INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 16, Number 1, Pages 116–138

A SPACE-TIME PARALLEL METHOD FOR THE OPTIC FLOW ESTIMATION IN LARGE DISPLACEMENTS AND VARYING ILLUMINATION CASE

DIANE GILLIOCQ-HIRTZ AND ZAKARIA BELHACHMI

Abstract. We consider a unified variational PDEs model to solve the optic flow problem for large displacements and varying illumination. Although, the energy functional is nonconvex and severely nonlinear, we show that the model offers a well suited framework to extend the efficient methods we used for small displacements. In particular, we resort to an adaptive control of the diffusion and the illumination coefficients which allows us to preserve the edges and to obtain a sparse vector field. We develop a combined space-time parallel programming strategy based on a Schwarz domain decomposition method to speed up the computations and to handle high resolution images, and the *parareal* algorithm, to enhance the speedup and to achieve a lowest-energy local minimum. This full parallel method gives raise to several iterative schemes and allows us to obtain a good balance between several objectives, e.g. accuracy, cost reduction, time saving and achieving the "best" local minimum. We present several numerical simulations to validate the different algorithms and to compare their performances.

Key words. Optic flow estimation, large dislacements, variable illumination, adaptive finite elements, parallel and parareal computations, domain decomposition.

1. Introduction

The optic flow problem is a central research topic in computer vision since the last few decades. It is used in many fields, e.g. robotic for obstacle detection, in video compression and also in medical imaging. In the small displacements case, several approaches have been used, e.g. statistical methods, learning techniques, PDEs, etc ([2], [10], [27], [28]), see [22], [9] and the references therein. Despite the fact that these methods have been extended to the large displacements case, a lot of challenging questions remain, in both modelling and computational grounds. In fact, for small displacements (i.e. successive frames in the sequence are close), the constraint expressing the conservation of the pixel's intensity is usually linearized in a single step, which leads essentially to solve a linear system of PDEs. This is no longer true for large displacements and the problem requires, in the variational framework, to work with highly nonlinear and nonconvex energy functionals possessing possibly a lot of minima. Minimizing such functionals inevitably raises many difficulties (existence of -local- minimizers, regularity of solutions, well suitable optimization algorithms, cost of computations, etc). Moreover, phenomena like occlusions, change in the illumination within the scene, etc, should prevent from using techniques with too much smoothing effects (the solutions develop singularities which are important features to describe the motion).

In this article, we extend the approach considered in [18] for small displacements to the challenging problem of large displacements and varying illumination. It is a variational method which relies on the framework introduced in the seminal works of [24] and [20] and widely pursued by the community of mathematical image

Received by the editors in revised form March 19, 2018.
2000 *Mathematics Subject Classification.* 35K15, 35K55, 65M22, 65M55, 68Y05, 68U10, 65D18.

analysis, e.g. without being exhaustive [2, 8, 28], (see also [9] and the references therein). In this common framework, one minimizes an energy functional which consists of a data term that expresses the fundamental constraint of the optic flow estimation, namely the constancy of the pixel's intensity, and a regularization term among a family of well known regulaziers (e.g. Tychonov, TV, ...). In all cases, this last part represents a diffusion term with prescribed diffusion coefficient/function.

In the article, we adopt the approach of control of the diffusion introduced in [6] which takes a spatially variable diffusion function, adjusted locally and adaptively to decrease the amount of diffusion near the singularities (high gradients areas).

To take into account the varying illumination, several approaches were suggested, e.g. constancy of the magnitude of the gradient of the images [8]. Gennert et al [15] suggested a "law" for such variations which relaxes the constancy of the intensity. In this work, we extend this method to construct a varying illumination function from the simple initial "law" suggested in [15]. Using the control approach, we obtain both a diffusion and a change of illumination with sophisticated profiles, i.e. piecewise smooth with non trivial jump sets (those giving some relevant informations on occlusions areas, shadow, ...). Mathematically speaking, the method is convergent, in the Γ-convergence sense, to (special) bounded variation functions solutions [7].

We show that variational methods and the finite element discretisation offer a powerful and complete framework to solve the optic flow problem in the large displacement case. In fact, this setting allows us to decrease significantly the cost of computation by using coarse elements in the homogeneous regions where the flow field is smooth and refined meshes near the edges. Thus, we distinguish the geometry of the vector field (the optic flow), hence the meshes associated to it, and the meshes associated to the images in the sequence (constrained by the resolution and generally very fine and isotropic even if the motion is not), so that even for high resolution images, we may end up with meshes with only few elements and a sparse optic flow.

In order to increase the efficiency of the method with respect to additional constraints like fast computations and the high resolution sequences, it is necessary to use advanced tools of high performance computing, particularly the parallel programming. In fact, we have two main objectives in using parallel programming approaches: on one hand the (usual) goal of obtaining a significant time saving, and, in another hand, the aim of handling high resolution images. In addition, as the problem is nonconvex, we aim at computing a lowest -energy minimum. This last objective motivate our use of a parallel in time method (parareal algorithm) which may be considered as a time multi-grid method [13], and allows us to reach such lowest-energy solution for a given accuracy. This fact may be proven in special and simple problems (as for a standard multi-grid algorithms) but in our case it is a numerical evidence rather than a rigorous proof which is still open question. Another reason for the use of the parareal approach in our problem is the ability to use a different numerical scheme at the fine time scale than the one used at the coarse grid, or even the use of different physics between the two scales, i.e. a simplified version of the system of equations at one level, ... This gives a supplementary flexible way to discretize such complex systems.

We study and compare three parallel computing methods: an MPI parallel method, in space, using a Schwarz domain decomposition technique which is clearly

a well suited method to handle high resolution frames and for which tools of parallel implementation have reached a high level of maturity. Such a method is implemented under the openGL freeware FreeFem++ [11], [21] and its MPI facilities. The aim here is twofold: managing large sequences of frames with high resolution and obtaining high speedup. We show that the method in this case is nearly scalable. In fact, we perform a detailed task analysis to exhibit the "non scalable" actions for further improvements of the method.

The second parallel method, the so-called parareal algorithm [23], consists of two time-grids (a fine and coarse scale) and mainly allows one to use a sequential coarse scheme and a parallel solver at the fine grid. The speedup within this method is limited ([14]). We build several iterative schemes and compare them to select the one which fulfils as best as possible the aims of achieving a desired accuracy with a good local minimum (compared to the one given by a sequential algorithm), time saving, ...

The space-time parallel method couples the two previous ones and turns out to be an efficient strategy to solve variational problems with severe nonlinearities and nonconvex energies. It is then important to notice that the article aims to show that the parallel computing is not used only to decrease the computational efforts (mainly the computations time) but to design efficient schemes to solve the problem as well.

The article is organized as follows: In Section 1, we state the problem under a variational form. We describe the iterative fixed point scheme to solve the associated nonlinear PDEs system. In Section 2, we study the corresponding discrete problem. We develop the anisotropic and adaptive method of the control of the diffusion matrix which is essential in our approach to preserve the edges of the flow field. We finish the section with some numerical results in the sequential framework to show the accuracy of the model in determining the optic flow for large displacements and varying illumination with preserving the edges. This furnishes also the "reference" solutions to compare with the results of the parallel methods of the next sections, since there is no ground-truth available. Section 3 is devoted to the first parallel method, namely the overlapping Schwarz domain decomposition method. We describe in detail this parallel approach and its implementation for the considered problem. Numerical simulations are performed for the speedup, the convergence and the accuracy. In particular, a detailed study of the time of computations is given. In Section 4, we consider a parallel in time method (parareal method), the principle as well as the implementation are presented and some numerical results are given. Finally, in Section 5, we combine the two parallel methods for a fully parallel approach and we perform some numerical simulations to show the efficiency of this coupling which conjugate computation efficiency and accurate solving of the optic flow problem.

2. The model problem

2.1. Variational formulation. We define a function I from an image domain $\Omega \subset \mathbb{R}^2$ to $\mathbb{R}$ representing the intensity of a pixel (x, y) at an instant t by

$$\begin{array}{rccc} \text{I:} & \Omega \times [0,T] & \to & \mathbb{R} \\ & (x,y),t & \mapsto & I(x,y,t). \end{array}$$

According to the work of Barron [4], we slightly smooth the sequence thanks to a convolution with a Gaussian kernel K_σ of standard deviation σ. The new function is noted

$$f(x,y,t) = (K_\sigma \star I)(x,y,t).$$

The estimation of the optical flow consists in finding the vector field $\mathbf{u} = (u_1, u_2)$, called the optic flow, describing the motion of each pixel between two frames of a given sequence. Most methods for the determination of $\mathbf{u}$ are based on the assumption that the intensity of a pixel is constant between two successive frames. More precisely, the optic flow defines trajectories $\chi(t, \mathbf{x})$ along which this constancy assumption holds,

$$\begin{cases} \dot{\chi} = \mathbf{u}, \\ \dot{\mathbf{u}} = 0, \\ \chi(0) = \mathbf{x} \end{cases}$$

and

$$\frac{d(f \circ \chi)}{dt} = 0,$$

which is familiarly written in the computer vision community

$$f(x + u_1, y + u_2, t + 1) = f(x, y, t). \tag{1}$$

In the small displacements case (the frames are very close), this equation is linearized with a Taylor expansion and reads

$$f_x u_1 + f_y u_2 + f_t = 0, \tag{2}$$

where the unknown is the vector field (u_1, u_2) and ϕ_x, respectively ϕ_y denotes the partial derivatives with respect to x, respectively y. With only one equation to determine two unknowns, this problem is ill-posed. It is called the aperture problem. Several methods have been considered among which the seminal approach of Lucas and Kanade [24] and its variational form due to Horn and Schunck [20], which consists in minimizing, over the Sobolev space $H^1(\Omega)^2$, the energy functional

$$\mathcal{E}(\mathbf{u}) = \int_\Omega K_\rho \star (f_x u_1 + f_y u_2 + f_t)^2 + \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) dxdy, \tag{3}$$

where K_ρ is a gaussian kernel of standard deviation ρ and α is a nonnegative constant.

Remark 2.1. *(1) The convolution with the gaussian kernel K_ρ acts like a convolution with the function $\mathbb{1}_{\mathcal{V}(x,y)}$ and gives a larger weight to the pixels very close to (x, y).*

(2) The main shortcoming of this model is that it produces continuous solutions and do not preserves the edges which are essential in image processing. Following [7], we will consider a variable coefficient α which is chosen locally in order to decrease the diffusion near the singularities (the edges).

In the case of large displacements the frames are distant so that effects like occlusions arise (objects which appear/disappear in the scene), which mainly imply that the constancy assumption could not be linearized without loss of important informations. Consequently, the data term in the energy functional becomes a least square fitting of the difference $f(x+u_1, y+u_2, t+1) - f(x, y, t)$. Another constancy assumption, implicitly hidden in (1), is the brigthness invariance which means that eventual changes in the illumination are neglected and are taken constant. When this assumption become too stringent, many authors relax it for example by adding a specific constraint (conservation of the modulus of the gradient of f) [8], or, for colored images, by working with variables which are less sensitive to such illumination changes [25]. Another approach suggested by Gennert and Negahdaripour [15]

amounts to model this change with a new unknown. More precisely, the conservation law (1) is modified as follows

$$f(x+u_1, y+u_2, t+1) = M(x,y,t)f(x,y,t) + T(x,y,t), \tag{4}$$

that is to say the varying illumination is taken as a linear transformation of the intensity. For small displacement and in many physical situations this assumption is reasonable and coherent with the laws of the optic. However, for large displacements it might be too restrictive. In our model, we include (4) in the energy functional but as we apply the method of ([7]) to modify the diffusion coefficients, the modelling of the illumination changes is more involved as we will show. For simplicity, we set $T = 0$, thus the model for the estimation of the optical flow in large displacements and with varying illumination consists in finding a minimum of the non-linear functional

$$\begin{aligned}\mathcal{E}(\mathbf{u}, M) = &\int_\Omega \underbrace{(f(x+u_1, y+u_2, t+1) - Mf(x,y,t))^2 dxdy}_{\text{non-linear data term}} \\ &+ \int_\Omega \underbrace{\alpha_1(x,y)|\nabla u_1|^2 + \alpha_2(x,y)|\nabla u_2|^2) + \lambda(x,y)|\nabla M|^2 dxdy}_{\text{linear regularization term}}.\end{aligned} \tag{5}$$

It should be noted that the energy functional is defined over the space $H^1(\Omega)^3$, however our adaptive approach to control the diffusion yields solutions which are functions of bounded variations (i.e. in the larger space BV, see [7] for details). In particular, they have a jump set (edges). Hence, we have α_1, α_2, and λ which are real functions, such that there exist two constants a_0, a_1

$$0 < a_0 \leq \alpha_1(x,y), \alpha_2(x,y), \lambda(x,y) \leq a_1.$$

The Euler-Lagrange system related to the minimisation of the functional $\mathcal{E}$ is as follows

$$\begin{cases} -\operatorname{div}(\alpha_1(\mathbf{x})\nabla u_1) + L(f,U)(\mathbf{x}) f_x(x+u_1, y+u_2, t+1) = 0, & \text{in } \Omega \\ -\operatorname{div}(\alpha_2(\mathbf{x})\nabla u_2) + L(f,U)(\mathbf{x}) f_y(x+u_1, y+u_2, t+1) = 0, & \text{in } \Omega \\ -\operatorname{div}(\lambda(\mathbf{x})\nabla M) - L(f,U)(\mathbf{x}) f(x,y,t) = 0, & \text{in } \Omega \\ \frac{\partial u_1}{\partial n} = \frac{\partial u_2}{\partial n} = \frac{\partial M}{\partial n} = 0, & \text{on } \partial\Omega \end{cases} \tag{6}$$

where $L(f,U) = f(x+u_1, y+u_2, t+1) - Mf(x,y,t)$.

2.2. Iterative scheme and linearization. To solve the PDEs system (6), which is strongly nonlinear, we use a fixed point method. Note that the non-convexity of the energy and the existence of (possibly) a lot of local minima, as well as the non explicit form of the nonlinearity (which depends on the data f), prevent from strong convergence results of the iterative scheme. In fact, to solve the system, we proceed in two steps, a semi-implicit linearization step, the iterative scheme is then still nonlinear, but the remaining nonlinear term concerns the transition from the iteration k to $k+1$, thus we can use the linearization methods used in the small displacements case. More precisely, we have the fixed point algorithm

$$\begin{cases} -\operatorname{div}(\alpha_1(\mathbf{x})\nabla u_1^{k+1}) + (L(f,U))^{k+1}(\mathbf{x}) f_x(x+u_1^k, y+u_2^k, t+1) = 0 & \text{in } \Omega \\ -\operatorname{div}(\alpha_2(\mathbf{x})\nabla u_2^{k+1}) + L(f,U))^{k+1} f_y(x+u_1^k, y+u_2^k, t+1) = 0 & \text{in } \Omega \\ -\operatorname{div}(\lambda(\mathbf{x})\nabla M^{k+1}) - L(f,U))^{k+1} f(x,y,t) = 0, & \text{in } \Omega \\ \frac{\partial u_1^{k+1}}{\partial n} = \frac{\partial u_2^{k+1}}{\partial n} = \frac{\partial M^{k+1}}{\partial n} = 0, & \text{on } \partial\Omega \end{cases} \tag{7}$$

Then, we set

$$\begin{aligned} u_1^{k+1} &= u_1^k + \delta u_1 \\ u_2^{k+1} &= u_2^k + \delta u_2 \\ M^{k+1} &= M^k + \delta M. \end{aligned}$$

We denote for simplicity

$$f_x^k = f_x(x+u_1^k, y+u_2^k, t+1), f_y^k = f(x+u_1^k, y+u_2^k, t+1), f^k = f(x+u_1^k, y+u_2^k, t+1)$$

Next, we use a Taylor expansion, and obtain the linear system corresponding to the minimization of the functional (5)

$$\left\{ \begin{array}{ll} -\operatorname{div}(\alpha_1(\mathbf{x})\nabla\delta u_1) + (f_x^k\delta u_1 + f_y^k\delta u_2 - f(\mathbf{x},t)\delta M)f_x^k & \\ \qquad = \operatorname{div}(\alpha_1(\mathbf{x})\nabla u_1^k) - (L(f,U))^k f_x^k & \text{in } \Omega \\ -\operatorname{div}(\alpha_2(\mathbf{x})\nabla\delta u_2) + (f_x^k\delta u_1 + f_y^k\delta u_2 - f(\mathbf{x},t)\delta M)f_y^k & \\ \qquad = \operatorname{div}(\alpha_2(\mathbf{x})\nabla u_2^k) - (L(f,U))^k f_y^k & \text{in } \Omega \\ -\operatorname{div}(\lambda(\mathbf{x})\nabla\delta M) - (f_x^k\delta u_1 + f_y^k\delta u_2 - f(\mathbf{x},t)\delta M)f(\mathbf{x},t) & \\ \qquad = \operatorname{div}(\lambda(\mathbf{x})\nabla M^k) + (L(f,U))^k f(\mathbf{x},t) & \text{in } \Omega \\ \frac{\partial\delta u_1}{\partial n} = \frac{\partial\delta u_2}{\partial n} = \frac{\partial\delta M}{\partial n} = 0, & \text{on } \partial\Omega \end{array} \right.$$

It is more convenient to use a vectorial notation, we set

$$\delta\mathbf{U} = \begin{pmatrix} \delta u_1 \\ \delta u_2 \\ \delta M \end{pmatrix}, \ \mathbf{A}^k = \begin{pmatrix} A_{1,1}^k & A_{1,2}^k & A_{1,3}^k \\ A_{1,2}^k & A_{2,2}^k & A_{2,3}^k \\ A_{1,3}^k & A_{2,3}^k & A_{3,3}^k \end{pmatrix},$$

where

$$\begin{aligned} A_{1,1}^k &= f_x^2(x+u_1^k, y+u_2^k, t+1) \\ A_{2,2}^k &= f_y^2(x+u_1^k, y+u_2^k, t+1) \\ A_{3,3}^k &= f^2(x,y,t) \\ A_{1,2}^k &= f_y(x+u_1^k, y+u_2^k, t+1)f_x(x+u_1^k, y+u_2^k, t+1) \\ A_{1,3}^k &= -f(x,y,t)f_x(x+u_1^k, y+u_2^k, t+1) \\ A_{2,3}^k &= -f(x,y,t)f_y(x+u_1^k, y+u_2^k, t+1), \end{aligned}$$

and

$$\mathbf{F}^k = \begin{pmatrix} -(f(x+u_1^k, y+u_2^k, t+1) - M^k f(x,y,t))f_x(x+u_1^k, y+u_2^k, t+1) \\ -(f(x+u_1^k, y+u_2^k, t+1) - M^k f(x,y,t))f_y(x+u_1^k, y+u_2^k, t+1) \\ (f(x+u_1^k, y+u_2^k, t+1) - M^k f(x,y,t))f(x,y,t). \end{pmatrix}.$$

Now, given u^k, we compute u^{k+1} by solving

$$\left\{ \begin{array}{ll} -\operatorname{div}(\mathbf{\Lambda}(\mathbf{x})\nabla\delta\mathbf{U}) + \mathbf{A}^k\delta\mathbf{U} = \operatorname{div}(\mathbf{\Lambda}(\mathbf{x})\nabla\mathbf{U}^k) + \mathbf{F}^k, & \text{in } \Omega \\ \frac{\partial\delta\mathbf{U}}{\partial n} = 0, & \text{on } \partial\Omega \end{array} \right. \tag{8}$$

Note that in this Neumann boundary value problem (8), the matrix $\mathbf{A}^k$ may degenerates in $\omega \subset \Omega$, a connected component subset of Ω. For the well-posedness, we write the datum $\mathbf{F}^k$, under the form $\mathbf{F}^k = \mathbf{A}^k\mathbf{h} + \mathbf{g}$, where $\mathbf{h}$ is such that $(\mathbf{A}^k)^{\frac{1}{2}}\mathbf{h} \in L^2(\Omega)$, $\operatorname{supp}\mathbf{g} \subset \Omega$ and $\int_\omega \mathbf{g}\,dx = 0$. Then we have (see [6] for details)

Proposition 2.1. *Problem* (8) *admits a weak solution* $\delta\mathbf{U}$ *in* $H^1(\Omega)^3$.

Proof. The existence of a weak solution $\delta\mathbf{U}$ follows from the Lax-Milgram Lemma. The only point to check is the continuity of the linear form, and precisely the part $\mathbf{v} \mapsto \int_\Omega \mathbf{F}^k \cdot \mathbf{v}\, d\mathbf{x}$. We have

$$\begin{aligned}
|\int_\Omega \mathbf{F}^k \cdot \delta\mathbf{U}\, d\mathbf{x}| \leq & \int_\Omega |\mathbf{A}^k\mathbf{h} \cdot \delta\mathbf{U}|\, d\mathbf{x} + |\int_\Omega \mathbf{g}\, \delta\mathbf{U}\, d\mathbf{x}| \\
& \leq \left(\int_\Omega |h|^2\, d\mathbf{x}\right)^{\frac{1}{2}} \left(\int_\Omega |A^k\delta\mathbf{U}|^2\, d\mathbf{x}\right)^{\frac{1}{2}} \\
& \quad + C\left(\int_\omega |\mathbf{grad}\, \delta\mathbf{U}|^2\, d\mathbf{x}\right)^{\frac{1}{2}} \left(\int_\omega |\mathbf{g}|^2\, d\mathbf{x}\right)^{\frac{1}{2}} \\
& \leq C\|\delta\mathbf{U}\|_{H^1(\Omega)^3},
\end{aligned}$$

where C is the Poincaré-Wirtinger inequality applied in $H^1(\omega, \mathbb{R}^2)^3 \setminus \mathbb{R}^2$ and the smooth set ω is such that $\operatorname{supp} \mathbf{g} \subset \omega \subset \Omega$. □

We emphasize that the transition from the iteration k to $k+1$ corresponds to the method that we have developed for the small displacements case. It consists in solving the linear problem (8) with respect to $\delta\mathbf{U}$. However, within this method, the diffusion matrix Λ is modified in an adaptive algorithm to slow the diffusion near the edges (see [6]- [18]). Thus, Thus, we have to modify slightly the right-hand side by setting $\Lambda(x) = \Lambda_0 I_d$, with Λ_0 is constant and I_d is the identity matrix. Therefore, we set the $\Lambda(x) = \Lambda_0\, I_d$, at the right-hand side, Λ_0 being a constant (and I_d is the identity matrix).

3. The discrete problem

3.1. finite elements formulation. Let $\mathcal{T}_h$ be a regular triangular mesh, we define the space of approximation of the solution

$$\mathcal{V}_h = \{\mathbf{V}_h \in (C(\overline{\Omega}))^3, \quad \mathbf{V}_h|_K \in (P_1(K))^3\}$$

where h represents the maximal diameter of the cells K, $C(\overline{\Omega})$ is the space of continuous functions on $\overline{\Omega}$ and $P_1(K)$ is the set of the polynomial functions of degree less than or equal to 1 in K.

For simplicity we drop the index k in the sequel. We define the bilinear form

$$a_{\mathbf{\Lambda}}(\delta\mathbf{U}_{\mathbf{\Lambda}}, \mathbf{V}) = \int_\Omega \mathbf{\Lambda}(\mathbf{x})\nabla\delta\mathbf{U}_{\mathbf{\Lambda}}\nabla\mathbf{V} d\mathbf{x} + \int_\Omega \mathbf{V}^t\mathbf{A}\delta\mathbf{U}_{\mathbf{\Lambda}} d\mathbf{x}$$

and the linear form

$$L(\mathbf{V}) = \int_\Omega \mathbf{F}\mathbf{V} + \mathbf{\Lambda}_0\nabla\mathbf{U}\nabla\mathbf{V} d\mathbf{x}.$$

The variational formulation leads to find the weak solution $\delta\mathbf{U}_{\mathbf{\Lambda},h} \in \mathcal{V}_h$ such that

$$a_{\mathbf{\Lambda},h}(\delta\mathbf{U}_{\mathbf{\Lambda},h}, \mathbf{V}_h) = L_h(\mathbf{V}_h) \quad \forall\mathbf{V}_h \in \mathcal{V}_h, \tag{9}$$

where $a_{\mathbf{\Lambda},h}$ and $L_h(\mathbf{V})$ are the discrete counterparts of $a_{\mathbf{\Lambda}}$ and $L(\mathbf{V})$ respectively. In the discrete bilinear form $\mathbf{A}$ is replaced by a finite element approximation $\mathbf{A}_h$.

Similarly to the continuous case, we have

Proposition 3.1. *Problem* (9) *admits a solution* $\delta\mathbf{U}_h$ *in* $\mathcal{V}_h$.

Note that in the practice we solve the problem (9) by adding ϵId, (Id identity matrix) to $\mathbf{\Lambda}$ which ensures the uniqueness of the solution.

In what follows we will assume the diffusion matrix Λ piecewise constant in the sens that each coefficient α_i, $i = 1, 2$ satisfies

$$\alpha_i(x) = \alpha_{i,K} \text{ (constant)}, \quad \forall K \in \mathcal{T}_h$$

3.2. Control of the diffusion matrix and the adaptive algorithm. Based on the method of Belhachmi and Hecht [6], we implement an adaptive control of the diffusion matrix Λ. In this article, we keep the parameter λ constant such that $\lambda(x, y) = \lambda_0$. The control of this parameter is considered in [16]. Since the displacements are large, instead of considering the same regularization process for u_1 and u_2 as in [6], we are going to adapt the parameter α_1 related to u_1 and the one related to u_2 independently to enforce the anisotropy of the method. Denoting $F_{1,h}$ and $F_{2,h}$ a finite elements approximation of the two first components of $\mathbf{F}$ and $A_{i,j,h}$ an approximation of $A_{i,j}$, we define the error indicators by

$$\eta_K^1 = (\alpha_{1,K})^{-\frac{1}{2}} h_K \left\| F_{1,h} + \alpha_{1,K}\Delta u_{1,h} + (A_{1,1,h}u_{1,h}) + (A_{1,2,h}u_{2,h}) \right\|_{L^2(K)} + \frac{1}{2} \sum_{e \in \mathcal{E}_K} (\alpha_{1,e})^{-\frac{1}{2}} h_e^{\frac{1}{2}} \left\| [\alpha_{1,K} \nabla u_{1,h} \cdot \mathbf{n}_e]_e \right\|_{L^2(e)} \tag{10}$$

and

$$\eta_K^2 = (\alpha_{2,K})^{-\frac{1}{2}} h_K \left\| F_{2,h} + \alpha_K^2 \Delta u_{2,h} + (A_{1,2,h}u_{1,h}) + (A_{2,2,h}u_{2,h}) \right\|_{L^2(K)} + \frac{1}{2} \sum_{e \in \mathcal{E}_K} (\alpha_{2,e})^{-\frac{1}{2}} h_e^{\frac{1}{2}} \left\| [\alpha_{2,K} \nabla u_{2,h} \cdot \mathbf{n}_e]_e \right\|_{L^2(e)} \tag{11}$$

where $\mathcal{E}_K$ represents the set of all edges e of K, the diameter of K is noted h_K and the diameter of an edge e is h_e. $\mathbf{n}_e$ represents the unit normal vector on e. $\alpha_{i,e}$ is the maximum between the values of $\alpha_{i,K}$, $i = 1, 2$ at the two neighbouring elements sharing the edge e (recall that $\Lambda(x)$ is piecewise constant). We denote by $[.]_e$, the jump through the edge e. These residual error indicators are equivalent to the H^1-norm the finite element discretization error [26]. In our method, we use the error indicators in order to apply on one hand a geometric adaptation (mesh adaption). This allows us to obtain tight location of the singularities -edges, corners-) [7]. Next, we perform a functional adaptation to decrease the diffusion in this area thus to prevent the smoothing of the edges. In fact, the value of these indicators is large on the discontinuities of the scene since the value of $\nabla u_{1,h}$, and $\nabla u_{2,h}$ are large. Thus, the local choice of α_1 et α_2 is then given for $\ell = 1, 2$ by

$$(\alpha_{\ell,K})^{n+1} = \max\left(\frac{(\alpha_{\ell,K})^n}{1 + \kappa \max\left(\frac{\eta_K^\ell}{\|\eta^\ell\|_\infty} - \zeta, 0 \right)}, \alpha_s \right). \tag{12}$$

where κ is an arbitrary control parameter and α_s is a threshold. This formula means that, if the relative error is greater than $\zeta \in [0, 1]$ we reduce the value of $\alpha_{i,K}$. On the other hand, if it is less than ζ the denominator is equal to one and so $\alpha_{i,K}$ is unchanged.

The analysis of this method is performed in [7], in particular it is shown that it preserves the edges.

We present below the different steps of the algorithm implemented with the software *FreeFem++*.

(1) Initialisation: $\mathbf{U}^0 = 0$, $i = 0$.
(2) Compute $f(\mathbf{x} + \mathbf{U}^i)$ and $\mathbf{A}^i$.
(3) Resolution of the problem (8) for the iteration i.
(4) Local choice of the parameter α and mesh adaptation.
(5) Update the solution: $\mathbf{U}^{i+1} = \mathbf{U}^i + d\mathbf{U}^i$.
(6) $i = i + 1$, go back to step 2.

The algorithm stops when the L^2 error between $\mathbf{U}^i$ and $\mathbf{U}^{i+1}$ is less than 10^{-2}.

4. The sequential case

We give in this section some numerical results in the sequential case. We emphasize that the following results provide us with a reference solution to evaluate the parallel methods of the next sections, but yet constitute a novelty for the optic flow estimation in a large displacements and varying illumination case.

4.1. Evaluation of the model. For the evaluation of the model for large displacement, we use some sequences provided by the Middleburry website at `www.vision.middleburry.edu/flow/`. As the visualisation of a dense vector field could be difficult, we also use the color map proposed by Middleburry (see figure 1). Instead of representing a vector with an arrow, we assign a color to each vector with respect to its norm and its orientation.

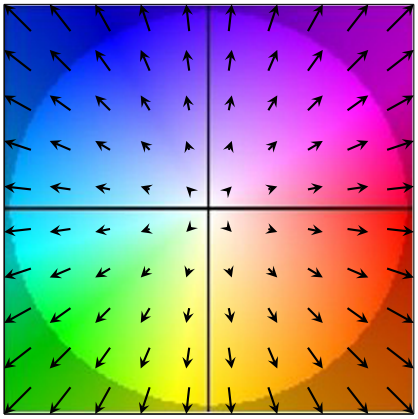
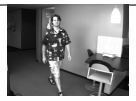


FIGURE 1. Vector field and its corresponding color map.

To work with such large displacement sequences, we use the frames 7 and 14 of each corresponding test. Note that there is no ground truth available in Middleburry plateform. The test cases considered in this article and their specifications are presented in the table 1.

These sequences highlight the principal issues of the optical flow in large displacements such as the varying illumination effects and the large occlusion areas. We present on the figure 2 the results obtained with our algorithm with and without the adaptive method presented in the section 3.2. The use of a local control of the regularization parameters α_1, α_2, allows us to obtain sharp edges and thus, gives

TABLE 1. Particularity of the different test cases used.

Test case		Specificity
RubberWhale		Synthetic images with a lot of details
DogDance		Real photographs with large occlusions
Walking		

a better approximation of the optical flow as a piecewise smooth solution with a jump set.

FIGURE 2. Optical flow obtained with (right) and without (middle) adaptation of α.

We have also applied a mesh adaptation and we present on the figure 3 the adapted meshes after 10 iterations. They are well refined around the edges and are coarser on the homogeneous areas. This is one strong point of the method which allows us to reduce significantly the number of degrees of freedom. As one can see in Figure 4, the number of degrees of freedom decreases from the second iteration and consequently reduces notably the time of computations as shown in the table 2.

4.1.1. Convergence issue. The functional (5) is not a convex function and is severely nonlinear. The nonlinearity depends on the data f, thus without additional assumptions, we are not able to prove the convergence of the fixed point algorithm. Nevertheless, as usual in such cases, we stop the algorithm when the L^2

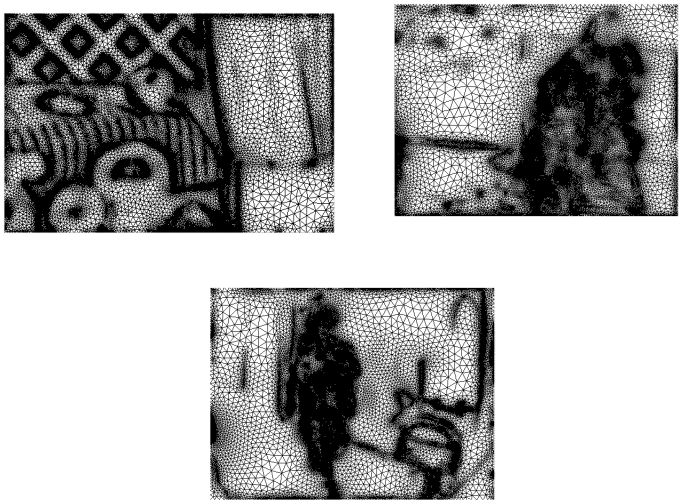


FIGURE 3. Adapted meshes.

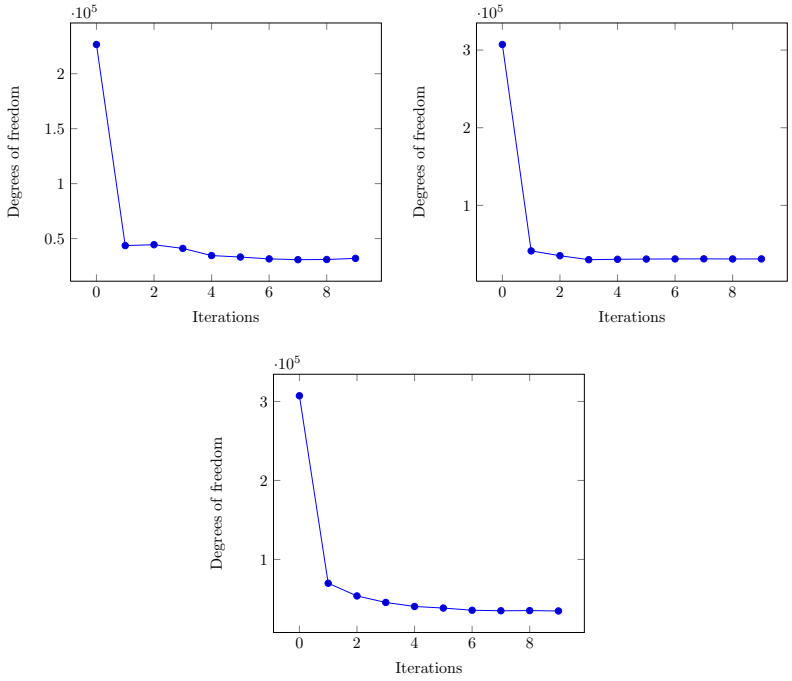


FIGURE 4. Evolution of the number of degrees of freedom with respect to the adaptation iterations.

error between the solutions of two successive iterations is small.

In Figure 5, we give the evolution of the error

TABLE 2. Computation times with and without mesh adaptation.

Test Case	Without mesh adaptation	With mesh adaptation
RubberWhale	650s	232s
Walking	1022s	339s
DogDance	978s	336s

$$err = \left\| U^{n+1} - U^n \right\|_{L^2} .$$

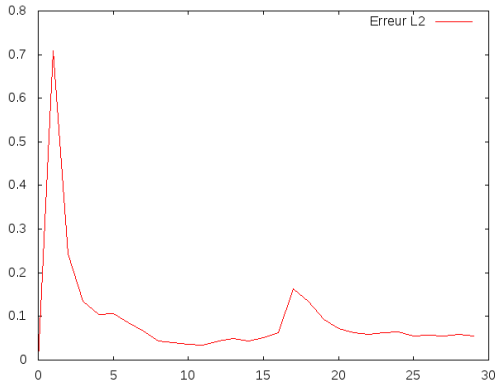


FIGURE 5. Evolution of the error $\left\| U^{n+1} - U^n \right\|_{L^2}$.

As one may check in Figure 5, numerical evidences show a convergence of the method (to a local minimum). In all cases, we observed a convergence process until a given iteration (here in the Rubberwhale example iteration 15), then the global error start to increase but if we stop the mesh adaptation the convergence is continued. The reason for this is that the mesh adaptation adjust the mesh to the optic flow u, so the bilinear interpolation (i.e. $f(\mathbf{x}, \mathbf{U})$) increases as the mesh for the image itself is the initial one and is dependent only on f. Nevertheless, we have seen in Figure 4 that the mesh adaptation detect the singular sets after one or two iterations, and thus we can stop it earlier.

We emphasize that real-time computations is not an objective of this work at this stage (and anyway seems difficult to achieve in the framework of variational methods for such functionals). Nevertheless, since our approach allows us to reduce significantly the degrees of freedom, we see in Figure 5 and Figure 4 how coarsening the mesh in the homogeneous areas and refining it near the edges preserve the accuracy and permit a substancial gain in time even for the sequential approach. We develop in the following section a framework for high performance computing by the parallelization methods.

5. Parallel (in space) method using a domain decomposition

Solving the optical flow problem in the large displacements case can be a time consuming, especially for high resolution sequences. In this section we are going to consider first an overlapping Schwarz domain decomposition technique. Other domain decomposition techniques may be used. We implement the method with the openGL freeware Freefem++ and we will use its MPI [11] facilities.

5.1. Domain decomposition method. The domain decomposition consists of splitting the whole image into several subdomains. The estimation of the optical flow on each subdomain is then computed in parallel by several *Central Processing Unit* (noted CPU) (see figure 6).

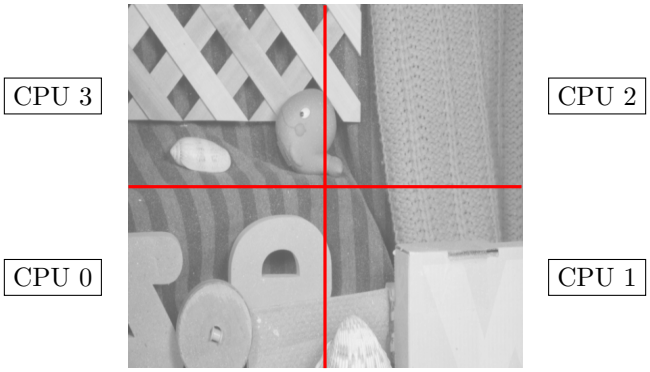


FIGURE 6. Decomposition of an image for four CPU.

As it is shown in the figure 7, we denote Ω_i the part of the image corresponding to the CPU$_i$ and $\mathcal{J}_i$ the set of all indexes j which are neighboors of i. Then, we define

$$\Sigma_{i,j} = \Omega_i \cap \Omega_j \text{ and } \Gamma_{i,j} = \partial\Sigma_{i,j} \backslash \partial\Omega_j.$$

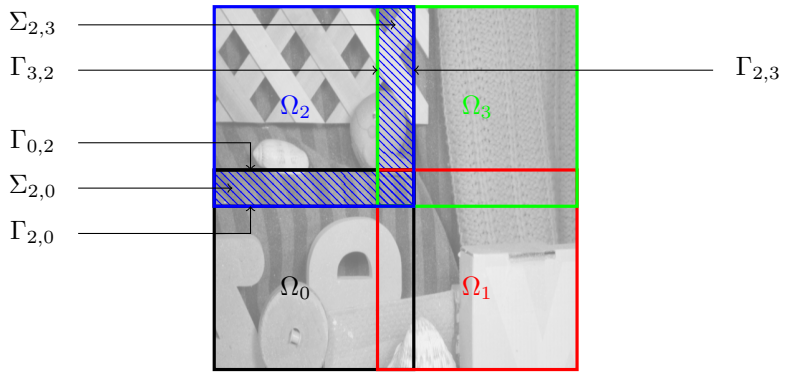


FIGURE 7. Example of notations for CPU $i = 2$.

Within this method, to find an estimation $\mathbf{U}_i$ of the optical flow in the domain Ω_i we have to solve the problem (at each global iteration k):

$$\text{(13)} \qquad \begin{cases} -\text{div}(\mathbf{\Lambda}^k \nabla d\mathbf{U}_i^k) + A^k d\mathbf{U}_i^k = \text{div}(\mathbf{\Lambda}_0 \nabla d\mathbf{U}_i^k) + \mathbf{F}^k \text{ in } \Omega_i \\ \dfrac{\partial d\mathbf{U}_i^k}{\partial n} = 0 \text{ on } \partial\Omega_i \backslash \Gamma_{i,j}, \ \forall j \in \mathcal{J}_i \\ d\mathbf{U}_i^k = d\mathbf{U}_j^{k-1} \text{ on } \Gamma_{i,j}, \ \forall j \in \mathcal{J}_i. \end{cases}$$

As it is known in the domain decomposition literature, the overlapping Schwarz algorithm is convergent (each sequence $(\mathbf{U}_i^k)$ converges to its corresponding solution

$\mathbf{U}_{|\Omega_i}$) and the speed of this convergence depends on the size of the overlap. The larger the overlap is, the faster the convergence will be. However, a large overlapping zones implies also an increasing computations time. In our tests, we have found (by trial and error) that a five pixel overlap is a good balance between this two objectives.

In order to combine the adaptive strategy of the regularization parameter with the domain decomposition method, we consider each subdomain as a single image and apply the adaptive algorithm on each part. It means that we compute the error indicator on each part of the splitted image independently and we perform a local choice of the diffusion coefficients, and the mesh adaptation at the subdomain level. This choice implies that the unstructured submeshes don't match on the overlapping interfaces which add a supplementary interpolation evaluation. At the end, to store and to draw the solution, we perform a last interpolation on the initial regular mesh to obtain a motion vector for each pixel of the image. This is a bit time consuming and may be improved.

5.2. Numerical results. On the figure 8, we present the solution obtained after four iterations of the additive Schwarz algorithm. We have split the initial sequence in four parts. We can see that the interfaces between each subdomains are correctly merged. One can also see that the local adaptation of α and the mesh adaptation give sharper edges on the image.

FIGURE 8. Solution with the domain decomposition method for 4 subdomains after 4 iterations of the Schwarz algorithm.

We give the adapted mesh obtained with this method in the figure 9. We can see that again, the mesh has been refined on the edges.

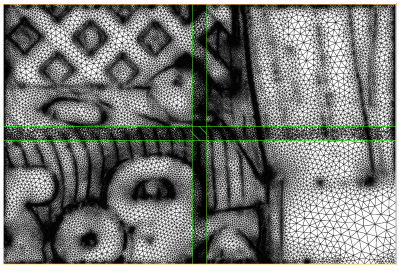


FIGURE 9. Final adapted mesh of the *RubberWhale* test case after 4 iterations of adaptation.

Finally, we present in the figure 10 the computation times for different decomposition of the image. We have detailed the time needed for the principal parts of the

algorithm. We can see that the interpolation evaluations and the communication between the processors increase the time of computations whereas the factorization and the mass matrix assembly decrease in the same proportion as the number of CPUs increase. Besides, if we denote λ the width of an interface and L its length, we may check that both the communication and interpolation time remain bounded by a factor $4L\lambda$ for any decomposition with 2^i, $i \geq 1$ subdomains, which means that the method is nearly scalable in the sens that the perfect scalability is only limited by the bounded time dedicated to communication and the interpolation (this last task may be significantly reduced and so is the mesh adaptation process). We refer the reader to [18] for comparison with the case of small displacement (where the interpolation is almost not required).

To end this section, we notice that we have performed the simulations in an academic parallel environment to serve as a proof-of-the-concept but it is ready to implement in high performance computing context.

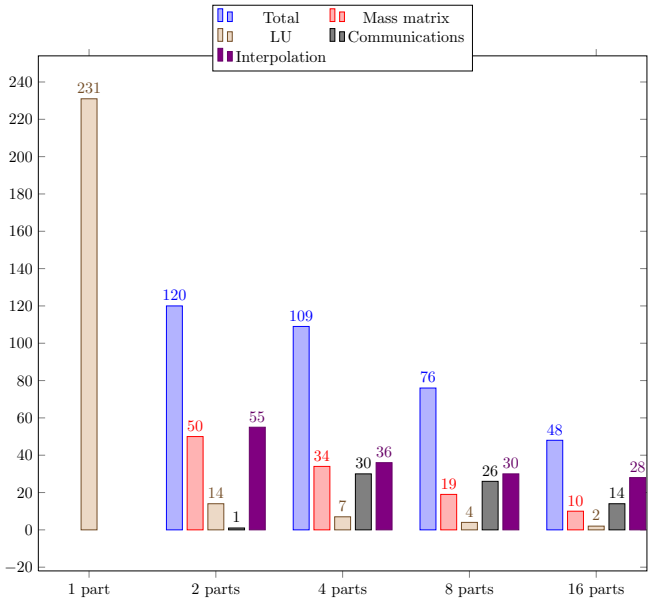


FIGURE 10. Detailed computation times for the domain decomposition method after four iterations of the Schwarz method.

6. Parallel in time approach: parareal method

6.1. principle of the method. The parareal method was introduced in 2001 by J.-L. Lions, Y. Maday, and G. Turinici [23] (see also ([14] and [13]). This is a parallel in time method that consists in coupling a coarse and a fine solvers where the computations of the fine solver are performed in parallel. Both the parareal and the coupling with a domain decomposition method were considered in the few last years, see [5, 12, 19] and the references therein. For optic flow problems, only the small displacements case were considered [18].

Several implementations of the parareal algorithm were compared (Aubanel [1]), including the distributed algorithm presented as the most effective. In this article we have used the classical approach to implement the parareal method. The novelty of our approach relies on the use of the algorithm as a tool for modelling in the

sens that we consider three type of schemes which differ by the system of equations chosen in the fine solver. The convergence analysis is beyond the scope of the article as well as a rigorous proof that the algorithm, as numerical evidences suggest it, allows us for a given accuracy to compute a solution of lowest energy compared to the one given by the sequential algorithm. Such an analysis is necessary and will be considered in a further work.

For simplicity we consider the "unstationnary" optic flow system (τ is a pseudo time)

$$\frac{\partial \mathbf{U}}{\partial \tau} - \operatorname{div}(\mathbf{\Lambda}(\mathbf{x})\nabla \mathbf{U}) + (\mathbf{f}(\mathbf{x}+\mathbf{U}) - M\mathbf{f}(\mathbf{x}))\tilde{\mathbf{f}}(\mathbf{x}+\mathbf{U}) = 0, \tag{14}$$

with an initial and the boundary conditions and where we set

$$\tilde{\mathbf{f}}(\mathbf{x}+\mathbf{U}) = \begin{pmatrix} f_x(x+u_1, y+u_2, t+1) \\ f_y(x+u_1, y+u_2, t+1) \\ f(x,y,t) \end{pmatrix}.$$

First, we consider $0 = T_0 < T_1 < \cdots < T_N = T$ a subdivision of the time interval $[0,T]$. Then, we define a coarse solver $\mathcal{G}$ of time step $\Delta T = \frac{T}{N}$ where N is the total number of coarse time steps. The solver $\mathcal{G}$ solves the following coarse problem on the whole interval $[0,T]$

$$\begin{cases} \mathbf{U}^g_{n+1} = \mathcal{G}(\mathbf{U}^g_n), & n = 0, \ldots, N \\ \mathbf{U}^g_0 = \mathbf{U}^g(T_0 = 0) = 0. \end{cases}$$

We also define a fine solver $\mathcal{F}$ of time step $\Delta\tau = \frac{\Delta T}{M}$ where M is the total number of fine time steps and we solve the fine problem in parallel on each interval $[T_n, T_{n+1}]$ starting from the corresponding coarse solution $\mathbf{U}^g_n$

$$\begin{cases} \mathbf{U}^f_{n,m+1} = \mathcal{F}(\mathbf{U}^f_{n,m}), & m = 0, \ldots, M \\ \mathbf{U}^f_{n,0} = \mathbf{U}^g_n. \end{cases}$$

According to G. Bal, and Y. Maday [3] scheme, we then update the values by using the prediction-correction formula

$$\mathbf{U}^g_{n+1} = \mathcal{G}(\mathbf{U}^f_{n,M}) - \mathbf{U}^g_{n+1} + \mathbf{U}^f_{n,M}.$$

After having sequentially solved the coarse problem, the global iterative process starts including the resolution of the fine problem in parallel on each interval $[T_n, T_{n+1}]$ and the update part. We call these iterations the global iterations of the parareal algorithm.

In the figure 11, we present the L^2 error between the sequential and the parareal method for 4 CPU. We use 4 coarse time steps and 10 fine time steps. We give one graph per global iteration. We can see that 3 global iterations of the method are enough to obtain a good L^2-convergence, and 4 iterations are sufficient to obtain an almost exact similitude between the sequential and parareal solutions. By almost exact similarity we mean that the jump set of the optic flow is computed with the same accuracy (in fact, the energy value of the parareal solution is a bit smaller). Hence, N is the maximum number of global iterations. We notice that a quantitative comparison with previous works on the parareal algorithm have not been yet performed because of the complex nature of our functional setting (nonlinearity and non convexity) but should be conducted in the futur.

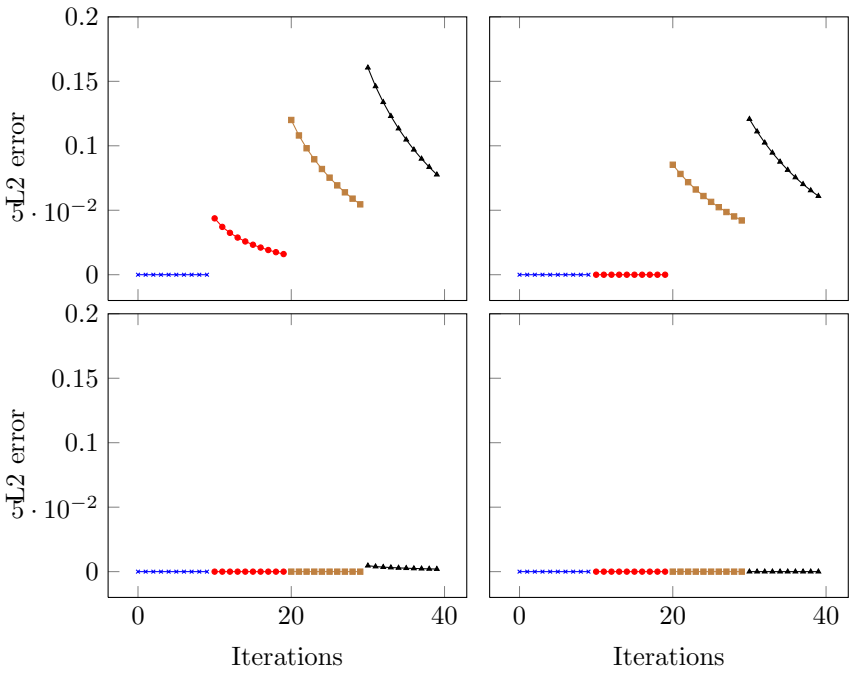


FIGURE 11. Evolution of the L2 error between the parareal method and the corresponding sequential algorithm for each global iteration.

6.2. Strategies of discretization. The choice of the solver $\mathcal{G}$ and $\mathcal{F}$ is an important part of the process. In this article, we implement and compare three different solvers combinations. For all of them, we consider the same coarse solver obtained with the semi-implicit discretisation (7). We emphasize that the parallel in time strategy is not only intended to speedup the computations but is a part of designing an iterative method where the problems solved at the two grids (coarse and fine) may differ with the objective of keeping high accuracy, in the sens of finding a local minimum with the lowest possible energy, and a realistic cost compared to the sequential solver.

Denoting $\delta\mathbf{U}_n^g = \mathbf{U}_{n+1}^g - \mathbf{U}_n^g$, $n \geq 0$, we have the system of equations (on the unknown $\delta\mathbf{U}$),

$$\begin{cases} \frac{\delta\mathbf{U}_n^g}{\Delta T} - \operatorname{div}(\mathbf{\Lambda}(\mathbf{x})\nabla\delta\mathbf{U}_n^g) + \mathbf{A}_n^g\delta\mathbf{U}_n^g = \operatorname{div}(\mathbf{\Lambda}_0\nabla\mathbf{U}_n^g) + \mathbf{F}_n^g, & \text{in } \Omega \\ \frac{\partial\mathbf{U}_n^g}{\partial n} = 0, & \text{on } \partial\Omega \\ \mathbf{U}_{n,0}^g = \mathbf{U}_0, & \text{in } \Omega \end{cases}$$

6.2.1. Semi-implicit discretization for the fine solver. In a first time, we consider the same semi-implicit solver for the fine resolution, which reads: find $\delta\mathbf{U}_{n,m}^f$, $m \geq 0$ solution of

$$\begin{cases} \frac{\delta \mathbf{U}^f_{n,m}}{\Delta T} - \mathrm{div}(\mathbf{\Lambda}(\mathbf{x})\nabla \delta \mathbf{U}^f_{n,m}) + \mathbf{A}^f_{n,m}\delta \mathbf{U}^f_{n,m} = \mathrm{div}(\mathbf{\Lambda}_0 \nabla \mathbf{U}^f_{n,m}) + \mathbf{F}^f_{n,m}, & \text{in } \Omega \\ \frac{\partial \mathbf{U}^f_{n,m}}{\partial n} = 0, & \text{on } \partial\Omega \\ \mathbf{U}^f_{n,0} = \mathbf{U}^g_n, & \text{in } \Omega \end{cases}$$

In the following, we note this method, the GDSI method.

6.2.2. Semi-implicit linear scheme for the fine solver. The second method used in this article consists of using a semi-implicit linear scheme where the non-linearity is fully explicit. In this way, the system (7), with the previous notations, and starting from $\mathbf{U}^f_{n,0} = \mathbf{U}^g_n$, is now given by $m \geq 0$,

$$\begin{cases} \frac{\mathbf{U}^f_{n,m+1} - \mathbf{U}^f_{n,m}}{\Delta t} - \mathrm{div}(\mathbf{\Lambda}(\mathbf{x})\nabla \mathbf{U}^f_{n,m+1}) = \mathbf{F}_{n,m} & \text{in } \Omega \\ \frac{\partial \mathbf{U}^f_{n,m+1}}{\partial n} = 0, & \text{on } \partial\Omega \\ \mathbf{U}^f_{n,0} = \mathbf{U}^g_n, & \text{in } \Omega \end{cases}$$

where

$$\mathbf{F}_{n,m} = \begin{pmatrix} -(f(x+u^k_1, y+u^k_2, t+1) - M^k f(x,y,t)) f_x(x+u^k_1, y+u^k_2, t+1) \\ -(f(x+u^k_1, y+u^k_2, t+1) - M^k f(x,y,t)) f_y(x+u^k_1, y+u^k_2, t+1) \\ (f(x+u^k_1, y+u^k_2, t+1) - M^k f(x,y,t)) f(x,y,t) \end{pmatrix}.$$

We call this method GDEx. This scheme shows how the parareal method allows us to use a "simplified" physics at a given scale (here the finer one).

6.2.3. Using a small displacement model for the fine solver. Finally, we consider that the fine solver is fine enough to use the model of the optical flow problem in small dispacements [18], [16] given in the stationary case by

$$\begin{cases} -\mathrm{div}(\mathbf{\Lambda}(\mathbf{x})\nabla \mathbf{U}) + \mathbf{A}\mathbf{U} = \mathbf{F} \text{ in } \Omega \\ \frac{\partial \mathbf{U}}{\partial n} = 0 \text{ on } \partial\Omega \end{cases} \tag{15}$$

where

$$\mathbf{U} = \begin{bmatrix} u_1 \\ u_2 \\ m_t \end{bmatrix}, \ \mathbf{A} = \begin{bmatrix} f_x(x,y,t)^2 & f_x(x,y,t)f_y(x,y,t) & -f_x(x,y,t)f(x,y,t) \\ f_y(x,y,t)f_x(x,y,t) & f_y(x,y,t)^2 & -f_y(x,y,t)f(x,y,t) \\ -f(x,y,t)f_x(x,y,t) & -f(x,y,t)f_y(x,y,t) & f(x,y,t)^2 \end{bmatrix}$$

and

$$\mathbf{F} = \begin{bmatrix} -f_x(x,y,t)f_t(x,y,t) \\ -f_y(x,y,t)f_t(x,y,t) \\ f(x,y,t)f_t(x,y,t). \end{bmatrix}.$$

The fine solver of this third method called GDPD is given by

TABLE 3. Comparison of the computation time between the GDSI and GDEx algorithms and an equivalent sequential algorithm (Seq).

Method	Time (s)
Seq	1955
GDSI	1774
GDEx	1534
GDPD	1300

$$
(16)\qquad \begin{cases} \frac{\mathbf{U}^{m+1}-\mathbf{U}^{m}}{\Delta t} - \operatorname{div}(\mathbf{\Lambda}(\mathbf{x})\nabla \mathbf{U}^{m+1}) + \mathbf{A}\mathbf{U}^{m+1} = \mathbf{F}^{m}, & \text{in } \Omega \\ \frac{\partial \mathbf{U}^{f}_{n,m+1}}{\partial n} = 0, & \text{on } \partial\Omega \\ \mathbf{U}^{f}_{n,0} = \mathbf{U}^{g}_{n}, & \text{in } \Omega \end{cases}
$$

This scheme is rather different than the two previous ones. In fact, the coefficients $\mathbf{A}$ and the nonlinearity $\mathbf{F}$ in this scheme are evaluated at all the time instants $(T_i)_i$ of the parareal subdivision while the first schemes use only image deformations at the fixed time instants $T_0 = t$ and $T_N = t+1$. Therefore, the equation obtained at the fine scale now is different from the others. A closer look to this scheme reveals that the Large displacements model (M) is now decomposed into a given number of small displacements $(S_i)_i$ and finding a minimum of (M) is obtained by following a path which passes by the minima of $(S_i)_i$. Since the local energies (of (S_i)) are strictly convex each minimum of an S_i is unique, the aim is to get this way a better chance to select a good local minimum among those of the global energy (of (M)). Numerical results tend to confirm this point, however we have no mathematical proof of this fact.

In this parareal method, we also want to use the adaptive control of the regularization α_i and the mesh adaptation since we have seen that it is essential to preserve the edges. We apply these adaptations, explained in section 2, during the coarse resolution. Thus, it is performed once and serves to detect the singularities and prevents over-smoothing near the edges without increasing the cost of computations. The new mesh and the new function $\alpha_i(\mathbf{x})$ are communicated between all processors in the same time that the values $\mathbf{U}^{g}_{n}$.

6.3. Numerical results. In order to evaluate the strategies GDSI, GDEx and GDPD we again use the test case *RubberWhale* from Middleburry. Comparing the computation times (see table 3) with an equivalent sequential algorithm, we can see that both parareal methods allow to save time.

The GDEx and GDPD methods are faster compared to the GDSI method but on the figure 12, we can see that even if the solution for these methods presents some sharp edges, the global solution with the GDSI method is more accurate. Thus, we will keep the GDSI method in the following.

7. A space-time parallel method

7.1. Strategy. The speeudup of the parareal method si limited in the sens that it does not permit high performane computing and the scalability. The speedup is smaller than domain decomposition (see [18] and [17] for details). However, with its

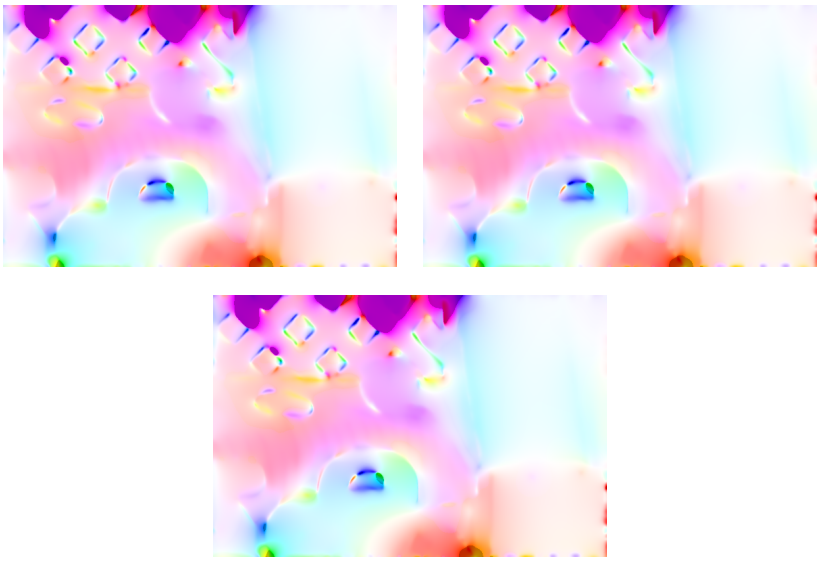


FIGURE 12. Estimation of optical flow for large displacement *RubberWhale* test case. Left: GDSI method. Right: GDEx method. Bottom: GDPD method.

ability of using different schemes and physics between the fine and the coarse solver, it remains a relevant choice to solve complex systems (nonlinear, non-convex, ...).

Instead of assigning a single processor to a subdomain of the image, we now attribute a group of CPU. The repartition of the different processors in each group is done thanks to the formula

$$\text{groupe}(\text{CPU}_i) = \frac{\text{CPU}_i}{\text{nbPart}}.$$

In this way we can apply the parareal method presented in the section 6 in each subdomain of the image. The interfaces between these parts are treated with the domain decomposition method presented in the section 5.1.

7.2. Numerical results. On the figure 13, we present the solution obtained with the coupling between the parareal method (GDSI) and the domain decomposition method. We can see that the solution keeps the regularity of the parareal method and the sharp edges given by the domain decomposition method.

FIGURE 13. Optical flow obtained with the coupling between the parareal and the domain decomposition method.

More, on the figure 14, we show that this coupling allows to strongly reduce the computation time of the GDSI method.

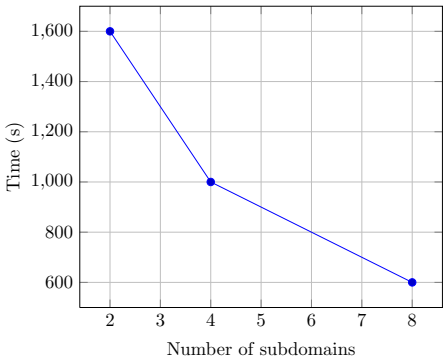


FIGURE 14. Computation times of the coupling between the parareal and the domain decomposition method for different numbers of subdomains.

Conclusion

In this article, we have introduced a novel approach to solve the optic flow problem in large displacements and varying illumination case based on the extension of the variational framework of the small displacements. A first expectation is to describe accurately the motion in a given scene by preserving relevant features, such as the edges, in the computed solution. Secondly, to find discrete strategies fulfilling several objectives, among which cost reduction and time saving computational environnement. The problem is then very challenging and encompasses several difficulties, non convexity, strong nonlinearity, etc. We have seen that the use of a variational formulation and a finite element method yields satisfactory results within an elegant framework. The use of unstructured meshes and adaptive control of the parameters, allow us to fit more accurately the "geometry of the motion field" and to obtain a sparse optic flow as well as a more realistic varying illumination modelling than the linear case.

In a second step, we have implemented two parallel methods, an overlapping Schwarz domain decomposition method and the parareal method. The two approaches appear complementary, in the sens that the domain decomposition is not only efficient to parallelize the solver but also to treat high resolution images. It is also "nearly scalable" and allows a satisfactory speedup. The parareal method, even not the top-of-the-art algorithm in this article ([1]), is very suitable to modelize the large displacements as a "sum" of small displacements and also it permits to use different physics at the different scales which is a promising feature for solving minimization problems of complex energy functionals.

From the computational point of view, each parallel method reduces the cost and the time of the resolution and their combination results in a very efficient and flexible fully parallel method. We have implemented these ideas in an academic computational environnement as a proof-of-the concept and showed that a more involved high performance computing will increases the time saving. We think that the methods presented here have also great potential for solving other variational problems where the energy functional is not convex and severely nonlinear.

References

[1] E. Aubanel. Scheduling of tasks in the parareal algorithm. *Parallel Computing*, 37:172–182, 2011.

[2] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60:156–182, 1999.

[3] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In *Recent Developments in Domain Decomposition Methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 189–202. Springer Berlin Heidelberg, 2002.

[4] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.

[5] M. Bedez, Z. Belhachmi, 0. Haeberlé, A. Greget, S. Moussaoui, JM. Bouteiller, and S. Bischoff. A fully parallel in time and space algorithm for simulating the electrical activity of a neural tissue. *Journal of Neuroscience Methods*, 257 15:17–25, 2016.

[6] Z. Belhachmi and F. Hecht. Control of the effects of regularization on variational optic flow computations. *Journal of Mathematical Imaging and Vision*, 40:1–19, 2011.

[7] Z. Belhachmi and F. Hecht. An adaptive approach for segmentation and TV denoising in the optic flow estimation. *JMIV*, 54 (3):358–377, 2016.

[8] A. Bruhn. *Variational optic flow computation: Accurate modelling and efficient numerics*. PhD thesis, Department of Mathematics and Computer Science, Saarland University, Saarbrucken, Diss, 2006.

[9] A. Bruhn, T. Brox, N. Papenberg, and J. Weickert. A survey on variational methods for small displacements. In Vol 10 O. Scherzer (Ed.) Mathematics in Industry, editor, *Mathematical Models for Registration and Applications to Medical Imaging*, pages 103–136. Springer Berlin Heidelberg, 2006.

[10] A. Bruhn, J. Weickert, and C. Schnorr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61:211–231, 2005.

[11] O. Pironneau F. Hecht. In *FreeFem++*. www.freefem.org.

[12] M.-J. Gander, Y.-L. Jiang, and R.-J Li. Schwarz waveform relaxation methods. In R. Bank, M. Holst, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XX*, volume 91 of *Lecture Notes in Computational Science and Engineering*, pages 451–458. Springer Berlin Heidelberg, 2013.

[13] M.-J. Gander and S. Vandewalle. On the superlinear and linear convergence of the parareal algorithm. In O. B. Widlund and D. E. Keyes, editors, *Domain Decomposition Methods in Science and Engineering XVI*, pages 291–298. Springer Berlin Heidelberg, 2007.

[14] M.J. Gander, Y.-L. Jiang, and R.-J Li. Parareal schwarz waveform relaxation methods. In *Domain Decomposition Methods in Science and Engineering XX*, volume 91 of *Lecture Notes in Computational Science and Engineering*, pages 451–458. Springer Berlin Heidelberg, 2013.

[15] M.A. Gennert and S. Negahdaripour. Relaxing the brightness constancy assumption in computing optical flow. *Technical Report*, 1987.

[16] D. Gilliocq-Hirtz. *Techniques variationnelles et calcul parallle en imagerie. Estimation du flot optique avec luminosit variable en petits et larges dplacements*. PhD thesis, Laboratoire de Mathmatiques, Informatique et Applications, Universit de Haute Alsace, Mulhouse, France, 2016.

[17] D. Gilliocq-Hirtz and Z. Belhachmi. Coupling parareal and adaptive control in optical flow estimation with application in movie's restoration. pages 1–6. IEEE, Jan 2015.

[18] D. Gilliocq-Hirtz and Z. Belhachmi. A massively parallel multi-level approach to a domain decomposition method for the optical flow estimation with varying illumination. *SMAI Journal of Computational Mathematics*, 2:121–140, 2016.

[19] R. Guetat. Coupling parareal with non-overlapping domain decomposition method. *https://hal.archives-ouvertes.fr/hal-01312528/*, 2016.

[20] B.K. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[21] P. Jolivet, V. Dolean, F.; Hecht, F. Nataf, C. Prud'Homme, and N. Spillane. High performance domain decomposition methods on massively parallel architectures with freefem++. *J. Numer. Math.*, 20 (3-4):287C302, 2012.

[22] D. Kondermann, A. Abraham, L. Brostow, W. F?rstner, S. Gehrig, A. Imiya, B. J?hne, F. Klose, M. Magnor, H. Mayer, R. Mester, T. Pajdla, R. Reulke, and H. Zimmer. On performance analysis of optical flow algorithms. In F. Dellaert, J.-M. Frahm, M. Pollefeys,

B. Rosenhahn, and L. Leal-Taixe, editors, *Outdoor and Large-Scale Real-World Scene Analysis, Lecture Notes in Computer Science (7474)*, pages 329–355. Springer Berlin Heidelberg, 2012.

[23] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *Comptes Rendus de l'Académie des Sciences, Série I, Mathematics*, 332:661–668, 2001.

[24] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, pages 674–679. Morgan Kaufmann Publishers Inc., 1981.

[25] Y. Mileva, A. Bruhn, and J. Weickert. Illumination-robust variational optical flow with photometric invariants. In F. Hamprecht, C. Schnorr, and B. Jahne, editors, *Pattern Recognition*, pages 152–162. Springer Berlin Heidelberg, 2007.

[26] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *Journal of Computational and Applied Mathematics*, 50:67–83, 1994.

[27] J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: From continuous models to algorithms. In Las Palmas de Gran Canaria L. Alvarez, IWCVIA03, editor, *International Workshop on Computer Vision and Image Analysis*, volume 3, pages 1–6, 2003.

[28] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14:245–255, 2001.

LMIA-IRIMAS, Université de Haute Alsace et Université de Strasbourg, 6 rue des Frères Lumière, 68093 Mulhouse, France.

E-mail: `diane.gilliocq-hirtz@uha.fr`

LMIA-IRIMAS, Université de Haute Alsace et Université de Strasbourg, 6 rue des Frères Lumière, 68093 Mulhouse, France

E-mail: `zakaria.belhachmi@uha.fr`