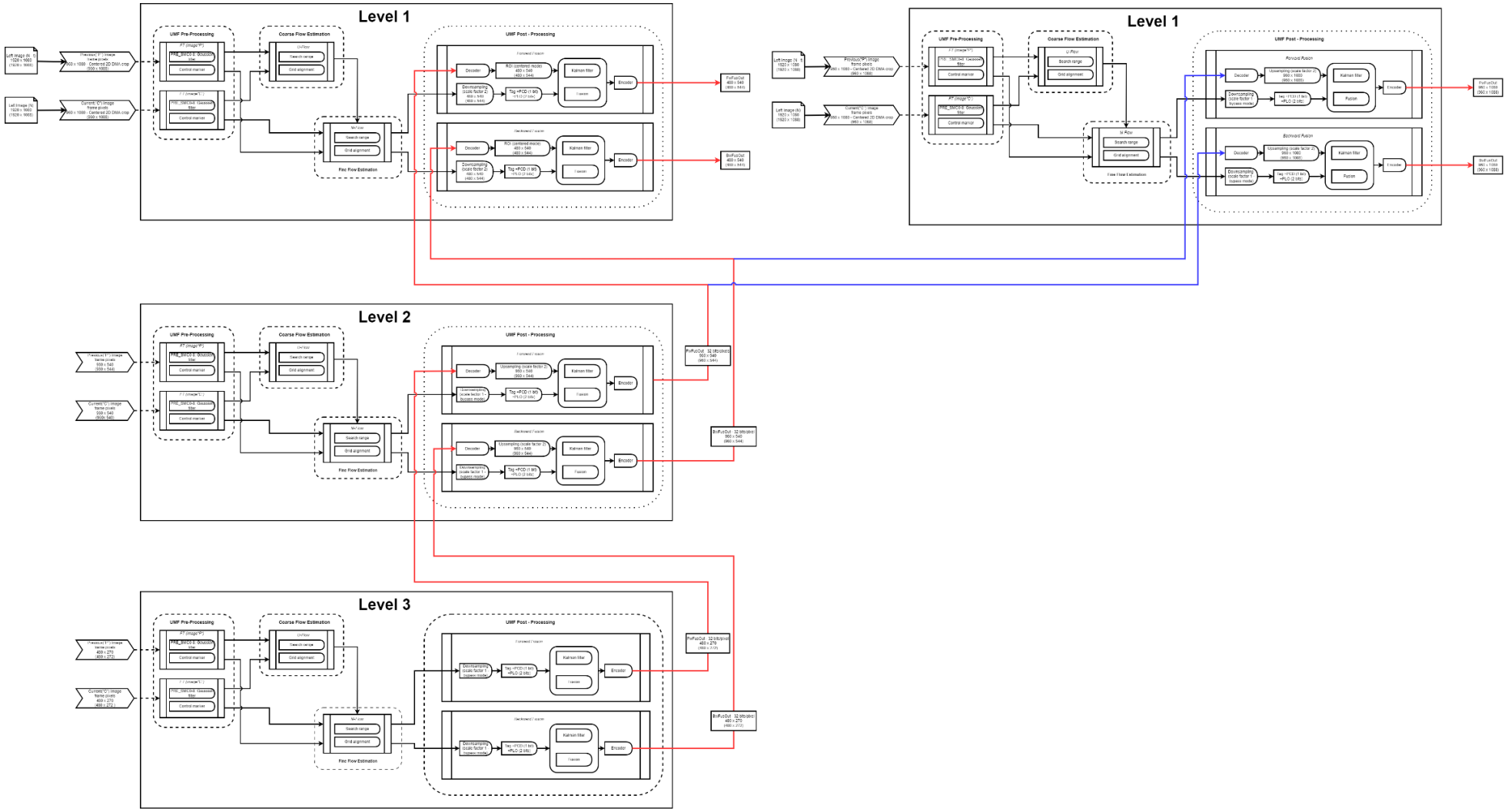
PYRAMID LEVEL

SUBLINE

DATE 09/13/2022
NAME: NGUYEN LE KHAC ANH
NGUYEN QUANG HUNG
R-CAR VISION 2, SOC SW,
VISION IP
RENESAS ELECTRONICS CORPORATION

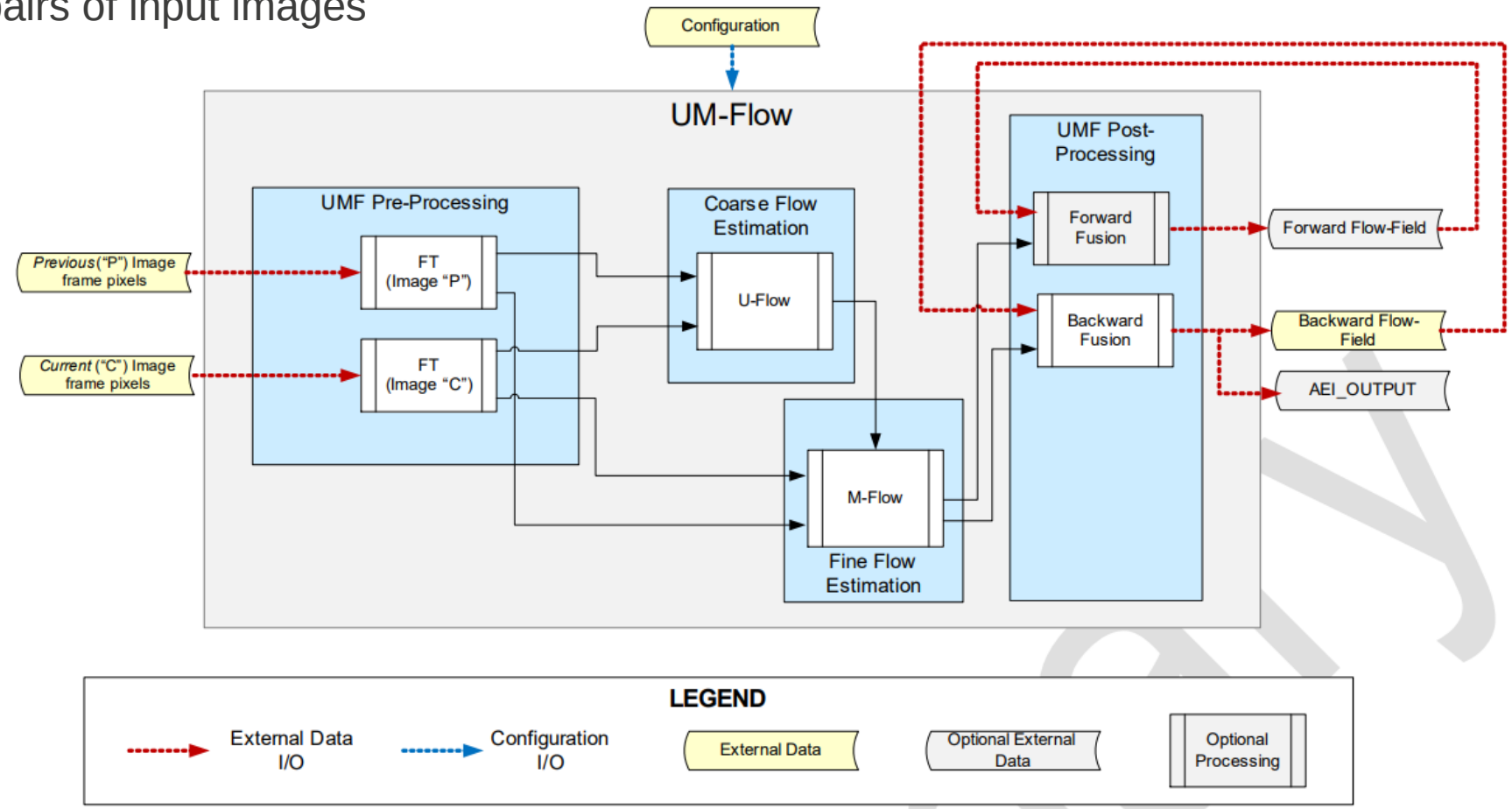
AGENDA

LEVEL PYRAMID DIAGRAM



OVERVIEW

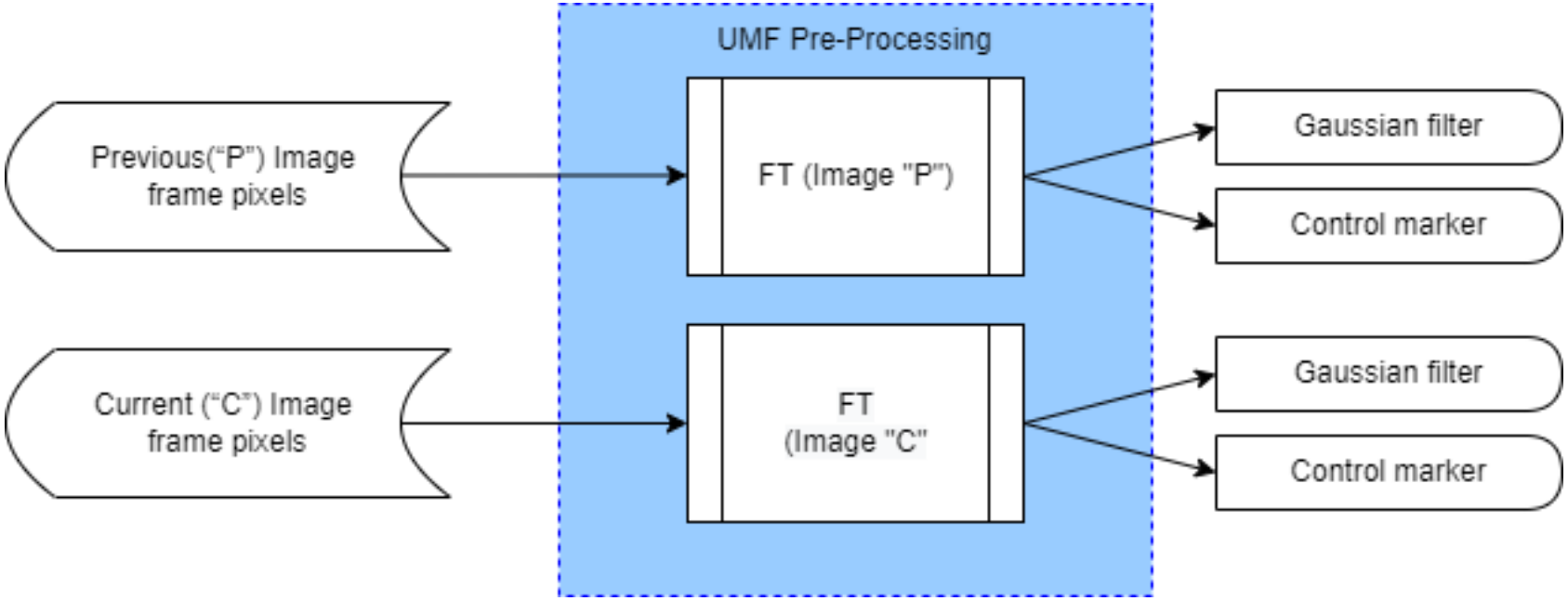
The UMF_CORE is designed in order to sequentially support optical flow estimation on different resolution levels of the same pairs of input images



FRAME TRANSFORMATION

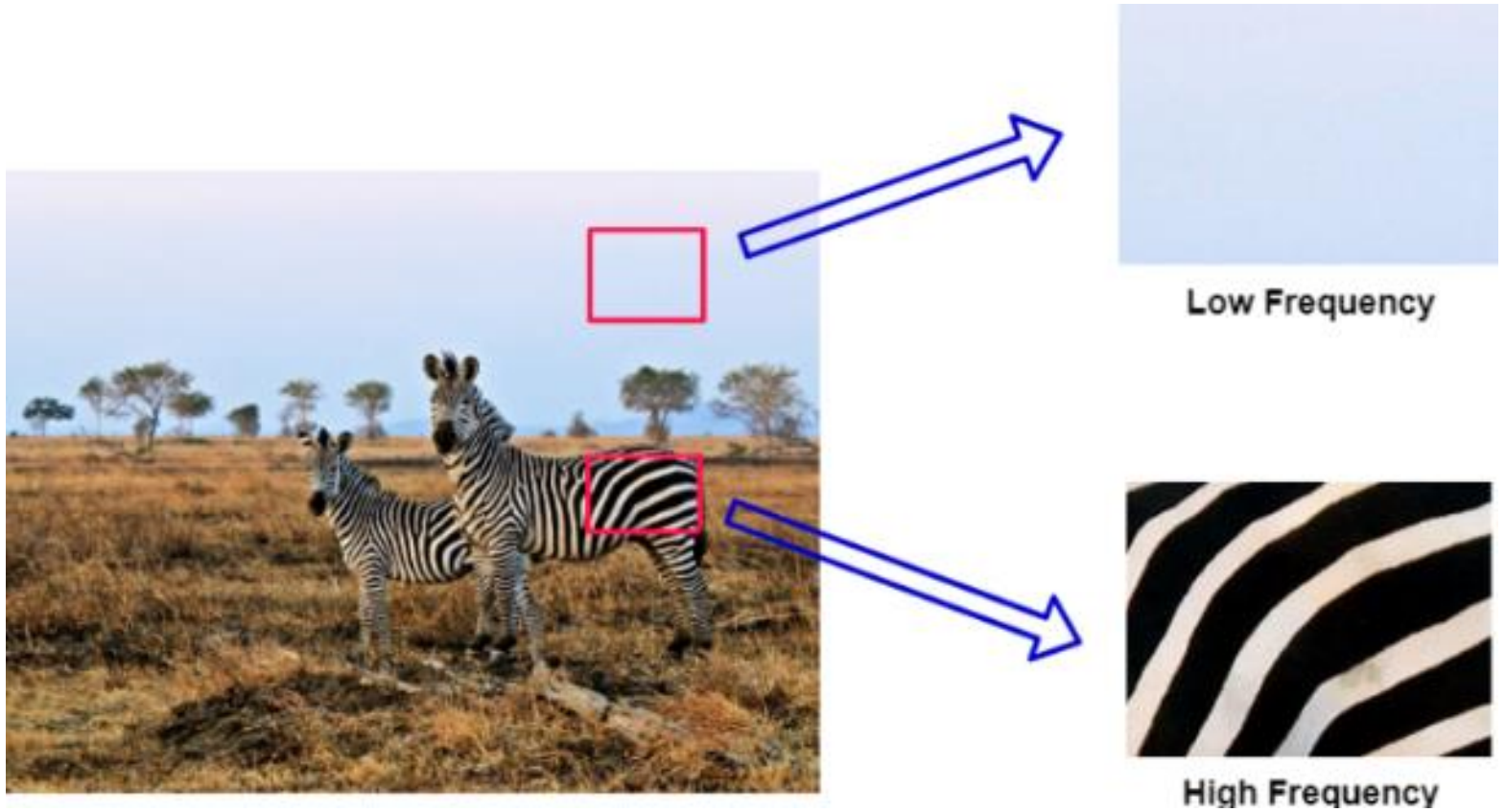
Frame transformation (FT) contains two identical blocks: FT_P is dedicated to process the previous image frame, and FT_C is dedicated to process the current image

Feature: **Smooth filtering of gray images**



FRAME TRANSFORMATION

A Gaussian Filter is a low-pass filter used for reducing noise (high-frequency components) and blurring regions of an image.



FRAME TRANSFORMATION

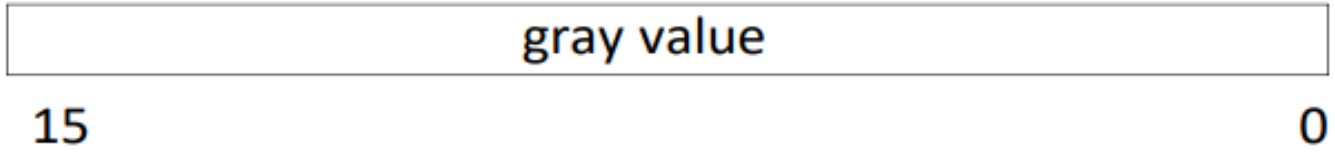
A Gaussian Filter is a low-pass filter used for reducing noise (high-frequency components) and blurring regions of an image.

Original

Smooth

THE GAUSSIAN FILTER

A 3x3 kernel filter with programmable coefficients is applied to the input gray image data



The input image consists of maximum 16 bits per pixel. In case of 12 bits/pixel, the 4 MSBs are unused. The programmable values of the filter coefficients PRE_SMC0-8 are 0, 1, 2, 4 and 8

Recommended coefficients values are

[0 0 0 0 1 0 0 0 0] : no smoothing

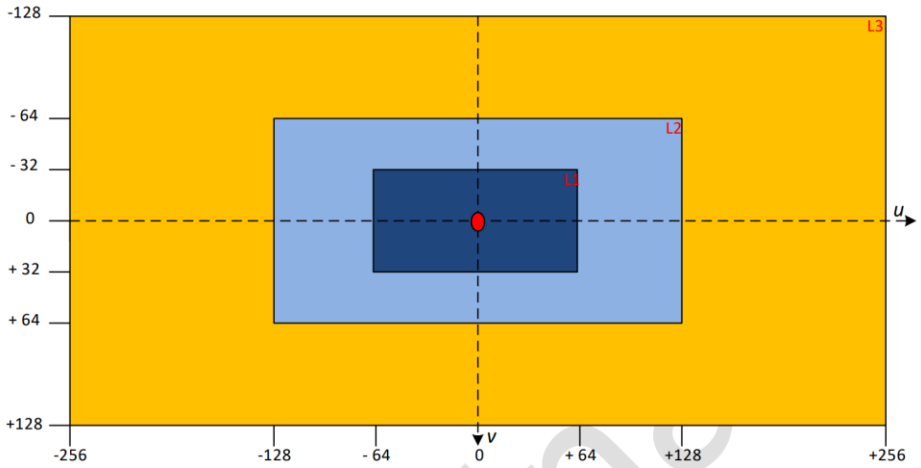
[1 2 1 2 4 2 1 2 1] : Gaussian filter

[1 1 1 1 1 1 1 1 1] : Box filter

3x3 kernel filter:

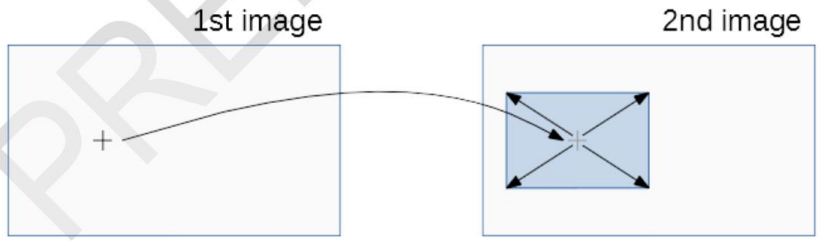
1	2	1
2	4	2
1	2	1

COARSE FLOW ESTIMATION – FINE FLOW ESTIMATION

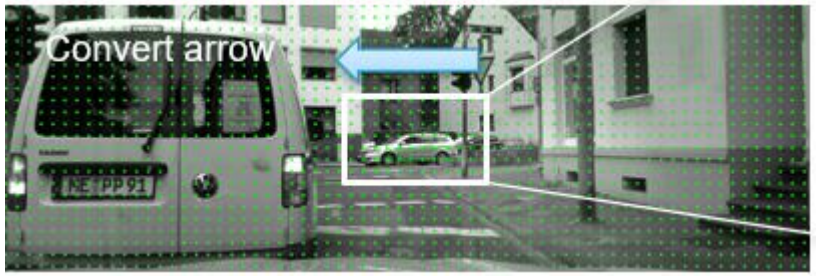


(u, v) = (0,0)

[SRL,SRR,SRU,SRD,Vshift] = [64,64,31,31,0]



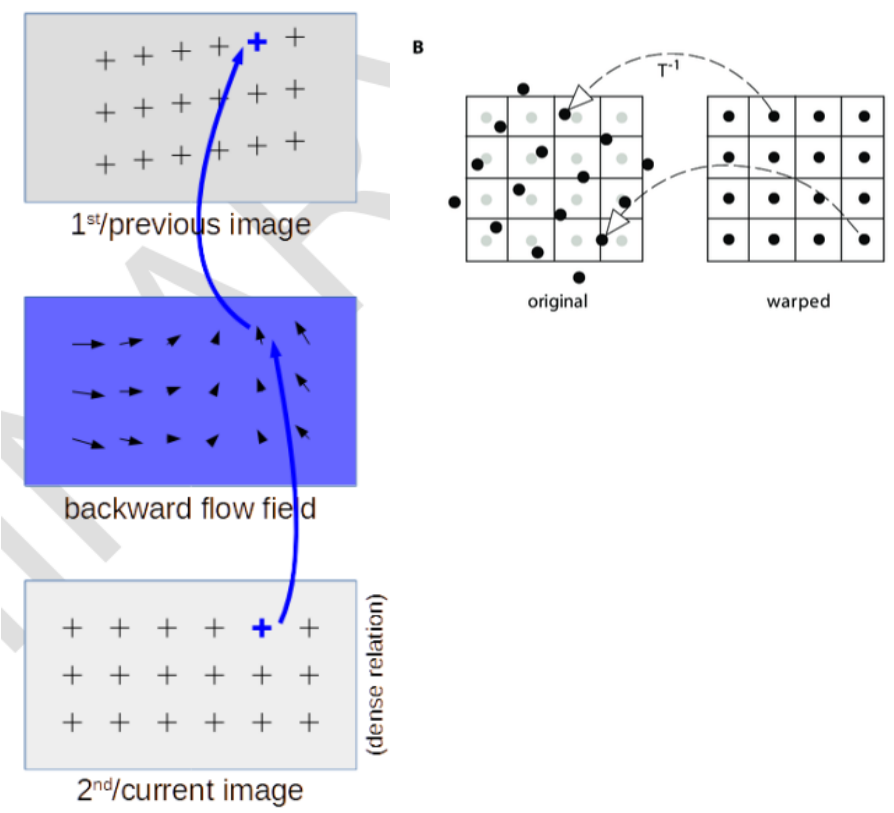
Grid alignment



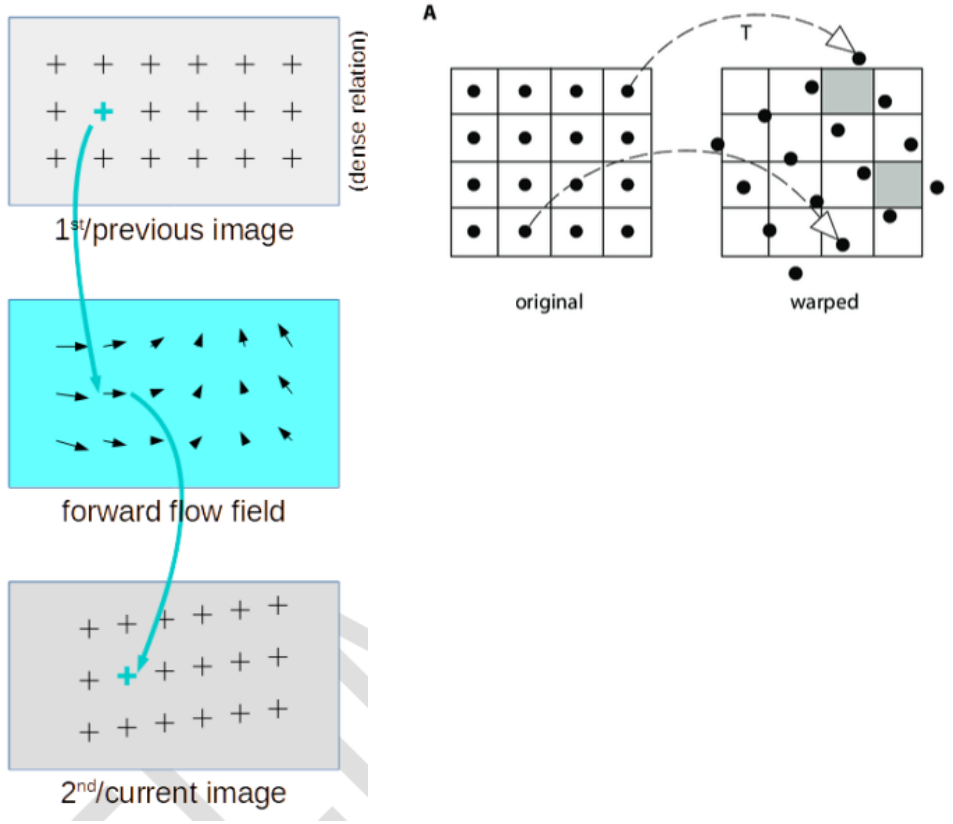
Seach range

BACKWARD & FORWARD FLOW FIELD

Backward flow field



Forward flow field



DOWN-SAMPLING AND UP-SAMPLING

The Gaussian Pyramid:

A representation in multiple scales

The goal is to define a representation in which image information at different scales is explicitly available

DOWN-SAMPLING AND UP-SAMPLING

Down-sampling and Up-sampling in DOF use the Gaussian pyramid method

Down-sampling

Take every 2nd pixel from I for g_1

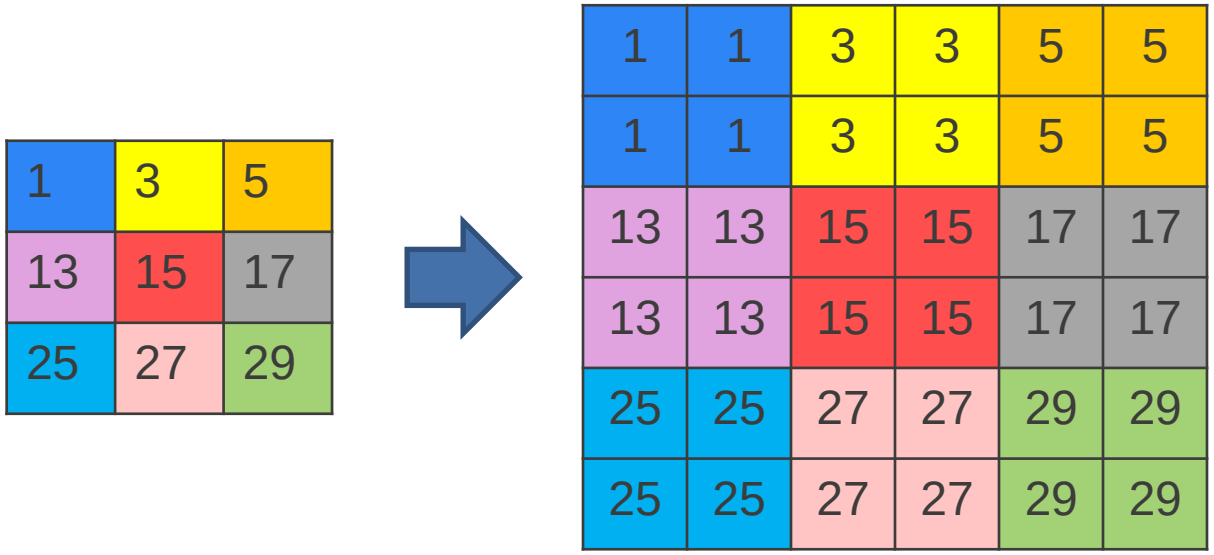
1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

1	3	5
13	15	17
25	27	29

DOWN-SAMPLING AND UP-SAMPLING

Down-sampling and Up-sampling in DOF use the Gaussian pyramid method

Up-sampling



DECODER – TAG (PCD + PLO) - ENCODER

Input data formats

Field name	Bit field	Description
Previous / Current grey image input format		
gii	15:0	grey scale pixel. If < 16 bit, then right aligned.
Backward / Forward flow input format		
u	31:22	horizontal motion component (format: sint10)
v	21:12	vertical motion component (format: sint10)
FLE	11:10	*floEncoding* information Gives the number of less significant bits of *u* and *v* used for sub-pixel data. Note that the same encoding applies to both *u* and *v*. **0**: 4 bit for sub-pixel, format: SQ5.4, range (-32….+31.9375), accuracy: ε ≤ 1/32 **1**: 3 bit for sub-pixel, format: SQ6.3, range (-64….+63.875), accuracy: ε ≤ 1/16 **2**: 2 bit for sub-pixel, format: SQ7.2, range (-128….+127.75), accuracy: ε ≤ 1/8 **3**: 1 bit for sub-pixel, format: SQ8.1, range (-256….+255.5), accuracy: ε ≤ 1/4
-	9:3	Reserved
PCD	2	*pyraConfirmed* information The output result (originating from level *pyraLevelOrigin)* is confirmed by the next level (*pyraLevelOrigin+1)* **0**: conflicting result or no result on next level **1**: confirmed result (best case)
PLO	1:0	*pyraLevelOrigin* information Index of the pyramid level (if any) where this flow vector was computed (highest resolution = smallest number wins) **0**: invalid (no flow result) **1**: originates from pyramid level 1 (best case) **2**: originates from pyramid level 2 (medium resolution) **3**: originates from pyramid level 3 (lowest resolution)

Output data formats

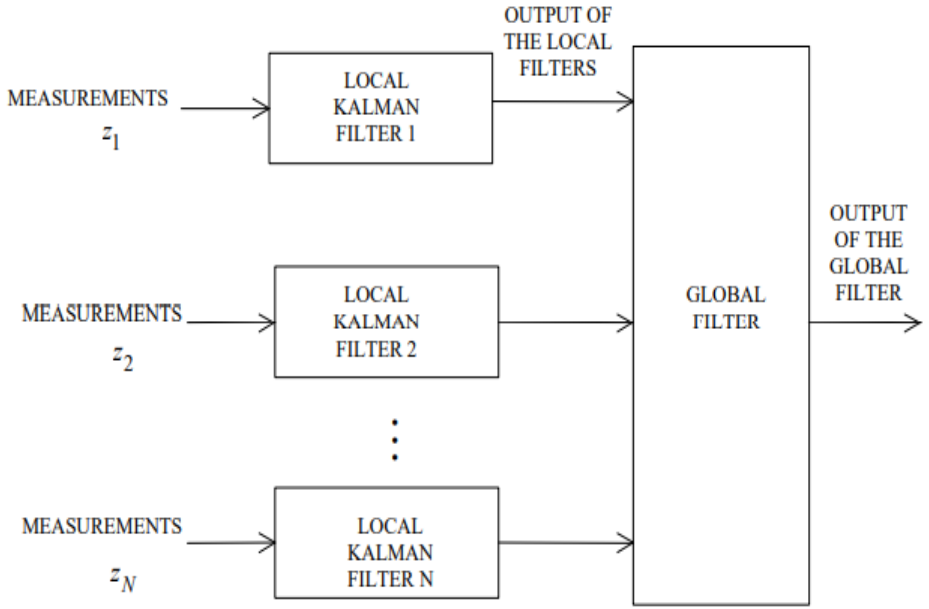
Field name	Bit field	Description
Backward flow / forward Flow / PS flow output format		
u	31:22	horizontal motion component (sint10)
v	21:12	vertical motion component (sint10)
FLE	11 :10	*flowEncoding* information Gives the number of less significant bits of u and v used for sub-pixel data. Note that the same encoding applies to both u and v **0**: 4 bit for sub-pixel, format: SQ5.4, range: (-32….+31.9375), accuracy: ε ≤ 1/32 **1**: 3 bit for sub-pixel, format: SQ6.3, range: (-64….+63.875), accuracy: ε ≤ 1/16 **2**: 2 bit for sub-pixel, format: SQ7.2, range: (-128….+127.75), accuracy: ε ≤ 1/8 **3**: 1 bit for sub-pixel, format: SQ8.1, range: (-256….+255.5), accuracy: ε ≤ 1/4
-	9:3	Reserved
PCD	2	*pyraConfirmed* information The output result (originating from level *pyraLevelOrigin) is confirmed by the next level (pyraLevelOrigin+1)* **0**: conflicting result or no result on next level **1**: confirmed result (best case)
PLO	1:0	*pyraLevelOrigin* information Index of the pyramid level (if any) where this flow vector was computed (highest resolution = smallest number wins) **0**: invalid (no flow result) **1**: originates from pyramid level 1 (best case) **2**: originates from pyramid level 2 (medium resolution) **3**: originates from pyramid level 3 (lowest resolution)

- Decoder: The FUSION Decoder module decompresses, 10 bits each for U and V components, 2 bits for the flow encoding (FLE) , 2 bits for pyramid level (PLO) and 1 bits for quality measure (PCD) by the Low Priority. The decompressed values are derived depending on the value of FLE by shifting both values by FLE bits to the left, the two FLE bits are then removed from the output.
- Encoder: the pixel data is encoded back to the 32-bit compressed format used at the Path A input interface.

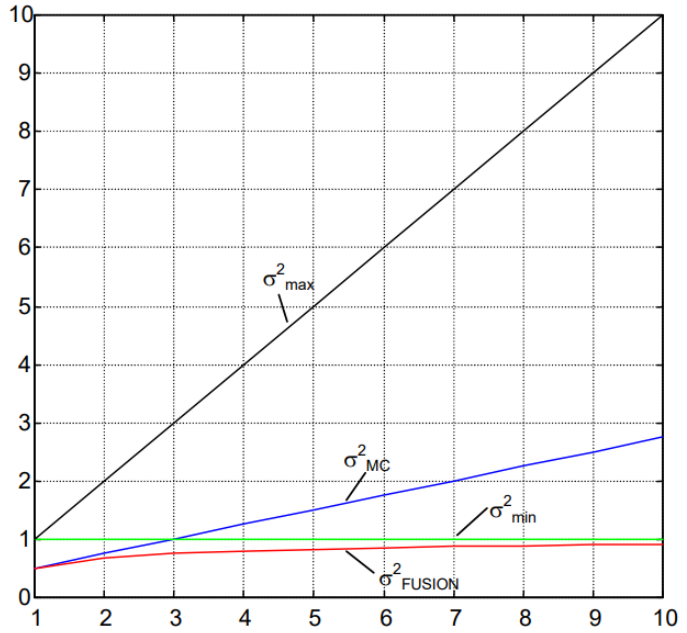
FUSION

The FUSION module gets 2 flow-field estimate frames, and combines them into one resulting flow-field estimate frame, according to pixel based criteria, i.e., the output value is based locally only on the respective input values, not on their surrounding estimate values

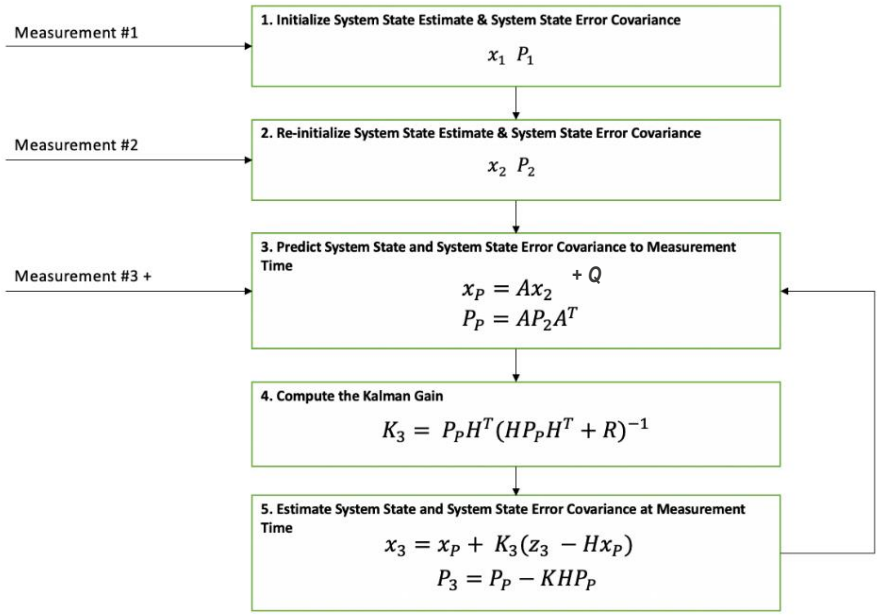
Evolution of the different variances



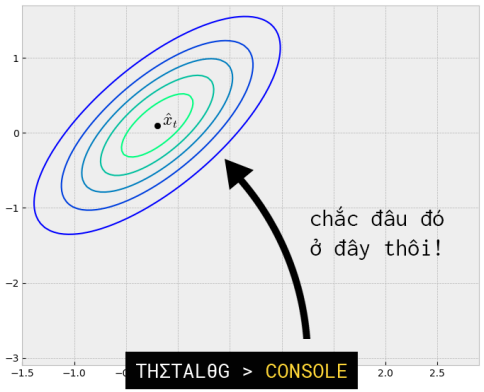
Decentralized Kalman filter

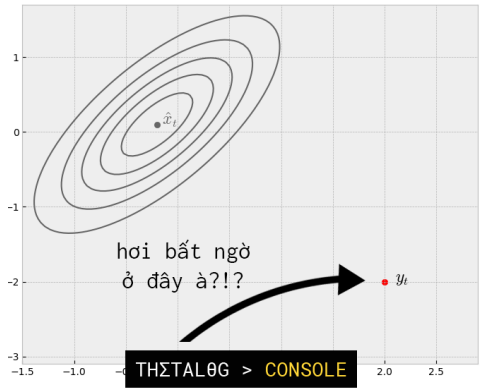


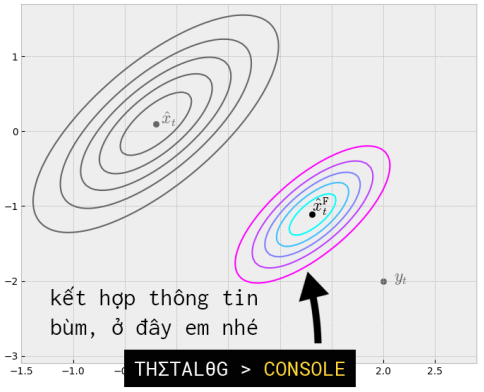
KALMAN FILTER

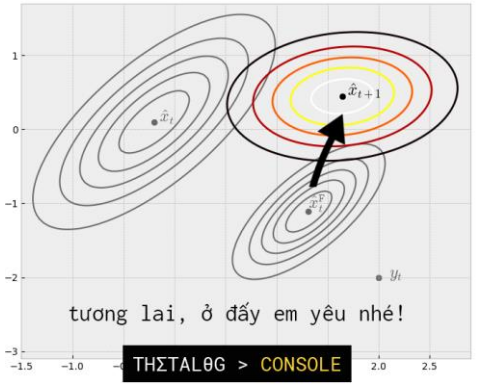


x	state variable	n x 1 column vector	Output
P	state covariance matrix	n x n matrix	Output
z	measurement	m x 1 column vector	Input
A	state transition matrix	n x n matrix	System Model
H	state-to-measurement matrix	m x n matrix	System Model
R	measurement covariance matrix	m x m matrix	Input
Q	process noise covariance matrix	n x n matrix	System Model
K	Kalman Gain	n x m	Internal



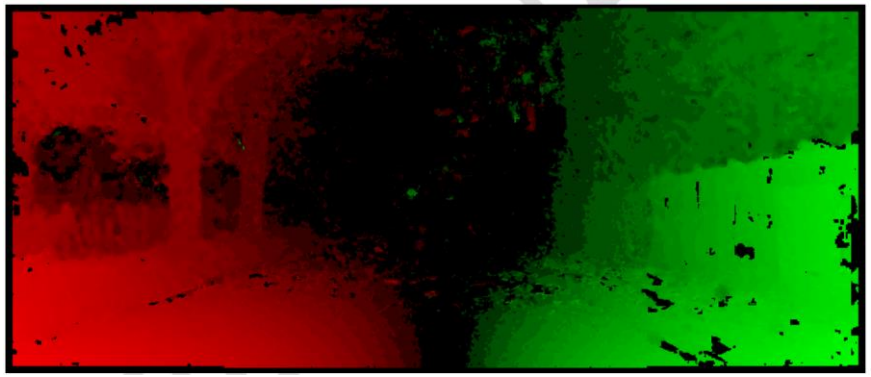




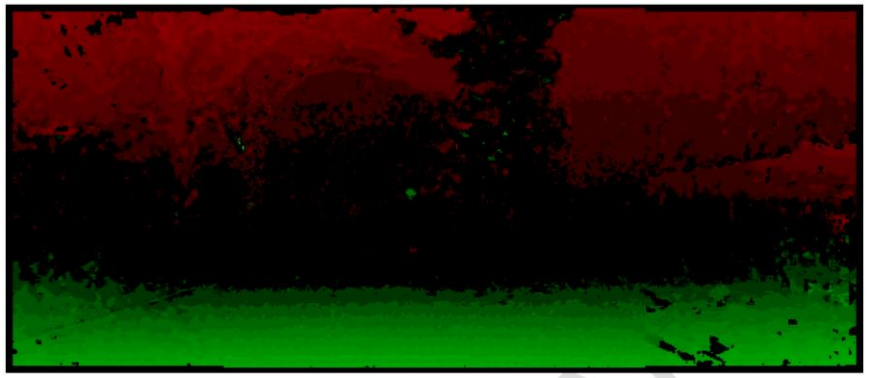


DENSE OPTICAL FLOW OUTPUT

Flow field horizontal



Flow field vertical



- Real world scenario

- Single component flow fields

Red: negative component values with the brightness referring to the magnitude of the component

Green: positive component values with the brightness referring to the magnitude of the component

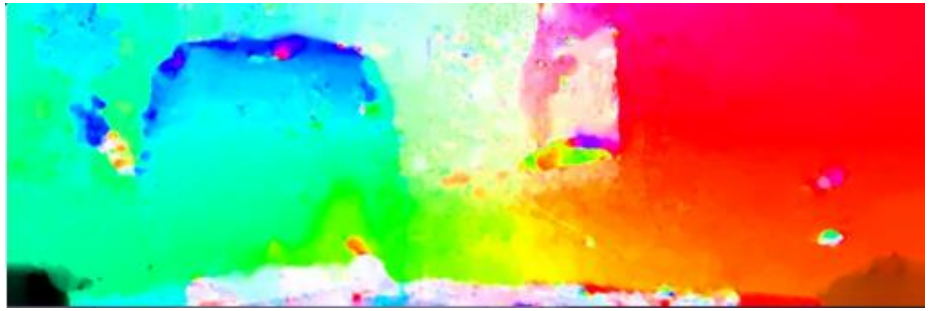
DENSE FLOW MAP RESULT

Level maps(PLO)

- black – flow vector is invalid, no result exists
- white – flow vector from L3
- light gray – flow vector from L2
- dark gray – flow vector from L1

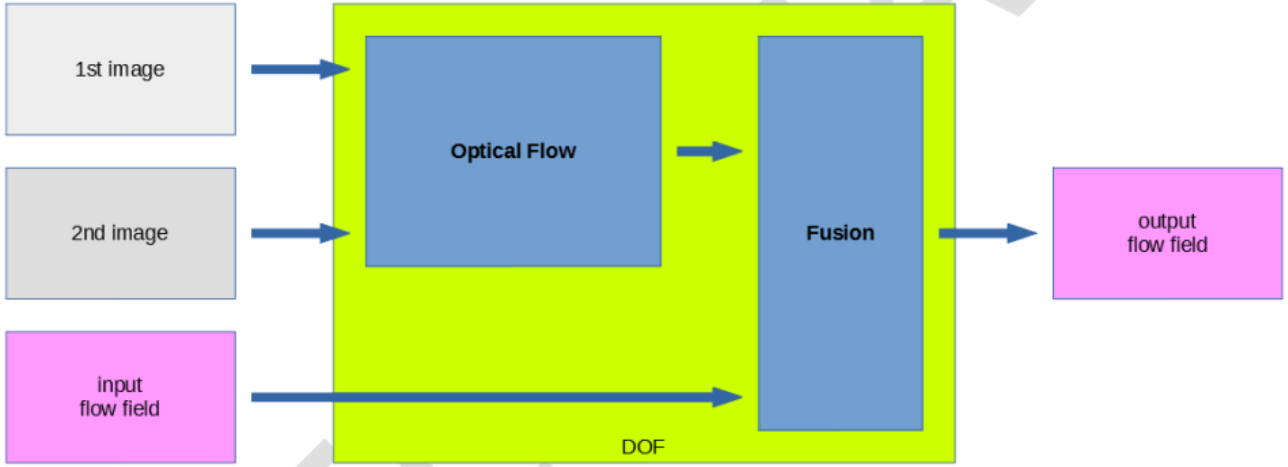
Confirmation maps(PCD)

- black – flow vector is invalid (after fusion)
- yellow – flow vector at Ln is valid but not at Ln+1 or vector at Ln does not correspond to vector at Ln+1
- green – flow vector at Ln and Ln+1 are valid and correspond to each other

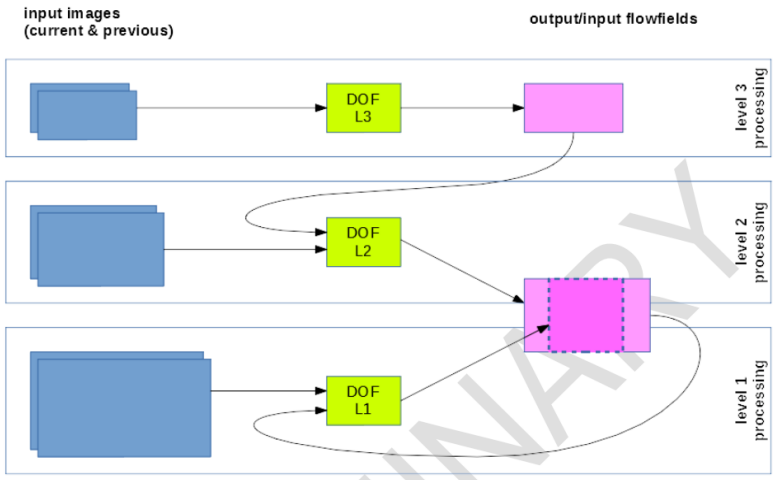


SUMMARY

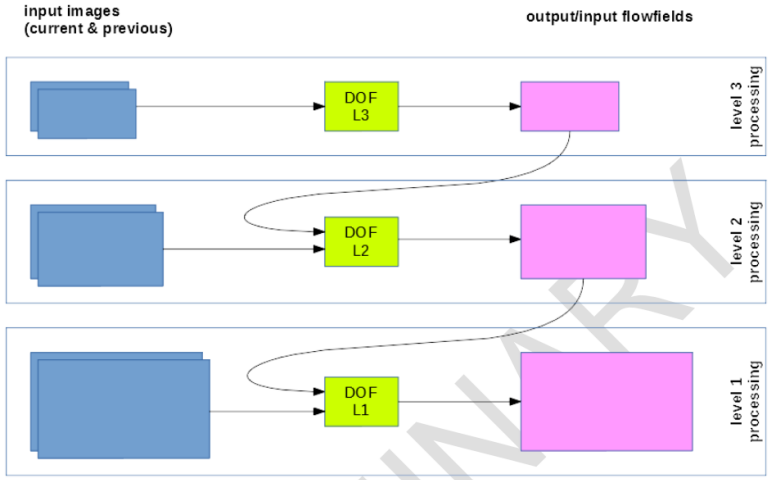
Optical flow and fusion



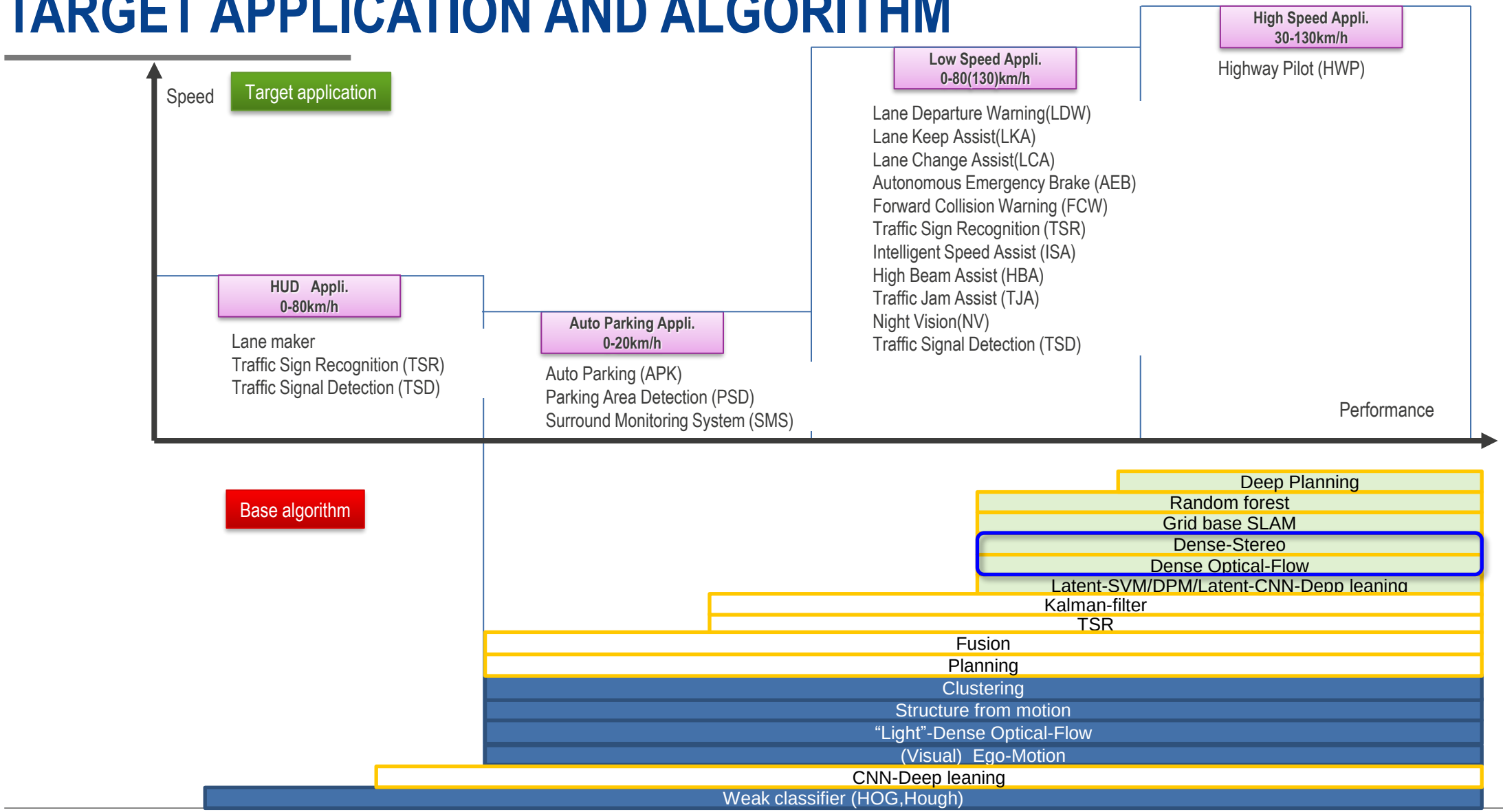
No dedicated L1 flow field buffer, with L1 to L2 downscaling



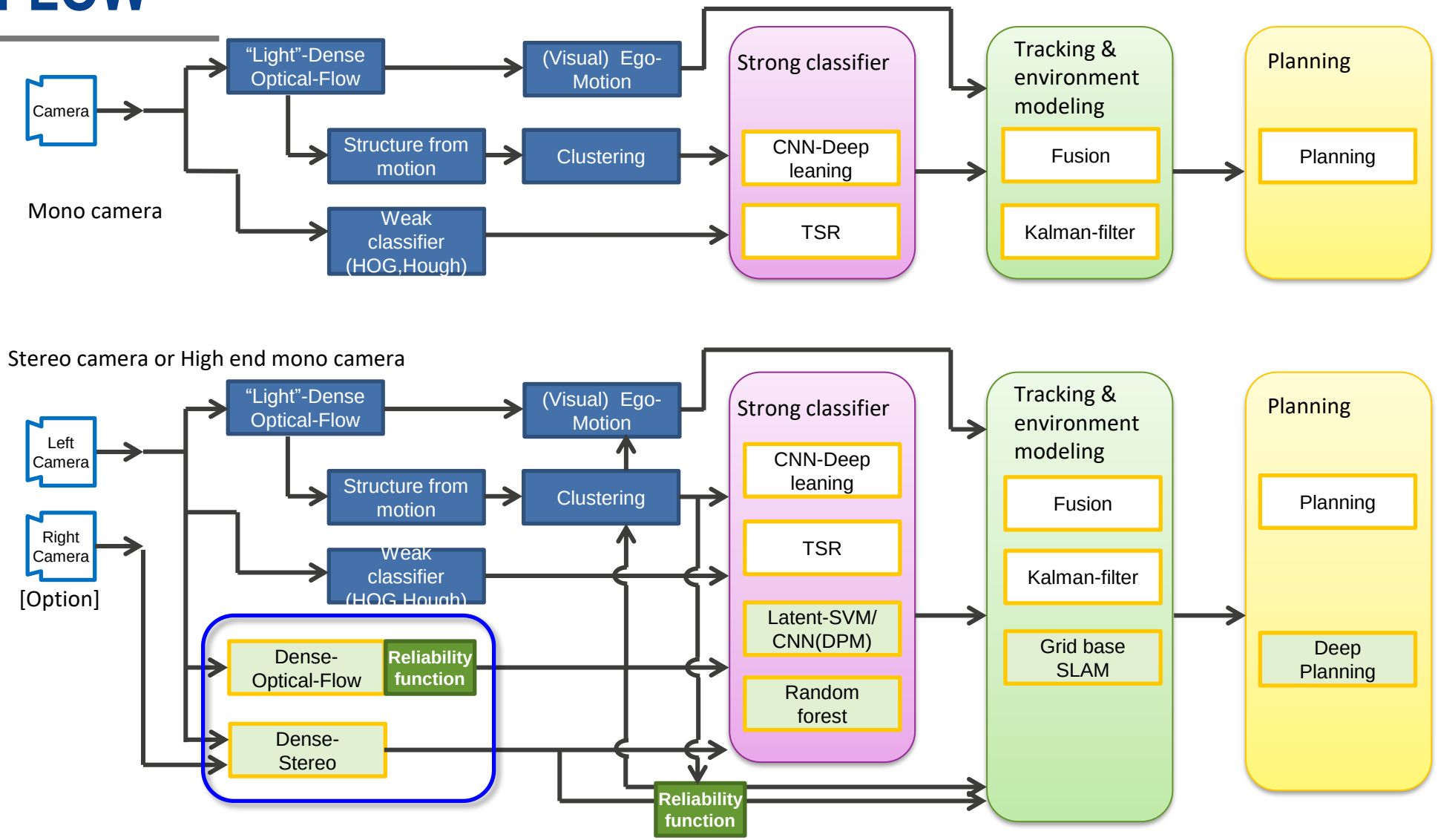
Separate buffers, no L1 to L2 downscaling



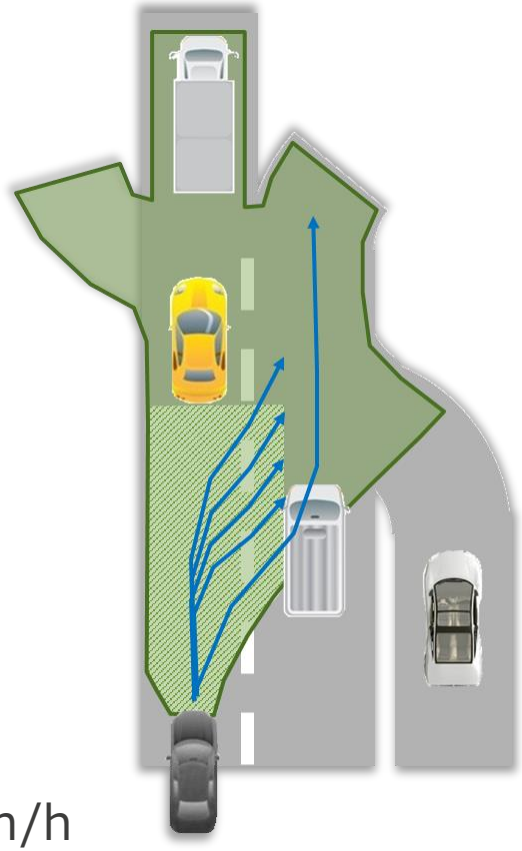
TARGET APPLICATION AND ALGORITHM



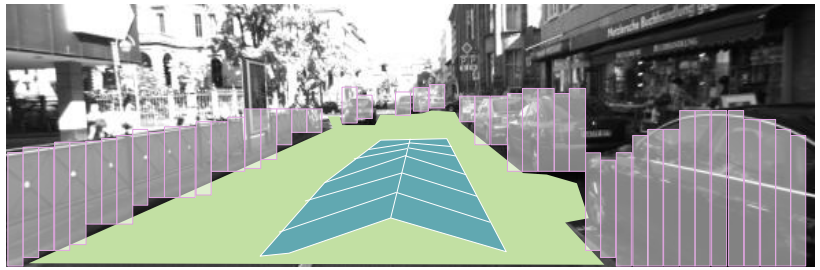
DATA FLOW



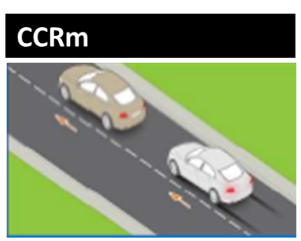
WHY NEED DENSE LEVEL ?

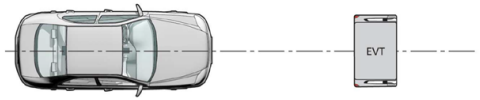


- ✓ Vehicle control in sensitive scene
- ✓ Become a very attractive sensing technique for obtaining 3D information



- ✓ 3D information keep criteria targeted 130km/h





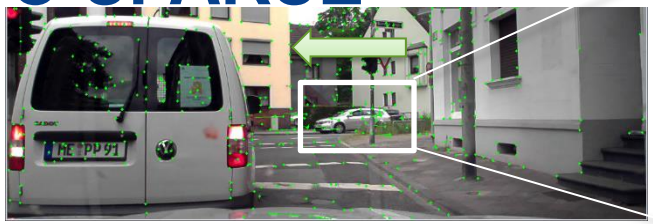
CCRm (moving target 20km/h)	AEB+FCW combined	
	AEB	FCW
AEB city	-	-
AEB Inter-urban	30 ~ 130 (80) km/h	50 ~ 130 (80) km/h

COMPARE DENSE TO SPARSE

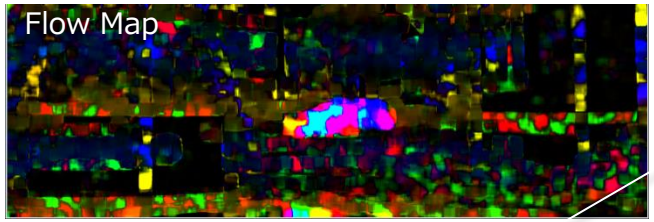
Sparse

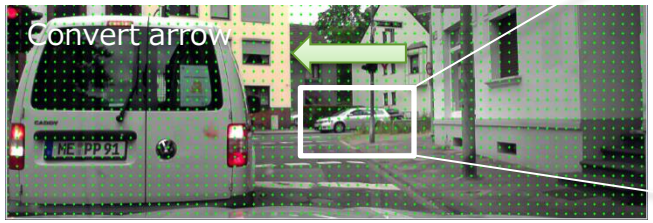
Dense

LK

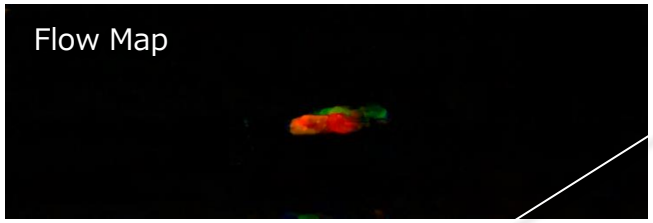


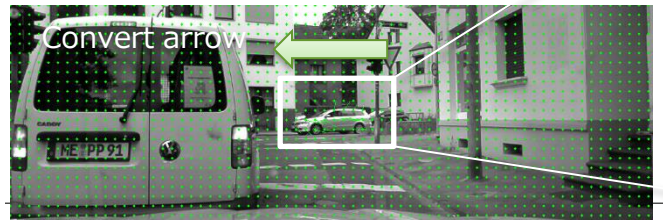
Farneback





Renesas (TVL1 base)

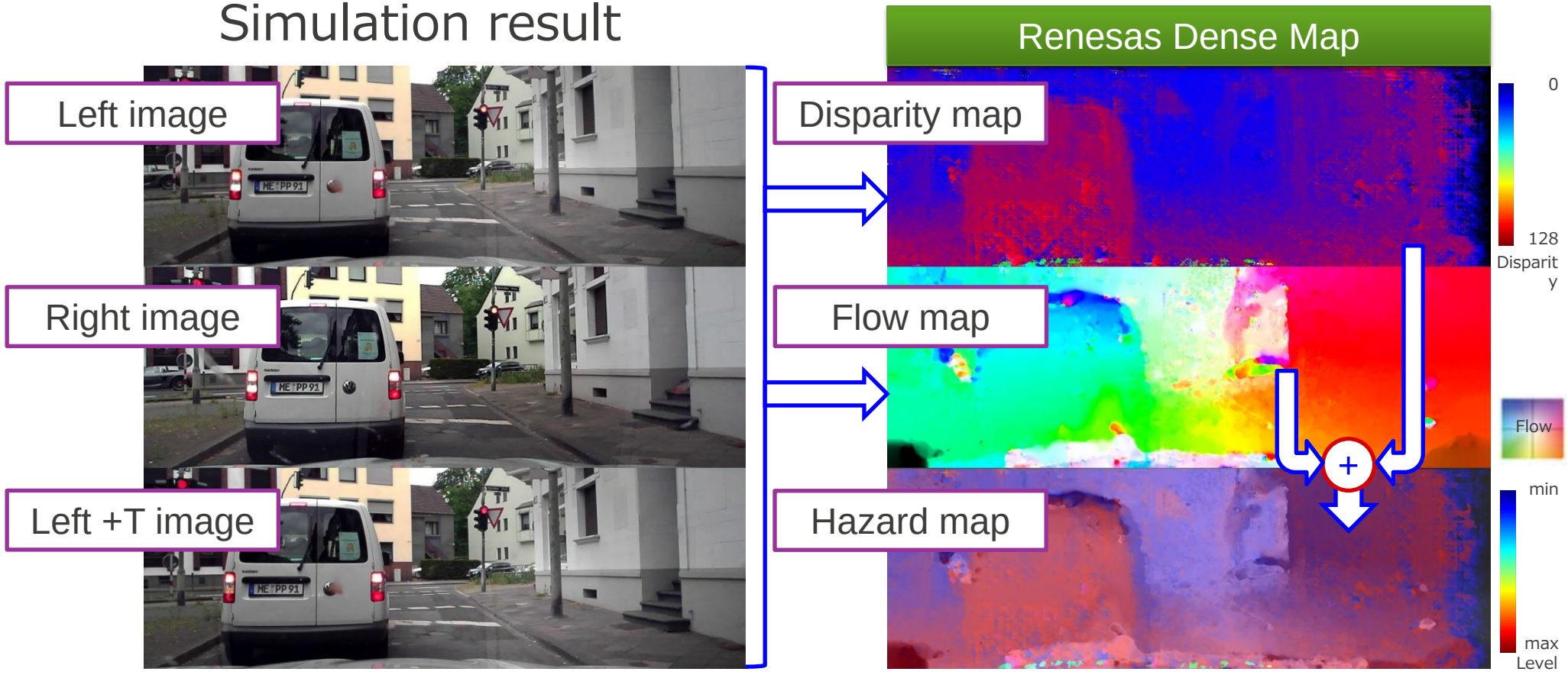




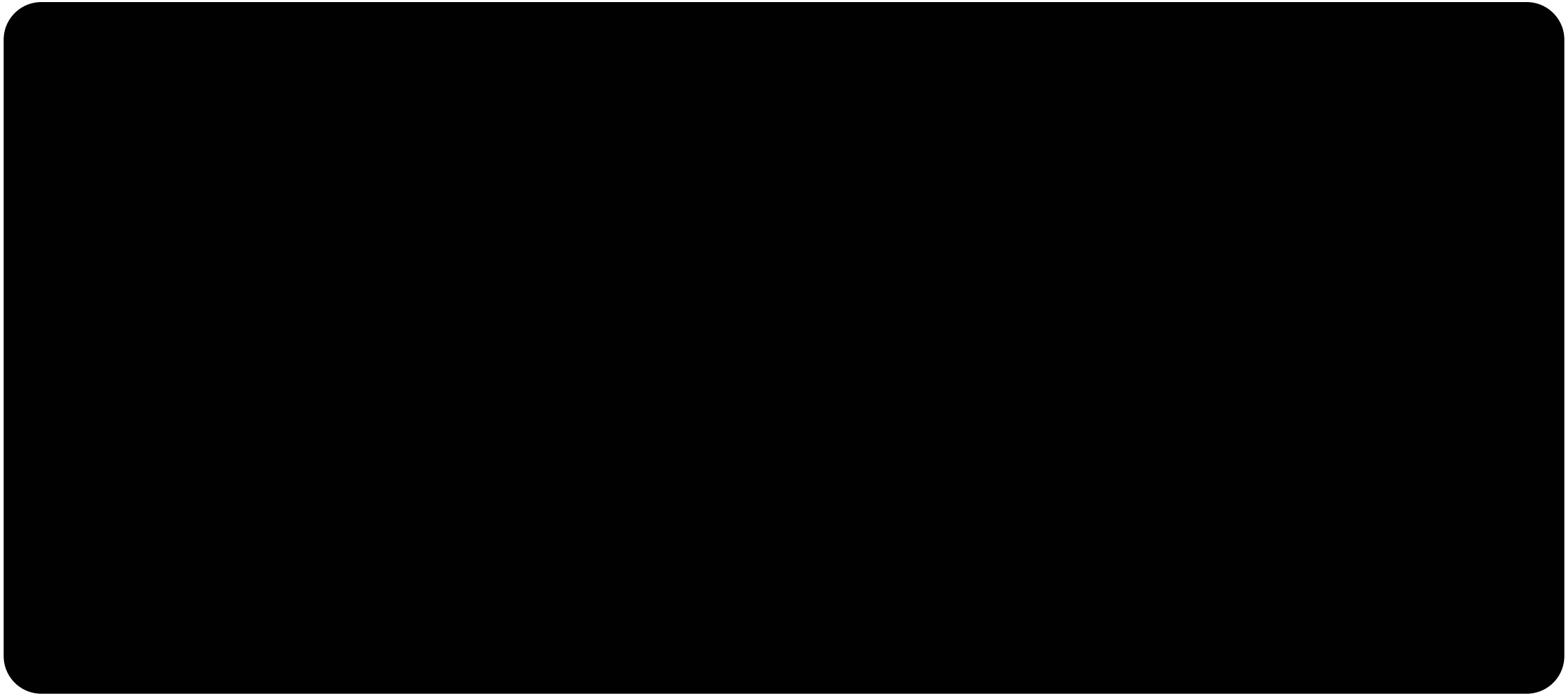
DENSE MAP

IP	Target	Criteria
Dense-Optical-Flow	2Mpix 16ms	0-130Km/h High speed pilot
Dense-Stereo	2Mpix 33ms	0-130Km/h High speed pilot

Simulation result



DENSE MAP SIMULATION DEMONSTRATION



HEADLINE

SUBLINE