OPTRO 2020
Real-Time Embedded Video Denoiser Prototype

Andrea Petreto

Sorbonne Université, CNRS, LIP6 – LHERITIER - Alcen

2020/01/29

Plan

Context

- Noise Apperance in bad conditions : Low light, Low contrast
- Physic limitation : Modern sensors quasi-perfect with very low read noise levels $\Rightarrow$ Noise $=$ Photon noise.
- Bigger optics :
  - Bigger systems.
  - Heavier
  - More expensive
- Need to propose a new solution : Overcome physics limitation with software.
- Proposed solution : Embedded real-time Spatial-temporal filter (25 fps).

State of the art

- Only a few articles on video denoising
  - A lot more on image denoising.
  - Good displacements estimation needed for robust video denoising [1].
  - [Zuo 2016] Existing algorithms very heavy [2].

    ⇒ Kalman + bilateral method proposed.
    ⇒ Movement estimation using block matching.
    ⇒ No timing indicatons for this solution.
- Recent reconsideration for this problem
  - VBM3D [3] and VBM4D [4] references for a long time (2007, 2011).
  - Visually more efficient new methods:
    - VNLnet 2018 [5].
    - TOF denoising 2017 [6].
    - UNet 2018 [7].

    ⇒ Issue: Compute time too much important (Patchs + CNN).
  - Real-time methods
    - STMKF 2017 [8].
    - Google 2018 [9].

    ⇒ Issue : Only for low noise level situations (like video compression).

Denoising chain : RTE-VD

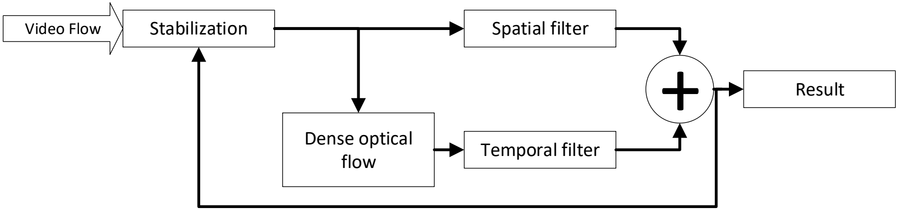

Figure: Main denoising steps.

- Stabilization : Lucas Kanade global approach.
- Dense optical flow : TV-L1.
- Spatial filtering : Separated bilateral filter.
- Temporal filtering : Lateral filter.

Optimizations

Algorithmic Optimizations

- Various transormations applied for all steps:
  - SIMDization (Inefficient vectorization → Code hand-writen in SIMD Neon).
  - Multi-task parallelism with OpenMP.
  - Operators fusion → Reduce the number of memory access.
  - Operators pipeline → Enhanced memory locality.
  - Cache blocking → Reduce memory footprint and enhanced memory locality.
- Other transformations more specific to each algorithm.
  - Lucas-Kanade stabilization.
    - Convolution computation using Integral images (summed area table).
    - Parallel computation of the convolution using partial integral images.
  - TV-L1 dense optical flow estimation.
    - Iterations pipeline (Dasip 2018) [10].
    - Fixed number of iterations chosen by studying the convergence speed.
    - Unbalanced distribution of the number of iterations between scales (3-20-80).
  - Spatial-temporal trilateral filter.
    - Approximation using separated filter [11].

Optimizations

Impact of the optimizations (1/2)

Algorithm	Slow 1C	Fast 1C	Fast 8C	speedup
Stab (LK)	6.66	1.59	0.37	$\times 18$
Flow (TVL1)	260.73	107.93	27.59	$\times 10$
Filter	840.08	1.39	0.25	$\times 3\ 360$
Total	1 107.47	110.90	28.21	$\times$**39**

Table: Execution time (ms) and speedup for RTE-VD on AGX CPU.

- **Slow 1C** : Naive mono core implementation (vectorization on).
- **Fast 1C** : Fast mono core implementation.
- **Fast 8C** : Fast 8 cores parallel implementation.

- Mono core gain : $\times 10$.
- Total gain : $\approx \times 40$.
- Major gain on the filtering due to its approximation with separated filter.
- Optical flow = critical step : 98% total computation time.

Denoising efficiency (1/2)

- Evaluation on a well known database : *Derf's Test Media Collection*.
- Comparison with other state of the art algorithms.
  - STMKF: State of the art for real-time methods.
  - VBM3D & VBM4D: State of the art for denoising efficiency (slow).
  - RTE-VD: This work.

Noise	Method	crowd	park_joy	pedestrians	station	sunflower	touchdown	tractor	overall
$\sigma = 20$	STMKF	26.25	25.59	28.34	26.66	26.97	28.87	25.37	26.70
	RTE-VD	26.38	25.65	30.58	30.98	32.51	30.17	29.38	**28.73**
	VBM3D	28.75	27.89	35.49	34.19	35.48	32.85	31.44	31.34
	VBM4D	28.43	27.11	35.91	35.00	35.97	32.73	31.65	31.11
$\sigma = 40$	STMKF	20.80	20.75	20.70	20.41	20.70	20.86	19.80	20.56
	RTE-VD	22.55	21.64	25.72	27.76	27.87	27.05	25.99	**24.85**
	VBM3D	24.81	23.78	30.65	30.62	30.88	30.21	27.82	27.43
	VBM4D	24.65	23.22	31.32	31.53	31.39	30.09	28.09	27.35

Table: PSNR comparison on 7 *Derf's Test Media Collection*'s sequences with other state of the art algorithms.

- Until +7dB over STMKF. Average: +4dB ($\sigma = 40$).
- Maximum -6dB under VBM3D/4D. Average: -2.5dB ($\sigma = 40$).

Denoising efficiency (2/2)

Figure: Visual comparison on the pedestrian scene (Gaussian noise: $\sigma = 40$).

- Stronger denoising than STMKF.
- Less efficient denoising than VBM3D/4D.
- Weakness on details rendering for background.

Time and energy consumption (1/3)

Execution time comparison with other state of the art algorithms

- Implementation on various platforms :

Board	Process	CPU	Fmax (GHz)	Idle Power (W)
TX2	16 nm	4×A57 + 2×Denver2	2.00	2.0
AGX	12 nm	8×Carmel	2.27	6.3
NANO	12 nm	4×A57	1.43	1.2
XEON	14 nm	2×10C/20T Skylake	2.20	–

Table: Technical specification of tested platforms.

- Method comparison:

Algorithm	Time (s)	Platform
STMKF	0.004 5	Xeon
RTE-VD (this work)	0.009 7	Xeon
VBM3D	2.0	Xeon
VBM4D	45	Xeon
STMKF	0.015	AGX
RTE-VD (this work)	0.037	AGX

Table: Time per qHD images (960×540 pixels).

- 200× faster than VBM3D.
- 4600× faster than VBM4D.
- 2.5× slower than STMKF.
- Embedded real-time in qHD

Time and energy consumption (2/3)

Time vs energy efficiency: Dynamic consumption

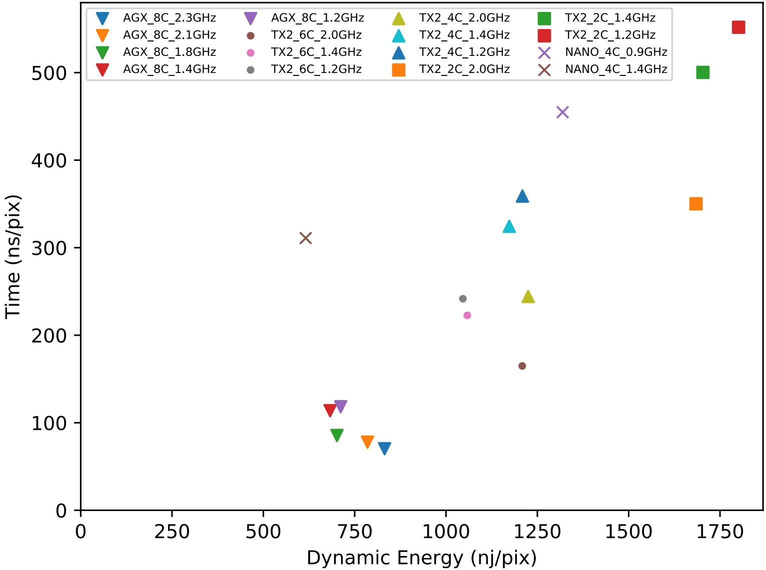

Figure: Time/energy efficiency of RTE-VD on CPU for various frequencies ($E_{dynamic}$).

Time and energy consumption (3/3)

Time vs Energy efficiency: Dynamic + static consumption

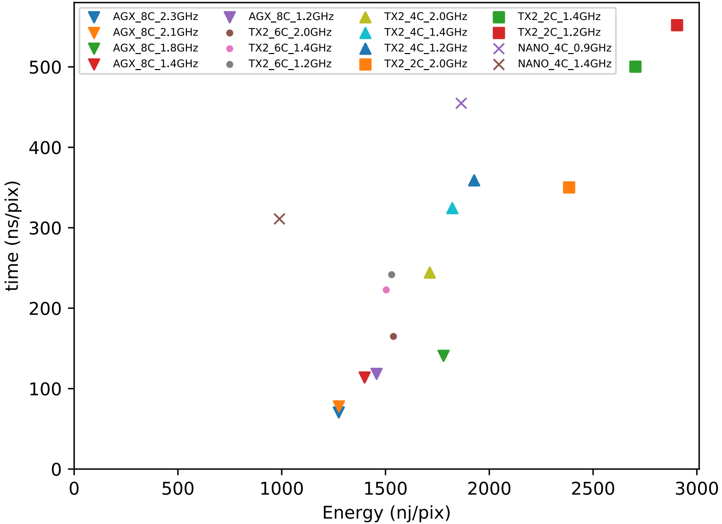

Figure: Time/energy efficiency of RTE-VD on CPU for various frequencies (E_{total}).

VIRTANS : Denoiser Prototype

Figure: VIRTANS : Video Real-Time Algorithm : Noise Suppression.

- TX2i based Architecture.
- Real-time 480×270 pixels video denoising.
- SDI input / HDMI Output.
- Ethernet communication with LHERITIER Cameras.

Conclusion & futur works

- Conclusion
  - Introduction to a new Real-Time Embedded Video Denoising algorithm
    - Slower than STMKF but denoising a lot more efficient
    - Embedded real-time perfomances for qHD videos (960×540 pixels)
  - Energy consumption study: positioning depending on the targeted system
  - RTE-VD based real-time video denoiser : VIRTANS

  $\Rightarrow$ RTE-VD is well positioned for speed/accuracy tradeoff

- Future works
  - GPU implementation and hybrid CPU/GPU computation
  - 32 - 16 bits hybrid computation
  - Reduce VIRTANS form factor even more.

Thank you !

References I

C. Liu and W. T. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *European Conference on Computer Vision*, pp. 706–719, Springer, 2010.

C. Zuo, Y. Liu, X. Tan, W. Wang, and M. Zhang, "Video denoising based on a spatiotemporal kalman-bilateral mixture model," *The Scientific World Journal*, vol. 2013, 2013.

K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *2007 15th European Signal Processing Conference*, pp. 145–149, IEEE, 2007.

M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4d nonlocal spatiotemporal transforms," in *Image Processing: Algorithms and Systems IX*, vol. 7870, p. 787003, International Society for Optics and Photonics, 2011.

A. Davy, T. Ehret, G. Facciolo, J.-M. Morel, and P. Arias, "Non-local video denoising by cnn," *arXiv preprint arXiv:1811.12758*, 2018.

References II

T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *arXiv preprint arXiv:1711.09078*, 2017.

S. Lefkimmiatis, "Universal denoising networks: a novel cnn architecture for image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3204–3213, 2018.

S. G. Pfleger, P. D. Plentz, R. C. Rocha, A. D. Pereira, and M. Castro, "Real-time video denoising on multicores and gpus with kalman-based and bilateral filters fusion," *Journal of Real-Time Image Processing*, pp. 1–14, 2017.

J. Ehmann, L.-C. Chu, S.-F. Tsai, and C.-K. Liang, "Real-time video denoising on mobile phones," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 505–509, IEEE, 2018.

References III

A. Petreto, A. Hennequin, T. Koehler, T. Romera, Y. Fargeix, B. Gaillard, M. Bouyer, Q. L. Meunier, and L. Lacassagne, "Energy and execution time comparison of optical flow algorithms on SIMD and GPU architectures," in *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pp. 25–30, IEEE, 2018.

T. Q. Pham and L. J. Van Vliet, "Separable bilateral filtering for fast video preprocessing," in *2005 IEEE International Conference on Multimedia and Expo*, pp. 4–pp, IEEE, 2005.

Optimizations

Impact of the optimizations (2/2)

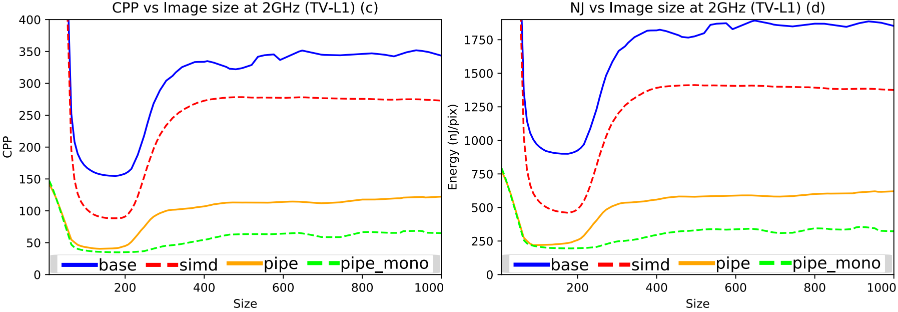

Figure: Impact of TV-L1 optimizations on speed and energy on TK1's CPU depending of the image size. Lower is better.

Version	Base	II	Pipe	MA	SIMD	OMP8	**Total**
Time (ms)	1742	56,0	33,5	29,0	9,2	1,6	–
Speedup	$\times1$	$\times31$	$\times1,75$	$\times1,2$	$\times3,1$	$\times5,9$	$\times1120$

Table: Impact of LK Stabilization optimizations for fullHD images on AGX's CPU.

Backup : Time and energy consumption

Time vs energy efficiency

Table: Best configurations for 25 fps

Configuration	Energy (nJ/pix)	Time (ns/pix)	Max size (#pix)	Freq (GHz)
NANO min energy NANO min time	616	311	358	1.4
TX2 min energy	1046	242	406	1.2
TX2 min time	1209	165	492	2.0
AGX min energy	683	114	592	1.4
AGX min time	832	70	754	2.3

- NANO : the most energy efficient
- AGX : the fastest and the most energy efficient
- TX2 : Penalized by its process but faster than NANO
- Greatest image at 25 fps : 754 $\times$ 754 pixels