GPU-Multi-core memetic algorithm to optical flow estimation

Abdelkhalek Mansouri, Jean-Charles Creput

Laboratoire Connaissance et Intelligence Artificielle Distribuées CIAD UMR 7533, Univ. Bourgogne Franche-Comté (UBFC),
Belfort, 90010, France.
abdelkhalek.mansouri@gmail.com, jean-charles.creput@utbm.fr

Keywords : *Optical flow, parallel local search, memetic algorithm, GPU, multi-core CPU.*

1 Introduction

Optical flow estimation [1] is a challenging computer vision task for which researchers and developers worked on for decades and proposed hundreds of solutions. In this paper, we propose a metaheuristic population-based framework that embeds GPU parallel local search, called *Memetic-PLS*, to optical flow estimation. The approach is implemented on GPU platform for all data computation (on image pixels) and CPU multi-core for kernel calls management and solution selection. The aim of the approach is to provide a good tradeoff between quality and computation time comparing to state-of-the-art programs, while preserving flow discontinuities at object boundaries, and allowing further accelerations as powerful new multi-processors become available.

2 Optical flow problem

In this paper, we apply a memetic algorithm to image processing optical flow problem by energy minimization. Let I_1 and I_2 be two images : $[W] \times [H] \to \mathbb{R}^+$, where $[N] = \{0, ..., N-1\}$, for any integer N. Let $\Omega = [W] \times [H]$ be their index domain that defines the image plane, with regular 4-neighbour topology. Let $u : [W] \times [H] \to \mathbb{R}^2$ be the flow variable that represents the pixel displacements of first image to match second image, and constitutes search space. The energy function to minimize is defined as follows :

$$E(u) = \sum_{i \in \Omega} |I_1(i) - I_2(i+u_i)| + \lambda \sum_{i \in \Omega} \sum_{j \in \mathcal{N}_i^r} |u_i - u_j| \times G_S(i-j) \times G_I(I_1(i) - I_1(j)) \times noc_j, \quad (1)$$

where $\mathcal{N}_i^r$ is a given neighborhood of radius r, G_S is a Gaussian function in the spatial domain and G_I is a Gaussian function in the intensity domain. Term noc_j is a test for non occlusion.

3 Memetic parallel local search algorithm (Memetic-PLS)

Memetic algorithms are extensions of genetic algorithms by employing local search in place of a standard mutation. The goal is to take benefits of local search while providing diversification mechanisms to avoid being trapped in local minima. The proposed Memetic-PLS [2] operates on a superpixel segmentation map of the first image obtained by k-means 3D. Given a population of solution flows $\mathcal{P}$, to each solution is associated a multi-core CPU process that executes GPU kernels at pixel level. All pixel operations are GPU. The algorithm starts by a (GPU) winner-take-all construction step to match superpixels to second image. Then, an improvement loop executes (GPU) parallel local search followed by (GPU) evaluation at each generation. The neighborhood operator consists of translation/rotation of a given superpixel. Evaluation computes the main energy function. A selection operator selects the single best and single worst solutions of population. A (GPU) crossover merges best into worst solution retaining superpixels that should favor best match. Two GPU post-treatments are applied. A denoising procedure acts as an erosion operation, followed by a bilateral flow filtering over a minimum spanning forest segmentation map of the first image.

4 Experimental results

We implement the algorithm with C++ and CUDA Toolkit v10.0. We test our approach against three state-of-the-art approaches : NVIDIA implementation of Horn and Schunck algorithm, Brox et al. algorithm, and Epicflow approach. We report the results of the experiments on Middlebury dataset, with nine test-cases, in Table 1. Population size is 10 for 20 generations performed. We report the endpoint error, angular error, and computing time in seconds, averaged on 10 runs, with standard deviation in parenthesis. The results show that Memetic-PLS provides a competitive quality/time tradeoff compared to the tested methods. Samples of visual results are provided in Figure 1 to show how well edge boundaries are preserved.

TAB. 1 –
OPTICAL FLOW EXPERIMENTAL RESULTS ON MIDDLEBURY BENCHMARKS.

Optical flow approaches	Platform	*AEE (stddev)*	*AAE (stddev)*	$t(s)$ *(stddev)*
Horn and Schunck	GPU	0.88	11.38	0.0002
Epicflow	CPU	0.86	11.10	7.56
Brox et al.	CPU	0.46	6.16	4.97
Memetic-PLS	GPU	0.62(0.014)	6.71 (0.17)	2.39 (0.09)

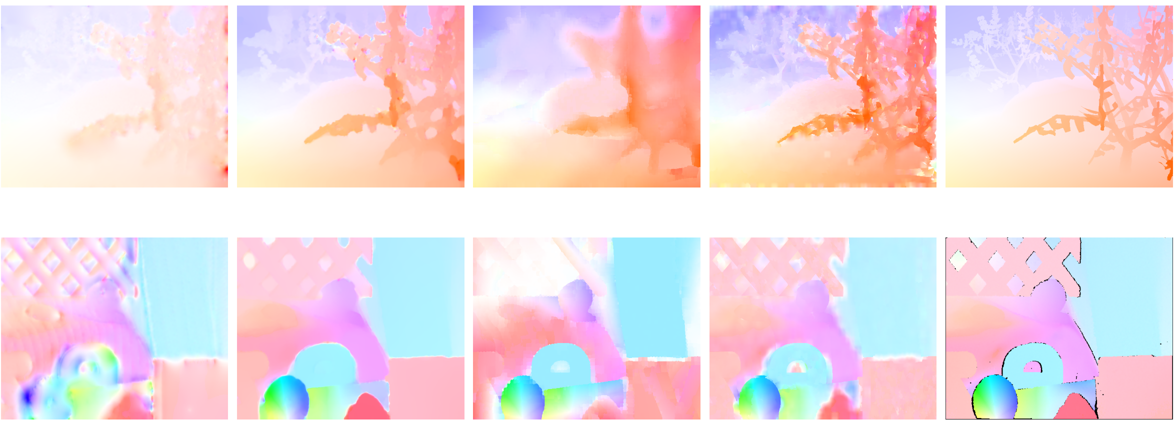

FIG. 1 – Visual comparative results (with standard color code) against three other state-of-the-art approaches. First row : Grove3 benchmark ; Second row : RubberWhale benchmark. From left to right : Horn and Schunck ; Brox et al. ; Epicflow ; Memetic-PLS ; ground truth flow.

References

[1] Sun, D. and Roth, S. and Black, M. J. Secrets of optical flow estimation and their principles *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[2] Abdelkhalek Mansouri. Generic heuristics on gpu to superpixel segmentation and application to optical flow estimation. *Ph.D. thesis*, Université de Bourgogne Franche-Comté (2020). URL http ://www.theses.fr/2020UBFCA012