Optical flow optimization using a parallel Genetic Algorithm

Olmo Zavala-Romero, Guillermo Botella, Anke Meyer-Baese and Uwe Meyer-Baese

Introduction

Optical flow is the displacement field from one image into another. It shows the velocities from a 3D space projected into a 2D space and it is being used in many different areas like medical imaging, tracking and object segmentation, etc. This work presents a parallel genetic algorithm (GA) that estimates the optimal parameters of an optical flow algorithm. The multi-channel gradient model (McGM), presented by Johnston et al.[3], is an optical flow algorithm that works well under different problems like static patterns, contrast invariance and different kinds of noise. This model depends on more than ten parameters which are being estimated by the parallel GA.

Multi-channel Gradient Model

This gradient model is based on the basic motion constrain

$$\frac{\partial I}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial t}+\frac{\partial I}{\partial t}=0 \qquad (1)$$

The velocity is computed as the derivative of space over time

$$v=\frac{d\mathbf{x}}{dt}=-\frac{\frac{\partial I}{\partial t}}{\frac{\partial I}{\partial \mathbf{x}}} \qquad (2)$$

The McGM takes into account that the human visual system uses several orders of spatial and temporal derivatives to obtain information on the environment. Therefore, it generates several filters from higher order derivatives to calculate the velocity:

$$v=-\frac{\partial^n I}{\partial \mathbf{x}^{n-1}t}\Big/\frac{\partial^n I}{\partial^n \mathbf{x}} \qquad (3)$$

A truncated Taylor expansion is used to approximate the brightness about a point $I(\mathbf{x},t)$. Each term of the expansion is expressed as filters which are generated using the kernel function:

$$K(r,t)=\frac{1}{4\pi\sigma}e^{-\frac{r^2}{4\sigma}}\frac{1}{\sqrt{\pi}\tau\alpha e^{r^2/4}}e^{-\left(\frac{ln(t/\alpha)}{\tau}\right)^2} \qquad (4)$$

The number of derivatives, the filters rotations and the parameters α, σ and τ, inside the kernel function, are some of the parameters estimated by the GA.

References

[1] S. Baker, D. Scharstein, J.P.Lewis, S.Roth, M.J.Black, and R.Szeliski. A database and evaluation methodology for optical flow. *ICCV93*, pages 1–8, 2007.

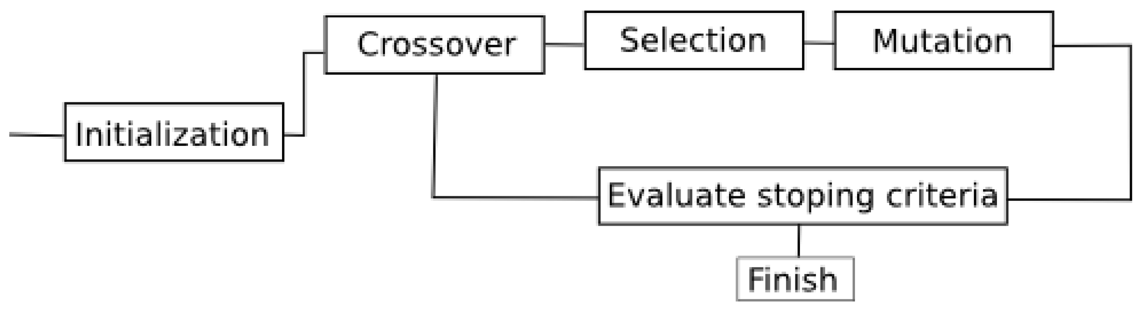
[2] J. Barron and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[3] A. Johnston, P. W. McOwan, and C. P. Benton. Robust velocity computation from a biologically motivated model of motion perception. *Proc. R. Soc. Lond.*, B 266:509–518, March 1999.

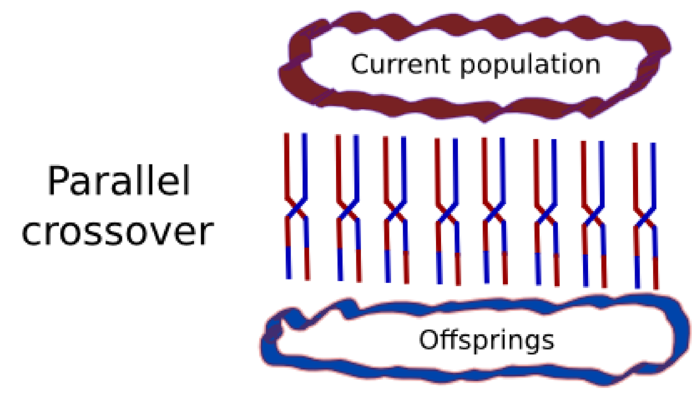
[4] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l^1 optical flow. *Proceedings of the British Machine Vision Conference*, 2009.

Parallel Genetic Algorithm

The proposed GA has four main steps: **initialization**, **crossover**, **selection** and **mutation** The GA iterates until it reaches the stopping criteria

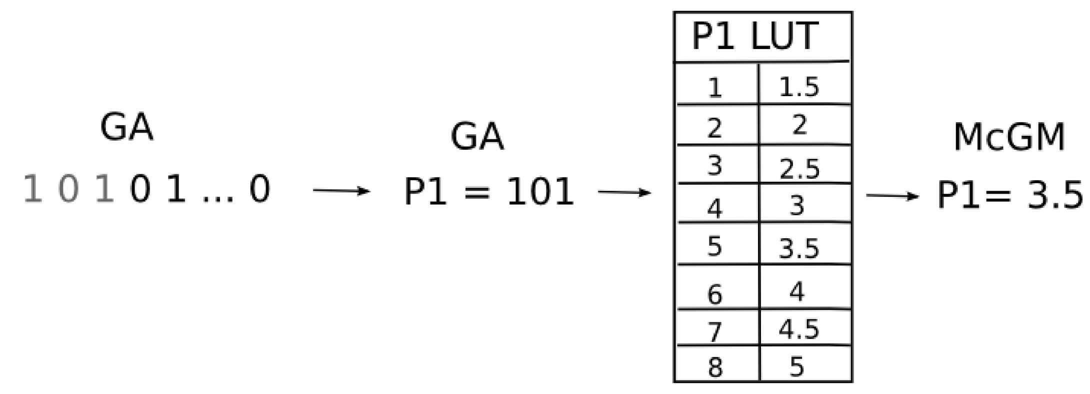


The GA creates individuals during the initialization and crossover steps. This process is the most computationally expensive part of the algorithm and it is implemented in parallel using OpenMP.



Genetic Algorithm with McGM

Individuals are represented by arrays of bits (1's and 0's) which are mapped into sets of parameters:

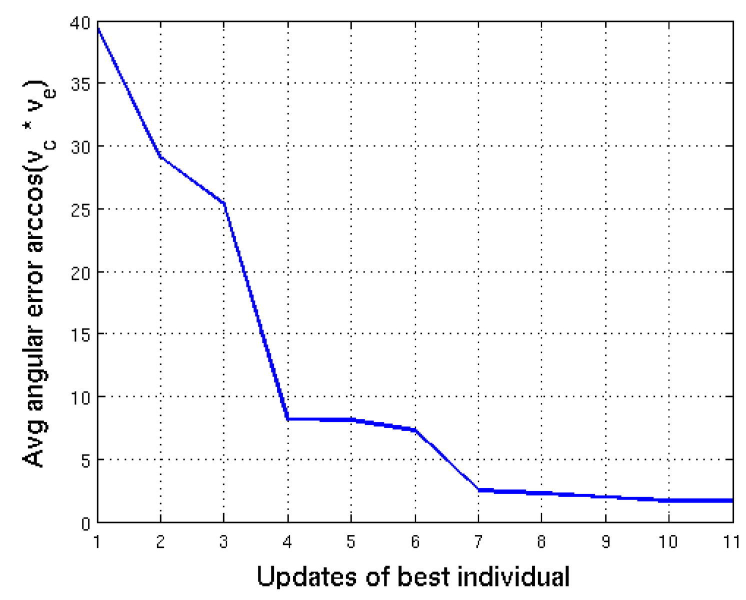


The GA uses the angular error (5), proposed by Barron et al. [2], to evaluate how close the optical flow is from the correct one.

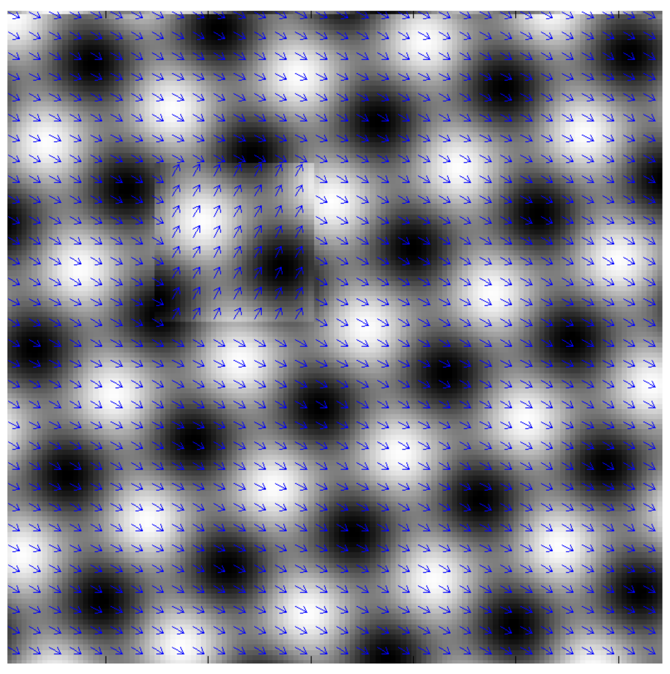
$$\varphi_E = arccos(\vec{v}_c\cdot\vec{v}_e) \qquad (5)$$

Where v_c represents the correct displacement vector and v_e the estimated displacement (the optical flow). The GA works directly with the array of bits so it is not coupled with the McGM and can be used to estimate optimal parameters of other problems.

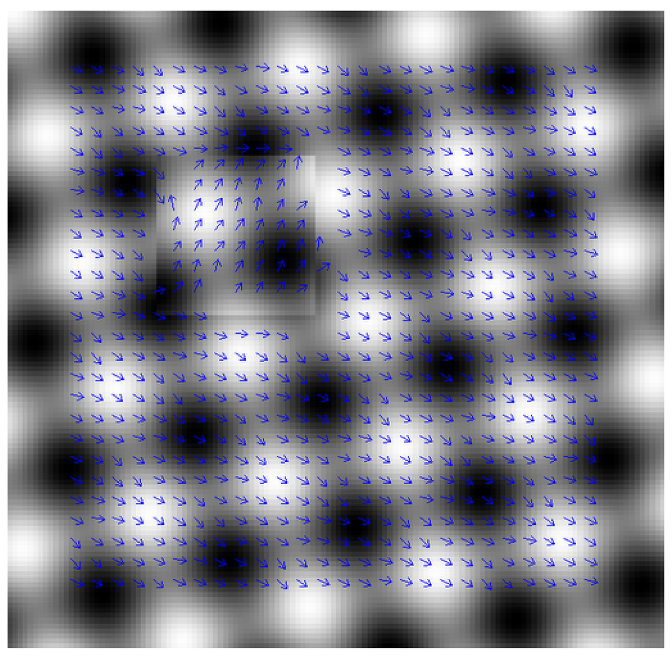
Results



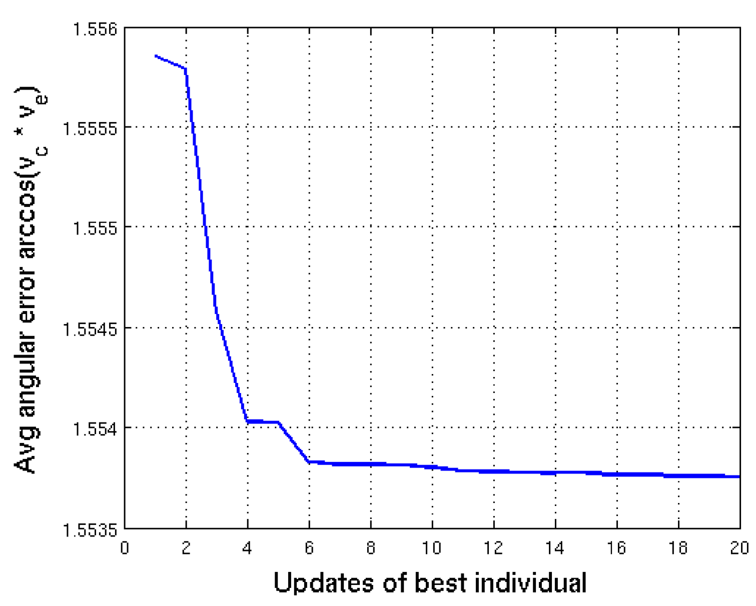
a) GA convergence



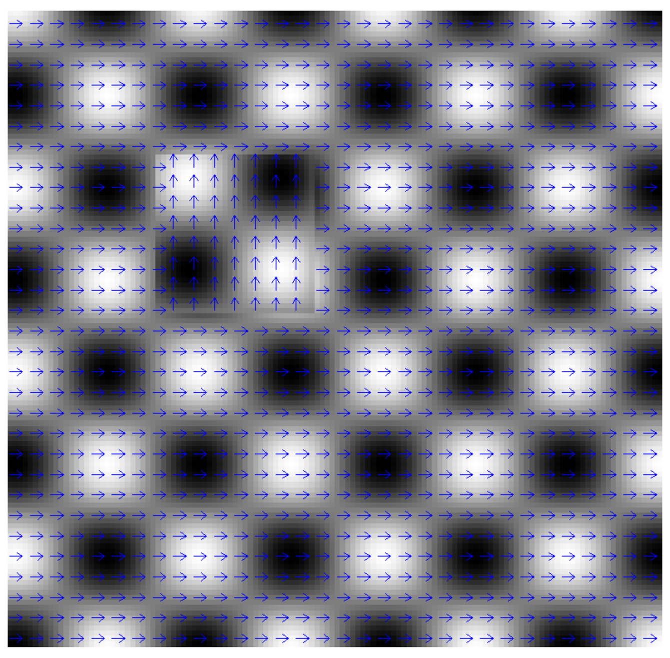
b) Correct optical flow



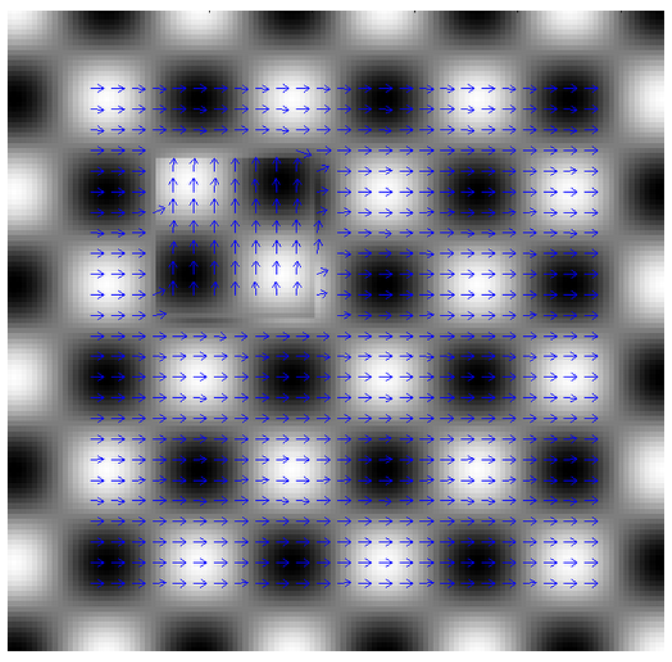
b) Estimated optical flow



a) GA convergence

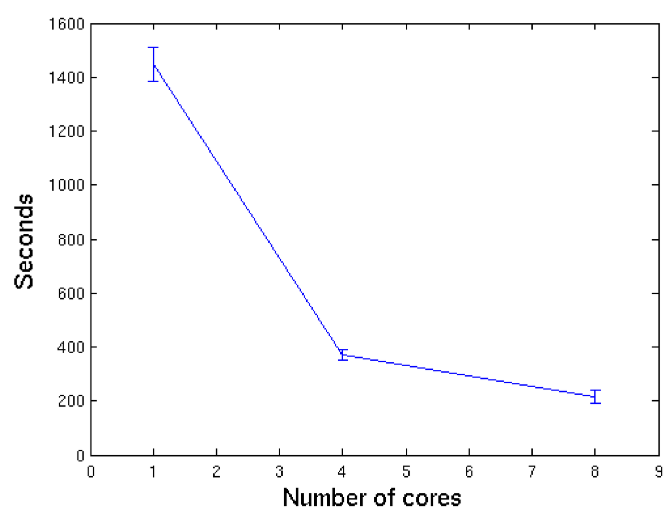


b) Correct optical flow



b) Estimated optical flow

The following results show the time that takes to initialize 100 random individuals using one, four and eight threads with the OpenMP library.



	4 Cores	8 Cores
Speed up	3.91	6.72
Efficiency	97 %	84 %

Conclusion and future work

The proposed GA obtained an average angular error below 1.55 with synthetic sequences of composite sines. The GA implementation achieved an average speedup of 6.72 using 8 threads and an average speedup of 3.91 using 4 threads.

Tests with more challenging videos, like the database proposed by S. Baker et al. [1], are necessary to compare the McGM against newer optical flow algorithms, for example, the Anisotropic Huver-L^1[4]. Another important enhancement is the implementation of the Message Passing Interface (MPI) for the parallel evaluation of individuals.