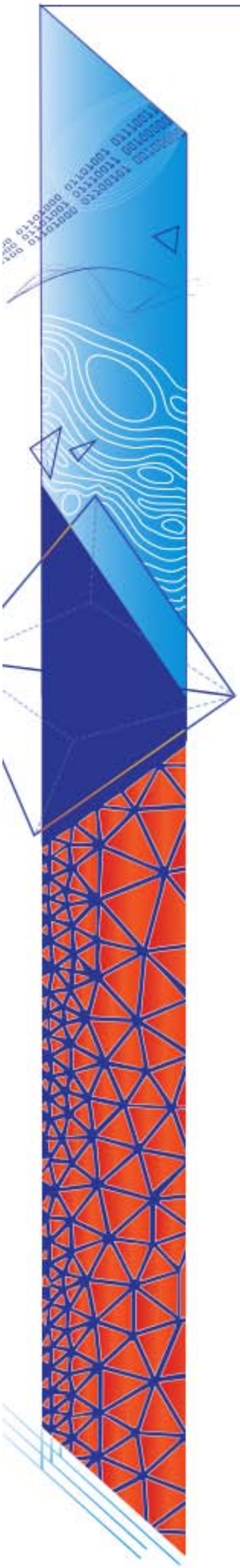

SMAI-JCM

SMAI Journal of Computational Mathematics

A massively parallel multi-level approach to a domain decomposition method for the optical flow estimation with varying illumination

Diane Gilliocq-Hirtz & Zakaria Belhachmi

Volume 2 (2016), p. 121-140.

<http://smai-jcm.cedram.org/item?id=SMAI-JCM_2016__2__121_0>

SMAI Journal of Computational Mathematics
Vol. 2, 121-140 (2016)

A massively parallel multi-level approach to a domain decomposition method for the optical flow estimation with varying illumination

Diane Gilliocq-Hirtz [1]
Zakaria Belhachmi [2]

[1] LMIA, 6 rue des Frères Lumière, 68093 Mulhouse, France
E-mail address: diane.gilliocq-hirtz@uha.fr
[2] LMIA, 6 rue des Frères Lumière, 68093 Mulhouse, France
E-mail address: zakaria.belhachmi@uha.fr.

Abstract. We consider a variational method to solve the optical flow problem with varying illumination. We apply an adaptive control of the regularization parameter which allows us to preserve the edges and fine features of the computed flow. To reduce the complexity of the estimation for high resolution images and the time of computations, we implement a multi-level parallel approach based on the domain decomposition with the Schwarz overlapping method. The second level of parallelism uses the massively parallel solver *MUMPS*. We perform some numerical simulations to show the efficiency of our approach and to validate it on classical and real-world image sequences.

Keywords. optical flow, varying illumination, domain decomposition, adaptive control, finite element method, variational method, multi-level parallelism.

1. Introduction

In the last decades, the estimation of the optical flow has become a popular and central problem in computer vision ([18], [9], [19], [2]). It is involved, for example, in almost all the movies and pictures compression processes, or in the obstacle detection in the new smart cars and in robotics. The modelling and the detection of the motion in a scene involve numerous difficulties (e.g. the aperture problem, occlusions, ... [18]). Among such difficulties, for the optical flow estimation, one with growing importance is the cost of the method in term of computation time (and the storage), which rises with the increasing resolution of the images due the technological devices progress. Up to now, there exist numerous methods for the optical flow estimation among which the Partial Differential Equations and particularly, the variational methods turn to be very efficient. They offer a complete framework which consists of mathematically founded continuous models, and a large number of numerical methods ([18], [9], [19], [2]). They allow to cope with the ill-posedness of the optical flow problem, due to the aperture problem [9], by including a large range of regularization procedures. In this article, we consider a variational method based on the linear Horn and Schunck approach [12] but with a variable regularization parameter. This method introduced in [5] is proved to be an efficient approach to solve the optical flow problem in the sense that it is edge preserving and low cost (in term of degrees of freedom) [6]. A large part of the research works in the literature deals with the determination of optical flow under an assumption of constant illumination. This assumption is not satisfied in general and becomes a source of inaccurate estimation and serious limitations in many applications. Besides, modelling a varying illumination is quite complex and increases the ill-posed character of the problem. In this article, following Gennert and Negahdaripour [10], we assume given an a priori law for the illumination variations and we introduce a supplementary variable, which models the illumination, in the system of equations. We show that the obtained partial differential equations system solved in the framework of the adaptive variational approach [5] is an efficient and innovative method for the optical flow estimation with varying illumination.

A classical criticism against the variational methods is their "complexity" and their potential time consuming character, particularly when using unstructured meshes and finite element method discretization. In a previous work [4], we have used a parallel in time algorithm to reduce the computational time. To go further, we are going to implement a decomposition domain method. The main contribution of this article is to propose and to validate an efficient massively parallel multi-level solver using domain decomposition method to solve the optical flow problem with varying illumination, following the variational adaptive approach proposed in [5]. We obtain the optical flow with an accurate estimation and a significant reduction of the time of computations. We validate our method on a classical benchmark and two real-world image sequences.

In Section 1, we recall the Horn and Schunck model for the optical flow estimation and we give the system of equations to solve in the case where we allow illumination variations. In Section 2, we briefly recall the finite element method and we will rewrite the discrete optical flow system. We also define the adaptive strategy and give the resulting algorithm. In Section 3, we will consider the domain decomposition method with overlapping and the additive Schwarz method used to solve the algebraic problem. The Section 4 concerns the numerical simulations. We present some results obtained with our method, in particular, we give in detail the optical flow estimation for the so-called RubberWhale sequence to show the performances of the approach. We also perform the analysis of the computation time of the massively parallel algorithm. Finally, we will present two examples of the computation of the optical flow for real-world sequences.

2. Optical flow problem

We consider a sequence of two successive frames where $\Omega \subset \mathbb{R}^2$ is the image domain. The intensity of a pixel (x, y) at an instant t is defined by the function:

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

As it is usually done [3], we use a convolution with a Gaussian kernel K_σ of standard deviation σ to work with smoothed images. We define the smoothed image sequence by:

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

The optical flow $\mathbf{u} = (u_1, u_2)$ represents the vector field describing the motion of each pixel in the sequence. Following Horn and Schunck [12], we first consider the brightness constancy assumption. It means that the brightness intensity stays the same between two successive frames:

$$f(x,y,t) = f(x+u_1, y+u_2, t+1) \tag{2.1}$$

which is equivalent to say that the variation in time is null:

$$\frac{\partial f}{\partial t}(x,y,t) = 0.$$

Since the displacements are supposed to be small, we can assume that f is $\mathcal{C}^1([0,\infty[\,;\mathbb{R})$. So by using a first order Taylor expansion, we obtain the fundamental constraint of the optical flow:

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} = 0. \tag{2.2}$$

The time derivatives of x and y represent the component u_1 and u_2 of the optical flow. Hence, with the notation $f_* = \frac{\partial f}{\partial *}$ the equation (2.2) becomes:

$$f_xu_1 + f_yu_2 + f_t = 0. \tag{2.3}$$

In this way, we have to determine two unknowns u_1 and u_2 with only one equation. This is the so-called *aperture problem* illustrated in the figure 2.1. On each sub-figure, we represent by an arrow the spottable motion in a local neighborhood. In the example 1, if the line is moving horizontally, we can't see any difference. The only visible motion is the vertical displacement. In the same way, in the example 2, only the horizontal motion can be detected. The example 3 represents the case where we are on a corner. In this case, both horizontal and vectical motions can be detected.

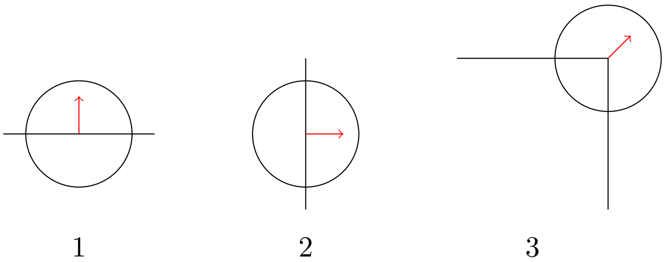

Figure 2.1. Representation of the aperture problem.

To go through this ill-posedness, Lucas and Kanade [14] assume that the motion is constant in a neighborhood of size ρ. Contrary to this local assumption, Horn and Schunck propose [12] a global approach to overcome the aperture problem. They introduce a regularization parameter α which acts as a penalizer and leads to a smoother flow field (the bigger α is, the smoother the flow is). The idea of Bruhn and Weickert [9] is, to combine both methods and minimize the functional:

$$\int_\Omega K_\rho \star (f_xu_1 + f_yu_2 + f_t)^2 + \alpha(|\nabla u_1|^2 + |\nabla u_2|^2)dxdy \tag{2.4}$$

where K_ρ is a Gaussian deviation of parameter ρ and α is a constant regularization parameter.

On real-world images, the assumption of the brightness constancy is no longer verified. Occlusions, shadows or glints don't meet this constraint. Hence, the estimation of the previous model is not accurate so we need to consider another assumption. Different approaches were proposed to model illumination variations, such as the assumption of the constancy of the gradient amplitude [8] or, in the case of color images, we can cite the work with variables less sensitive to such illumination changes [16]. In this article, following Gennert and Negahdaripour [10], we consider a varying illumination obeying an affine transformation. This assumption, even covering a wide range of applications, might appear as a strong one. Moreover, it introduces a supplementary unknown, enforcing the ill-posedness of the optical flow estimation. To balance such potential shortcomings, we couple this modelling with an adaptive control of the regularization of the parameter associated to the new unknown. In this article, we restrict ourselves to smooth variations, keeping the parameter large enough. The assumption allowing a linear motion of the brightness intensity between the two images becomes:

$$f(x + u_1, y + u_2, t + 1) = M(x, y, t)f(x, y, t) + T(x, y, t)$$

where M is the multiplier and T is the translator. In our case, we suppose that the translator is negligible. The smaller the displacement is, the closer to 1 M is. In this way, we can set:

$$M = 1 + \delta m \tag{2.5}$$

where δm tends to zero when the displacement is very small. That gives the new equation of the optical flow:

$$f_x u_1 + f_y u_2 + f_t - f m_t = 0 \tag{2.6}$$

where m_t is the derivative of M. For more details, we refer the reader to [10]. The optical flow problem allowing varying illumination consists finally of minimizing the functional:

$$\int_\Omega K_\rho \star (f_x u_1 + f_y u_2 + f_t - m_t f)^2 + \alpha(|\nabla u_1|^2 + |\nabla u_2|^2) + \lambda |\nabla m_t|^2 dxdy. \tag{2.7}$$

According to Euler-Lagrange equations, we have the system:

$$\begin{cases} -\mathrm{div}(\Lambda \nabla \mathbf{U}_\Lambda) + A_\rho \mathbf{U}_\Lambda = \mathbf{F} \text{ in } \Omega, \\ \dfrac{\partial \mathbf{U}_\Lambda}{\partial n} = 0 \text{ on } \partial\Omega, \end{cases} \tag{2.8}$$

with

$$\Lambda = \begin{bmatrix} \alpha \\ \alpha \\ \lambda \end{bmatrix}, \mathbf{U}_\Lambda = \begin{bmatrix} u_1 \\ u_2 \\ m_t \end{bmatrix}, \mathbf{A}_\rho = \begin{bmatrix} K_\rho \star (f_x)^2 & K_\rho \star (f_x f_y) & -K_\rho \star (f_x f) \\ K_\rho \star (f_y f_x) & K_\rho \star (f_y)^2 & -K_\rho \star (f_y f) \\ -K_\rho \star (f f_x) & -K_\rho \star (f f_y) & K_\rho \star (f)^2 \end{bmatrix},$$

and

$$\mathbf{F} = \begin{bmatrix} -K_\rho \star (f_x f_t) \\ -K_\rho \star (f_y f_t) \\ K_\rho \star (f f_t) \end{bmatrix}.$$

One can note that in the case where $f(x, y, t) = 0$, the coefficient f^2 of the matrix $\mathbf{A}_\rho$ is null which implies that this matrix is no longer degenerated. To prevent this issue, when we are in this situation we consider the term $f^2 + \varepsilon$ instead of f^2 where ε is a small parameter in practice equals to 10^{-8}.

It is well known that taking α and λ constant, even well chosen, leads to undesired oversmoothing and blurs the edges of an image. Thus, following the idea of ([5], [6]), we will consider the general setting where α and λ are discontinuous and piecewise constant functions in order to prevent these smoothing effects. Indeed, as proved in [6], choosing a small value of α in regions where there are edges gives sharper edges, then an improved restitution of the motion and its segmentation. We refer to [6] for more details. In particular, we consider a subdivision of the whole space Ω:

$$\Omega = \underset{i \in \mathcal{I}}{\cup} \Omega_i.$$

On this subdivision, we define the two piecewise functions α and λ by:

$$\alpha = (\alpha_i), \quad \lambda = (\lambda_i).$$

We denote by α_m and α_M, the non-negative minimal and maximal values of α. In the same way, we define λ_m and λ_M. Under these notations, we have the following theorem.

Theorem 2.1. *The problem* (2.8) *has a unique solution in* $H^1(\Omega)$*.*

The well-posedness is obtained from the Lax-Milgram lemma. By assuming that Ω is Lipschitz, we can work in the Hilbert space $H^1(\Omega)$. We define the norm:

$$\|\mathbf{U}\|^2_{\rho,\Lambda} = \|\Lambda^{\frac{1}{2}} \nabla \mathbf{U}\|^2_{L^2(\Omega)} + \|\mathbf{A}_\rho^{\frac{1}{2}} \mathbf{U}\|^2_{L^2(\Omega)}$$

where $\|\mathbf{A}_\rho^{\frac{1}{2}} \mathbf{U}\|^2_{L^2(\Omega)} = (\mathbf{A}_\rho \mathbf{U}, \mathbf{U})$. Let denote $\beta_M = \max(\alpha_M, \lambda_M)$ and
$\beta_m = \min(\alpha_m, \lambda_m)$, a straightforward extension of ([6], proposition 2.7) gives the following proposition.

Proposition 2.2. *Let $\boldsymbol{U}$ be a solution of (2.6). There exists $C > 0$ independent of α such that for $\boldsymbol{U}_\Lambda$, the solution of the optical flow problem where the regularization is a piecewise constant function, the following inequalities hold:*

$$\| \boldsymbol{U}_\Lambda \|_{\rho,\Lambda} \leq C \| A_\rho^{\frac{1}{2}} \boldsymbol{U} \|_{L^2(\Omega)}. \tag{2.9}$$

and

$$\| \boldsymbol{U} - \boldsymbol{U}_\Lambda \|_{\rho,\Lambda} \leq C \left(\frac{\beta_M}{\beta_m} \right)^{\frac{1}{2}} \| \alpha^{\frac{1}{2}} \nabla \boldsymbol{U}_\Lambda \|_{L^2(\Omega)}. \tag{2.10}$$

From now on, we denote $\mathbf{U}$ in place of $\mathbf{U}_\Lambda$ for brevity.

3. Finite elements implementation

3.1. Variational form

In order to solve the system (2.8), we use the finite element method. To do so, we need to write this equation under its variational form. We multiply the first equation of (2.8) by a test function $\boldsymbol{\varphi}$, we integrate over Ω and by using the Green formula on the first integral, we have:

$$\int_\Omega \Lambda \nabla \mathbf{U} \nabla \boldsymbol{\varphi} d\mathbf{x} + \int_\Omega \mathbf{A}_\rho \mathbf{U} \boldsymbol{\varphi} d\mathbf{x} = \int_\Omega \mathbf{F} \boldsymbol{\varphi} d\mathbf{x}, \qquad \forall \boldsymbol{\varphi} \in (H^1(\Omega))^3. \tag{3.1}$$

3.2. Discretization

We consider a regular triangular mesh $\mathcal{T}_h$ where each pixel of the image consists of two triangles as it is shown on the figure 3.1.

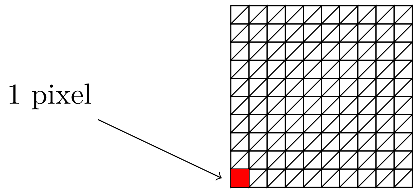

Figure 3.1. Representation of the initial mesh $\mathcal{T}_h$. Each square represents a pixel of the image and consists of two triangles.

We define the space of approximations by:

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

where $P_1(K)$ is the space of the linear functions on $K \in \mathcal{T}_h$. If we set $A_{\rho,h}$ a finite element approximation of A_ρ, the discrete problem of the optical flow states:

$$\begin{cases} \text{Find } \mathbf{U}_h \in \mathcal{V}_h^3, \text{ such that} \\ \displaystyle\int_{\mathcal{T}_h} \Lambda \nabla \mathbf{U}_h \cdot \nabla \mathbf{V}_h dxdy + \int_{\mathcal{T}_h} \mathbf{V}_h^T \mathbf{A}_{\rho,h} \mathbf{U}_h dxdy = \int_{\mathcal{T}_h} \mathbf{F}_h \cdot \mathbf{V}_h dxdy, \quad \forall \mathbf{V}_h \in \mathcal{V}_h^3. \end{cases} \tag{3.2}$$

We can show, by using the Lax-Milgram lemma that this weak formulation admits a unique solution.

3.3. Adaptive regularization

In this part, we are interested in controlling the parameter α. The regularization parameter is now considered as a piecewise constant function. This local choice of α is based on an a posteriori strategy analysis and was proposed by Belhachmi and Hecht [5]. The choice of the regularization is motivated by the fact that a small value is usefull to correctly approximate the Neumann boundary conditions on the edges of objects. However, it increases the maximum value of the optical flow. So, in order to have a better estimation, we prefer a large regularization. The local choice of α allows to decrease its value on regions where we need a small α and keep a large value in the rest of the image. In our case, we have the additional unknown m_t with its regularization parameter λ and in a first time, we keep this parameter constant.
Since we want to locally choose the regularization parameter, we will have a large ratio $\frac{\beta_M}{\beta_m}$, so we will use the inequality (2.10) in the error indicator. The control of the regularization is done through an error indicator which is given for each element $K \in \mathcal{T}_h$ by:

$$\eta_K = \Lambda_K^{-\frac{1}{2}} h_K \|\mathbf{F}_h + \operatorname{div}(\Lambda_K \nabla \mathbf{U}_{\Lambda,h}) + A_{\rho,h}\mathbf{U}_{\Lambda,h}\|_{L^2(K)^2} + \frac{1}{2}\sum_{e\in\varepsilon_K} \Lambda_e^{-\frac{1}{2}} h_e^{\frac{1}{2}} \|[\Lambda \nabla \mathbf{U}_{\Lambda,h}\cdot \mathbf{n}_e]_e\|_{L^2(e)^2} \quad (3.3)$$

where ε_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 normal vector from e, Λ_e is the maximum between the Λ of the two neighbors of an edge, and $[.]_e$ represents the jump over the edge e which means the difference between the outside and inside values. The error indicator η_K describes the finite element error and the model error. On the potential set of discontinuities, this value is large because $\nabla U_{\Lambda,h}$ is large. In fact, when the brightness is abruptly changing in an area, it means that we are close to an edge for the optical flow. So, to improve the solution, we decrease α from the two first components of Λ and the third component λ stays constant. The decreasing formula for α is given by:

$$\alpha_K^{n+1} = \max\left(\frac{\alpha_K^n}{1+\kappa\max\left(\frac{\eta_K}{\|\eta_K\|_\infty}-\eta, 0\right)}, \alpha_{th}\right) \quad (3.4)$$

where κ is an arbitrary control parameter and α_{th} is a threshold. In this way, if the relative error (measured with η_K) is greater than a given value η we reduce in K, the value of α. On the other hand, if it is less than η the denominator is equal to one and so α remains unchanged. This local and adaptive control of α is implemented with the algorithm below.

(1) Compute of a first approximation $\mathbf{U}_\alpha^0$ of the optical flow. This estimation is done on a cartesian grid T_h^0 where we have one cell per pixel. Define $i = 0$.

(2) $i = i + 1$. Build an adapted mesh T_h^i with the metric error indicator.

(3) Update of $\alpha_i(x, y)$ on T_h^i.

(4) Compute $\mathbf{U}_\alpha^i$ on T_h^i.

(5) Go to step 2.

The first value of the regularization α_0 is arbitrarily chosen to be a large value. Typically, we take $\alpha_0 = 2000$. In this way, the homogeneous part will be well estimated. On edges, the regularization will be reduced by the procedure in order to avoid a lack of precision on discontinuities. The convergence of this algorithm when the mesh size goes to zero is proved in [6]. In practice, the iterative algorithm stops when the L2 error between $\mathbf{U}_\Lambda^n$ and $\mathbf{U}_\Lambda^{n+1}$ reaches a threshold set at 10^{-3}. In the numerical result section we will see that only a few iterations of mesh adaptation is enough to obtain a well adapted mesh.

3.4. Adaptive control of parameter λ

As for the regularization parameter α, we propose to study the effect of a control of the parameter λ. In the problem (2.8), the variable m_t increases with respect to the illumination variation. We want to reduce the effect of this variable in regions where the brightness remains unchanged without changing anything where it is varying. To do so, we are going to apply the same strategy than the adaptation of α on the parameter λ using the following formula:

$$\lambda_K^{n+1} = \max\left(\frac{\lambda_K^n}{1+\zeta\max\left(\frac{\|m_t\|_2}{\|m_t\|_\infty}-\zeta',0\right)},\lambda_s\right) \tag{3.5}$$

with ζ a constant parameter controlling the speed of decrease of λ and ζ' a threshold. As we will see in the numerical results, even if this work is just at its beginning, the first estimations seem to be encouraging. This part will be more detailed in a forthcoming work.

4. Domain Decomposition

4.1. Image decomposition

In the case where we want to estimate the optical flow between two large images, we have implemented a domain decomposition. Indeed, as we have seen on the figure 3.1, if we have large number of pixels on the picture, we have a large number of triangles in the mesh. In the finite element method, the resolution consists of solving a large system with a large number of degrees of freedom. The aim is that every Central Processing Unit (CPU) computes the optical flow of one part of the image (see figure 4.1). So, the size of the problem to solve is reduced by a factor equivalent to the number of CPU used. The domain decomposition method allows us to obtain performances which are difficult to reach with classical variational methods and even worse with the adaptive process to control the parameter α.

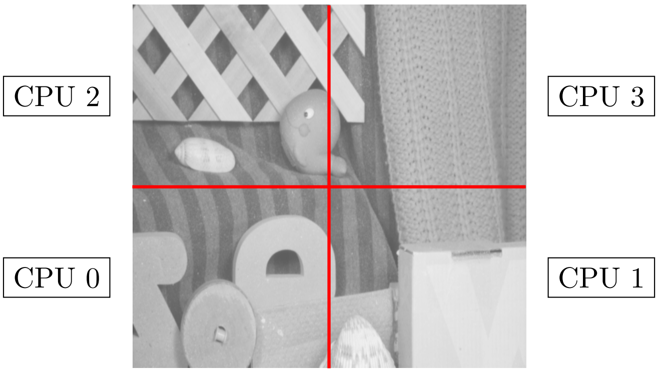

Figure 4.1. Decomposition of an image for four CPU.

Since the optical flow estimation is sensitive at the boundaries, we use an additive Schwarz method to improve the estimation on the interfaces. This method requires an overlapping between the subdomains.

4.2. Model decomposition

We note Ω_i the part of the image corresponding to the CPU$_i$ and $\mathcal{J}_i$ the set of all indexes j which are neighbors to i. We define $\Sigma_{i,j} = \Omega_i \cap \Omega_j$ and $\Gamma_{i,j} = \partial\Sigma_{i,j}\backslash\partial\Omega_j$ (See figure 4.2).

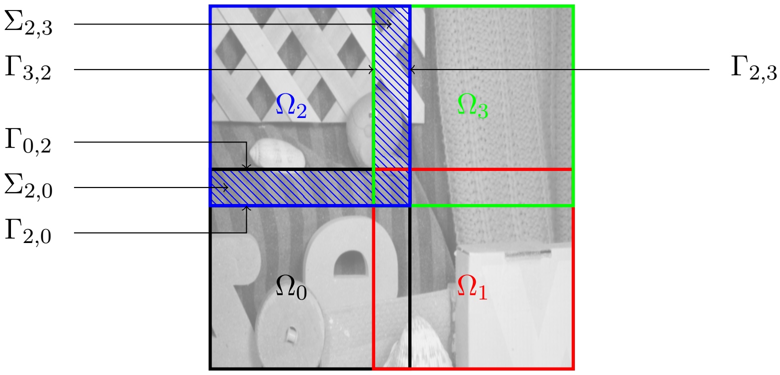

Figure 4.2. Example of notations for CPU $i = 2$.

By using the additive Schwarz method to find an estimation $\mathbf{U}_i$ of the optical flow in the part Ω_i, the iterative problem related to (2.8) is given at the iteration k by:

$$\begin{cases} -\mathrm{div}(\Lambda^k \nabla \mathbf{U}_i^k) + A_\rho \mathbf{U}_i^k = \mathbf{F} \text{ in } \Omega_i, \\ \dfrac{\partial \mathbf{U}_i^k}{\partial n} = 0 \text{ on } \partial\Omega_i \backslash \Gamma_{i,j},\ \forall j \in \mathcal{J}_i, \\ \mathbf{U}_i^k = \mathbf{U}_j^{k-1} \text{ on } \Gamma_{i,j},\ \forall j \in \mathcal{J}_i. \end{cases} \tag{4.1}$$

Thanks to this method, each sequence $(\mathbf{U}_i^k)$ converges to its corresponding solution $\mathbf{U}_{|\Omega_i}$. The rate of convergence increases with respect to the size of $\Sigma_{i,j}$. However, the size of the overlap should be smaller than the half size of a part otherwise all parts Ω_i which are not on edges would be fully covered by their neighbors. Moreover, the solution of the optical flow problem is sensitive to boundaries, so if we do too many splits, the size of each part will be too small to obtain satisfying results. For all these reasons, we don't choose a large overlap. We will see later that the best compromise in our case is an overlap of five pixels.

In order to combine the adaptive strategy of the regularization with the domain decomposition method, we consider each subdomain as a single image and we apply the adaptive algorithm on each part. It means that we compute the error indicator independently on each part of the split image and we perform a local choice of the regularization and the mesh adaptation on each subdomain. This choice implies that the unstructured submeshes don't match on the overlap so the corresponding value is found by FreeFem++ through an interpolation. At the end, to store and visualize the solution, we perform a last interpolation of the solution on the initial mesh to obtain a motion vector for each pixel of the image.

4.3. Multi-level parallel method

To solve the system with the finite element method we use the software *FreeFem++* [11]. The default solver of this software for linear systems is *UMFPACK* (Unsymmetric MultiFrontal method) which allows to use non-symmetric matrices. The drawback of this library is that it can only solve problems smaller than 4GB. It means that we can't deal with image subdomains larger than about 500 × 500 pixels per CPU. So we need to cut the whole image enough in the case of large images. This implies to have a large number of CPU which is not always the case depending on the machine we work with.

To overcome this issue, we use the *MUMPS* (MUltifrontal Massively Parallel sparse direct Solver) [1] library which allows the resolution of sparse linear problems in parallel. The advantage of this solver is that it is not limited by the size of the problem, so we are able to use high definition sequences. However, using the *MUMPS* library enforces to apply an initial LU decomposition of the mass matrix which can be time consuming (see figure 5.11). Combining the domain decomposition method with the use of the *MUMPS* library allows to implement a multi-level parallel method that can reduce the computation time of such high definition sequences.

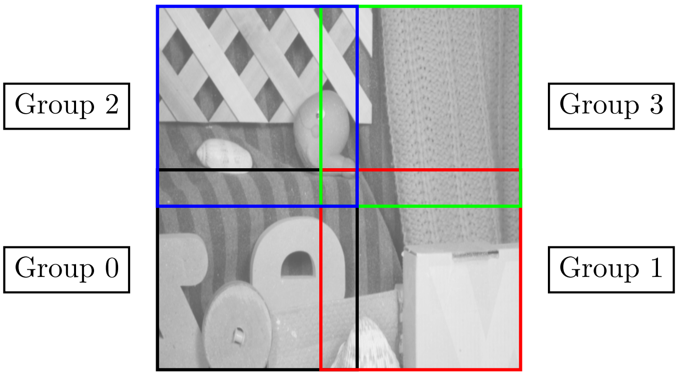

Figure 4.3. Decomposition of an image for four groups of CPU.

To implement this multi-level method, we split the domain Ω thanks to the same strategy used for the previous domain decomposition method. For each subdomain, we create a group of processors. As it is shown in the figure 4.3, each group of processors is assigned to a single subdomain. The processors are ordered as follow: the number of the group the CPU_i belongs to is given by:

$$\text{group}(\text{CPU}_i) = i$$

where the binary operator a by b and nbPart is the total number of subdomains of the split image. The rest of the implementation is the same that the domain decomposition method except that we now act on a group of processors instead of a single one. We perform the local choice of the regularization parameter and the mesh adaptation independently on each subdomain.

5. Numerical Results

We use the *RubberWhale* sequence (figure 5.1) given by the Middleburry website at `www.vision.middleburry.edu/flow/` in order to validate our algorithms. All the tests of this section are done using the method corresponding to the problem (2.8) which deals with illumination variations.

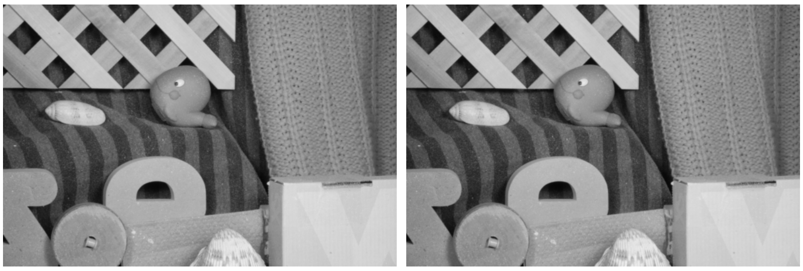

Figure 5.1. Frames 10 and 11 of the RubberWhale sequence.

Following their convention, we represent the estimated vector field thanks to a color map (figure 5.2) which assigns a color to each vector with respect to its orientation and its norm.

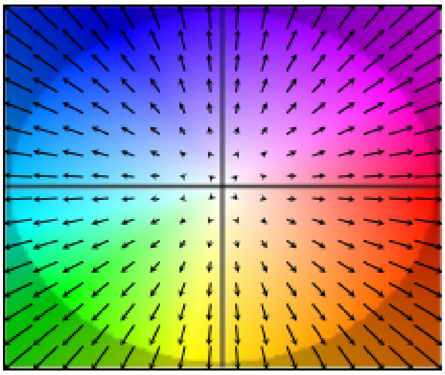

Figure 5.2. Vector field and its corresponding color map.

To validate our implementation, we compare our results with the exact solution also provided by Middelburry (figure 5.3).

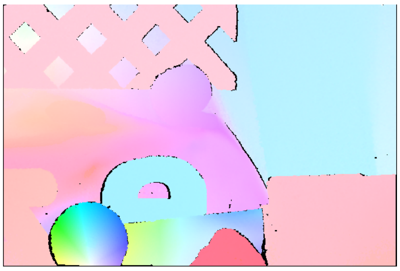

Figure 5.3. Ground truth solution of the RubberWhale sequence.

For optical flow problems, the accuracy of the method is usually evaluated by computing the *Average Angular Error* (AAE) given by:

$$AAE = acos \frac{u_{1,h}u_{1,e} + u_{2,h}u_{2,e} + 1}{\sqrt{(u_{1,h}^2 + u_{2,h}^2 + 1)(u_{1,e}^2 + u_{2,e}^2 + 1)}},$$

and the *Endpoint Error* (EE) given by:

$$EE = \sqrt{(u_{1,h} - u_{1,e})^2 + (u_{2,h} - u_{2,e})^2},$$

where $\mathbf{u}_h = (u_{1,h}, u_{2,h})$ is an approximation of the vector field and $\mathbf{u}_e = (u_{1,e}, u_{2,e})$ represents the exact optical flow. For the RubberWhale test case, using the resolution of the system (2.7) which deals with varying illumination, our algorithm reaches an average angular error equals to 20.89 and an endpoint error equals to 0.38. In the litterature, the best angular errors go from 1 to 15 and the endpoint errors from 0.07 to 0.39 for the equivalent evaluation test case called *Army* (see the evaluation table at `http://vision.middlebury.edu/flow/eval/results/results-a1.php`). There are two reasons to explain that our error is large compared to these values. First, we recall that *FreeFem++* [11] uses unstructured meshes and the computation of the angular error is not invariant with respect to the choice of the mesh. The other reason is that, even if we obtain a good approximation of the vector field direction (see the results presented below), due to the large value of the regularization in the non-egde areas, we under-estimate the vector norms. There exists some iterative strategy to improve this value but since we are principally interested in improving the computational time and in the adaptation of the regularization parameter, we don't use it. In addition, note that the results of Middleburry do not all use algorithms allowing varying illumination a priori.

We present in the following results the estimation obtained with our model solving the problem (2.8) with an adaptive control of the regularization parameter α. The initial value of α is set to 2000. For the parameter λ we choose a very large constant value, $\lambda = 10^6$. In a further work we will study a local adaptive method for this parameter to improve the solution when the brightness variation is not regular. The standard deviation parameters of the gaussian convolutions σ and ρ are both set to 2. All these parameters are found empirically.

To validate the domain decomposition, we need to verify the convergence of the Schwarz method which means that:

$$\mid \mathbf{U}^k - \mathbf{U}^{k+1} \mid \underset{k\to\infty}{\longrightarrow} 0.$$

We have split the image in four parts and tested the impact of the size of the overlap by using three different values. On the figure 5.4, we present the evolution of the error through ten iterations of the method for these three different overlaps.

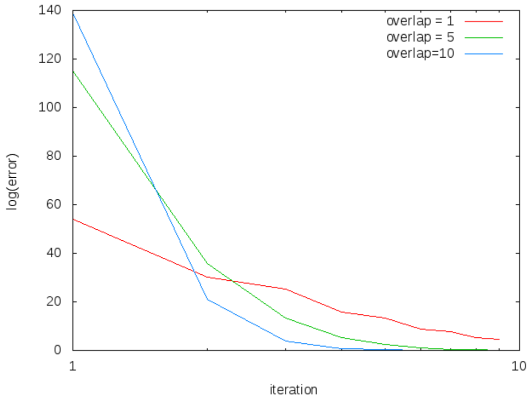

Figure 5.4. Convergence of the Schwarz method for three different overlaps.

We can see that the Schwarz method has a fast convergence and that the speed of convergence increases with the size of the overlap. However, if the overlap is large, the running time of the algorithm increases, see figure 5.5. Hence, we need to choose the size of the overlap considering an optimal ratio between the rate of convergence and the speed of computation time. Therefore, we choose an overlap of five pixels in the following tests.

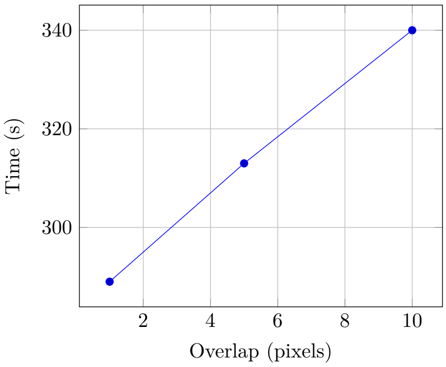

Figure 5.5. Computation time with respect to the overlap.

On the figure 5.6, we present the evolution of the estimation according to the iterations of the Schwarz method. We can see that since the third iteration, we already have a good estimation of the solution at the interfaces.

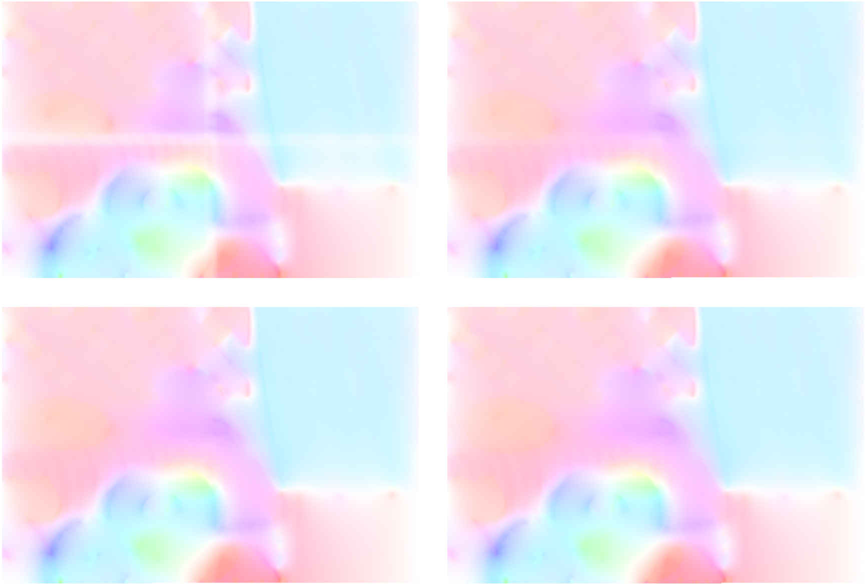

Figure 5.6. Results for a 2x2 separation, one image per iteration of Schwarz and with an overlap of five pixels.

We present in the figure 5.7 the final mesh after ten iterations of adaptation. Since the meshes of the subdomains don't match with each other, the communications of the values located at the intersections are made thanks to interpolations.

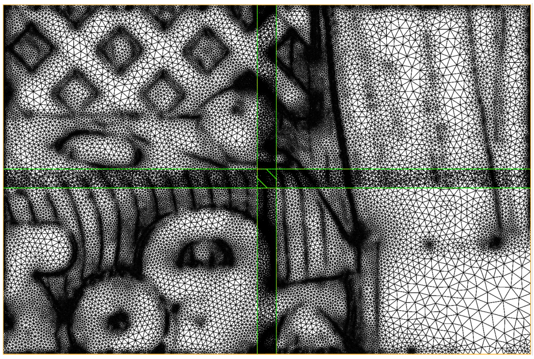

Figure 5.7. Final mesh of the *RubberWhale* test case after 10 iterations of adaptation.

In the figure 5.8, we give the evolution of the number of degrees of freedom. This number is strongly reduced during the first mesh adaptation. After that, it stays relatively constant. Hence, applying only a few iterations of mesh adaptation, in our case two, is enough.

In order to evaluate the implementation of the multi-level parallelism, we have launched the same test case for different splittings of the image (2, 4 and 8 parts) and different numbers of CPU Per Part (CPP). In all cases, we have done ten iterations of the Schwarz method and we have kept an overlap of five pixels. On the figure 5.9, we present the improvement of the global computation times.

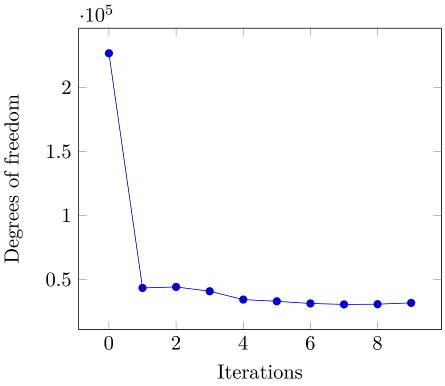

Figure 5.8. Degrees of freedom with respect to iterations of adaptation for the *RubberWhale* test case.

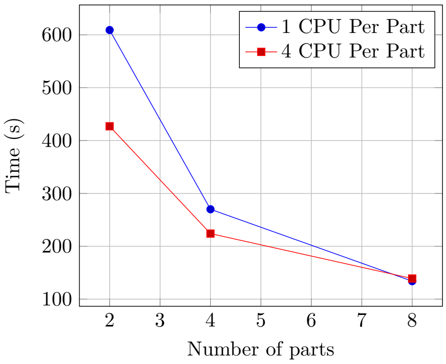

Figure 5.9. Computation times of different configurations of the multi-level parallelism.

To understand the low efficiency of the multi-level implementation in the case where we use 32 CPU and split the image in 8 subdomains, we first represent on the figure 5.10 the communication times obtained in the different cases.

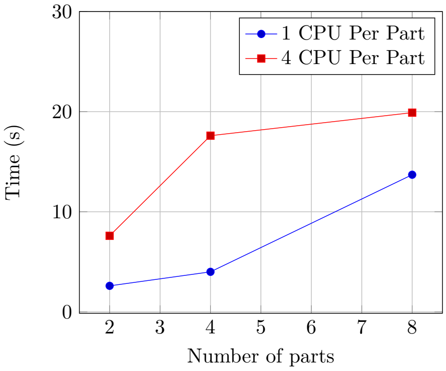

Figure 5.10. Communication times with respect to the number of CPU and the splitting (does not include the intern communications due to the *MUMPS* solver).

The increase of the communication times is not enough to explain the result obtained. Indeed, in every parallel implementations, the communication times increase with respect to the number of

CPU used but it is usually balanced with the time saved in the computations. So, in order to give more details, we present in the figure 5.11 the times of the two main parts of the computation: the construction of the mass matrix and the LU factorization with the resolution part. We can see that adding more CPU to a part slightly increases the time of construction of the mass matrix. However, it allows to decrease the time of the LU factorization. On our architecture, if we split the images into several parts, the main part of the computation is the mass matrix construction. If we use a large number of CPU per part, we reduce the difference between the time saved in the LU decomposition and the time lost in the matrix construction. More, we increase the communication times and the *MUMPS* parallel solver doesn't balance that.

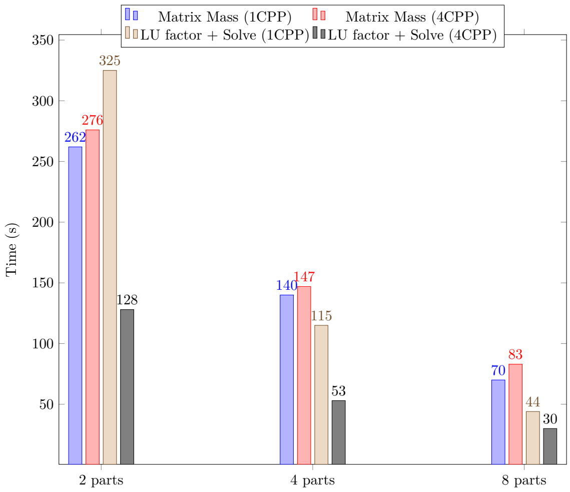

Figure 5.11. Detailed computations times.

This last result confirms that we must keep subdomains large enough to obtain a good efficiency with our multi-level implementation.

The next test consists of evaluating the efficiency of the adaptive regularization of the parameter α. The computation is done for the RubberWhale test case with a 2×2 domain decomposition. The local choice of α and the mesh adaptation are done in each subdomain independently. The adaptation steps are done with the Schwartz iterations. We can see on the figure 5.12 that we obtain a better definition of the edges without altering the solution on the homogeneous area.

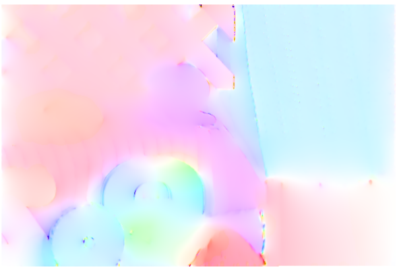

Figure 5.12. Estimated optical flow after ten iterations of adaptation (result for a 2x2 separation).

To evaluate the accuracy of the computed flow in the case of varying illumination, we first use the RubberWhale test case where we have modified the first frame by uniformly increasing the brightness intensity (see figure 5.13). For this test, we use the control of the regularization α but we keep the parameter λ constant. Since we haven't modified the motion between the two pictures, the exact solution is the same that in the previous test case.

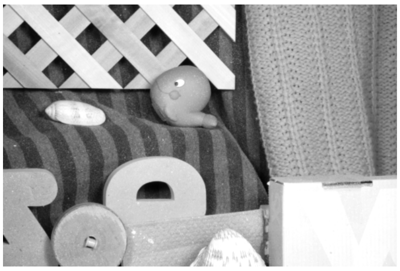

Figure 5.13. Modified first frame of the RubberWhale sequence for the varying illumination test case.

On the figure 5.14, we present the result obtained with and without the treatment of the brightness variation. On the right hand side, we can see that the solution without treatment is not well estimated. On the other hand, the model which uses the modified assumption (2.6) allowing the brightness variation gives a much more accurate optical flow.

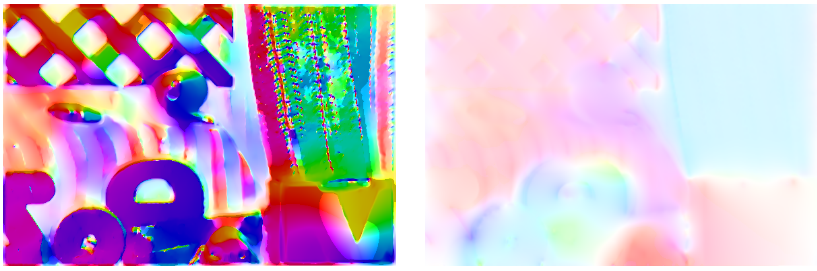

Figure 5.14. Left: Estimation with (right) and without (left) treatment of the varying illumination. The images were split in four parts.

On the figure 5.15, we can see that the adaptive regularization is still working for the test with non-constant brightness. In this case, the edges are even better defined.

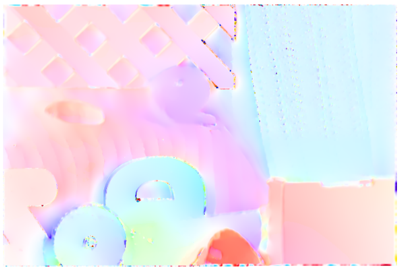

Figure 5.15. Estimated optical flow after ten iterations of adaptation for the test with varying illumination (results for a 2x2 separation).

To test the impact of the adaptation of the parameter λ, we now use an other modification of the RubberWhale sequence where variations of the brightness intensity are only localised on few spots in the second frame, see figure 5.16.

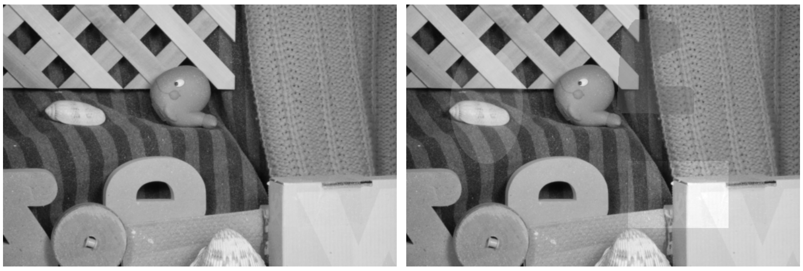

Figure 5.16. RubberWhale sequence with non regular brightness variation.

As we show in the figure 5.17, using a constant parameter λ is not optimal if the brightness variation is not uniform. If λ is chosen too big it leads to a non accurate optical flow and on the other hand, a smaller value of λ leads to a vector field strongly regularized.

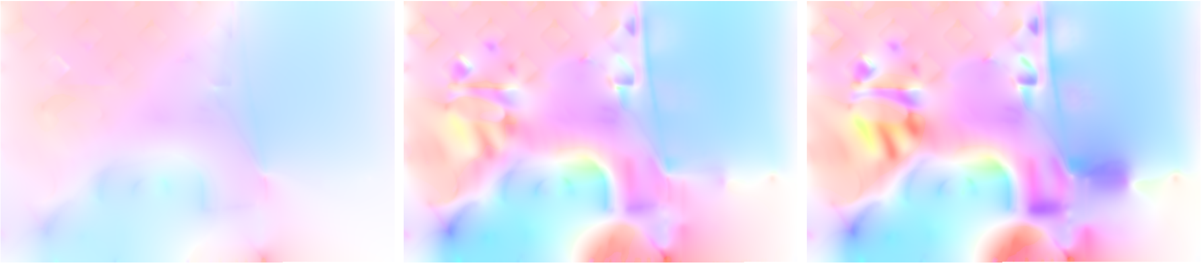

Figure 5.17. Optical flow estimation for the test case figure 5.16 with different values of the constant parameter λ. From left to right: $\lambda = 10^4$, $\lambda = 10^6$, $\lambda = 10^7$.

On the figure 5.18, we present the solution given using both adaptive choices of the parameters α and λ. It finally gives a good estimation of the optical flow in the whole image and still preserves defined edges on discontinuities.

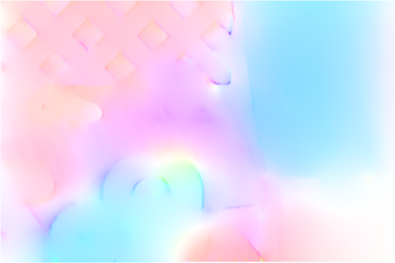

Figure 5.18. Optical flow estimation for the sequence of the figure 5.16 with an adaptive control of α and λ.

Finally, on the figures 5.19 and 5.20, we have used two sequences of real images provided by the *Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement* (Cerema). These pictures present two main interests. First, the high resolution. We have 2028×1098 pixels for the highway sequence and 1524×1092 pixels for the wall sequence which corresponds to a vector field of about two billions of pixels to determine. The second interest is that the brightness and the texture are natural.

The highway sequence presents a non-constant brightness with a more complex texture and a lot of occlusions as the white lines on the road or the large motion of the car. We can see on the figure 5.19, the difference between the model with and without the varying illumination. On the left hand side, we have the solution without treatment. Again, we see that this is not correctly estimated because of the variation of the natural lighting. On the right, despite the occlusion areas which could be improved with a large displacement algorithm, we obtain a better approximation.

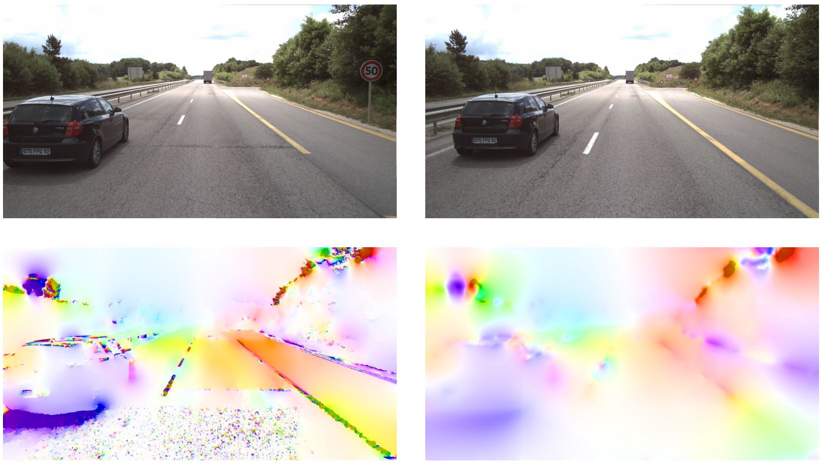

Figure 5.19. Up: Test case of the Highway sequence given by the Cerema. Bottom: Optical flow estimation obtained with (right) and without (left) treatment of the varying illumination. The images were divided in sixteen parts.

The figure 5.20 represents a wall of a tunnel. On this sequence the motion to estimate is linear. The challenge is to deal with irregular textures. This figure shows that our method is still efficient in this case. Indeed, the solution is quite smooth and linear except for the white square (bottom right) which may be improved by using a large displacement algorithm. This approach will be implemented in a further work.

In the figure 5.21, we give the computation times obtained with the two high definition tests. We recall that the pictures have about two billions of pixels and were not reduced. As in the previous cases, we perform ten iterations of the Schwarz algorithm.

6. Conclusion

In this article we have used the finite element method to solve the optical flow problem with varying illumination. Thanks to the additive Schwarz method, we have implemented the domain decomposition in order to parallelize the computations. We have shown that this method is an efficient way to decrease the computation time and to handle high resolution sequences.

We have also used a second level of parallelism in order to reduce the execution time even more and we have shown that such a massively parallel approach yields to an important gain of time. There exists other methods without overlapping which may be theoretically faster [13]. In a forthcoming work, we will consider this class of methods which requires a different managment of the boundary conditions (see [13]). However, as we have seen, an overlap of 5 pixels is enough so for large images the additional time can be neglected.

The finite element method has also permitted to use an adaptive strategy of the regularization parameter, hence to efficiently combine the quality of the optical flow estimation with a method that

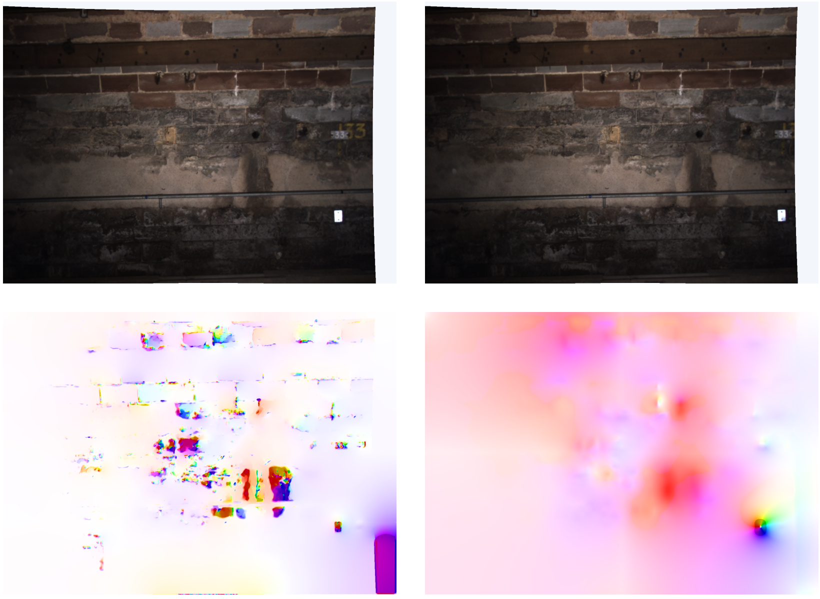

Figure 5.20. Up: Test case of the Wall sequence given by the Cerema. Bottom: Optical flow estimation obtained with (right) and without (left) treatment of the varying illumination. The images were divided in sixteen parts.

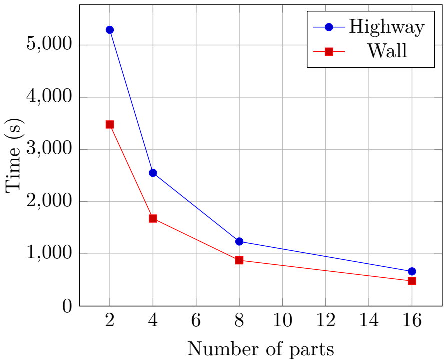

Figure 5.21. Computation times of the high definition tests for different splittings.

preserves the edges and the fine features of the computed flow. Finally, we have proposed a local choice of the parameter λ in order to obtain a good estimation of the optical flow even if the brightness variation isn't regular. This point will be the object of a forthcoming work.

Acknowledgements

The authors would like to thank the *Centre d'études et d'expertise sur les risques, l'environnement, la mobilité et l'aménagement (Direction territoriale Est, laboratoire de Strasbourg)* for their suggestions and their data. The authors also thank the anonymous reviewers for their valuable comments.

References

[1] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, and J. Koster. *Applied Parallel Computing. New Paradigms for HPC in Industry and Academia: 5th International Workshop, PARA 2000 Bergen, Norway, June 18–20, 2000 Proceedings*, chapter MUMPS: A General Purpose Distributed Memory Sparse Solver. Springer Berlin Heidelberg, 2001.

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

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

[4] Z. Belhachmi and D. Gilliocq-Hirtz. Coupling parareal and adaptive control in optical flow estimation with application in movie's restoration. In *Computer Vision and Image Analysis Applications (ICCVIA), 2015 International Conference on*, pages 1–6, Jan 2015.

[5] 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.

[6] Z. Belhachmi and F. Hecht. An adaptive approach for segmentation and TV denoising in the optic flow estimation. Working paper or preprint, 2014.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, pages 25–36. Springer Berlin Heidelberg, 2004.

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

[9] 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.

[10] M.A. Gennert and S. Negahdaripour. Relaxing the brightness constancy assumption in computing optical flow. *Technical Report, Massachusetts Institute of Technology Cambridge, MA, USA*, 1987.

[11] F. Hecht. New development in FreeFem++. *J. Numer. Math.*, 20:251–265, 2012.

[12] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185 − 203, 1981.

[13] P.-L. Lions. On the Schwarz alterning method. III: A variant for nonoverlapping subdomains. In *Third internationnal symposium on domain decomposition methods for partial differential equations*, volume 6, pages 202–223. SIAM Philadelphia, PA, 1990.

[14] 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*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[15] E. Mémin and P. Pérez. A multigrid approach to hierarchical motion estimation. In *Proc. Int. Conf. on Computer Vision, ICCV'98*, pages 933–938, Bombay, India, January 1998.

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

[17] P. Ruhnau, T. Kohlberger, C. Schnorr, and H. Nobach. Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids*, 38:21–32, 2005.

[18] J. Weickert, A. Bruhn, N. Papenberg, and T. Brox. Variational optic flow computation: From continuous models to algorithms. In *International Workshop on Computer Vision and Image Analysis (ed. L. Alvarez), IWCVIA-03, Las Palmas de Gran Canaria*, 2003.

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

[20] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel. Complementary optic flow. In D. Cremers, Y. Boykov, A. Blake, and Schmidt F., editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 207–220. Springer, 2009.