

COMPAS 2019

Débruitage temps réel embarqué pour vidéos fortement bruitées

Andrea Petreto (orateur), Thomas Romera, Florain Lemaitre, Ian Masliah,
Boris Gaillard, Manuel Bouyer, Quentin Meunier, Lionel Lacassagne

Sorbonne Université, CNRS, LIP6 – LHERITIER - Alcen

28 juin 2019



Contexte

- ▶ Apparition de bruit en conditions difficiles : faible luminosité, faible contraste
- ▶ Limitation physique : Capteurs modernes quasi-parfaits avec un bruit de lecture très faible
⇒ bruit = bruit de photon
- ▶ Besoin de proposer une nouvelle solution : Outrepasser les limites de la physique grâce au logiciel
- ▶ Solutions existantes
 - ▶ Essentiellement axées sur le traitement des images et non des vidéos
 - ▶ Inadaptées à l'application souhaitée
 - ▶ Dimension temporelle supplémentaire amenée par la vidéo
- ▶ Solution proposée : filtre spatio-temporel temps réel embarqué (25 ips)

État de l'art

- ▶ Faible littérature sur le dé-bruitage de vidéos
 - ▶ Traitement des images bien plus présent
 - ▶ Liu 2010 [1] Estimation de flot optique nécessaire pour dé-bruitage robuste
 - ▶ Zuo 2016 [2] Algorithmes existants extrêmement lourds
 - ⇒ Proposition d'une méthode kalman + filtre bilatéral
 - ⇒ Estimation du mouvement par block matching
 - ⇒ Pas d'indication de performance pour cette solution
- ▶ Regain d'intérêt pour ce problème
 - ▶ VBM3D [3] et VBM4D [4] références de longue date (2007, 2011)
 - ▶ Nouvelles méthodes visuellement plus efficaces
 - ▶ VNLnet 2018 [5]
 - ▶ TOF denoising 2017 [6]
 - ▶ UNet 2018 [7]
 - ⇒ Problème : temps de calculs beaucoup trop importants (Patches + CNN)
 - ▶ Méthodes adaptées au temps réel
 - ▶ STMKF 2017 [8]
 - ▶ Google 2018 [9]
 - ⇒ Problème : solutions pour de faibles bruits seulement (ex : compression)

Chaîne de débruitage : RTE-VD

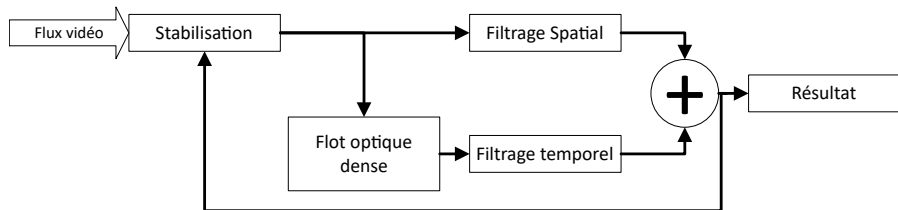


FIGURE – Principaux maillons de la chaîne de débruitage.

- ▶ Stabilisation : Lucas Kanade approche globale
- ▶ Flot optique dense : TV-L1
- ▶ Filtrage spatial : Filtre bi-latéral séparé
- ▶ Filtrage temporel : Filtre latéral

Effacité du débruitage



FIGURE – Comparaison visuelle sur la scène pedestrians (Bruit gaussien $\sigma = 40$)

Optimisations algorithmiques

- ▶ Différentes transformations appliquées à tous les niveaux : [10, 11, 12]
 - ▶ SIMDisation (vectorisation inefficace → Écriture du code SIMD Neon)
 - ▶ parallélisation Multi-tâches (avec OpenMP)
 - ▶ Fusion d'opérateurs → réduit le nombre d'accès mémoire
 - ▶ Pipeline d'opérateurs → améliore la localité mémoire
 - ▶ Blocage de cache → réduit l'empreinte mémoire et améliore la localité

- ▶ Autres transformations plus spécifiques à chaque algorithme
 - ▶ Stabilisation par Lucas-Kanade
 - ▶ Calcul de convolution par image intégrale
 - ▶ parallélisation du calcul d'image intégrale → calculs partiels
 - ▶ Calcul du flot optique par TV-L1
 - ▶ Pipeline d'itérations [13, 14]
 - ▶ Nombre d'itérations fixe choisi en étudiant vitesse de convergence
 - ▶ Répartition inégale des itérations entre les échelles (3-20-80)
 - ▶ Filtrage spatio-temporel tri-latéral
 - ▶ Approximation en filtre séparable

Temps et consommation énergétique

Impact des optimisations

Algorithm	Slow 1C	Fast 1C	Fast 8C	speedup
Stab (LK)	6.66	1.59	0.37	×18
Flow (TVL1)	260.73	107.93	27.59	×10
Filter	1 717.85	1.39	0.25	×6 871
Total	1 985.24	110.90	28.21	×70

TABLE – Temps d'exécution (ms) et accélération de RTE-VD sur le CPU de l'AGX

- ▶ **Slow 1C** : implémentation naïve mono coeur (vectorisation possible)
- ▶ **Fast 1C** : implémentation rapide mono coeur
- ▶ **Fast 8C** : implémentation rapide parallèle sur 8 coeurs
- ▶ Gain mono coeur : ×18
- ▶ Gain total : ×70
- ▶ Gain sur le filtrage très important grâce à l'approximation en filtre séparable
- ▶ **Flot optique** = point dur : 98% du temps total de calcul

Temps et consommation énergétique

Comparaison du temps d'exécution à l'état de l'art

► Implémentation sur différentes plateformes :

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 – Spécifications techniques des plateformes utilisées.

► Comparaison des méthodes :

Algorithm	Time (s)	Platform
STMKF	0.0045	Xeon
RTE-VD (this work)	0.0097	Xeon
VBM3D	2.0	Xeon
VBM4D	45	Xeon
STMKF	0.015	AGX
RTE-VD (this work)	0.037	AGX

- 200× plus rapide que VBM3D
- 4600× plus rapide que VBM4D
- 2.5× plus lent que STMKF

TABLE – Temps par images qHD (960×540 pixels)

Temps et consommation énergétique

Efficacité temps vs énergie : Consommation dynamique

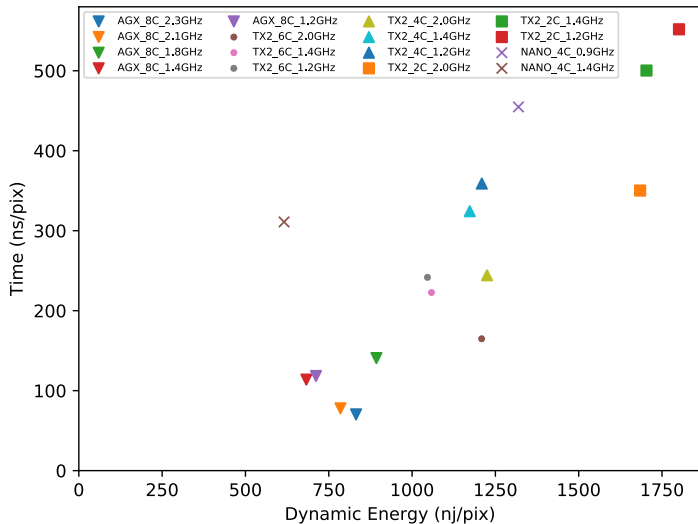


FIGURE – Efficacité temps/énergie de RTE-VD sur CPU pour différentes fréquences

Conclusion et travaux futurs









▶ Conclusion

- ▶ Mise en place d'une chaîne de débruitage temps réel embarquée
 - ▶ Plus lent que STMKF mais débruitage beaucoup plus efficace
 - ▶ Temps réel embarqué pour des vidéos qHD (960×540 pixels)
- ▶ Étude énergétique : positionnement en fonction du système embarqué visé

⇒ Positionnement intéressant en termes de compromis vitesse/qualité

▶ Travaux futurs

- ▶ Implémentation GPU et hybridation des calculs CPU/GPU
 - ▶ Démonstrateur temps réel à base de Nvidia Jetson : TRL 4-5.
 - ▶ Portages des calculs en flottants 16 bits [15]
 - ▶ Segmentation de la carte des vitesses pour clusterisation [16]
- ▶ Plus de Détails dans DASIP 2019 [17] (à paraître)

- 
- 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.
- 
- 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. Pflieger, 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.



L. Lacassagne, D. Etiemble, A. Hassan-Zahraee, A. Dominguez, and P. Vezolle, "High level transforms for SIMD and low-level computer vision algorithms," in *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pp. 49–56, 2014.



H. Ye, L. Lacassagne, J. Falcou, D. Etiemble, L. Cabaret, and O. Florent, "High level transforms to reduce energy consumption of signal and image processing operators," in *IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pp. 247–254, 2013.



D. Etiemble and L. Lacassagne, "16-bit FP sub-word parallelism to facilitate compiler vectorization and improve performance of image and media processing," in *International Conference on Parallel Processing (ICPP)*, pp. 1–4, IEEE, 2004.



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.



A. Petreto, A. Hennequin, T. Koehler, T. Romera, Y. Fargeix, B. Gaillard, M. Bouyer, Q. Meunier, and L. Lacassagne, "Comparaison de la consommation énergétique et du temps d'exécution d'un algorithme de traitement d'images optimisé sur des architectures simd et gpu," 2018.



S. Piskorski, L. Lacassagne, S. Bouaziz, and D. Etiemble, "Customizing CPU instructions for embedded vision systems," in *Computer Architecture, Machine Perception and Sensors (CAMP5)*, pp. 59–64, IEEE, 2006.



K. Aneja, F. Laguzet, L. Lacassagne, and A. Merigot, "Video rate image segmentation by means of region splitting and merging," in *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 437–442, 2009.



A. Petreto, T. Romera, F. Lemaitre, I. Masliah, B. Gaillard, M. Bouyer, Q. L. Meunier, and L. Lacassagne, "A new real-time embedded video denoising algorithm," in *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pp. 1–6, 2019.