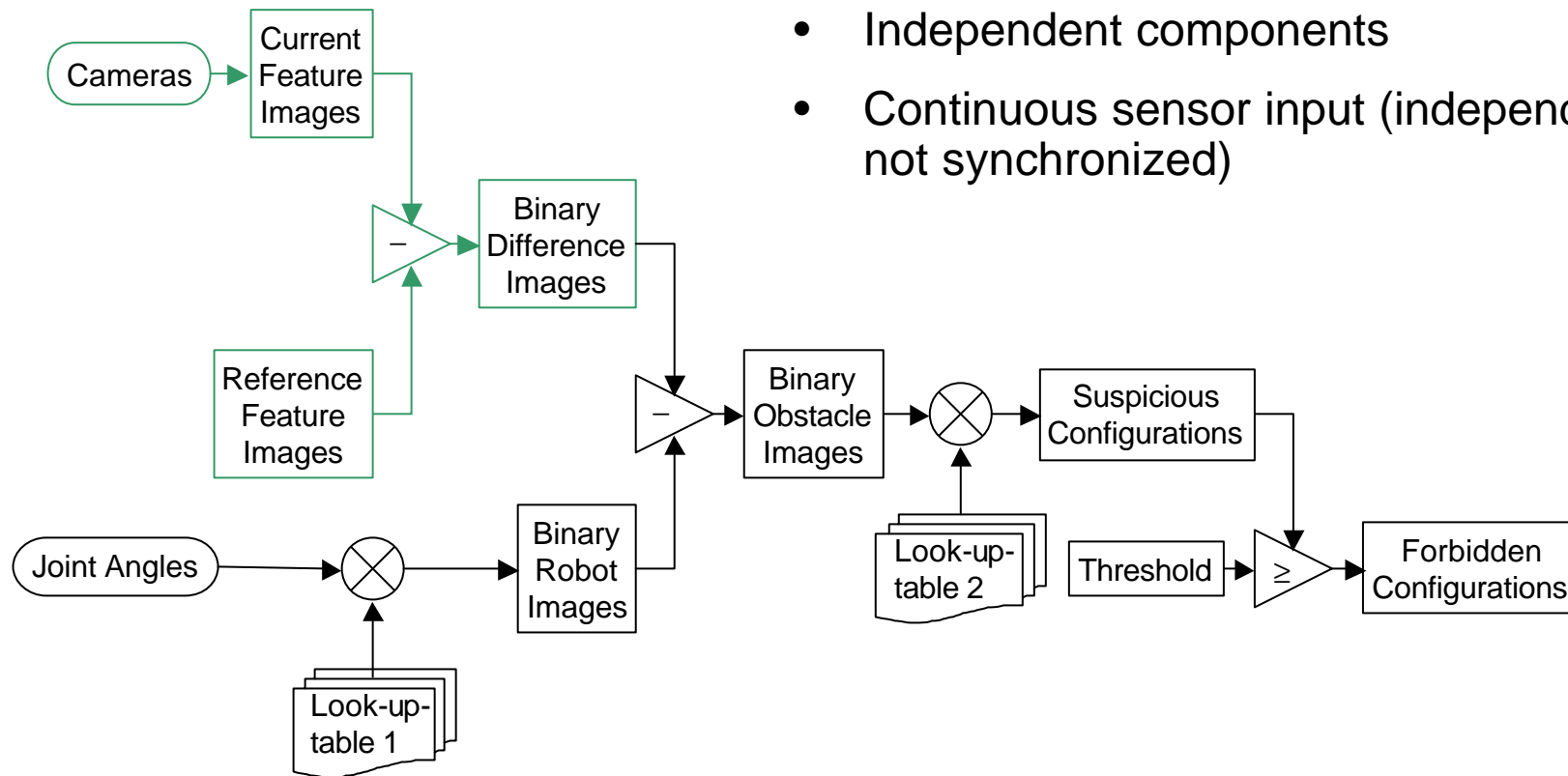
## Challenge

- Develop a framework for algorithms, that
  - process sensor data
  - work distributed
  - provide safety mechanisms



# Basic Algorithm Concept [ebert2001]

- Interaction of Data streams and operators
- Independent components
- Continuous sensor input (independent, not synchronized)



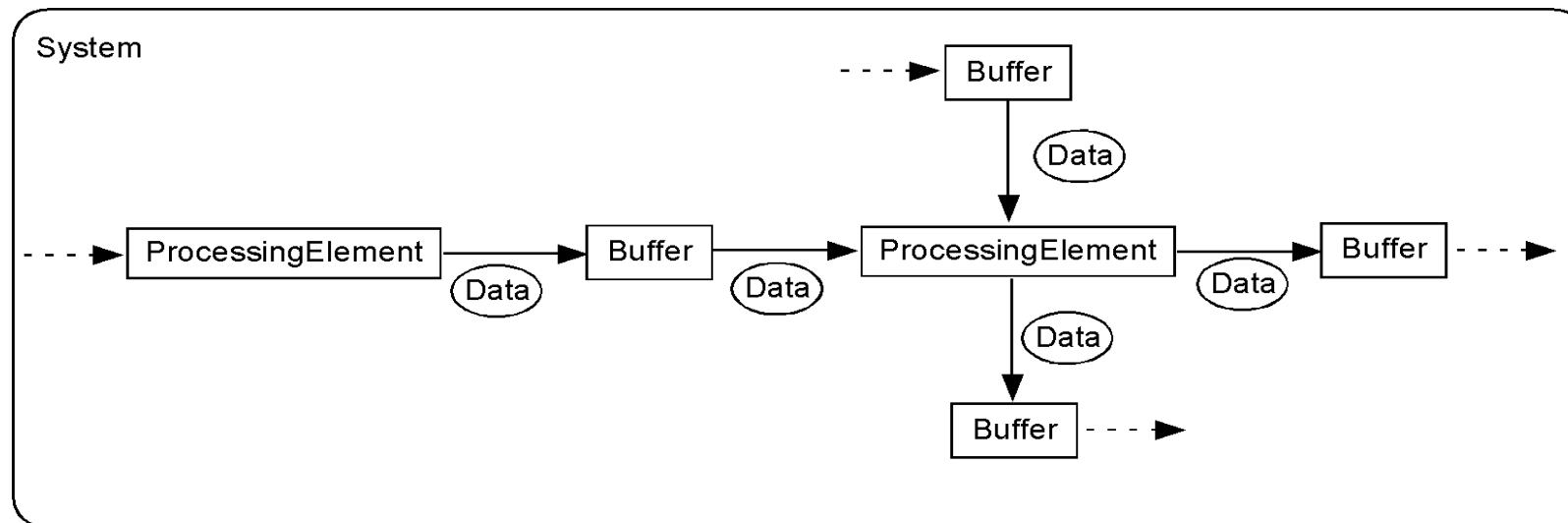
# Basic Design Considerations and Decisions

## Multithreading (Concurrency, Parallelism)

- Asynchronicity of sensor input and actuator output
- Speed improvements on parallel architecture

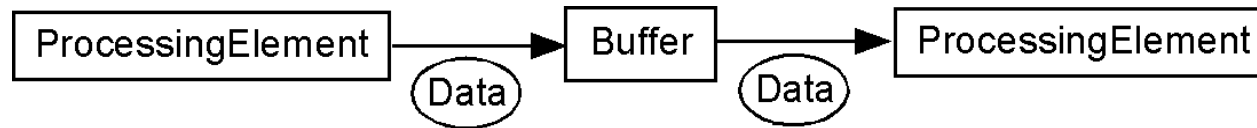
## Data-flow / data-operator oriented concepts

- Concept of operators on data (realized in ProcessingElement class)
  - \* Distinct Input / Output operators (Sensor and Actuator class)
- Concepts of buffered data channels (realized in Buffer class)



# Details of Major System Components

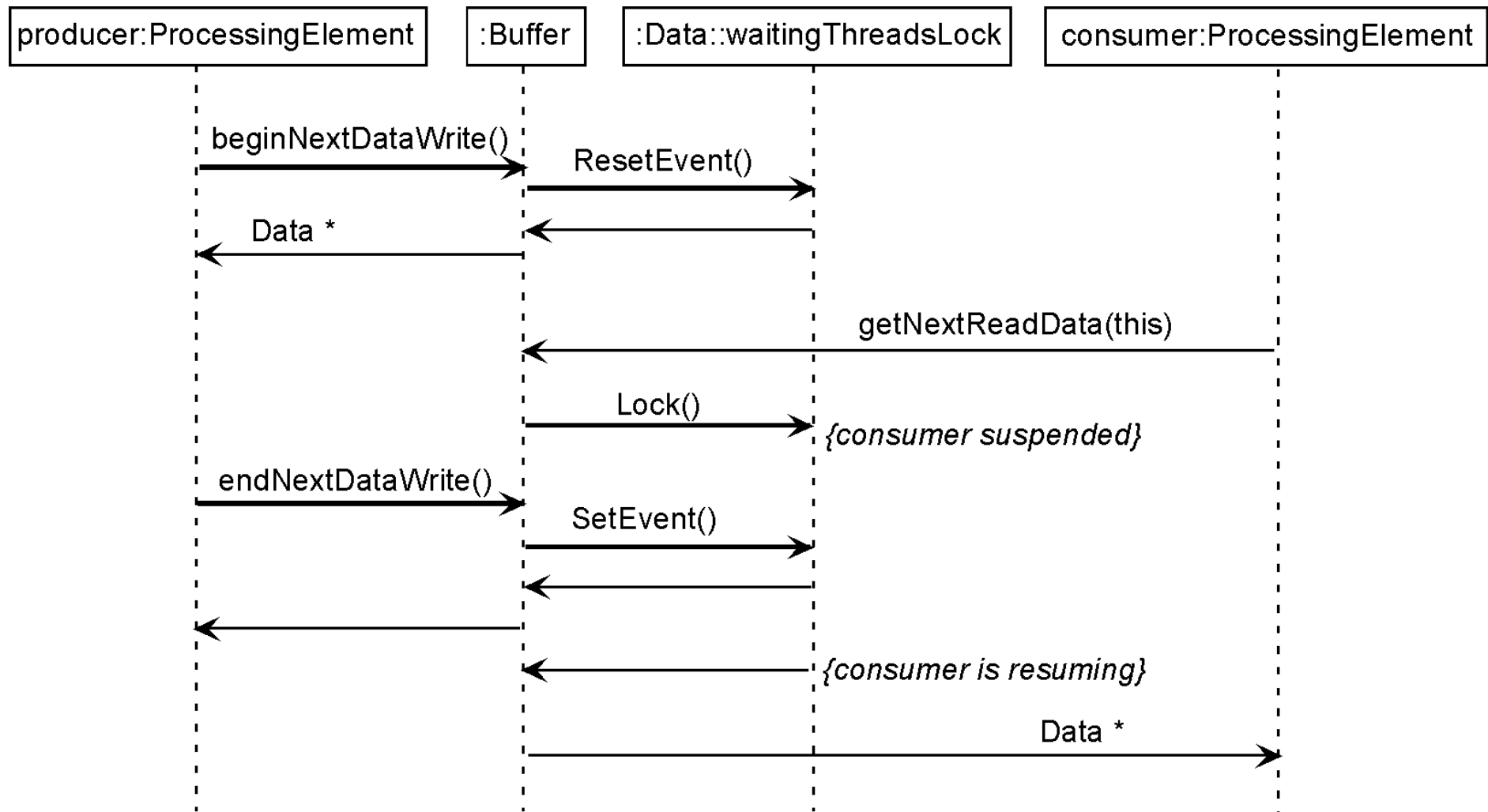
---



## Concepts

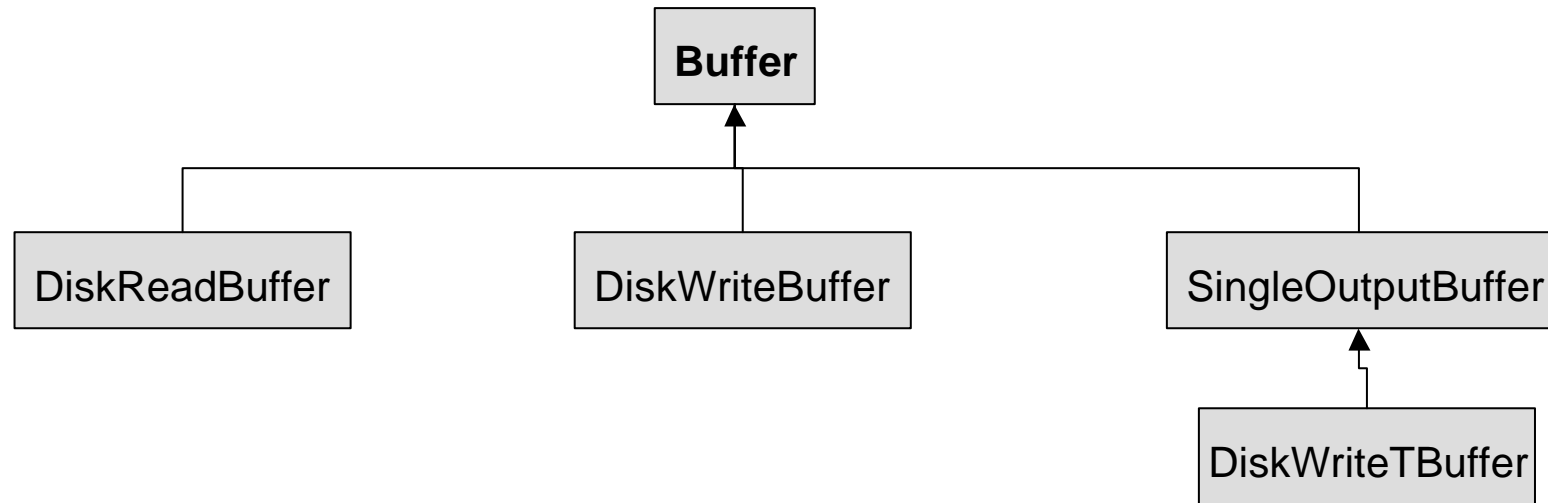
- Buffer class owns data objects (controls creation and destruction)
- Data class -> piece of coherent information
- ProcessingElements can be threads
- Buffers synchronize ProcessingElements on Data objects

# Sequence Diagram: ProcessingElement-Buffer-Interaction



# Disk Storage Framework

---



## Challenge

- Disk storage as special data flow channel concept
- Transparency to data operators (ProcessingElements)

## Solution

- Storage functionality in sub-classes of Buffer class
- Data class provides storage interface to Buffer class

# Z-Function and Compression

## Problem

- Memory-intensive lookup tables

## Solution

- Compression with RLE on linearized image regions

## Linearization with “Z-Function”

- Bit interleaving of every coordinate
- Interleaving from LSBs to MSBs
- Pattern in 2-dim leads to naming

## Results

- Compression rate achieved ~ 66 %

## Additional benefits of Linearization

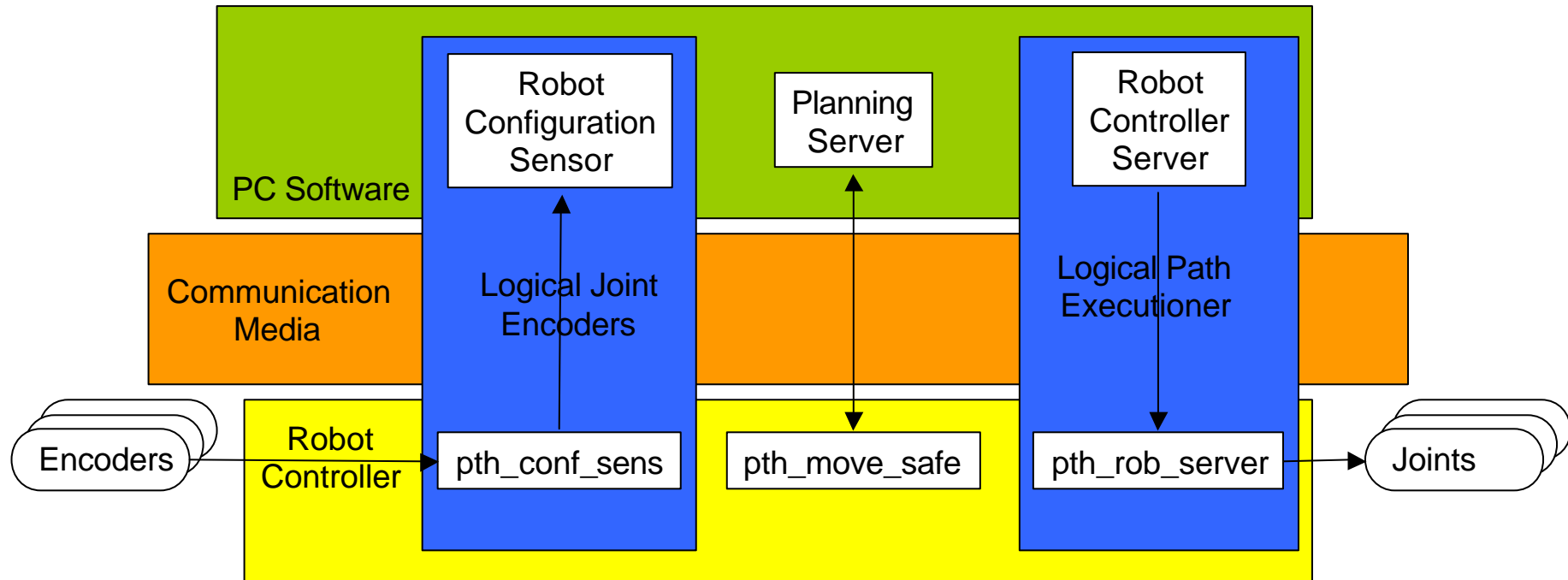
- Simplify algorithms covering every pixel

	0	1	2	3
	00	01	10	11
0	0	1	4	5
00	0000	0001	0100	0101
1	2	3	6	7
01	0010	0011	0110	0111
2	8	9	12	13
10	1000	1001	1100	1101
3	10	11	14	15
11	1010	1011	1110	1111





# Robot controller part design

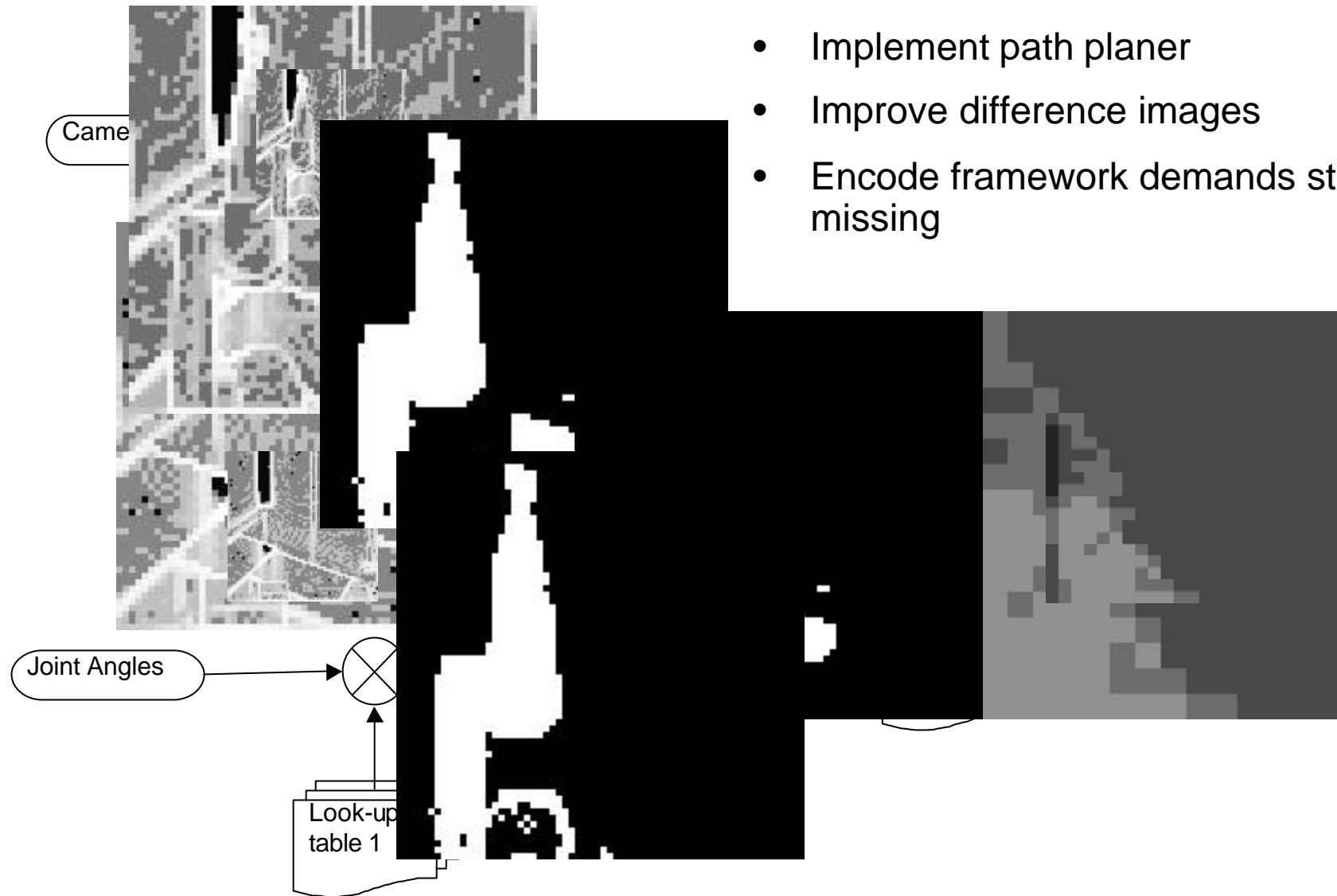


## The module `pth_move_safe`

- provides path planning service to other V+ - tasks
- contacts `PlanningServer` on PC side
- Controls tasks `pth_conf_sens` and `pth_rob_server`
- secures the connection via a network heartbeat
- provides additional services

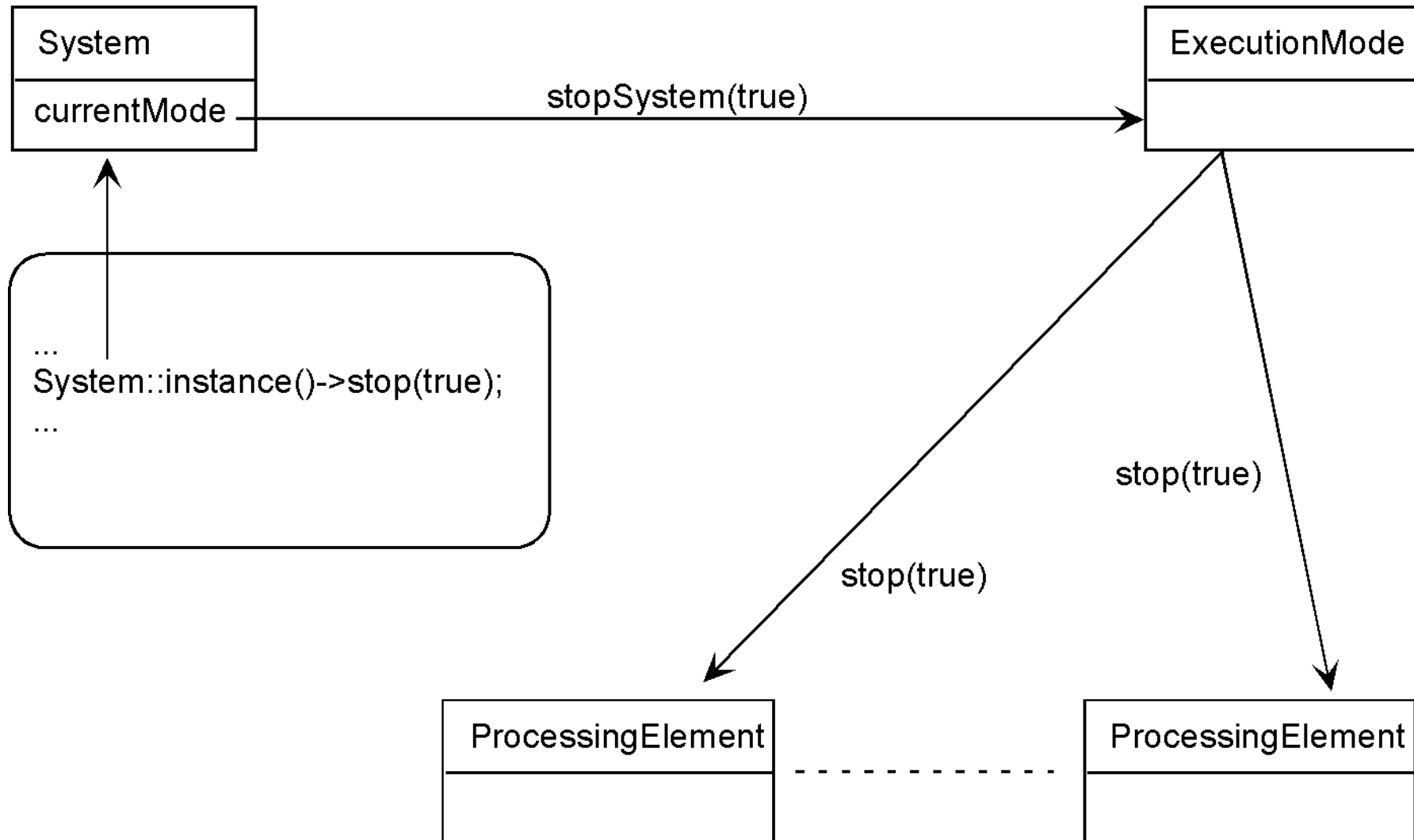
# Results and future work

---



- Implement path planer
- Improve difference images
- Encode framework demands still missing

# System Safety Mechanism



# Bibliography

---

**[ebert2001] Ebert,Dirk; Henrich, Dominik „Safe Human-Robot-Cooperation: Problem Analysis, System Concept and Fast Sensor Fusion“ in Proc. of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2001 Baden-Baden Germany) pp. 239-244**

