REAL-TIME FUSION OF VISIBLE AND THERMAL INFRARED IMAGES IN SURVEILLANCE APPLICATIONS ON SOC HARDWARE.

Fraunhofer
IOSB

Konrad Moren, Thomas Perschke

Institute of Optronics, System Technologies and Image Exploitation (IOSB)
Ettlingen Germany

konrad.moren@iosb,fraunhofer.de , thomas.perschke@iosb.fraunhofer.de

SPIE Berlin 11.09.2018

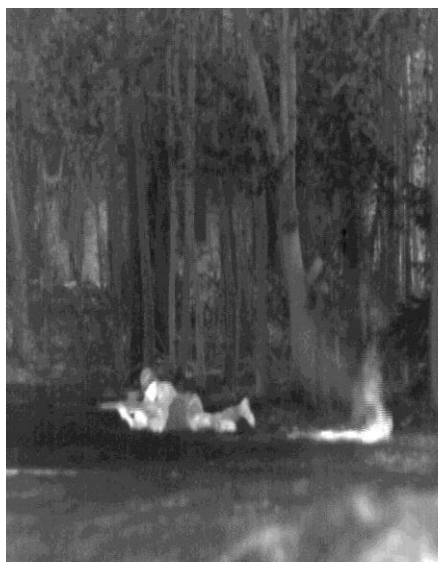
Fusion of visible and thermal infrared images - Applications

https://www.flir.com/products/duo/

http://www.dailymail.co.uk/sciencetech/article-2640869/Google-glass-war-US-military-reveals-augmented-reality-soldiers.html

https://www.bhphotovideo.com/c/product/1373861-REG/bosch_mic_9502_z30wqs_mic_ip_fusion_9000i.html

https://www.optik-pro.de/



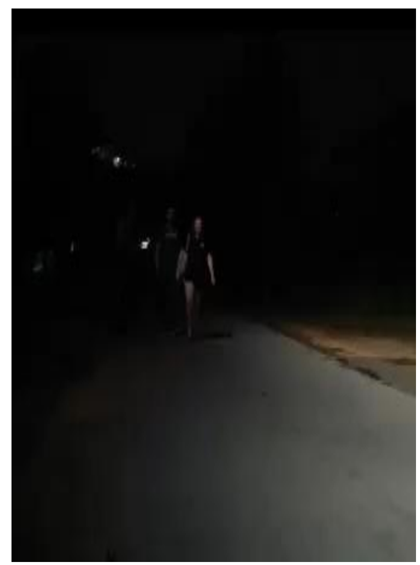
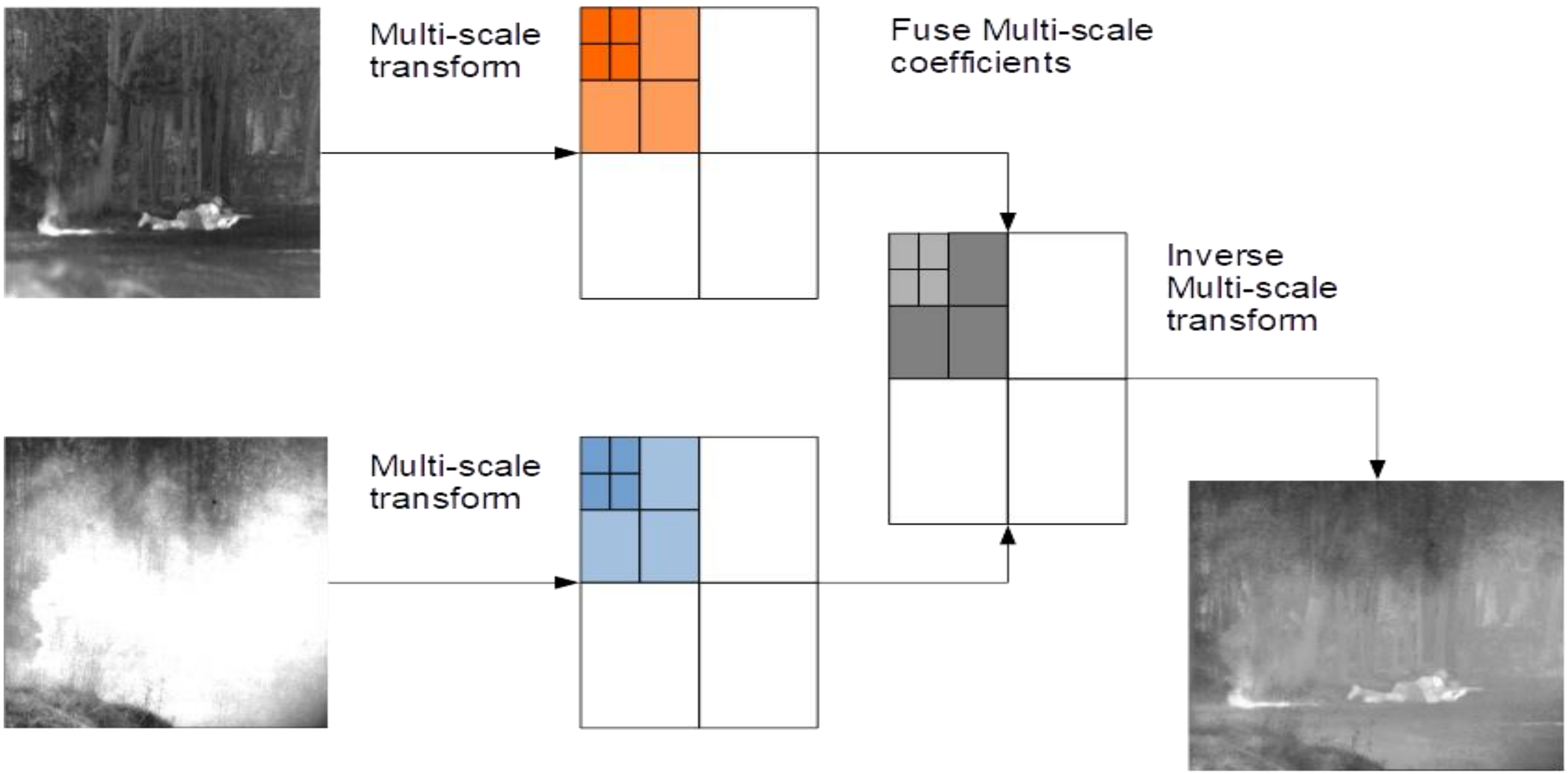


Image Fusion – Visual and IR Images



Input images

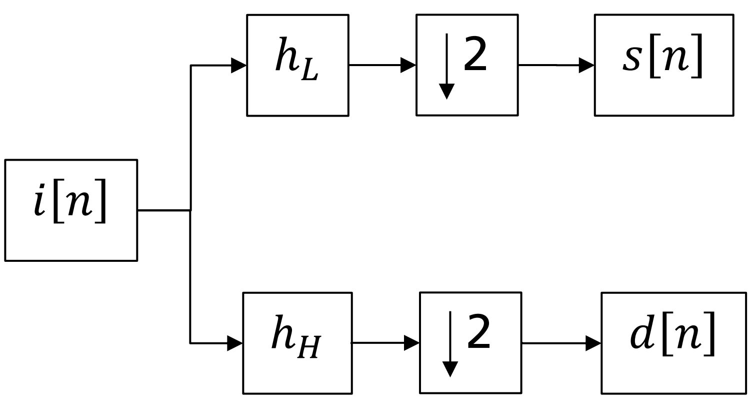
Fused image

Fraunhofer IOSB

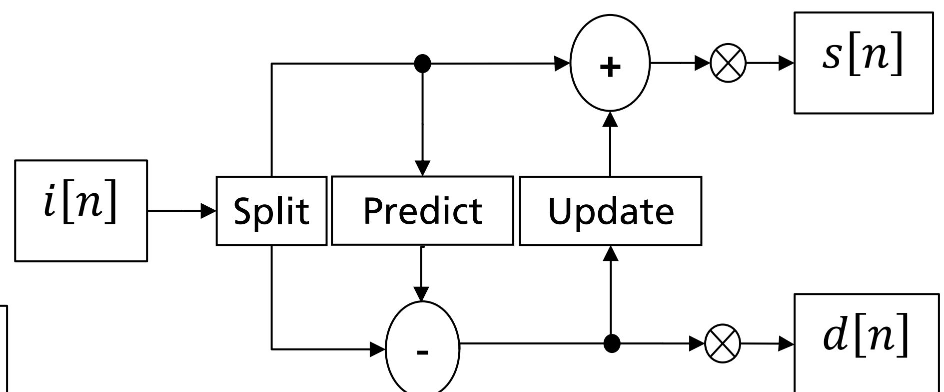
Multi-scale transformations

- Discrete Wavelet Transformation (DWT)
  - First generation wavelets - DWT Mallat Scheme
    - Standard method used to create the multi-scale representation
  - Second generation wavelets - DWT Lifting Scheme
    - Optimized method with reduced number of arithmetic operations

DWT first generation



DWT second generation



Fraunhofer IOSB

DWT Convolution

- 2D convolution with a low and a high pass (FIR) filters
- Filters are separable – Decompose filtering into a vertical and horizontal 1D-convolution

n	0	1	2	3	4
Low pass $h_L[n]$	0.06029	0.26684	-0.07822	-0.01686	0.0267
High pass $h_H[n]$	0.5575	0.2956	-0.0287	-0.0456	0

- Coefficients for the Cohen-Daubechies-Feauveau 9/7 wavelet

- Filters are symmetric :
  - $s[n] = i[2n] \cdot h_L\ [0] + \sum_{k=1}^{4}(i[2n+k] + i[2n-k]) \cdot h_L[k]$
  - $d[n] = i[2n+1] \cdot h_H\ [0] + \sum_{k=1}^{3}(i[2n+1+k] + i[2n+1-k]) \cdot h_H[k]$

- Complexity minimum effort : 23 MAD = 9 MUL + 14 ADD per output pixel pair ($s[n]$, $d[n]$)

DWT Lifting

- Lifting method reduces computational complexity of convolutions
- Convolution operation changed in series of lifting steps
  - Split, Update, Predict, Normalize
- Complexity, minimum effort: 14 MAD = 6 MUL + 8 ADD per output pixel pair ($s[n]$, $d[n]$)

n	0	1	2	3
$\boldsymbol{\alpha_n}$	1.58613	0.05298	0.88291	0.44351
$\boldsymbol{\beta_n}$	0.81289	0.61508	/	/

Split

$$s_0[n] = i[2n]$$
$$d_0[n] = i[2n+1]$$

Update, Predict

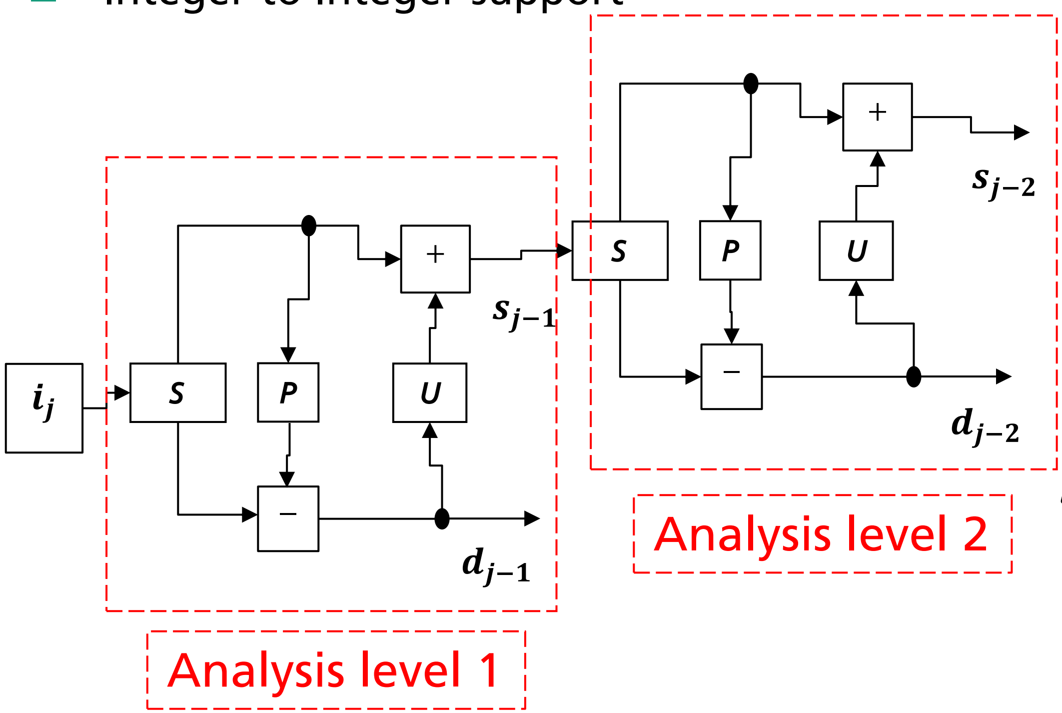
$$d_1[n] = d_0[n] - \alpha_0 \cdot (s_0[n+1] + s_0[n])$$
$$s_1[n] = s_0[n] - \alpha_1 \cdot (d_1[n] + d_1[n-1])$$
$$d_2[n] = d_1[n] - \alpha_2 \cdot (s_1[n+1] + s_1[n])$$
$$s_2[n] = s_1[n] + \alpha_3 \cdot (d_2[n] + d_2[n-1])$$

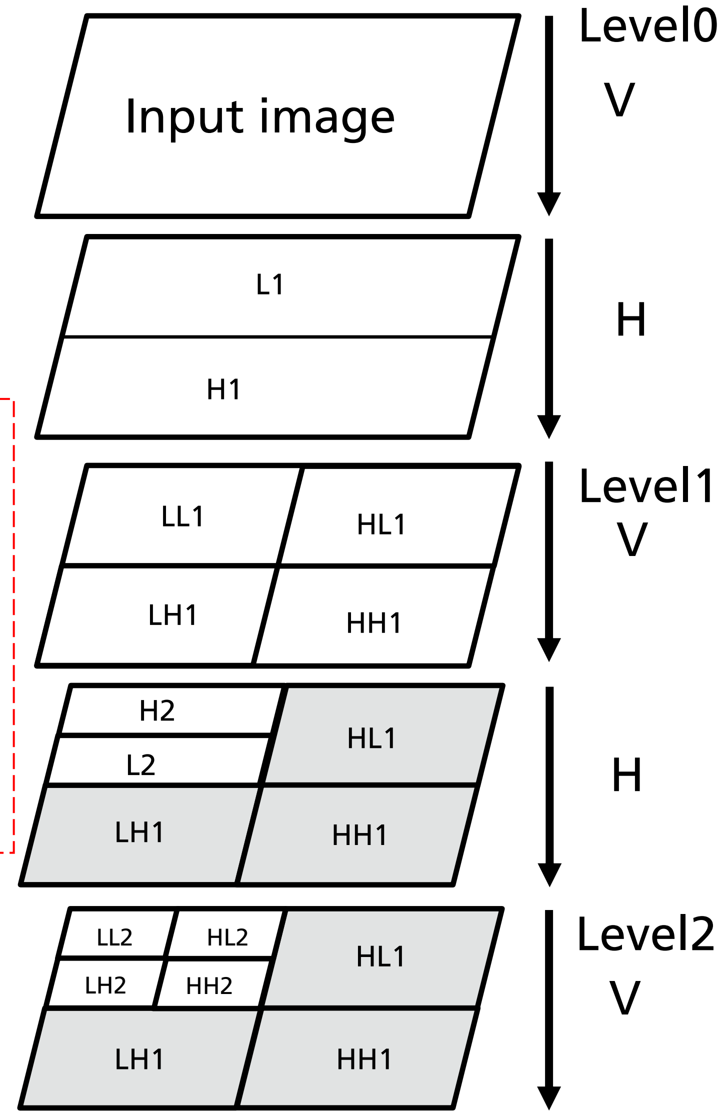
Normalize

$$\mathrm{s}[n] = \beta_0 \cdot s_2[n]$$
$$\mathrm{d}[n] = \beta_1 \cdot d_2[n]$$

Multi-level DWT Lifting

- Transformation in-place no auxiliary memory needed
- Efficient boundary extension handling
- Simple decoder structure , reversed order of encoder
- Integer to Integer support

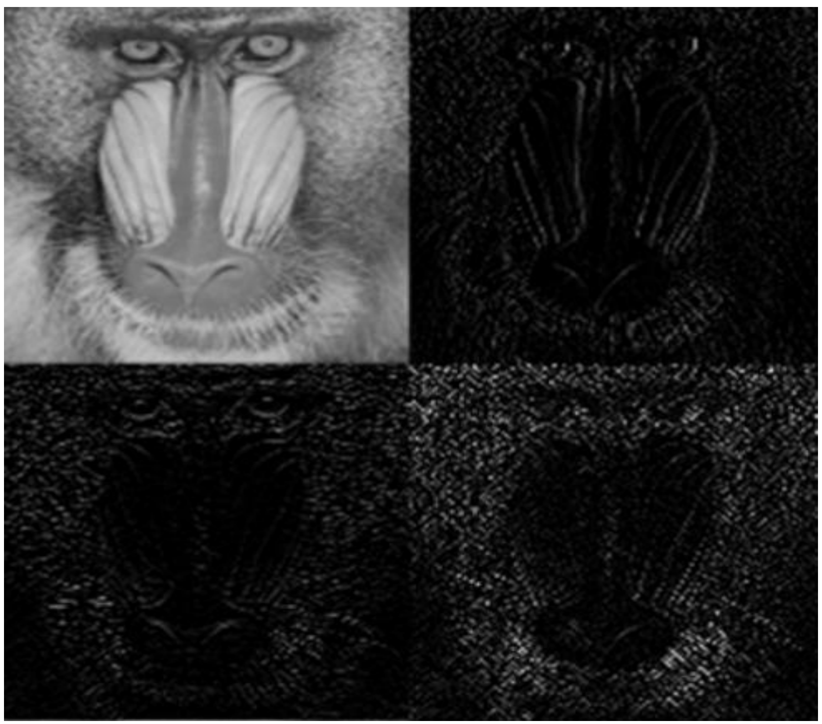




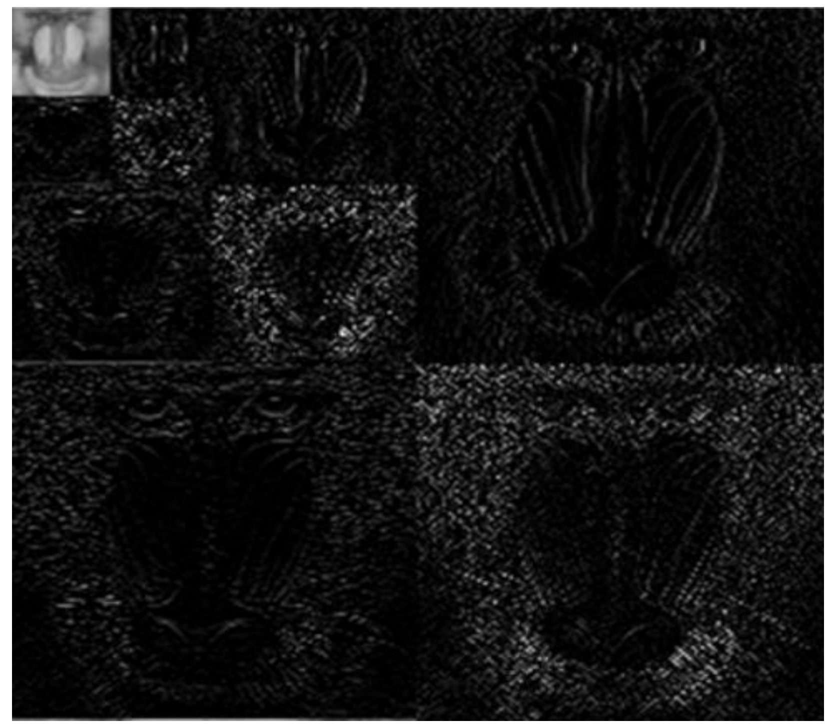
Fraunhofer IOSB

Multi-level fusion

- Pixel-wise combine functions:
  - Average, Abs-max fusion $F_j[n] = Max(Abs((V_j[n] + IR_j[n]) * 0.5)))$
  - Fusion operators independent at different DWT analysis/synthesis levels



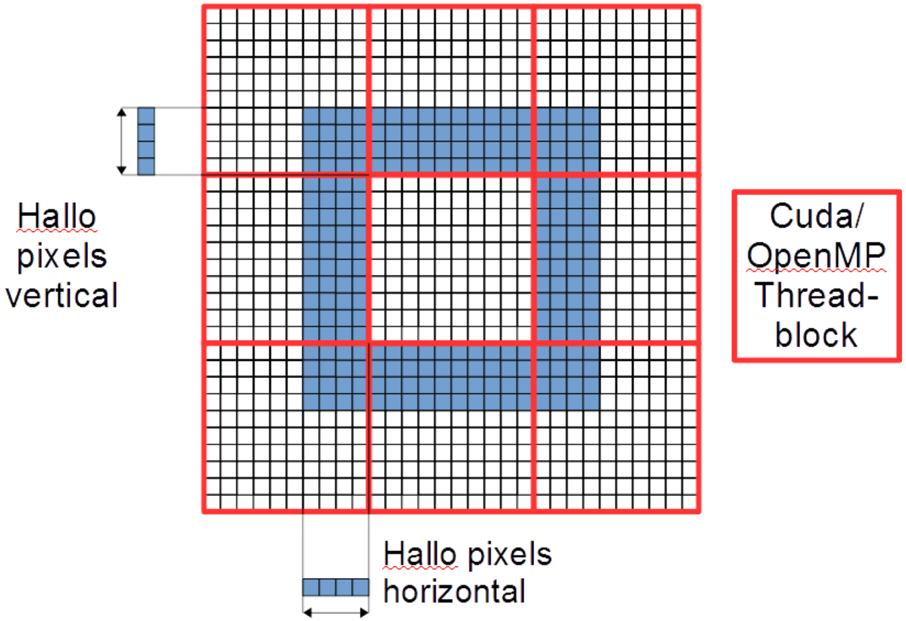
...

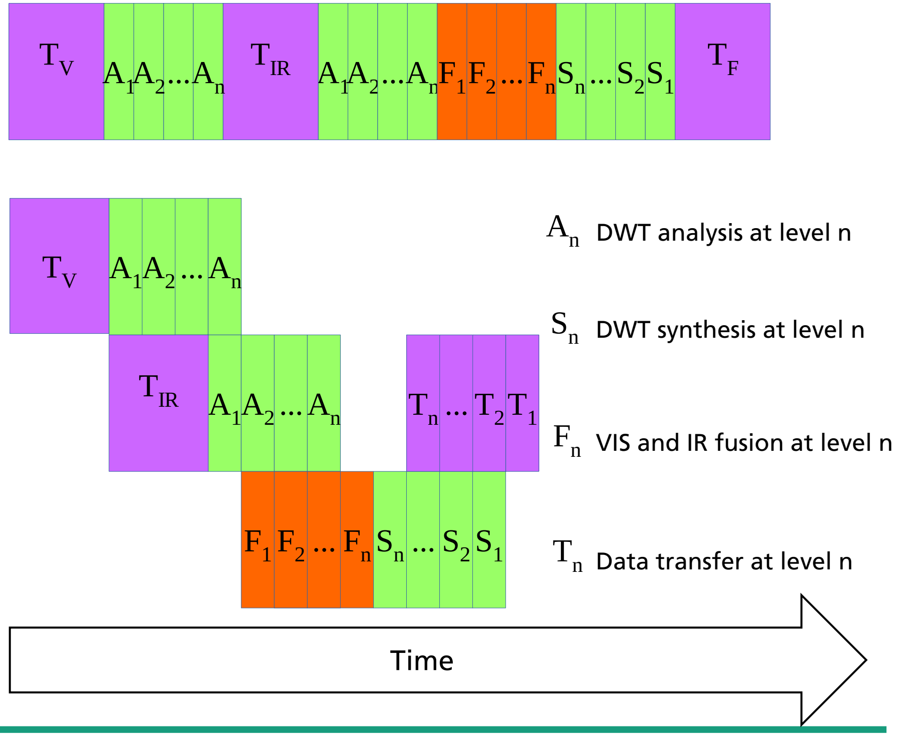


Fraunhofer IOSB

Data parallel, pipelined Fusion

- DWT- Lifting parallelized on CPU and GPU
  - Each output pixel pair computed independently
- Pipelined data transfers between I/O and CPU/GPU
  - Overlapped data transfers with computation

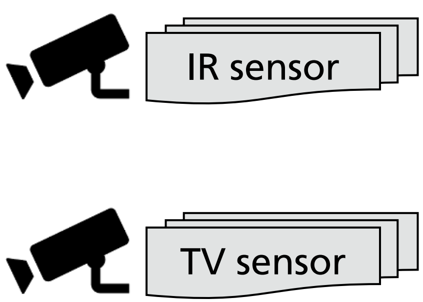


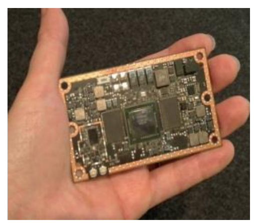


Fraunhofer IOSB

Evaluation - SoC Hardware

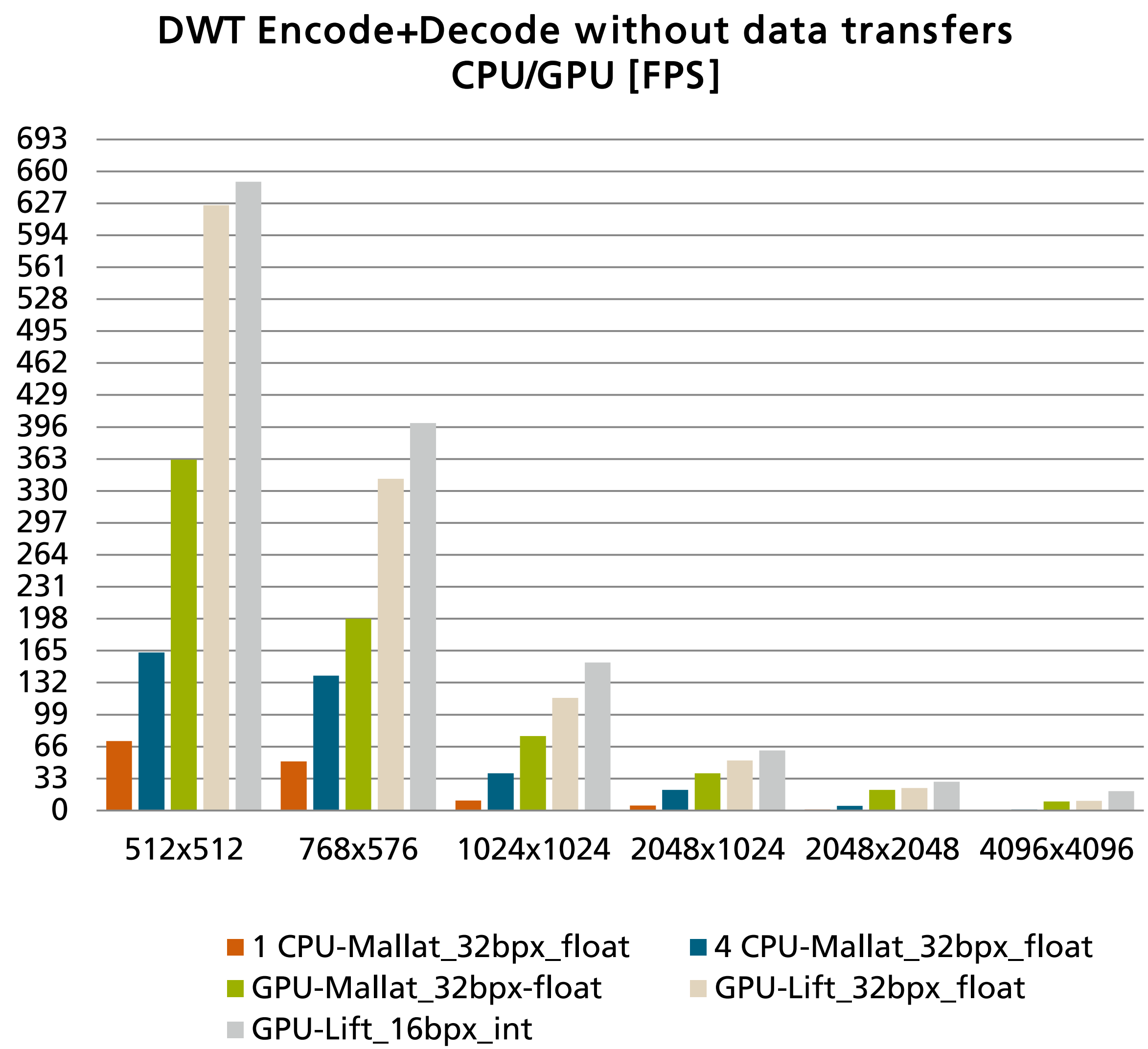
- Jetson TX2 NVidia
  - Programmable multi-core CPU / GPU
  - More than a 1 TFLOP/s of performance at 7.5Watt





Results

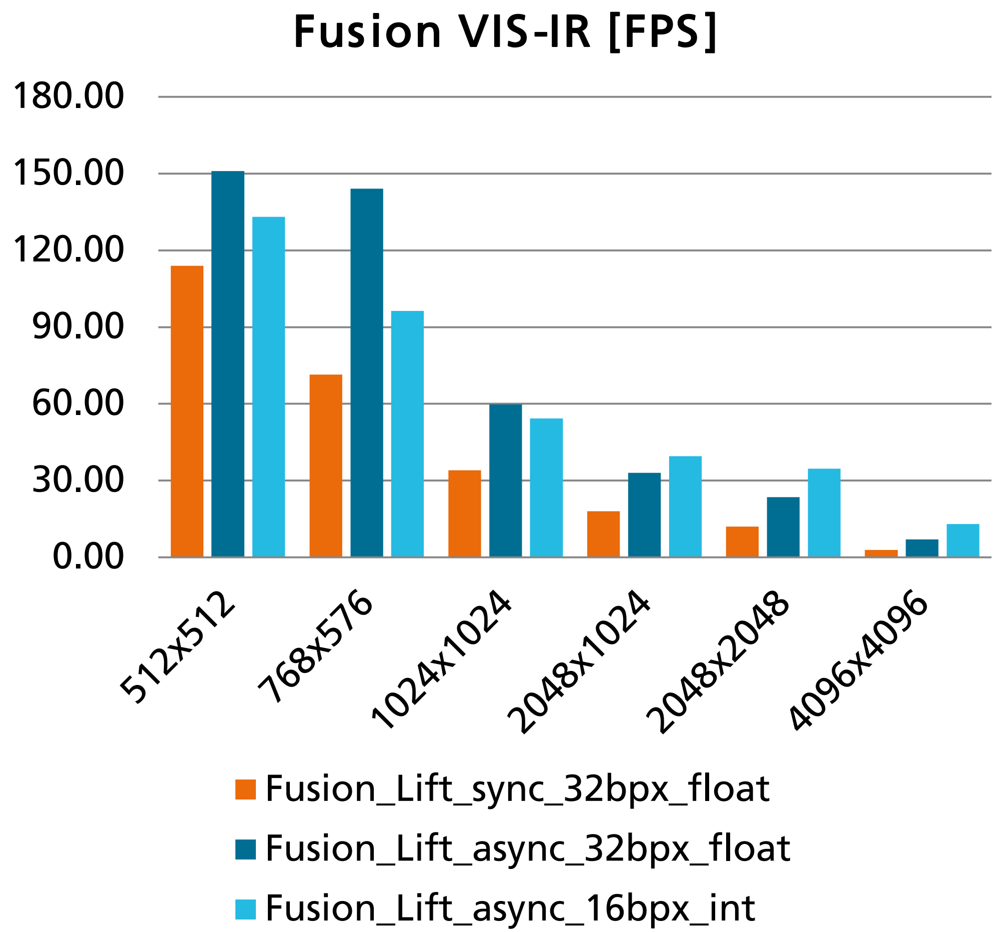
- Performance DWT:
  - DWT-Lifting performance 1.41x better than DWT-Mallat
  - GPU performance 4.7x higher than CPU
  - Real-time (30 Hz) performance with GPU for Full-HD resolution

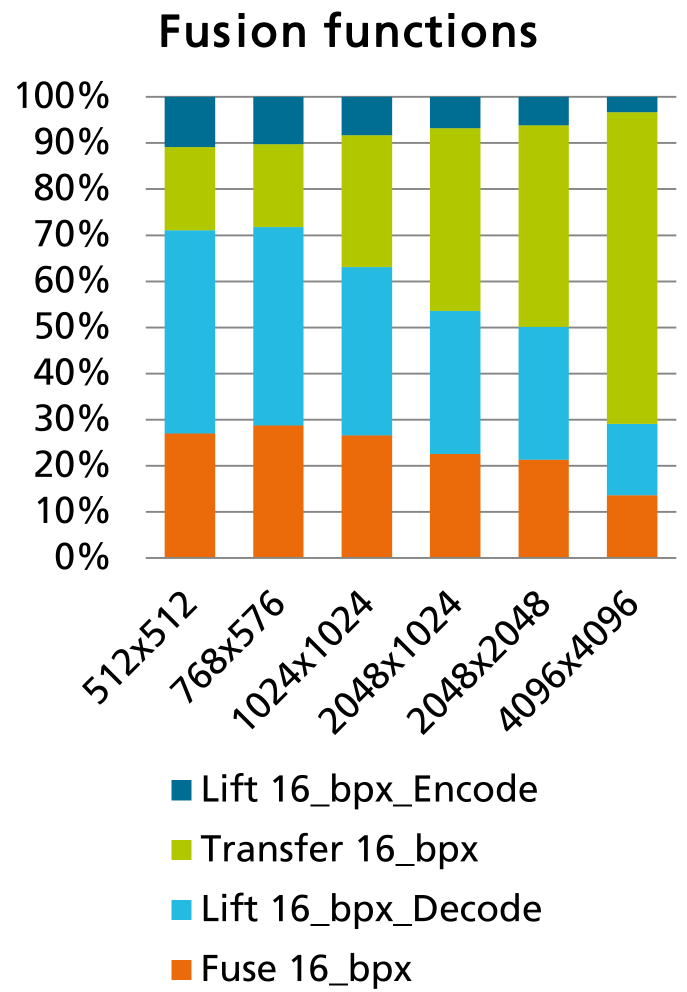


*Mean values of 100 tests per image , 5 DWT levels , CDF 9/7 Wavelet

Fraunhofer IOSB

Results





*Mean values of 100 tests per image , 5 DWT levels , CDF 9/7 Wavelet

Conclusion, Future work

- We show that the pipelined fusion method provides a real-time performance.
- The CPU is less suited than the GPU for a data-parallel DWT.
- The higher memory throughput of GPU is crucial for performance.
- The data transfers are bottleneck to further improve the performance.
- We plan to optimize the data transfers between I/O sensors and processing units.

Thanks for you attention! Questions?