

RESEARCH

Open Access

# On the optical flow model selection through metaheuristics

Danillo R Pereira<sup>1\*</sup>, José Delpiano<sup>2</sup> and João P Papa<sup>1</sup>

## Abstract

Optical flow methods are accurate algorithms for estimating the displacement and velocity fields of objects in a wide variety of applications, being their performance dependent on the configuration of a set of parameters. Since there is a lack of research that aims to automatically tune such parameters, in this work, we have proposed an optimization-based framework for such task based on social-spider optimization, harmony search, particle swarm optimization, and Nelder-Mead algorithm. The proposed framework employed the well-known large displacement optical flow (LDOF) approach as a basis algorithm over the Middlebury and Sintel public datasets, with promising results considering the baseline proposed by the authors of LDOF.

**Keywords:** Optimization methods; Evolutionary algorithms; Optical flow methods

## 1 Introduction

Optical flow estimation is one of the most important research areas in computer vision, and it aims at identifying the patterns of motion of objects and surfaces in a visual scene, i.e., to approximate the motion field from a time-varying image intensity. The literature is wide, being some very recent works related to optical flow estimation using Laplacian mesh structures [1], total generalized variation [2], probabilistic motion detection [3], and as an optimization problem in a high-dimensional motion field [4], just to name a few. The importance of optical flow estimation can be evidenced in image segmentation [5], rigid object reconstruction [6], cell tracking [7], video stabilization [8], among others. Some parallel-based implementations can be found in [9-11] as well.

Recently, Sun et al. [12] stressed that the theoretical foundations of a broad range of optical flow methods have changed little since the seminal work of Horn and Schunck [13]. Basically, they argued that, although the results have improved over the past years, the vast majority of optical flow methods rely on the same basis of the work proposed by Horn and Schunck. Another shortcoming related to the optical flow-based techniques relies on the estimation of their parameters, which poses a

big challenge to the field. Since most of techniques are parameter-dependent, a grid-search for a set of near-optimal/optimal parameters in video content may not be a viable task [14]. Therefore, many works often set the parameters by hand, which may limit our understanding about how well the considered optical flow method can generalize unseen data. As a matter of fact, the problem of estimating the parameters of optical flow techniques may be seen as a large-scale learning problem. Albeit, we usually need to estimate a few parameters only, the huge amount of data to be processed for such estimation in video datasets demands a high computational effort.

Although the reader can face several works that cope with the problem of estimating/calibrating camera parameters, only a few of them deal with the problem of parameter estimation in optical flow techniques. Heas et al. [15] and Krajsek and Mester [16], for instance, employed a Bayesian optimization framework for such purpose, and Li and Huttenlocher [17] presented an interesting stochastic optimization approach based on Markov random fields for optical flow parameter estimation. The authors state several arguments concerning the advantages of optimizing an error criterion instead of using a maximum likelihood approach for that, as employed by the work of Roth and Black [18]. The reader can refer to a few other works that model the task of optical flow parameter estimation as an optimization task by means of metaheuristic techniques. Delpiano et al. [19], for instance, proposed a

\*Correspondence: dpereira@ic.unicamp.br

<sup>1</sup>São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, 17033-360, Bauru, SP, Brazil

Full list of author information is available at the end of the article

multi-objective approach for parameter estimation aiming at optimizing both the training loss and the computational load. Later on, Pereira et al. [20] applied metaheuristic optimization algorithms for the same task considering the large displacement optical flow (LDOF) technique [21], being the results of social-spider optimization (SSO) [22], harmony search (HS) [23], and particle swarm optimization (PSO) [24] compared against each other in the well-known Middlebury dataset. Although one can find several other optical flow-based implementations out there [25-27], we opted to use LDOF due to its simplicity, reliability, and good rank in Sintel website. LDOF implementation is very accurate, but computationally expensive, thus being an interesting choice for applications that require high accuracy, but does not require very short execution times.

In order to fill the lack of research regarding model selection in optical flow environments, we extended the work of Pereira et al. [20] by adding two more optimization techniques, being one of them based on exact computations called Nelder-Mead (NM) [28] and the other one a 'baseline' using the parameters proposed by the authors of the LDOF technique [21], as well as we added one more dataset to the experimental section. Additionally, the work of Pereira et al. [20] proposed a local approach to estimate the parameters of LDOF: in short, the idea of their work is to optimize each sequence separately and then to employ the best set of parameters to optimize the remaining sequences. We propose here to optimize the techniques globally, which means we consider all sequences from a given dataset for parameter optimization, being the results more accurate than the ones reported by Pereira et al. [20]. The remainder of this paper is organized as follows: Section 2 presents a brief theoretical background about optical flow, and Section 3 revisits the techniques employed in this paper for comparison purposes. The proposed methodology and experimental results are discussed in Sections 4 and 5, respectively. Section 6 states the conclusions and future works.

## 2 Optical flow

Optical flow (OF) is a vector field representing 'the distribution of apparent velocities of movement of brightness patterns in an image' [13]. The idea contains two basic assumptions: the 'grey value constancy' and the 'smooth flow of the intensity values' between two successive images. Some articles still maintain the grey value constancy (as an example, see [29]), while other works report the necessity to loosen this assumption [30].

The OF constraint, given in Equation 1, is derived from the 'grey value constancy' assumption. It relates the spatial and temporal derivatives of a 2D image  $g$  at time step  $t$  and the OF vector  $\phi$ , and it has a strong analogy with mass conservation in fluid mechanics, shown

in Equation 2, where  $\phi$  is the fluid speed and  $\rho$  is the fluid density. As fluid mass, image intensity is often supposed to remain constant under deformation and motion. However, Equations 1 and 2 would be equivalent when  $\nabla\phi = 0$  only. This condition matches the smooth flow assumption that is considered when regularizing the flow field:

$$\frac{\partial g}{\partial t} + \phi \nabla g = 0, \quad (1)$$

$$\frac{\partial \rho}{\partial t} + \nabla(\rho\phi) = 0. \quad (2)$$

The early work presented in [13] stated the need for an extra constraint to compute the optical flow field from an image sequence and proposed one *ad hoc* constraint based on the assumption of flow smoothness. Another research work [31] proposed to consider the OF equation for several neighboring pixels in order to avoid the need for an extra constraint. More than 10 years later, Barron et al. [32] provided a comparison of several OF methods, mainly with respect to their average angular error (AAE) when applied to some image sequences. The experiments showed that the method in [31] was one of the most reliable methods at the moment.

Recently, several image datasets have been compiled for a more precise evaluation and comparison of OF methods [33,34]. Many shortcomings of the original methods have been overcome, and the accuracy of OF methods on the top of the rankings has grown continuously. Additionally, several researchers have tried to preserve the discontinuity of natural motion fields [35], overcoming the original assumption of OF smoothness in [13]. After the work by [32], there have been further attempts to compare different methods. Liu et al. [36], for instance, showed a trade-off between computational time and angular error using operation curves to compare different OF techniques. It is also interesting to consider the time comparison among OF algorithms given by [37], since the authors provide a picture of the computational load of some OF algorithms. More recently, a group of researchers presented a series of real image sequences and their respective ground truth obtained by tracking hidden fluorescent textures [33]. The authors also suggest a method to evaluate OF-based algorithms.

### 2.1 Large displacement optical flow

Given a sequence of  $m$  frames  $\mathcal{I} = \{I_1, I_2, \dots, I_m\}$ , let  $\phi = (a, b)^T$  be the optical flow for a pair of consecutive frames  $I_i, I_{i+1}$ ,  $i = 1, 2, \dots, m - 1$ , being such frame pre-smoothed using a Gaussian filter with parameter  $\sigma$ . The large displacement optical flow method proposed by Brox and Malik [21] solves the energy functional given by:

$$E(\phi) = E_{\text{color}}(\phi) + \gamma E_{\text{gradient}}(\phi) + \alpha E_{\text{smooth}}(\phi) + \beta E_{\text{match}}(\phi, \phi_1) + E_{\text{desc}}(\phi_1), \quad (3)$$

where the term  $E_{\text{color}}$  represents the common assumption of grey value or color constancy;  $E_{\text{gradient}}$  represents gradient constancy, which is invariant to a uniform illumination change;  $E_{\text{smooth}}$  enforces regularity of the resulting optical flow;  $E_{\text{match}}$  stands for an energy related to point correspondences; and the minimization of  $E_{\text{desc}}$  assures descriptor matching. The quantity  $\phi_1$  is an auxiliary variable which allows integrating descriptor matching into a continuous approach. The implementation available for LDOF [38] has a reduced number of parameters, which means we can consider all of them for optimization purposes. Such implementation allows the user to fine-tune four parameters: (i)  $\sigma$  is related to the Gaussian pre-smoothing of the images (pre-processing parameter), (ii)  $\alpha$  controls the importance attributed to smoothness of the resulting optical flow, (iii)  $\beta$  enforces the matching of points in both images, and (iv)  $\gamma$  regulates the penalization of violations to the gradient constancy assumption. It is important to highlight that this set of parameters influences significantly the accuracy (consequently the error metrics) and the computational load.

### 3 Optimization background

In this section, we describe the techniques employed in this paper for comparison purposes. The methods can be divided in two classes: (i) metaheuristic algorithms and (ii) exact methods. Concerning the former approaches, we used social-spider optimization, particle swarm optimization, and harmony search, and with respect to exact methods, we employed the Nelder-Mead, which is a deterministic algorithm for convex functions that employs a simplex for optimization purposes.

#### 3.1 Social-spider optimization

Social-spider optimization is based on the cooperative behavior of social spiders [22], and it takes into account two genders of search spiders: males and females. Depending on the gender, each agent is conducted by a set of different operators emulating a cooperative behavior in a colony. The search space is assumed as a communal web, and a spider's position represents an optimal (near optimal) solution.

An interesting characteristic of social spiders is the female-biased population. The number of male spiders hardly reaches 30% of the total colony members. The number of females  $N_f$  is randomly selected within a range of 65% to 90% of the entire population  $N$ , being calculated as follows:

$$N_f = [(0.9 - \xi 0.25)N], \quad (4)$$

where  $\xi \sim \mathcal{U}(0, 1)$ . The number of male spiders  $N_m$  is given by:

$$N_m = N - N_f. \quad (5)$$

Each spider  $i$  receives a weight  $\phi_i$  according to the fitness value of its solution:

$$\phi_i = \frac{\text{fitness}_i - \text{worst}}{\text{best} - \text{worst}}, \quad (6)$$

where  $\text{fitness}_i$  is the fitness value obtained by the evaluation of the  $i$ th spider's position  $i = 1, 2, \dots, N$ . The worst and best mean the worst fitness value and best fitness value of the entire population, respectively.

The communal web is used as a mechanism to transmit information among the colony members. The information is encoded as small vibrations and depends on the weight and distance of the spider which have generated them:

$$V_{i,j} = \phi_j e^{-d_{i,j}^2}, \quad (7)$$

where  $d_{i,j}$  is the Euclidean distance between the spider  $i$  and  $j$ . We can consider three special relationships:

- The vibrations  $V_{i,c}$  are perceived by the spider  $i$  as a result of the information transmitted by the member  $c$  who is the nearest member to  $i$  and possesses a higher weight  $\phi_c > \phi_i$ ;
- The vibrations  $V_{i,b}$  perceived by the spider  $i$  as a result of information transmitted by the spider  $b$  holding the best weight of the entire population;
- The vibrations  $V_{i,f}$  perceived by the spider  $i$  as a result of the information transmitted by the nearest female  $f$ .

Social spiders perform cooperative interaction over other colony members depending on the gender. In order to emulate the cooperative behavior of the female spider, a new operator is defined in Equation 8. The movement of attraction or repulsion  $\varphi_i$  of a female spider  $i$  at time step  $t + 1$  is developed over other spiders according to their vibrations, which are emitted over the communal web:

$$\varphi_i(t+1) = \begin{cases} \varphi_i(t) + \alpha * V_{i,c} * (s_c - \varphi_i(t)) + \beta * V_{i,b} * (s_b - \varphi_i(t)) + \gamma * (\text{rand} - \frac{1}{2}) & \text{if } \theta < \text{PF}; \\ \varphi_i(t) - \alpha * V_{i,c} * (s_c - \varphi_i(t)) - \beta * V_{i,b} * (s_b - \varphi_i(t)) + \gamma * (\text{rand} - \frac{1}{2}) & \text{if } \theta \geq \text{PF}, \end{cases} \quad (8)$$

where  $\theta, \alpha, \beta, \gamma$ , and  $\text{rand}$  are uniform random numbers between  $[0, 1]$ , PF is an input parameter, and  $s_c$  and  $s_b$  represent the nearest member to  $i$  that holds a higher weight and the best spider of the entire population, respectively.

The male spider population is divided into two classes: dominant and non-dominant. The dominant class spider has better fitness in comparison to non-dominant,

and they are attracted to the closest female spider in the communal web. On the other hand, non-dominant male spiders tend to concentrate in the center of the male population as a strategy to take advantage of resources that are wasted by dominant males. The movement of male spiders is given by:

$$\delta_i(t+1) = \begin{cases} \delta_i(t) + \alpha * V_{if} * (s_f - \delta_i(t)) + \gamma * (\text{rand} - \frac{1}{2}) & \text{if } \phi_{N_f+i} > \tilde{\phi}; \\ \delta_i(t) + \alpha * \left( \frac{\sum_{h=1}^{N_m} \delta_h(t) * \phi_{N_f+h}}{\sum_{h=1}^{N_m} \phi_{N_f+h}} \right) & \text{if } \phi_{N_f+i} \leq \tilde{\phi}, \end{cases} \quad (9)$$

where  $s_f$  represents the nearest female spider to the male spider  $i$  and  $\tilde{\phi}$  is the median weight of male spider population. Thus, the reader can observe that we have distinct movement equations for male and female spiders. Notice that we are using  $\phi_{N_f+i}$  to denote the male spiders, since we consider  $\phi$  as a vector containing the fitness of every spider within the web, being the first  $N_f$  spiders the female ones.

Mating is performed by dominant males and female members in a social-spider colony. Considering  $r$  (calculated by Equation 10) as being the radius, when a dominant male spider locates female members inside  $r$ , it mates, forming a new brood:

$$r = \frac{\sum_{j=1}^n l_j^{\text{high}} - l_j^{\text{low}}}{2n}, \quad (10)$$

where  $n$  is the dimension of the problem, and  $l_j^{\text{high}}$  and  $l_j^{\text{low}}$  are the upper and lower bounds, respectively. Once the new spider is formed, it is compared to the worst spider of the colony. If the new spider is better, the worst spider is replaced by the new one.

### 3.2 Harmony search

Harmony search is a metaheuristic technique based on the improvisation process of musicians searching for a good harmony [39]. The main idea is to generate a new harmony  $h_{\text{new}} = (h_{\text{new}}^1, h_{\text{new}}^2, \dots, h_{\text{new}}^N)$  at each iteration, based on memory considerations and pitch adjustment. In this case,  $N$  stands for the number of decision variables to be optimized.

The idea of the memorization step is to model the process of creating songs, in which the musician can use his/her memories of good musical notes to create a new song. This process is modeled by the harmony memory considering rate (HMCR), as follows:

$$h_{\text{new}}^j \leftarrow \begin{cases} h_{\text{new}}^j \in \{h_{\text{new}}^1, \dots, h_{\text{new}}^M\} & \text{with probability HMCR} \\ h_{\text{new}}^j \in \psi_j & \text{with probability (1-HMCR)}, \end{cases} \quad (11)$$

where  $M$  and  $\psi_j$  are the number of harmonies and the set of ranges for each decision variable  $j$ , respectively. Therefore,  $\text{HMCR} \in [0, 1]$  is the probability of choosing one value from the historic values stored in the harmony memory, and  $(1-\text{HMCR})$  is the probability of randomly choosing one feasible value. Further, if the new harmony has been created with probability HMCR, every component  $j$  of the new harmony vector  $h_{\text{new}}$  is examined to determine whether it should be pitch-adjusted or not, which is controlled by the pitch adjusting rate (PAR) variable:

$$h_{\text{new}}^j \leftarrow \begin{cases} \text{Yes, with probability PAR} \\ \text{No, with probability (1-PAR)}. \end{cases} \quad (12)$$

The pitch adjustment is often used to improve solutions and to avoid local optima. This mechanism concerns shifting the neighbouring values of some decision variable in the harmony. As such, if the pitch adjustment decision for the decision variable  $h_{\text{new}}^j$  is Yes, then  $h_{\text{new}}^j$  is replaced as follows:

$$h_{\text{new}}^j \leftarrow h_{\text{new}}^j + \delta_j \tau, \quad (13)$$

where  $\tau$  is an arbitrary distance (bandwidth) for the continuous design variable, and  $\delta_j \sim \mathcal{U}(0, 1)$  is an *ad hoc* parameter.

Recently, several researches have focused on developing variants of traditional HS. In our implementation, we employed the novel global harmony search (NGHS) [40], which has demonstrated better results than vanilla HS in our experiments. The NGHS does not employ PAR and HMCR parameters, but it introduces a new parameter  $P$  that denotes the probability of occurring an improvisation schema during a new harmony's creation, and therefore modifies the improvisation process. Another difference between NGHS and the HS is that a new harmony always replaces the worst one, even when the new one does not improve the worst harmony.

### 3.3 Particle swarm optimization

Particle swarm optimization can be seen as a search algorithm based on stochastic processes [24], where the learning of social behavior allows each possible solution (particle) 'fly' onto that space (swarm) looking for other particles that have the best features and thus minimizing or maximizing the objective function.

Each particle has a memory that stores its best local solution (local maxima or minima) and the best global solution (global maximum or minimum). Besides, each particle has the ability to imitate others that provide the best positions in the swarm. This mechanism can be summarized in three principles: (i) evaluation, (ii) comparison, and (iii) imitation. Each particle can evaluate others within your neighborhood through some objective function; it

can compare with your own value and finally decide whether it is a good choice to imitate it or not.

The swarm is modeled as a multidimensional space  $\mathbb{R}^N$ , where each particle  $l_i = (\lambda_i, \kappa_i) \in \mathbb{R}^N$  has two main features: (i) position  $\lambda_i$  and (ii) velocity  $\kappa_i$ . The best local  $\hat{\lambda}_i$  and global  $\hat{G}$  solutions (position in the swarm) are also known. After setting the size of the swarm (the number of particles), each particle is initialized with random values for both velocity and position. Each particle is then evaluated with respect to some objective function, and its local maxima/minima is updated. The global maximum/minimum value is updated with the particle that reached the best position in the swarm. This process is repeated until some convergence criterion is met. The position and velocity of the particle  $l_i$  at time step  $t + 1$  are updated by Equations 14 and 15, respectively:

$$\lambda_i^{t+1} = \lambda_i^t + \kappa_i^t, \tag{14}$$

and

$$\kappa_i^{t+1} = \Psi \kappa_i^t + c_1 r_1 (\hat{\lambda}_i - \lambda_i^t) + c_2 r_2 (\hat{G} - \lambda_i^t) \tag{15}$$

where  $\Psi$  is the inertia force that controls the interaction power between particles, and  $r_1, r_2 \in [0, 1]$  are random variables that give the idea of stochasticity concerning PSO. The constants  $c_1$  and  $c_2$  are also used to guide the particles (input parameters for the algorithm) onto good solutions.

### 3.4 Nelder-Mead method

The Nelder-Mead is an iterative heuristic of direct search approach (it does not compute derivatives) used to find stationary points (minimum or maximum) in multidimensional unconstrained functions [28]. This approach is commonly used in problems where the derivative is not known, or when the computational cost to compute it is prohibitive.

Given a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and an initial guess  $\mathbf{x}^0$ , the Nelder-Mead method creates a simplex  $S^0 = \{p_0, p_1, \dots, p_n\} \in \mathbb{R}^n$  around the initial guess  $\mathbf{x}^0$  with  $n + 1$

sample points. There are different approaches to generate an initial simplex  $S^0$ , and its size can influence the solution to be obtained. In our implementation, we generate the initial simplex  $S^0$  using the classical approach described by Equation 16:

$$\begin{aligned} \mathbf{p}_0 &= \mathbf{x}^0 \text{ and} \\ \mathbf{p}_j &= \mathbf{p}_0 + s e_j \quad j \in \{1, 2, \dots, n\}, \end{aligned} \tag{16}$$

where  $s$  is the step size that determines the simplex size and  $e = \{1, 1, \dots, 1\} \in \mathbb{R}^n$  is a diagonal vector with size  $\sqrt{n}$ . Thus, the initial simplex  $S^0$  has all edges with the same size  $s$ .

After the construction of simplex  $S^i$ , the Nelder-Mead starts the iterative process to find a stationary point  $\mathbf{x}^*$ . The first step is to compute all sample values  $f_j = f(\mathbf{p}_j) \forall 0 \leq j \leq n$ . Next, we determine the indices  $w, v$ , and  $b$ , which represent the worst, second worst, and best samples' indexes, respectively. Soon after, we compute the centroid  $c = \frac{1}{n} \sum_{j \neq w} p_j$  of all sample points except the worst once.

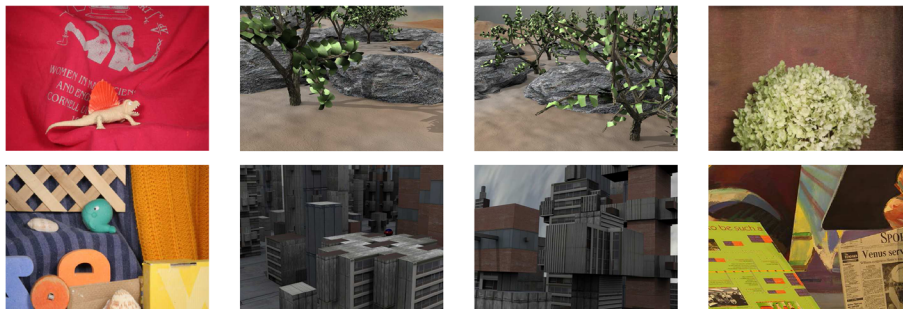
Further, we compute the reflect point  $\mathbf{p}_r = c + \vartheta(c - \mathbf{p}_w)$ : if  $f_b \leq f_r < f_w$ , then we replace the simplex sample  $\mathbf{p}_w$  by  $\mathbf{p}_r$ , and the iteration ends. Otherwise, if  $f_r < f_b$ , we compute the expansion point  $\mathbf{p}_e = c + \zeta(\mathbf{p}_r - c)$  and its sample value  $f_e = f(\mathbf{p}_e)$ . If  $f_e < f_r$ , then we select  $p_e$  and discard  $p_w$ ; otherwise, we accept  $p_r$  and discard  $p_w$ . Now, if  $f_r \geq f_w$ , we compute the contraction point  $\mathbf{p}_c$  (Equation 17) using the best sample between  $\mathbf{p}_r$  and  $\mathbf{p}_v$ :

$$\begin{aligned} \mathbf{p}_c &= c + \varphi(\mathbf{p}_r - c) && \text{if } (f_v \leq f_r \leq f_w) \\ \mathbf{p}_c &= c + \varphi(\mathbf{p}_w - c) && \text{if } (f_r \geq f_w). \end{aligned} \tag{17}$$

We denote  $\mathbf{p}_{brv}$  as the point with the lowest sample value between  $\mathbf{p}_r$  and  $\mathbf{p}_v$ , i.e.,  $\mathbf{p}_{brv} = \mathbf{p}_r$  if  $f_r \leq f_w$ , and  $\mathbf{p}_{brv} = \mathbf{p}_w$  otherwise. If  $f_c \leq f_{brv}$ , we accept  $\mathbf{p}_c$ ; otherwise, it is necessary to create a new shrink simplex, which can be calculated by updating the vertices as follows:

$$\mathbf{p}_j = \mathbf{p}_j + \rho(\mathbf{p}_j - \mathbf{p}_b), \quad \forall j \neq b, \tag{18}$$

where  $j = 1, 2, \dots, n$ . The iterative process is repeated until the maximum number of iterations is reached,



**Figure 1** Images from the Middlebury dataset used in the experiments. From left to right: Dimetrodon, Grove2, Grove3, Hydrangea, Urban2, Urban3, RubberWhale, and Venus.



**Figure 2** Images from the Sintel dataset used in the experiments. From left to right: alley\_1, ambush\_2, bamboo\_1, bandage\_1, cave\_2, market\_2, mountain\_1, shaman\_1, sleeping\_1, and temple\_2.

or some convergence criterion is met. Notice that the Nelder-Mead algorithm has the following parameters:  $\vartheta$ ,  $\varphi$ ,  $\rho$ , and  $\zeta$ .

#### 4 Methodology

This section describes the experimental setup employed in this paper to validate the optimization algorithms to set up parameters of LDOF. We used two well-known public datasets composed of image sequences and their respective ground truths: Middlebury [33,41] and Sintel [42,43], which have been frequently used to evaluate different OF methods [21,33]. The Middlebury dataset contains eight synthetic and laboratory sequences with a dense ground truth (Figure 1), and the Sintel dataset contains artificial naturalistic video sequence (Figure 2).

We employed the LDOF technique (Section 2.1) together with our implementation of SSO, NGHS, PSO, and NM. The main reason behind the use of such techniques is to alleviate the high computational burden often required by optimization techniques. In light of such shortcoming, we opted to use techniques with easy implementation, which usually reflects in their complexity. For the sake of comparison, we computed the average of ‘end point error’ (EPE) [44] values obtained over five runnings for each optimization technique, which is basically the difference between the ground truth and estimated optical flow.

Let  $u_e = (u_e, v_e)$  be the estimated optical flow, and  $u_{gt} = (u_{gt}, v_{gt})$  be the ground truth of the optical flow. Therefore, the EPE can be calculated as follows:

$$\text{EPE} = \sqrt{(u_e - u_{gt})^2 + (v_e - v_{gt})^2}. \quad (19)$$

Table 1 presents the parameters used for each of them: NM parameters were set according to the work of Lagarias et al. [45]. Additionally, we also employed LDOF with the parameters recommended by Brox and Malik [21], in which we refer here as the ‘baseline.’ SSO, NGHS, and PSO parameters were fine-tuned according to the work of Pereira et al. [20]. A search space with 20 agents and 200

iterations for SSO, NGHS, and PSO, and 100 iterations for NM.<sup>a</sup> Since the solution of NM algorithm is strongly influenced by the initial guess, we used random initial guesses for that.

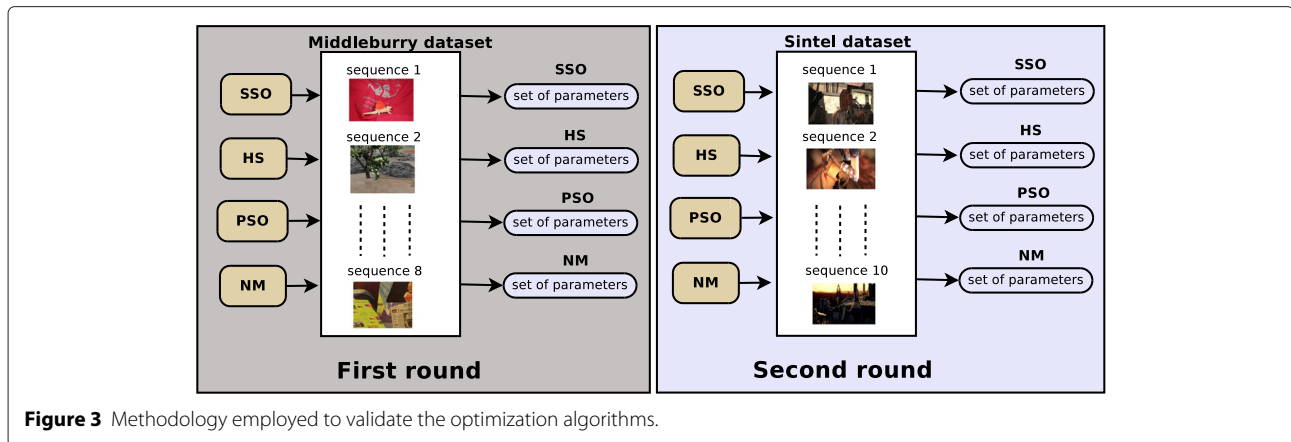
Roughly speaking, the main idea is to find out the set of LDOF parameters that minimize the EPE measure. Therefore, instead of employing a random or empirical approach for that, we make use of an optimization framework to perform a faster and more reliable search for such parameters. As such, the fitness function to be minimized is the one given by EPE measure. The experiments were divided in two rounds, as depicted in Figure 3. In the first round, we estimated the best set of parameters (the ones with minimum EPE) using the aforementioned optimization algorithms applied on the eight Middlebury sequences. In the second round, we applied the same algorithms on ten sequences of images from the Sintel dataset. In order to compare the optimization algorithms, we also employed a ‘baseline’ set of parameters proposed by Brox and Malik [21].

The methodology employed in this paper differs from the one used by Pereira et al. [20], which optimized each dataset image individually, i.e., they aimed at fine-tuning LDOF for each image, being the final result the average over all images considering the AAE metric. In this work, we conducted the optimization process over the whole dataset, i.e., we aimed at fine-tuning LDOF considering all images of the dataset at the same time. Therefore, the fitness function adopted in this work was the one given by the average of EPE values of all dataset images.

**Table 1** Parameters used for each optimization technique

Technique	Parameters
SSO	PF = 0.5
NGHS	$\rho = 0.1$
PSO	$c_1 = c_2 = 2.0$ and $\psi = 0.9$
NM	$\vartheta = 1, \varphi = \rho = 0.5$ and $\zeta = 2$





### 5 Experimental results

This section presents the results obtained by SSO, NGHS, PSO, and NM for optical flow parameter optimization purposes. We would like to stress that we did not consider the runtime (computational load), since our goal is to minimize the EPE metric only. Furthermore, the parameters to be optimized have a strong influence on both EPE and runtime.

In regard to the first round of experiments, Table 2 shows the EPE values concerning SSO, NGHS, PSO, NM, and LDOF baseline in the eight ground-truth image sequences of the Middlebury dataset. In the first round, PSO obtained the best average results with EPE equals to 0.325, followed by SSO (EPE equals to 0.330). Notice both methods presented better results than the baseline approach. Additionally, PSO obtained the best results in three out of eight Middlebury sequences (RubberWhale, Urban3, and Urban2), SSO achieved the best results in three out of eight Middlebury sequences (Dimetrodon, Grove3, and Venus), and the baseline achieved the best results for two sequences (Grove2 and Hydrangea). NGHS and NM did not achieve the best result in any image sequence. Although Brox and Malik [21] did not present the methodology used to find the baseline parameters,

this experiment highlighted the need for a fine-tune of parameters using optimization algorithms.

In regard to the second round of experiments, Table 3 shows the EPE values concerning SSO, NGHS, PSO, NM, and baseline on ten image sequences considering the Sintel dataset. In this experiment, we used the first two frames of the following sequences: alley\_1, ambush\_2, bamboo\_1, bandage\_1, cave\_2, market\_2, mountain\_1, shaman\_1, sleeping\_1, and temple\_2. We can observe that the optimization techniques presented similar results, being all of them more accurate than the baseline (except for cave\_2, where the baseline approach achieved similar results to PSO). PSO obtained the best results in four out of ten Sintel sequences (alley\_1, bamboo\_1, cave\_2, and temple\_2), SSO achieved the best results in three out ten Sintel sequences (bandage\_1, market\_2, and shaman\_1), NGHS obtained the best result in one out of ten Sintel sequences (ambush\_2).

Figure 4 depicts the average EPE values considering all sequences for the Middlebury and Sintel datasets, as well as the average between these two. Therefore, the main idea of this work is to highlight the importance of using optimization algorithms to fine-tune the parameters for OF-based techniques. Considering the average

**Table 2 Results obtained by SSO, NGHS, PSO, NM, and LDOF baseline [21] over Middlebury dataset**

Sequences	SSO	NGHS	PSO	NM	Baseline
Dimetrodon	<i>0.108 ± 0.02</i>	0.114 ± 0.01	0.114 ± 0.01	0.135 ± 0.03	0.115
Grove2	0.194 ± 0.00	0.205 ± 0.01	0.191 ± 0.01	0.224 ± 0.03	<i>0.176</i>
Grove3	<i>0.670 ± 0.04</i>	0.747 ± 0.02	0.685 ± 0.01	0.761 ± 0.05	0.701
Hydrangea	0.211 ± 0.00	0.211 ± 0.02	0.202 ± 0.01	0.235 ± 0.05	<i>0.184</i>
RubberWhale	0.126 ± 0.01	0.157 ± 0.00	<i>0.117 ± 0.00</i>	0.149 ± 0.04	0.129
Urban2	0.401 ± 0.04	0.426 ± 0.06	<i>0.346 ± 0.01</i>	0.459 ± 0.12	0.381
Urban3	0.614 ± 0.06	0.756 ± 0.11	<i>0.561 ± 0.14</i>	0.867 ± 0.14	0.818
Venus	<i>0.319 ± 0.02</i>	0.756 ± 0.01	0.381 ± 0.00	0.403 ± 0.03	0.377
Mean	0.330 ± 0.01	0.372 ± 0.00	<i>0.325 ± 0.01</i>	0.384 ± 0.02	0.360

The best EPE values for each sequence are in italics.

**Table 3 Results obtained by SSO, NGHS, PSO, NM, and LDOF baseline [21] over Sintel dataset**

Sequences	SSO	NGHS	PSO	NM	Baseline
alley_1	0.214 ± 0.02	0.263 ± 0.00	0.209 ± 0.01	0.226 ± 0.00	0.212
ambush_2	15.532 ± 0.50	13.947 ± 1.57	16.210 ± 1.39	14.53 ± 2.05	19.64
bamboo_1	0.246 ± 0.01	0.275 ± 0.02	0.236 ± 0.01	0.239 ± 0.00	0.238
bandage_1	0.740 ± 0.06	0.990 ± 0.04	0.915 ± 0.20	0.817 ± 0.04	0.785
cave_2	1.857 ± 0.07	1.992 ± 0.10	1.819 ± 0.04	1.899 ± 0.04	1.820
market_2	0.846 ± 0.00	0.928 ± 0.03	0.972 ± 0.16	0.853 ± 0.02	0.852
mountain_1	0.865 ± 0.06	0.798 ± 0.04	1.076 ± 0.33	0.741 ± 0.06	0.891
shaman_1	0.230 ± 0.02	0.297 ± 0.03	0.249 ± 0.01	0.278 ± 0.01	0.256
sleeping_1	0.114 ± 0.01	0.122 ± 0.01	0.110 ± 0.01	0.105 ± 0.01	0.102
temple_2	1.536 ± 0.05	1.612 ± 0.05	1.485 ± 0.01	1.563 ± 0.03	1.523
Mean	2.218 ± 0.05	2.122 ± 0.12	2.293 ± 0.16	2.185 ± 0.12	2.632

The best EPE values obtained for each sequence are in italics.

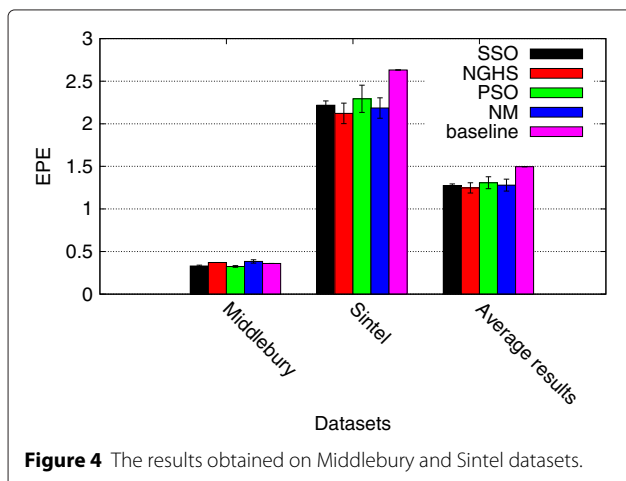
results of both experiments, all optimization techniques obtained better results than the baseline. Furthermore, the experiments show that the parameters shall be selected specifically for each dataset or application.

An additional experiment showed the computational load of each technique, which is measured here in terms of the number of calls to the LDOF algorithm and presented in Figure 5. If we are interested in a fast model selection, the best approach might be NGHS, since it has obtained reasonable results with less computational effort than swarm-based approaches. However, if we decide to apply an off-line fine-tuning, both SSO and PSO seem to be interesting approaches, being the former slightly more accurate.

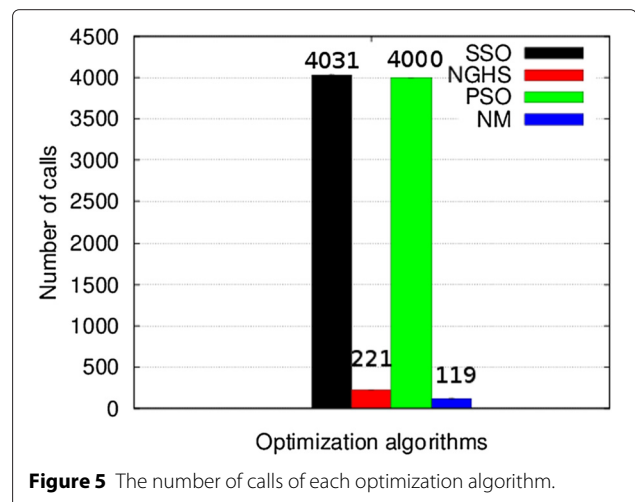
### 6 Conclusions

In this paper, we have validated the optimization algorithms in the context of model selection in optical flow-based applications, which play an important role in computer vision systems. The experimental section

compared the baseline parameters obtained by Brox and Malik [21] against with four optimization techniques: SSO, NGHS, PSO, and NM. Two rounds of experiments have been conducted over the well-known Middlebury and Sintel datasets: (i) the first round aimed at learning the best set of parameters (i.e., the ones that minimizes the end point error criterion) over the Middlebury dataset and (ii) the second phase performed the same over the Sintel dataset. In the first round, two optimization algorithms (SSO and PSO) achieved better results than the baseline parameters, and in the second round, all optimization algorithms achieved better results than the baseline. Therefore, this paper highlighted the need for an automatic fine-tuning of the parameters of optical flow techniques. In addition, the computational load of the compared techniques have been assessed in terms of the number of calls to the LDOF technique, evidencing the lower computational burden of NGHS and NM techniques.



**Figure 4** The results obtained on Middlebury and Sintel datasets.



**Figure 5** The number of calls of each optimization algorithm.



## Endnote

<sup>a</sup>The number of agents and iterations have been chosen based on previous experiments [20].

### Competing interests

The authors declare that they have no competing interests.

### Acknowledgements

The authors are grateful to FAPESP grants #2013/20387-7 and #2014/16250-9, CNPq grants #303182/2011-3, #470571/2013-6, and #306166/2014-3, and Universidad de los Andes FAI grant #05/2013.

### Author details

<sup>1</sup>São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, 17033-360, Bauru, SP, Brazil. <sup>2</sup>University of the Andes, Mons. Álvaro del Portillo, 12445, Santiago, Chile.

Received: 17 December 2014 Accepted: 14 April 2015

Published online: 09 May 2015

### References

- W Li, D Cosker, M Brown, T R, in *IEEE Conference on Computer Vision and Pattern Recognition*. Optical flow estimation using Laplacian mesh energy (IEEE Press, DC, USA, 2013), pp. 2435–2442
- R Ranftl, K Bredies, T Pock, in *European Conference on Computer Vision*. Lecture Notes in Computer Science, ed. by D Fleet, T Pajdla, B Schiele, and T Tuytelaars. Non-local total generalized variation for optical flow estimation, vol. 8689 (Springer, New York, 2014), pp. 439–454
- J An, SJ Ha, NI Cho, Probabilistic motion pixel detection for the reduction of ghost artifacts in high dynamic range images from multiple exposures. *EURASIP J. Image Video Process.* **2014**(1), 1–15 (2014)
- M Hornáček, F Besse, J Kautz, A Fitzgibbon, C Rother, in *European Conference on Computer Vision*. Lecture Notes in Computer Science, ed. by D Fleet, T Pajdla, B Schiele, and T Tuytelaars. Highly overparameterized optical flow using patchmatch belief propagation, vol. 8691 (Springer, New York, 2014), pp. 220–234
- M Narayana, A Hanson, E Learned-Miller, in *IEEE International Conference on Computer Vision*. Coherent motion segmentation in moving camera videos using optical flow orientations (IEEE Press, DC, USA, 2013), pp. 1577–1584
- E Ilg, R Kümmerle, W Burgard, T Brox, in *IEEE International Conference on Robotics and Automation*. Reconstruction of rigid body models from motion distorted laser range data using optical flow (IEEE Press, DC, USA, 2014), pp. 1–6
- G Dongmin, AL van de Ven, X Zhou, Red blood cell tracking using optical flow methods. *IEEE J. Biomed. Health Informatics.* **18**(3), 991–998 (2014)
- S Liu, L Yuan, P Tan, J Sun, in *IEEE Conference on Computer Vision and Pattern Recognition*. SteadyFlow: spatially smooth optical flow for video stabilization (IEEE Press, DC, USA, 2014), pp. 4209–4216
- F Valentiniotti, G Di Caro, B Crespi, Real-time parallel computation of disparity and optical flow using phase difference. *Machine Vision Appl.* **9**(3), 87–96 (1996)
- M Fleury, AF Clark, AC Downton, Evaluating optical-flow algorithms on a parallel machine. *Image Vision Comput.* **19**(3), 131–143 (2001)
- A Garcia-Dopico, JL Pedraza, M Nieto, A Pérez, S Rodríguez, J Navas, Parallelization of the optical flow computation in sequences from moving cameras. *EURASIP J. Image Video Process.* **2014**(1), 1–19 (2014)
- D Sun, S Roth, MJ Black, A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vision.* **106**(2), 115–137 (2014)
- BKP Horn, BG Schunck, Determining optical flow. *Artif. Intell.* **17**(1–3), 185–203 (1981)
- N Onkarappa, AD Sappa, Speed and texture: an empirical study on optical-flow accuracy in ADAS scenarios. *IEEE Trans. Intell. Transportation Syst.* **15**, 136–147 (2014)
- P Heas, C Herzet, E Memin, Bayesian inference of models and hyperparameters for robust optical-flow estimation. *IEEE Trans. Image Process.* **21**(4), 1437–1451 (2012)
- K Krajsek, R Mester, in *Pattern Recognition. Lecture Notes in Computer Science*, ed. by FA Hamprecht, C Schnörr, and B Jähne. Bayesian model selection for optical flow estimation, vol. 4713 (Springer New York, 2007), pp. 142–151
- Y Li, DP Huttenlocher, in *European Conference on Computer Vision. Lecture Notes in Computer Science*, ed. by D Forsyth, P Torr, and A Zisserman. Learning for optical flow using stochastic optimization, vol. 5303 (Springer New York, 2008), pp. 379–391
- S Roth, MJ Black, On the spatial statistics of optical flow. *Int. J. Comput. Vis.* **74**(1), 33–50 (2007)
- J Delpiano, L Pizarro, R Verschae, J Ruiz-del-Solar, in *9th International Conference on Computer Vision Theory and Applications*. Multi-objective optimization for characterization of optical flow methods, vol. 2 (IEEE Press, Odense, DK, 2014), pp. 556–573
- DR Pereira, J Delpiano, JP Papa, in *27th SIBGRAPI Conference on Graphics, Patterns and Images*. Evolutionary optimization applied for fine-tuning parameter estimation in optical flow-based environments (SciTePress, DC, USA, 2014), pp. 125–132
- T Brox, J Malik, Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 500–513 (2011)
- E Cuevas, M Cienfuegos, D Zaldivar, M Pérez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst. Appl.* **40**(16), 6374–6384 (2013)
- ZW Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*. (Springer, New York, 2009)
- J Kennedy, R Eberhart, in *Proceedings of the IEEE International Conference on Neural Networks*. Particle swarm optimization (IEEE Press, DC, USA, 1995), pp. 1942–1948
- L Xu, J Jia, Y Matsushita, Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(9), 1744–1757 (2012)
- D Sun, S Roth, MJ Black, in *IEEE Conference On Computer Vision and Pattern Recognition (CVPR) 2010*. Secrets of optical flow estimation and their principles (IEEE, DC, USA, 2010), pp. 2432–2439
- M Werlberger, W Trobin, T Pock, A Wedel, D Cremers, H Bischof, in *Proceedings of the British Machine Vision Conference (BMVC)*, London, UK. Anisotropic Huber-L1 optical flow (BMVA Press, Durham, 2009)
- JA Nelder, R Mead, A simplex method for function minimization. *Comput. J.* **7**, 308–313 (1965)
- A Bruhn, J Weickert, C Schnörr, Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *Int. J. Comput. Vis.* **61**(3), 211–231 (2005)
- N Cornelius, T Kanade, in *Proc. of the ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion: Representation and Perception*. Adapting optical-flow to measure object motion in reflectance and x-ray image sequences (Elsevier North-Holland, NY, USA, 1986)
- B Lucas, T Kanade, in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*. An iterative image registration technique with an application to stereo vision (Morgan Kaufmann Publishers Inc. CA, USA, 1981)
- JL Barron, DJ Fleet, SS Beauchemin, Performance of optical flow techniques. *Int. J. Comput. Vis.* **12**, 43–77 (1994)
- S Baker, D Scharstein, JP Lewis, S Roth, M Black, R Szeliski, A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **92**(1), 1–31 (2011). doi:10.1007/s11263-010-0390-2
- A Geiger, P Lenz, R Urtasun, in *IEEE Conference on Computer Vision and Pattern Recognition*. Are we ready for autonomous driving? The KITTI vision benchmark suite (IEEE Press, DC, USA, 2012)
- A Bruhn, J Weickert, A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *Int. J. Comput. Vis.* **70**, 257–277 (2006)
- H Liu, T Hong, M Herman, R Chellappa, Accuracy vs. efficiency trade-offs in optical flow algorithms. *Comp. Vision Image Underst.* **72**(3), 271–286 (1996)
- D Gibson, M Spann, Robust optical flow estimation based on a sparse motion trajectory set. *IEEE Trans. Image Process.* **12**(4), 431–445 (2003)
- LDOF implementation. <http://lmb.informatik.uni-freiburg.de/resources/software.php>. Accessed 4 Nov 2014
- ZW Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*. (Springer, New York, 2009)
- D Zou, L Gao, J Wu, S Li, Novel global harmony search algorithm for unconstrained problems. *Neurocomputing.* **73**, 3308–3318 (2010)

41. Middlebury dataset. <http://vision.middlebury.edu/flow/data/>. Accessed 4 Nov 2014
42. Sintel dataset. [sintel.is.tue.mpg.de/](http://sintel.is.tue.mpg.de/). Accessed 4 Nov 2014
43. DJ Butler, J Wulff, GB Stanley, MJ Black, in *European Conference on Computer Vision*. Part IV, LNCS 7577, ed. by A Fitzgibbon, et al. A naturalistic open source movie for optical flow evaluation (Springer, New York, 2012), pp. 611–625
44. M Otte, H-H Nagel, in *European Conference on Computer Vision*. Lecture Notes in Computer Science, ed. by J-O Eklundh. Optical flow estimation: advances and comparisons, vol. 800 (Springer, New York, 1994), pp. 49–60
45. JC Lagarias, JA Reeds, MH Wright, PE Wright, Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM J. Optim.* **9**, 112–147 (1998)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---