Valeo

SMART TECHNOLOGY
FOR SMARTER CARS

POINT-CLOUD PROCESSING USING HDL CODER

AGENDA

Introduction

- LiDAR Sensors in Automotive Industry

Point Cloud Processing

- Classic processing pipeline

HDL-Coder Workflow

- Hardware structure
- Examples on the usage

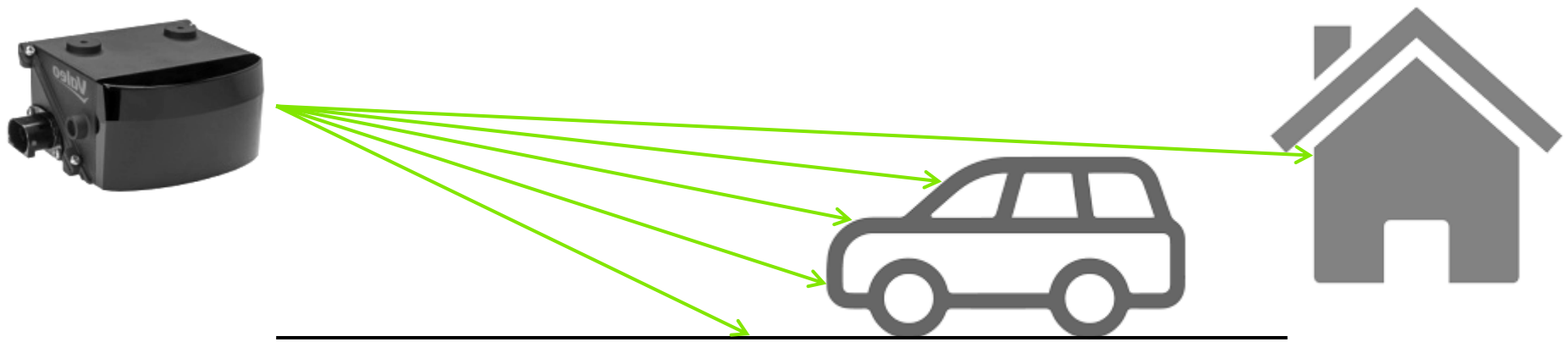
LIDAR SENSORS IN AUTOMOTIVE

Additional sensor technology offering

- High distance measurement
- Accurate point distance
- Working under bad circumstances (fog, night)
- Enhancing redundancy

Time of flight measurement for emmitted Laser Beam

$$d = \frac{\delta_t \cdot c}{2}, \quad \delta_t = \text{measured time till light is received}, c = \text{speed of light}$$



LIDAR SENSORS IN AUTOMOTIVE

Valeo SCALA first mass production LiDAR Sensor for automotive (2017)

- 58.000 points per second
- Next generation ~200.000 points per second

Research Utilities

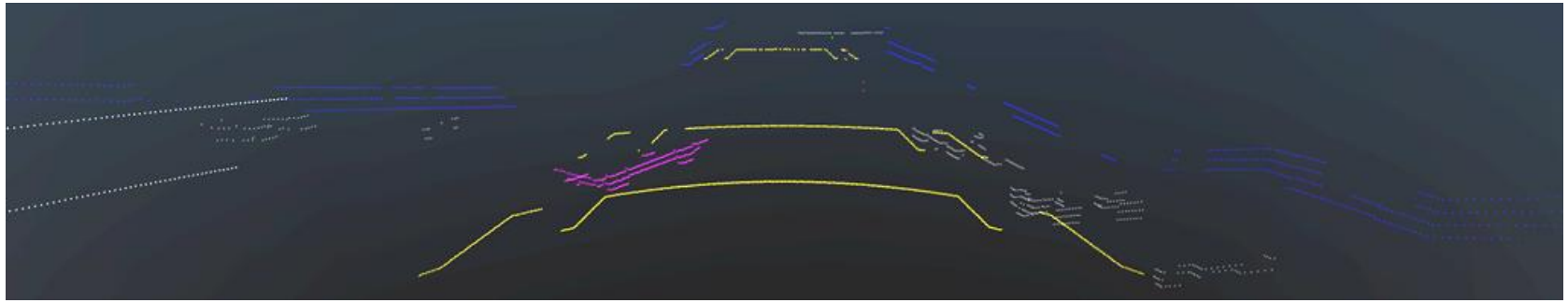
- Velodyne LiDAR Ultra Puck ~600,000 points per second
- Velodyne HDL-64E 2.2 million points per second

Number of points to be processed will increase

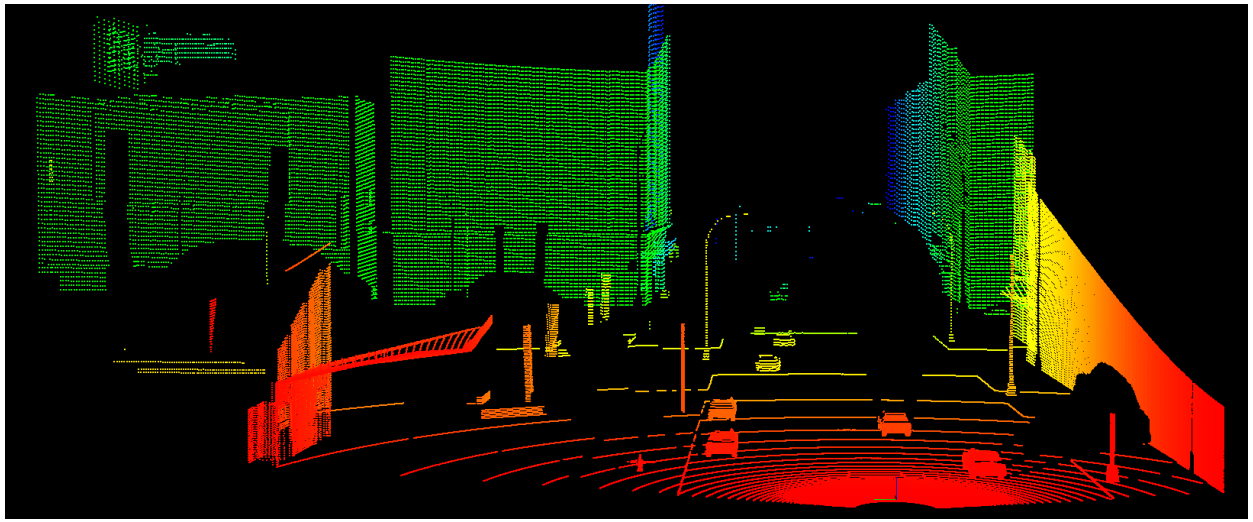
Possibility of getting more information out of the data

LIDAR SENSORS IN AUTOMOTIVE

State of the art



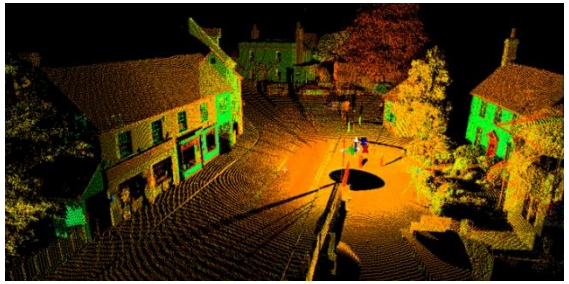
Possible future resolution



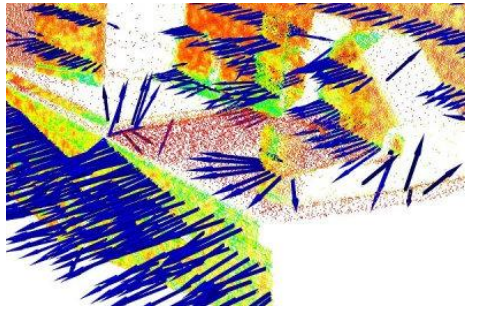
WHAT IS A POINT CLOUD

A point cloud is a set of discrete points in 3D space. Each point is represented by its cartesian coordiantes and may hold additional information

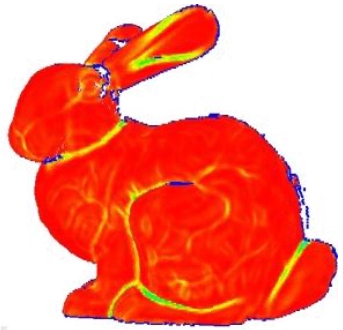
- XYZ coordinates
- RGB (Texture)
- Intensity
- Normals
- Curvature
- …



360° view

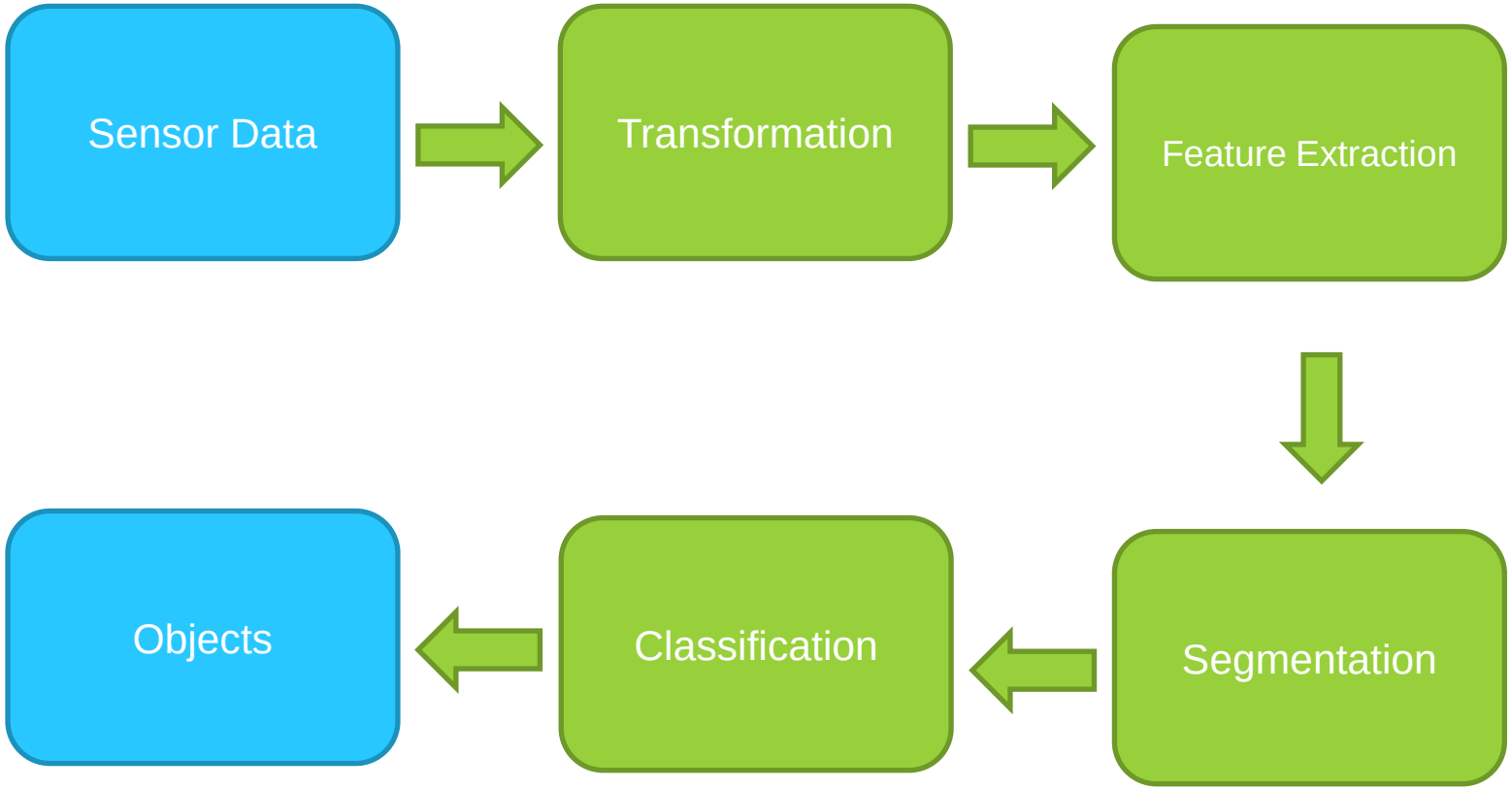


surface Normals



curvature

POINT CLOUD PROCESSING PIPELINE



POINT CLOUD - TRANSFORMATION

Transfor-mation

- Sensor Data distance measurement
- Known column and row index (n,m)

$$d(n,m), \quad \text{with} \quad n \in [0,1,...,\#rows], m \in [0,1,...,\#columns]$$

- Transformation to cartesian coordinates necessary

$$\begin{pmatrix} x(n,m) \\ y(n,m) \\ z(n,m) \end{pmatrix} = f(n,m) \cdot d(n,m) = \begin{pmatrix} \cos(pitch(n))\cos(yaw(m)) \\ \cos(pitch(n))\sin(yaw(m)) \\ \sin(pitch(n)) \end{pmatrix} \cdot d(n,m)$$

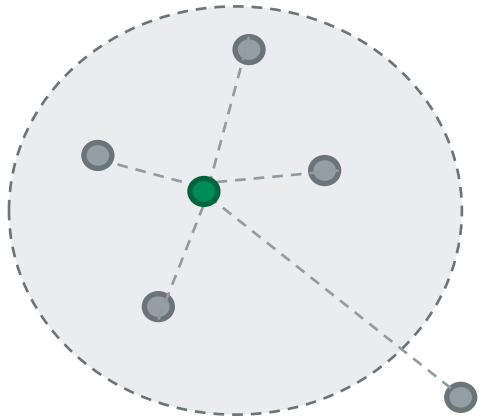
- pitch and yaw depend on resolution, mounting position

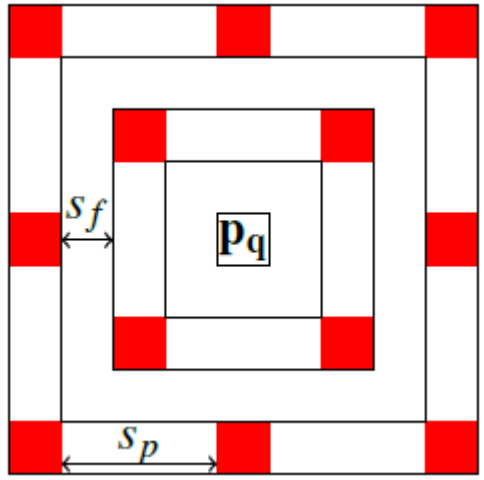
POINT CLOUD – FEATURE EXTRACTION

Feature Extraction

Normal estimation

- Nearest Neighbours for unorganized point cloud
  - Brute force
  - Kd-tree
  - Approximate nearest neighbours
- Nearest Neighbours for organized point cloud
  - Pixel like structure
  - Define a search mask
- Normal estimation using PCA
  - Surface normal corresponds to eigenvector of smallest eigenvalue





Curvature

- Estimated from Eigenvalues

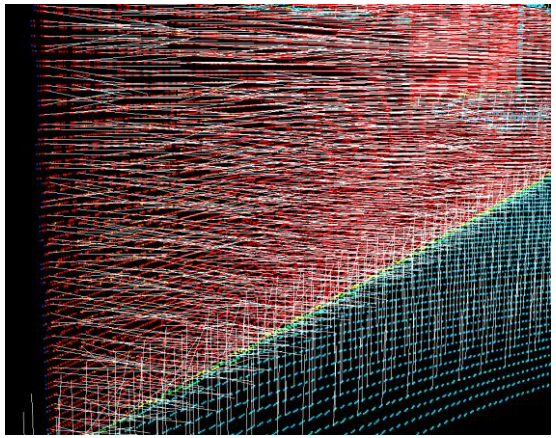
$$\frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad \text{with} \quad \lambda_0 \leq \lambda_1 \leq \lambda_2$$

POINT CLOUD – SEGMENTATION

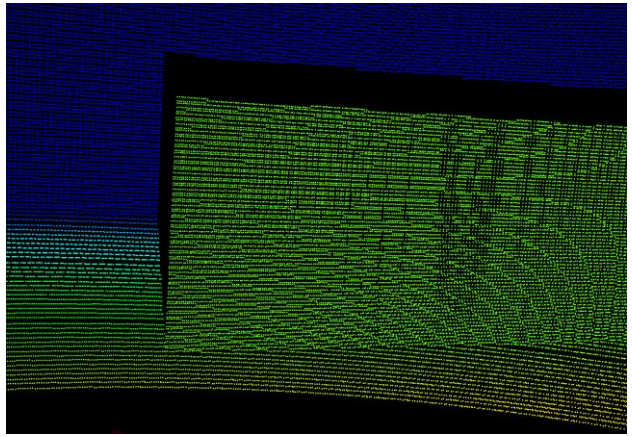
Segmen-
tation

- Evaluate features to detect
  - Depth changes

  - Edges



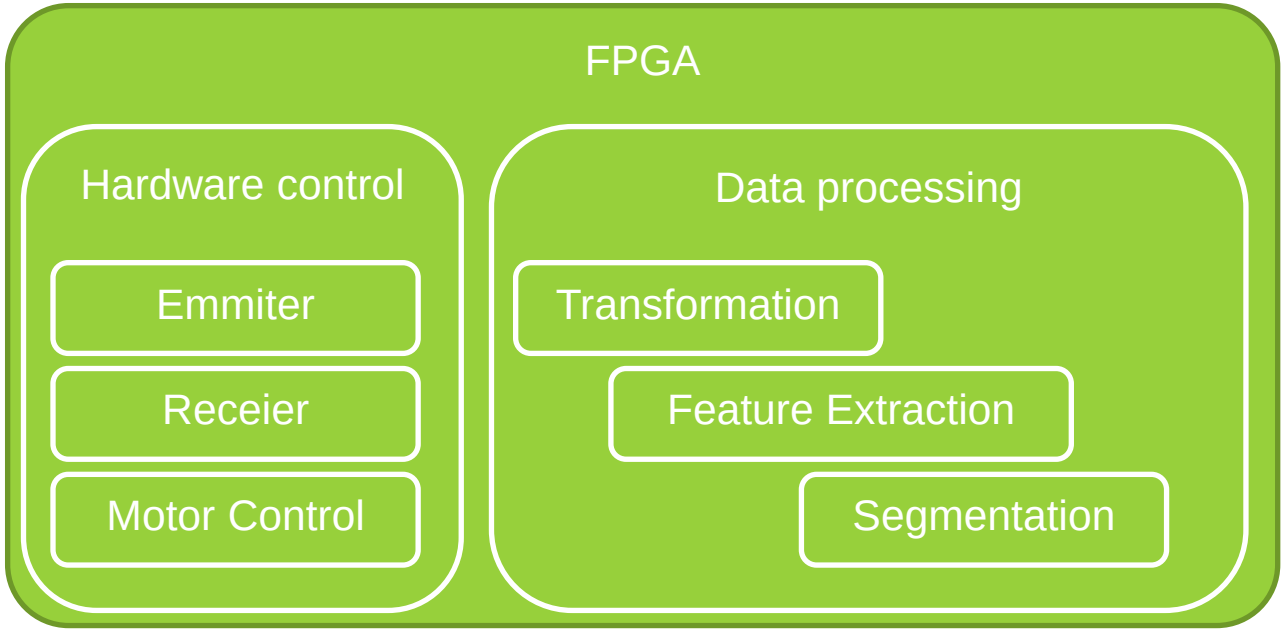
- Cluster connected points
  - Region growing algorithm



WHAT IS AN FPGA AND WHY DO WE USE IT?

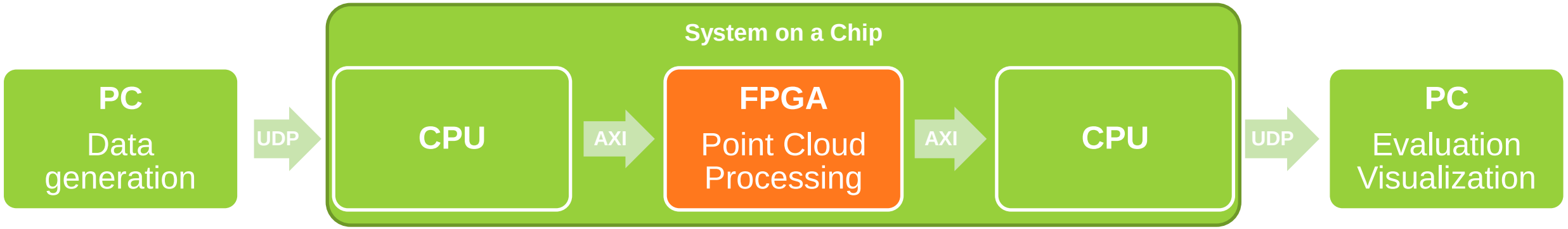
Field Progammable Gate Array

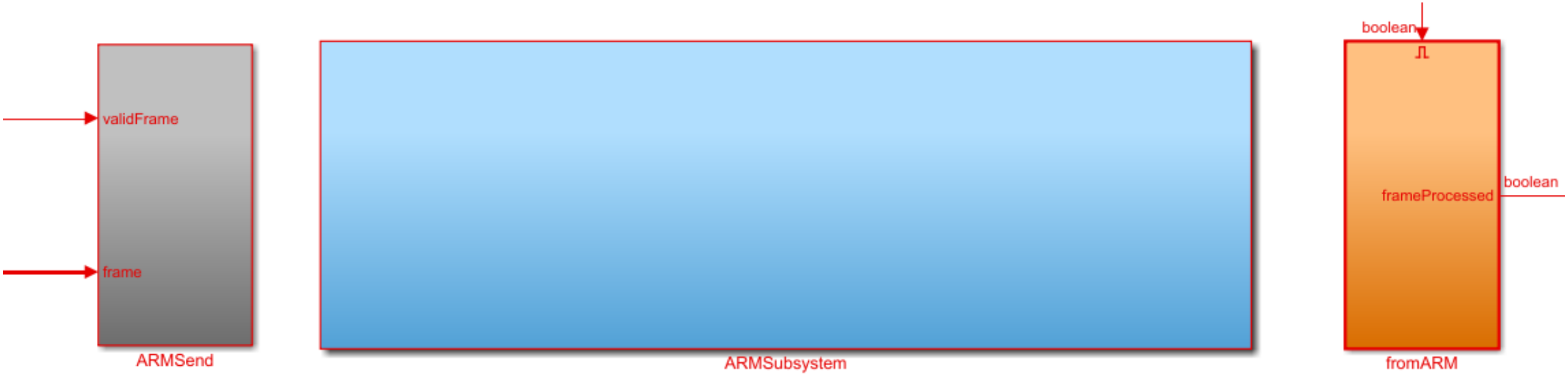
- Reconfigurable Integrated Circuit
  - Suitable for new projects
  - Implementation of bug fixes
- Parallel processing of
  - Algorithms for Hardware control
  - Each step in the data processing pipeline



POINT CLOUD PROCESSING ON FPGA

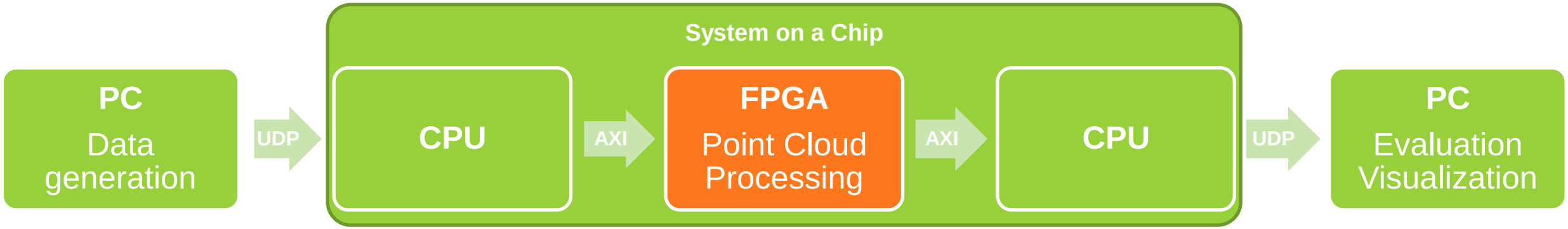
Possible data flow using Embedded-Coder and HDL-Coder

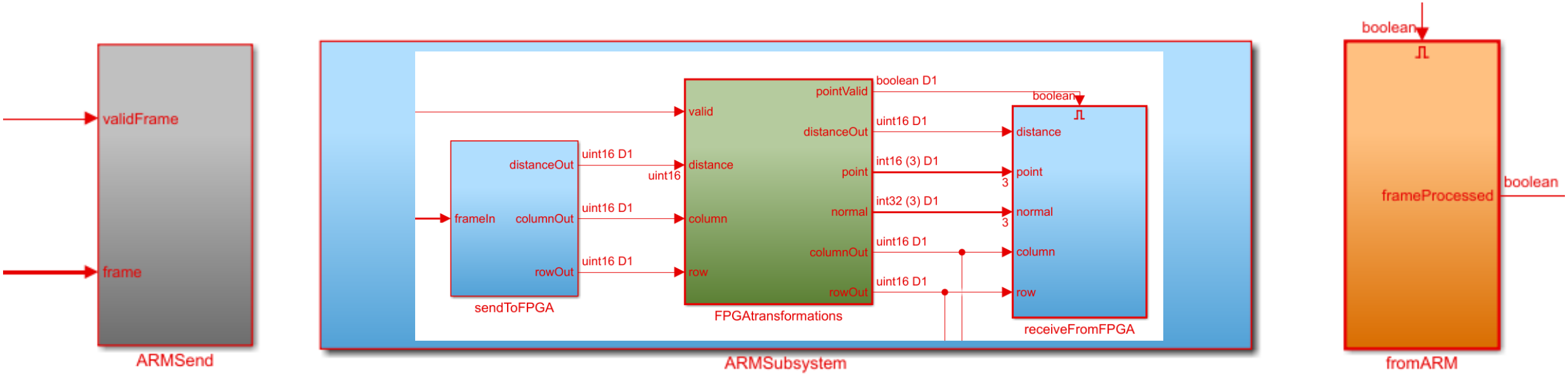




POINT CLOUD PROCESSING ON FPGA

Possible data flow using Embedded-Coder and HDL-Coder

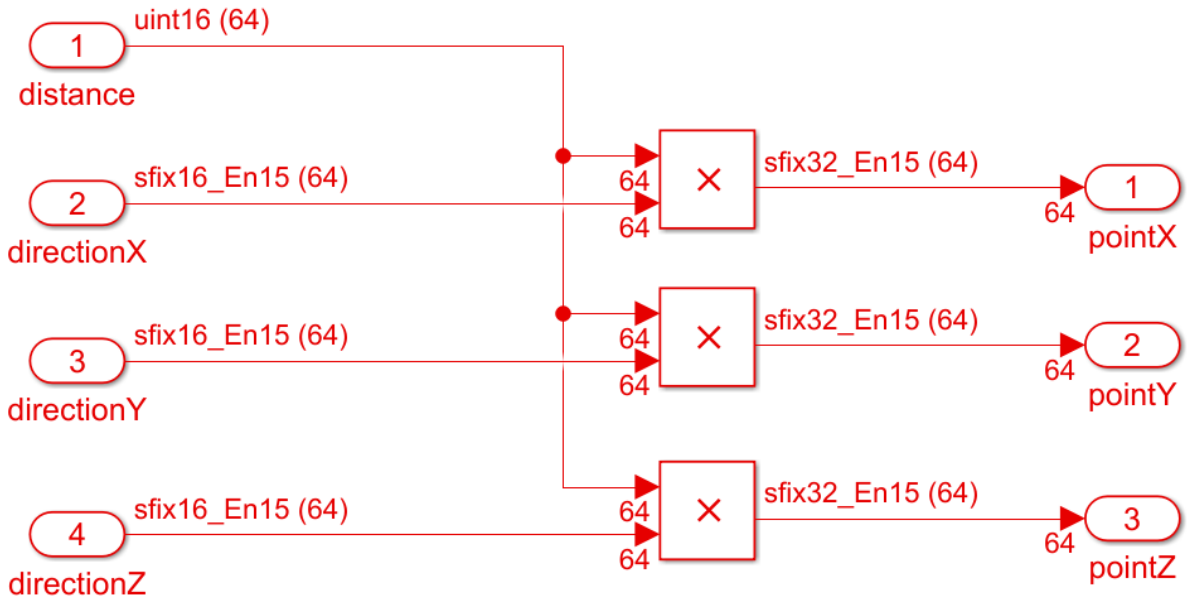




POINT TRANSFORMATION

Example:

- Column Wise data received from Sensor (64 scan points)
- Transformation multipliers from lookup table available



Summary

Multipliers	192
Adders/Subtractors	0
Registers	2
Total 1-Bit Registers	2
RAMs	0
Multiplexers	0
I/O Bits	7174
Static Shift operators	0
Dynamic Shift operators	0

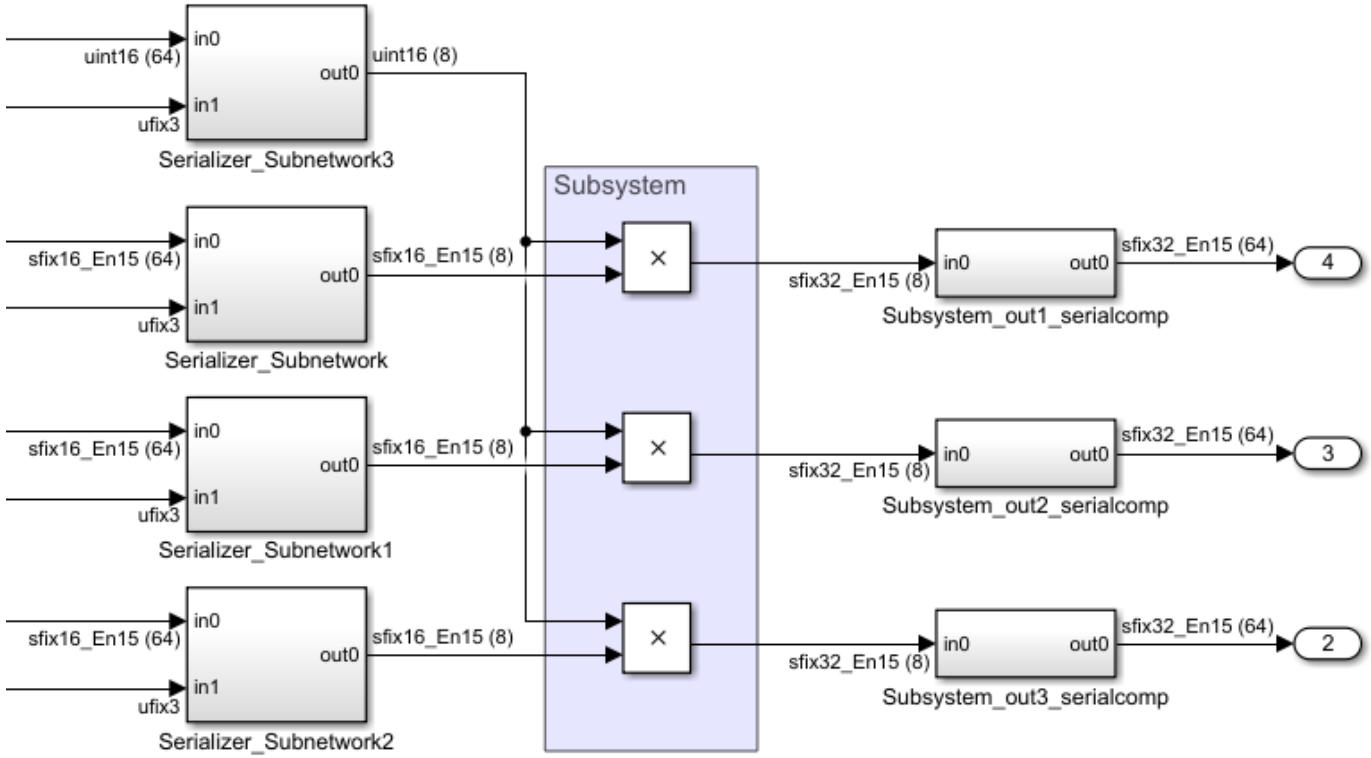
Critical Path Details

Id	Propagation (ns)	Delay (ns)	Block Path
1	3.9260	3.9260	Product2

POINT TRANSFORMATION

Streaming inserted

SharingFactor	0
StreamingFactor	8



Summary

Multipliers	24
Adders/Subtractors	4
Registers	385
Total 1-Bit Registers	11565
RAMs	0
Multiplexers	44
I/O Bits	7174
Static Shift operators	0
Dynamic Shift operators	0

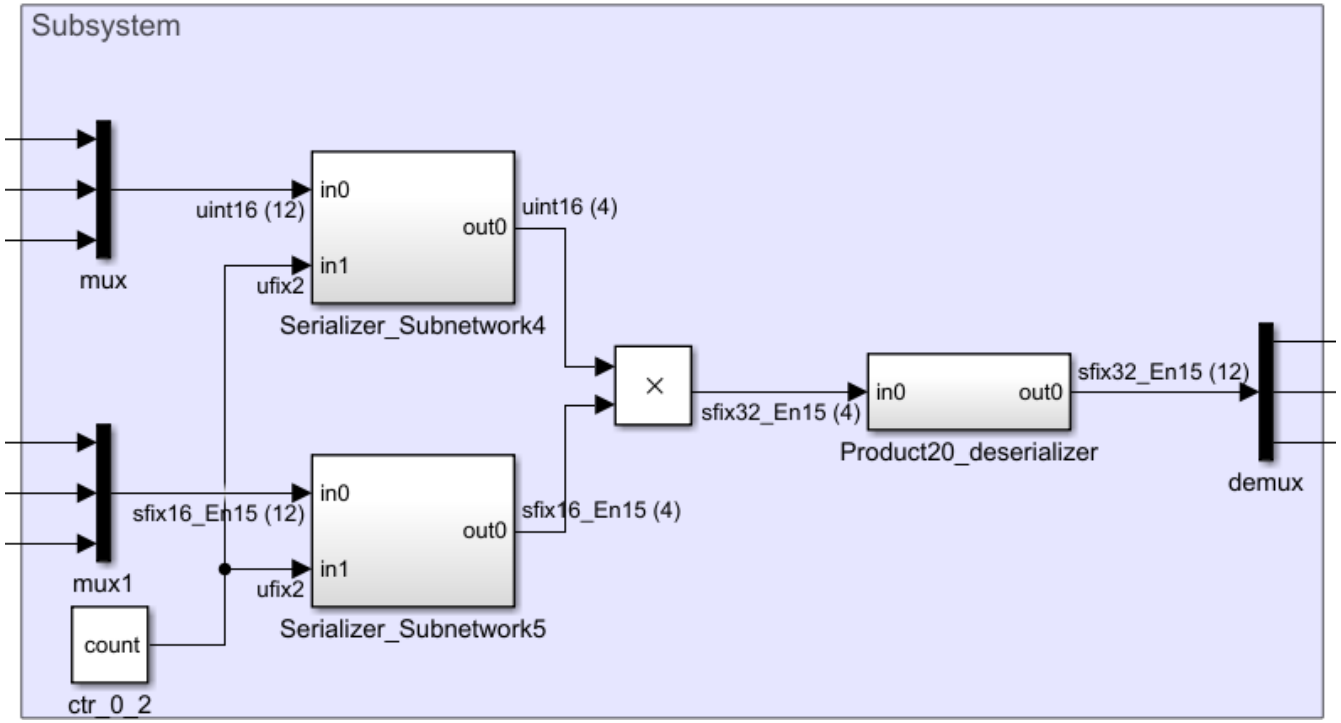
Critical Path Details

Id	Propagation (ns)	Delay (ns)	Block Path
1	0.4740	0.4740	*ctr_0_7*
2	1.3365	0.8625	*splitcomp_multiport*
3	5.2625	3.9260	*Product_out1_serialcomp_in_buff*
4	5.4320	0.1695	*Product_out1_serialcomp*

POINT TRANSFORMATION

Streaming and Sharing inserted

SharingFactor	3
StreamingFactor	16



Summary

Multipliers	4
Adders/Subtractors	6
Registers	618
Total 1-Bit Registers	18381
RAMs	0
Multiplexers	41
I/O Bits	7174
Static Shift operators	0
Dynamic Shift operators	0

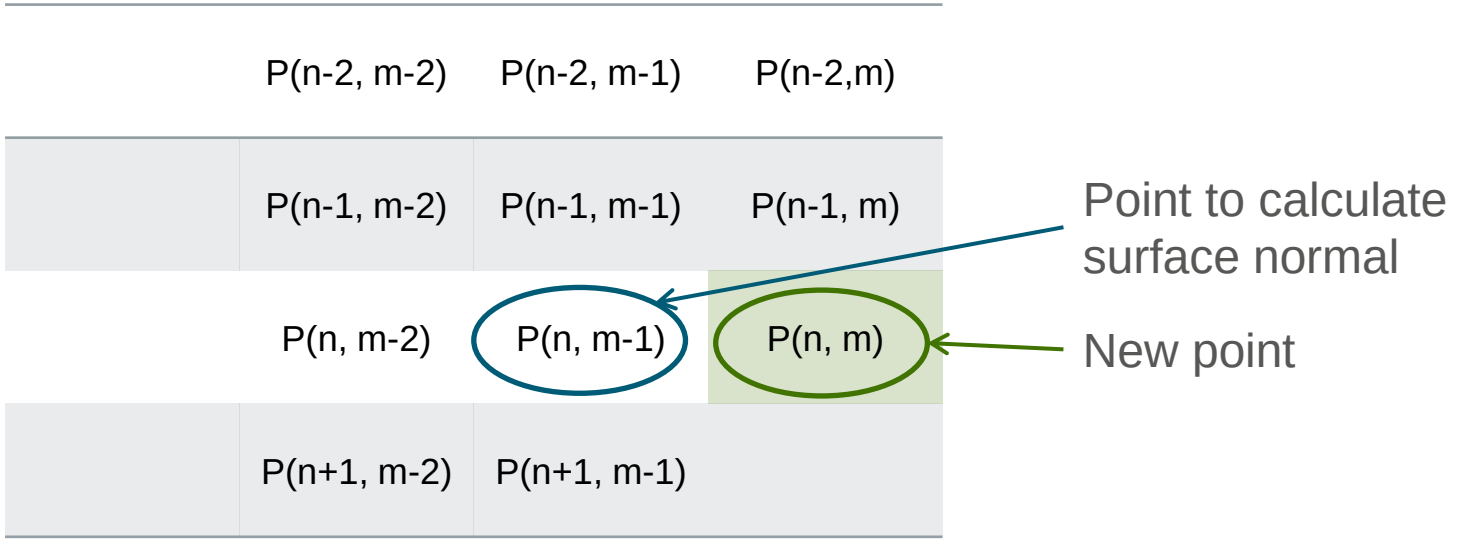
Critical Path Details

Id	Propagation (ns)	Delay (ns)	Block Path
1	0.4740	0.4740	ctr_0_15
2	1.1465	0.6725	splitcomp_multiport
3	1.1465	0.0000	mux
4	1.1465	0.0000	ratechange_splitcomp
5	1.8245	0.6780	splitcomp_multiport
6	5.7505	3.9260	Product20_deserializer_in_buff
7	5.9200	0.1695	Product20_deserializer

NORMAL ESTIMATION

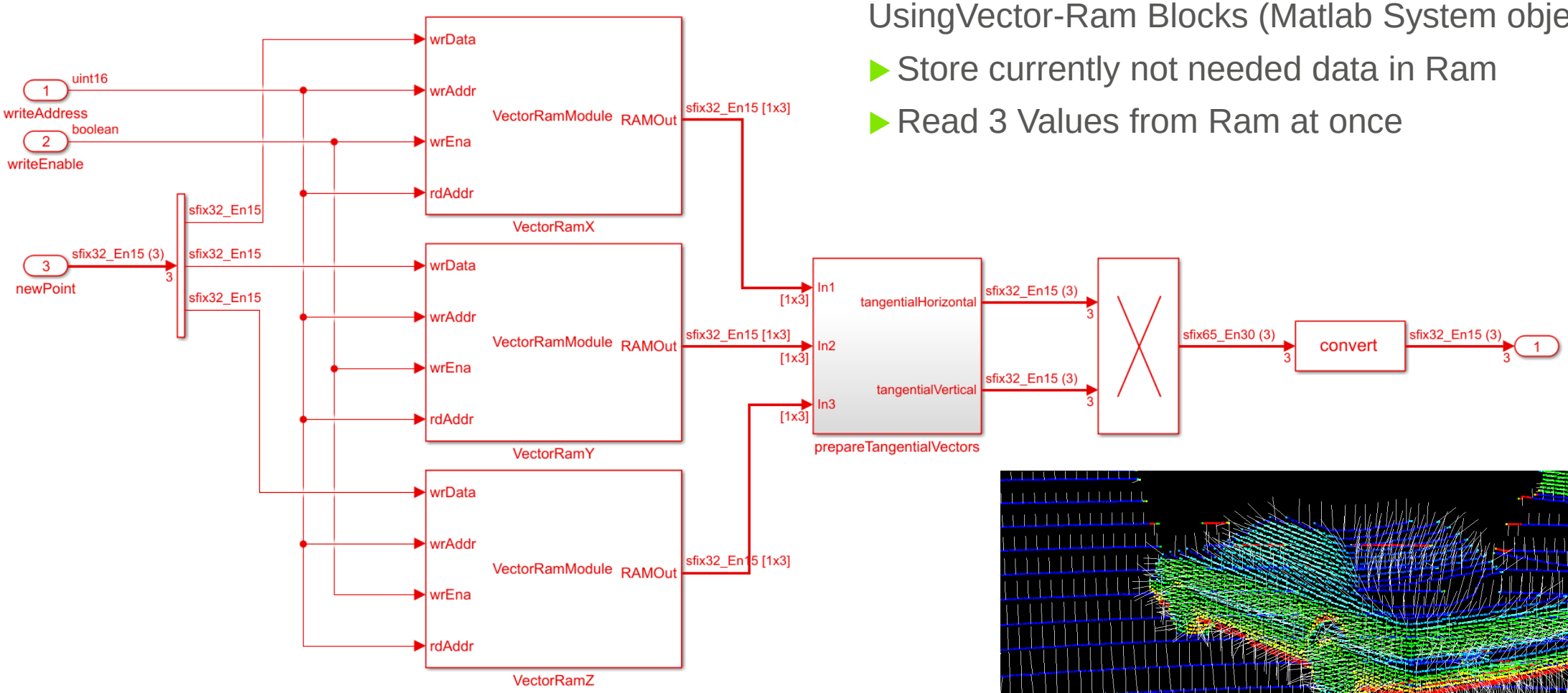
Example: Simple normal estimation

- Assumptions
  - Organized Point Cloud
  - Data processed point wise in increasing manner (column and row wise)



$$N(n, m-1) = \underbrace{[P(n,m) - P(n,m-2)]}_{\text{horizontal tangent}} \times \underbrace{[P(n+1,m-1) - P(n-1,m-1)]}_{\text{vertical tangent}}$$

NORMAL ESTIMATION



UsingVector-Ram Blocks (Matlab System object)

- Store currently not needed data in Ram
- Read 3 Values from Ram at once

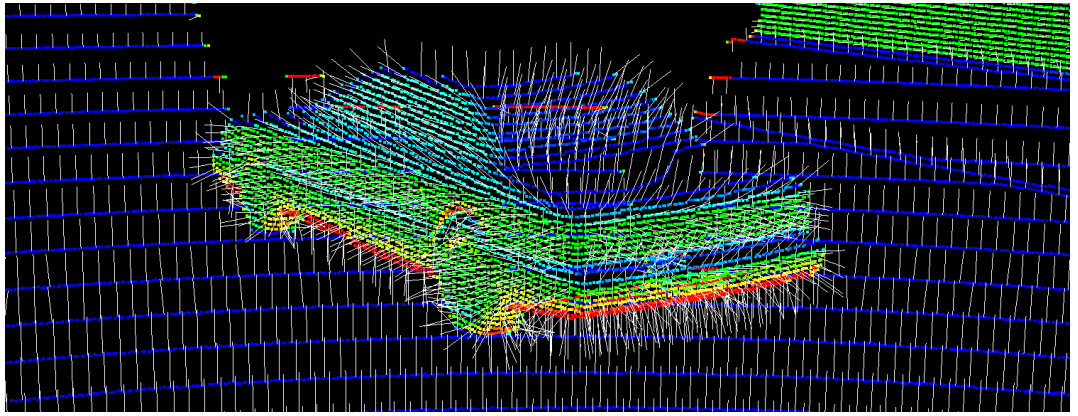
SUMMARY

HDL Coder

- Algorithm development for FPGA without VHDL coding
- Run as HiL-Setup directly
- Comparable to Simulink implementation

Challenges

- Algorithm selection
  - What can be implemented?
  - How to adopt algorithms?




Valeo
SMART TECHNOLOGY
FOR SMARTER CARS