



HAL
open science

Coupling parareal and adaptive control in optical flow estimation with application in movie's restoration

Zakaria Belhachmi, Diane Gilliocq-Hirtz

► **To cite this version:**

Zakaria Belhachmi, Diane Gilliocq-Hirtz. Coupling parareal and adaptive control in optical flow estimation with application in movie's restoration. 2015. <hal-01112935>

HAL Id: hal-01112935

<https://hal.science/hal-01112935v1>

Preprint submitted on 4 Feb 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Coupling parareal and adaptive control in optical flow estimation with application in movie's restoration

Zakaria Belhachmi
Université de Haute-Alsace,
LMIA, 6 rue des Frères Lumière
68100 Mulhouse, France
zakaria.belhachmi@uha.fr

Diane Gilliocq-Hirtz
Université de Haute-Alsace,
LMIA, 6 rue des Frères Lumière
68100 Mulhouse, France
diane.gilliocq-hirtz@uha.fr

Abstract—In this paper we use a variational method in order to estimate the optical flow. We use a combined Local-Global strategy coupled with a local choice of the regularization term. In order to improve the computation time we have implemented a parallel in time algorithm. As application of the optical flow we present an example of image reconstruction.

Keywords—Optical flow, variational method, regularization control, parareal, film restoration, complex Ginzburg-Landau equation.

I. INTRODUCTION

In this paper we present a new adaptive strategy for the determination of optical flow. We use the Combined Local-Global (CLG) strategy first presented by Andrés Bruhn, Joachim Weickert and Christoph Schnörr in 2005 [4]. It is a combination of Lucas and Kanade's method [7] which figures out the aperture problem by assuming a local constancy hypothesis and the global method of Horn and Schunck [5]. Our work follows Z. Belhachmi and F. Hecht's article [3] on the control of the regularization parameter. While most of methods use a priori estimation in order to choose the regularization parameter their method consists in a posteriori estimation of a parameter locally chosen. This becomes a piecewise continuous function. Concerning the algorithm in order to improve computation time we have used a parallel in time method first presented by J.-L. Lions, Y. Maday, and G. Turinici [6] in 2001. It is called "parareal" method. Contrary to well-known decomposition domain methods this one allows separating the work with respect to time steps. The principle is to first operate a fast coarse resolution and then use obtained solutions as initial values of each process. Finally we will make an application of the estimation of the optical flow following the idea described by Harald Grossauer in [9]. It is an example of film restoration using the combination of the optical flow and image inpainting. In our case we choose the complex Ginzburg-Landau equation to do the restoration. In the first part of this article we remind the optical flow problem and the combined Local-Global strategy. Then we recall Belhachmi and Hecht's adaptive control to expose in a second part the parareal algorithm to improve the computation time. In the last part we give an application of movies restoration using the optical flow and image inpainting.

II. OPTICAL FLOW PROBLEM

A. Variational optical flow estimation

We consider a sequence of successive images where $\Omega \subset \mathbb{R}^2$ represents the image domain. The intensity of a pixel at moment t is defined by the functional

$$I: \Omega \times [0, T] \rightarrow \mathbb{R} \\ (x, y), t \mapsto I(x, y, t) \quad (1)$$

Thanks to John Barron, David Fleet and Steven Beauchemin method [2] we use a convolution with a Gaussian K_σ of standard deviation σ to work with smoothed images. Thus we define the smoothed image sequence

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

The optical flow $u = (u_1, u_2)$ is the displacement of a pixel between two images. We will present numerical results obtained with two frames of the RubberWhale sequence (Fig. 1). This sequence is provided by the site of Middlebury www.vision.middlebury.edu/flow/. We give the ground truth solution in Fig. 2 and the color map in Fig. 3. The color map helps us to visualize the optical flow by assigning a color to each vector based on its orientation. The intensity of the color determines the vector's norm.



Figure 1. Frames 10 and 11 of the RubberWhale sequence.



Figure 2. Ground truth solution of the RubberWhale sequence.

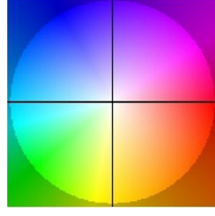


Figure 3. Color map used to visualize the optical flow.

Following the brightness constancy assumption brightness intensity stays the same between two successive frames

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

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

$$f_x u_1 + f_y u_2 + f_t = 0 \quad (2)$$

with the notation $f_r = \frac{\partial f}{\partial r}$.

In this way we have to determine two unknowns u_1 and u_2 with only one equation. This problem is called "aperture problem". To go through this ill posed-ness Lucas and Kanade [7] assume that every pixel of a neighborhood of size ρ moves similarly. Thus, we determine optical flow u by minimizing the functional

$$K_\rho \star ((f_x u_1 + f_y u_2 + f_t)^2)$$

where K_ρ is a Gaussian deviation of parameter ρ .

At the same time as Lucas and Kanade, Horn and Schunck proposed [5] another method to overcome the aperture problem. Contrary to the local assumption they proposed a global approach. They introduced a regularization part and consider u as a minimizer of the global energy

$$\int_{\Omega} \left((f_x u_1 + f_y u_2 + f_t)^2 + \alpha (|\nabla u_1|^2 + |\nabla u_2|^2) \right) dx dy$$

where α is the constant regularization parameter which acts as a penalizer and leads to smoother flow field (bigger is α smoother is the flow). The idea of the Combined Local-Global strategy is, as it is suggested, to combine both methods and minimize the functional

$$\int_{\Omega} \left(K_\rho \star (f_x u_1 + f_y u_2 + f_t)^2 + \alpha (|\nabla u_1|^2 + |\nabla u_2|^2) \right) dx dy.$$

According to Euler-Lagrange equations we have the system

$$\begin{cases} K_\rho \star (f_x)^2 u_1 + K_\rho \star (f_x f_y) u_2 + K_\rho \star (f_x f_t) - \alpha \Delta u_1 = 0 \\ K_\rho \star (f_y f_x) u_1 + K_\rho \star (f_y)^2 u_2 + K_\rho \star (f_y f_t) - \alpha \Delta u_2 = 0 \end{cases} \quad (3)$$

that we finally write as

$$\begin{cases} \alpha \Delta u + J_\rho u = f \text{ in } \Omega \\ \frac{du}{dn} = 0 \text{ on } \partial\Omega \end{cases} \quad (4)$$

with

$$J_\rho = \begin{pmatrix} K_\rho \star (f_x)^2 & K_\rho \star (f_x f_y) \\ K_\rho \star (f_y f_x) & K_\rho \star (f_y)^2 \end{pmatrix} \text{ and } f = \begin{pmatrix} -K_\rho \star (f_x f_t) \\ -K_\rho \star (f_y f_t) \end{pmatrix}.$$

B. Adaptive regularization

In this part we are interested to control the parameter α . The regularization parameter is now a function. This local choice of α is based on a posteriori strategy analysis. The idea was proposed by Z. Belhachmi and F. Hecht [3]. The general method for the choice of the regularization parameter consists on seeking an optimized value of α before the discretization. The choice of the regularization is an important part of the resolution because a small value is necessary to approximate correctly the Neumann boundary condition on edges of objects but it increases the maximum value of the optical flow. So in order to have a better estimation we prefer a larger 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.

We work in the space H^1 which is a Hilbert space for the scalar product

$$(u, v)_\rho = \int_{\Omega} \nabla u \nabla v \, dx dy + \int_{\Omega} J_\rho u \cdot v \, dx dy$$

and the norm

$$\|u\|_{H^1(\Omega)} = (u, u)_\rho^{1/2}.$$

Another choice of Hilbert space is given in [3]. $(\Omega_\ell)_{1 \leq \ell < L}$ is a subdivision of Ω . We define $\alpha(x, y)$ a piecewise function such that $\alpha(x, y) = \alpha_\ell$ on Ω_ℓ .

The weak formulation of the problem above with the adapted α is: for all $v \in H_\rho^1(\Omega)$, find $u \in H_\rho^1(\Omega)$, such that

$$\int_{\Omega} \alpha(x, y) \nabla u \cdot \nabla v \, dx dy + \int_{\Omega} v^T J_\rho u \, dx dy = \int_{\Omega} f \cdot v \, dx dy.$$

By posing

$$\begin{aligned} a_\alpha(u, v) &= \int_{\Omega} \alpha(x, y) \nabla u \cdot \nabla v \, dx dy + \int_{\Omega} v^T J_\rho u \, dx dy \\ f(v) &= \int_{\Omega} f \cdot v \, dx dy \end{aligned}$$

this problem finally becomes: find $u \in H_\rho^1(\Omega)$ such that

$$a_\alpha(u, v) = f(v). \quad (5)$$

The existence of a weak solution of (5) is due to the Lax-Milgram Lemma and is proved in [3]. We recall the proposition (a complete proof is given in [3])

Proposition 1

Let $u \in H_\rho^1(\Omega)$ be a solution of $J_\rho u = f$. For $\alpha > 0$ we have the two inequalities

$$\|u_\alpha\|_{\rho, \alpha} \leq c \left\| \int_{\rho}^{\frac{1}{\rho}} u \right\|_{L^2(\Omega)} \quad (6)$$

and

$$\|u - u_\alpha\|_{\rho,\alpha} \leq c \left(\frac{\alpha_M}{\alpha_m}\right)^{\frac{1}{2}} \left\| \alpha^{\frac{1}{2}} \nabla u_\alpha \right\|_{L^2(\Omega)} \quad (7)$$

with c a constant independent of α and α_m and α_M are respectively the minimal and the maximal value of $(\alpha_\ell)_{1 \leq \ell < L}$.

Since we want to locally choose the regularization parameter we will have a large ratio $\frac{\alpha_M}{\alpha_m}$. So we will use the inequality (7) in the error indicator.

Let $(Th)_h$ be a regular subdivision of Ω . Each element $K \in Th$ is a triangle with maximal size h . We define the space of approximations

$$V_h = \{v_h \in C(\bar{\Omega}), v_h|_K \in P_1(K)^2\}$$

where $J_{\rho,h}$ is an approximation of J_ρ and $a_{\alpha,h}$ of the bilinear term a_α where J_ρ is replaced by $J_{\rho,h}$. The discrete problem also reads: for all $v_h \in V_h$ find $u_{\alpha,h}$ such that

$$a_{\alpha,h}(u_{\alpha,h}, v_h) = f(v_h).$$

The control of the regularization is done through an error indicator which is given for each element $K \in Th$ by

$$\eta_K = \alpha_K^{-\frac{1}{2}} h_K \|f_h + \alpha_K \Delta u_{\alpha,h} + J_{\rho,h} u_{\alpha,h}\|_{L^2(K)^2} + \frac{1}{2} \sum_{e \in \varepsilon_K} \alpha_e^{-\frac{1}{2}} h_e^{\frac{1}{2}} \left\| [\alpha \nabla u_{\alpha,h} n_e]_e \right\|_{L^2(e)^2}$$

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 . n_e represents the normal vector from e , α_e is the maximum between the α of the two neighbors of an edge and $[r]_e$ represents the jump over the edge e that means the difference between the outside and the inside value.

The error indicator η_K shows the finite element error and the model error. On discontinuities this value is large so to improve the solution we decrease α following this formula

$$\alpha_K^{n+1} = \max \left(\frac{\alpha_K^n}{1 + \kappa \max \left(\frac{\eta_K}{\|\eta_K\|_\infty} - 0.1, 0 \right)}, \alpha_s \right)$$

where κ is an arbitrary control parameter and α_s is a sill. In this way, if the relative error is greater than 10% we reduce the value of α . In an other hand, if it is less than 10% the denominator is equal to one and so α stays the same.

To sum up we present different steps of the method.

- Compute a first approximation u_α^0 of the optical flow. This estimation is done on a Cartesian grid Th^0 where we have one cell per pixel. Define $i = 0$.
- $i = i + 1$. Build an adapted mesh Th^i with the metric error indicator.
- Local choice of $\alpha^i(x, y)$ on Th^i .
- Go to step 2.

C. Numerical Results

On the Fig. 4 we present solutions obtained with a constant regularization $\alpha = 3000$ and a locally adapted parameter with $\alpha^0 = 3000$.



Figure 4. Left: Solution with constant $\alpha = 3000$. Right: Solution after 20 iterations of adaptation and $\alpha^0 = 3000$.

As we can see on the Fig. 5 a small regularization implies a good estimation of the edges but there are too many details on the texture. That's why we have initially chosen a large α^0 . In this way the vector field is smoothed enough and after the adaptation on edges we can improve the estimation.



Figure 5. Solution with constant $\alpha = 100$.

To show the evolution of the estimation's accuracy with the adaptation of α we present in Fig. 6 the evolution of the Average Angular Error (AAE). We recall the formula of the angular error

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

where $(u_{1,h}, u_{2,h})$ is an approximation of the vector field and $(u_{1,e}, u_{2,e})$ represents the exact flow field.

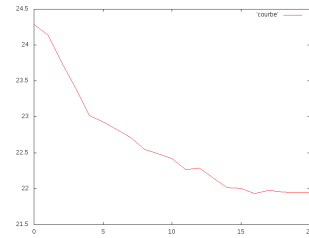


Figure 6. Evolution of AAE with respect to iterations of adaptation.

Finally we show in Fig. 7 the adapted mesh obtained after 20 steps of adaptation and the evolution of the number of degrees of freedom. The initial mesh is fine enough and we don't need to refine it. However, regions where the parameter α is large are sufficiently smoothed by the regularization to allow a larger

mesh. Moreover with the adaptation step we decrease the number of degrees of freedom. As we can see in Fig. 7.b. this number decreases significantly until the second iteration and stays relatively low. This implies a gain of computation time.

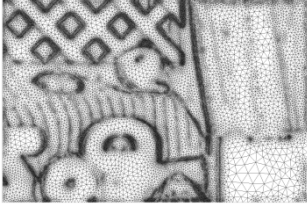


Figure 7.a. Adapted mesh after 20 iterations of adaptation.

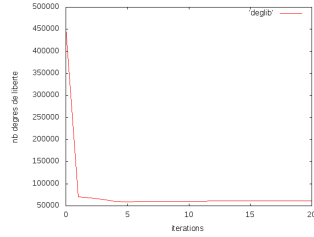


Figure 7.b. Evolution of the number of degrees of freedom.

III. PARAREAL METHOD FOR OPTICAL FLOW

A. Presentation of the algorithm

As we could have seen in the above section the adaptive strategy to control parameter α is a good method to have a fine characterization of the geometry which gives a better estimation of the flow field and to strongly reduce the number of degrees of freedom. Now we want to improve the computation time by parallelizing our program. However, classical methods of parallelization are not well adapted to our problem. As we are dependent of the mesh geometry we can't use usual domain decomposition methods and the parallelization on graphic cards is not adapted to the mesh adaptation. For these reasons we choose a parallel in time algorithm.

The word "parareal" was first stated by J.-L. Lions, Y. Maday and G. Turinici [6] to represent a parallel in time method. The aim is to couple coarse and fine resolutions to construct a predictor-corrector process [1] where the fine resolution is made in parallel. In order to use the parareal method we add a variation in time of the flow field. The problem to solve becomes

$$\begin{cases} \frac{\partial u}{\partial t} - \alpha \Delta u + J_\rho u = f \text{ in } \Omega, t \in [0, T] \\ u(0) = 0 \end{cases}$$

Let $0 = T_0 < T_1 < \dots < T_N = T$ be a partition of $[0, T]$. The idea is to first solve the problem with a coarse method on $[0, T]$ with a large time step ΔT . Then we solve the problem with a finer method of time step Δt (where $\Delta t < \Delta T$) on each interval $[T_n, T_{n+1}]$. For each fine resolution we take the initial condition $u_0 = u(T_n)$ which is the correspondent coarse solution. In this way we can do many fine problems in parallel since we have coarse solutions.

We set \mathcal{G} and \mathcal{F} a coarse and a fine solver of the optical flow problem where \mathcal{F} is at least as accurate as \mathcal{G} . In our case we took an Euler implicit method. We consider $\Delta T = T/N$ and $\Delta t = \Delta T/M$ corresponding time steps where N and M are respectively the total number of coarse time steps and the total number of fine time steps.

We first compute a coarse approximation of the flow field u on $[0, T]$

$$\begin{cases} U_{n+1} = \mathcal{G}(U_n), n = 0, \dots, N \\ U_0 = U(T_0). \end{cases}$$

At this point we start the global iterative part of the algorithm. We begin by the sequence $(U_n)_n$ of these coarse approximations and we compute a fine approximation on $[T_n, T_{n+1}]$ such that

$$\begin{cases} u_{n,m+1} = \mathcal{F}(u_{n,m}), m = 0, \dots, M \\ u_{n,0} = U_n. \end{cases}$$

The first remark is that the fine solution allows to correct the first coarse prediction in order to get a better approximation of $u(T_n)$. Indeed the initial value used for each coarse resolution can be updated with respect to the fine approximation

$$\tilde{U}_{n+1} = u_{n,M}.$$

After that we do a new coarse resolution which begins at $n = k + 1$ where k is the number of the current global iteration

$$\tilde{U}_{n+1} = \mathcal{G}(\tilde{U}_n).$$

Finally we can update the sequence of coarse approximations with the prediction-correction formula

$$(U_{new})_{n+1} = \tilde{U}_{n+1} - U_{n+1} + u_{n,M}.$$

We end by updated U to then start the next iteration

$$U = U_{new}.$$

Obviously the maximum number of global iterations is $k = N$. However, doing N iterations would correspond to the same result than the sequential algorithm. In fact it would be worse because of the MPI communications and update parts. Hence we check after each global iteration if two successive global results are significantly different and if they don't we stop our program.

In order to illustrate the program we present the pseudo code of the parareal algorithm.

1. Initialisations :
 - $U_{old} = 0$
 - Coarse resolution:
 - For $n = 0$ to $N - 1$
 - $U = \mathcal{G}(U_{old})$
 - $U_{old} = U$
 - $U_{end}[n] = U$
 - $u_{start}[n + 1] = U$
 - End For
2. Global iterative loop for parareal resolution:
 - $k = 0$
 - While error > tolerance and $k < N$
 - a. Parallel resolution (same for each CPU i):
 - $u_{old} = u_{start}[i]$
 - For $m = 0$ to $M - 1$
 - $u = \mathcal{F}(u_{old})$
 - $u_{old} = u$
 - End For

```

b. MPI communications: sharing data
   Each CPU sends his  $u_{old}$  in  $u_{temp}[i]$ 
c. Compute error
    $error = \|u_{temp}[mpisize] - u_{arret}\|_{L^2}$ 
d. Prediction-correction:
    $u_{start}[k + 1] = u_{temp}[k]$ 
   For  $I = k + 1$  to  $N - 1$ 
      $U_{old} = u_{start}[I]$ 
      $U = \mathcal{G}(U_{old})$ 
      $U_{old} = U$ 
      $\tilde{U} = U_{end}[I]$ 
      $U_{end}[I] = U_{old}$ 
      $u_{start}[I + 1] = U + u_{temp}[I] - \tilde{U}$ 
   End For
    $k++$ 
End While

```

The best way to show that the method works is to compare solution obtained with the original sequential algorithm. During the first global iteration the first Computational Process Unit (CPU) has the same start point than the sequential version and executes M small steps. So the M first steps are the same during the first iteration. Before the second global iteration, the start point of the second CPU is updated with the result obtained with the CPU 0. In this way for the second global iteration, results of the two first CPU are the same than results of sequential code and so on. In the Fig. 8 we show the error in L^2 norm between the two algorithms for each global iteration. The prediction-correction step improves the accuracy of all next CPUs.

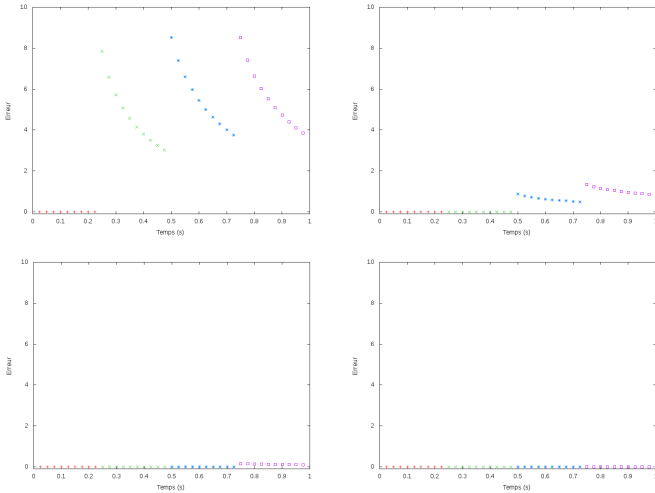


Figure 8. Evolution of the error in L^2 norm between the parallel and the sequential algorithms. From left to right, from up to down, we have the iteration 0 to 3. In red we have the CPU 0, in green the CPU 1, in blue the CPU 2 and in purple the CPU 3.

B. Numerical Results

Obviously for such a code the optical flow is near to the result obtained in sequential without adaptation. So in order to improve the accuracy of the result we apply an adaptation of the parameter α during the first coarse resolution. On the Fig. 9 we can see the difference between the two launches. The first

one is without adaptation and the second is with adaptation. The AAE is improved from 26.7 to 26.3 and we can observe on the picture that edges are more accurate.



Figure 9. Left: Final result for the parareal code without adaptation, AAE=26.7. Right: Final result for the parareal code with adaptation, AAE=26.3.

In the following table we give computation times with respect to the number of CPU used.

Table 1. Speedups $\left(\frac{\text{sequential time}}{\text{parallel time}}\right)$ in function of the number of CPU.

Number of CPU	Speedup
1	1
8	1.2
16	1.3

As we can see, the algorithm is non-scalable which means that we don't gain as many time of execution as the number of CPU used. There are two principal reasons for that. Firstly the multiple correction steps generate a lot of communications between processors. Then, this method is efficient when the correction step has acted several times for all CPUs. Indeed, we have seen above that estimations of all CPU are improved at each prediction-correction step. So if we have a big number of CPU and we do a big number of prediction-correction steps then the estimation is well done. However, as prediction-correction steps are expansive we need to have a big number of fine iterations for each CPU. This means that this method is more efficient for long problems in time with a very big number of iterations and a big number of CPU.

IV. MOVIE'S RESTORATION

A. Presentation of the method

We present now the application coupling optical flow and inpainting image first proposed by Harald Grossauer in [9]. In order to reconstruct a damaged image of a movie we need to identify which part of this image needs to be treated. We call it the "mask of damage". To identify this mask we consider three consecutives frames f_{n-1} , f_n and f_{n+1} . We compute the optical flow h_b from f_{n-1} to f_n and the optical flow h_f from f_{n+1} to f_n . So we have these following relations

$$f_{n-1}(x + h_b) \approx f_n(x) \approx f_{n+1}(x + h_f).$$

We consider that a pixel belongs to a map if we have

$$|f_{n-1}(x + h_b) - f_n(x)| > \lambda$$

and

$$|f_{n+1}(x - h_f) - f_n(x)| > \lambda.$$

for a chosen λ .

Once we have found the domain Σ where the image needs to be restored we solve the complex Ginzburg-Landau equation

$$\frac{\partial u}{\partial t} - \alpha \Delta u + \frac{1}{\varepsilon^2} (|u|^2 - 1)u + (u - f) = 0$$

on Σ to do image inpainting on this region. The complex function f is given by $f = f_0 + i\sqrt{1 - f_0^2}$ where f_0 represents the initial damaged picture.

The weak formulation of this problem states: find $u \in H^1(\Sigma)$ such that for all $v \in H^1(\Sigma)$ we have

$$\int_{\Sigma} \frac{du}{dt} v - \alpha \nabla u \cdot \nabla v + \frac{1}{\varepsilon^2} (|u|^2 - 1)u \cdot v + (u - f) \cdot v = 0.$$

B. Numerical Results

To present our result we use the sequence frame10-frame11-frame10 of the Venus test case of the site of *Middlebury* where we have manually degraded the frame 11 (Fig. 10).



Figure 10. Frame 11 of the Venus sequence manually altered.

In this particular case we know the exact flow field linking the frame 10 and frame 11. So we first have tested the reconstruction in the most ideal case with the exact optical flow and then with our computed optical flow. In Fig. 12 we show masks of damage found in both cases and in Fig. 13 we present results after the complex Ginzburg-Landau resolution [8].

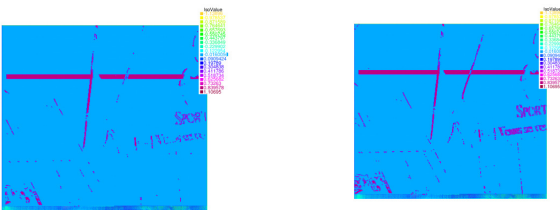


Figure 12. Mask of damage obtained using the exact optical flow (left) and our computed optical flow (right).

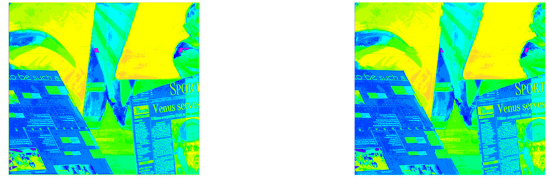


Figure 13. Reconstructed image using the exact optical flow (left) and our computed optical flow (right).

We can see that even with the evaluated optical flow, the mask of damage is well found and the reconstruction of the image is not so far of the reconstruction found by using the exact optical flow.

V. CONCLUSION

In this paper we have implemented a variational method to estimate the optical flow field. We have added an adaptive control of the regularization parameter of Horn and Schunck which is a way to have a good characterization of the geometry. This control is combined with a mesh adaptation which reduces significantly the number of degrees of freedom. Then, in order to improve the computation time we have implemented the parareal algorithm. We have seen that the method is efficient but it could be better on a longer problem in time and with more processors. Finally we have coupled the optical flow estimation with the complex Ginzburg-Landau equation in order to restore an altered movie. The computed optical flow has been very efficient to find the damaged area on the picture.

REFERENCES

- [1] Guillaume Bal and Yvon Maday. A parareal time discretization for nonlinear pde's with application to the pricing of American put. In Luca Pavarino and Andrea Toselli editors. Recent Development in Domain Decomposition Methods: Lecture Note in Computational Sciences and Engineering, 23:180,202, 2002.
- [2] John Barron, David Fleet, and Steven Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
- [3] Zakaria Belhachmi and Frédéric Hecht. Control of the effect of regularization on variational optic flow computation. *Journal of Mathematical Imaging and Vision*, 40:1–19, 2010.
- [4] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.
- [5] Berthold Horn and Brian Schunck. Determining optical flow. *Artificial Intelligence*, 17:185,203, 1981.
- [6] Jacques-Louis Lions, Yvon Maday, and Gabriel Turinici. Résolution d'edp par un schema en temps "pararéel". *Comptes Rendus de l'Académie des Sciences, Série I mathématiques*, 332(7) :661,668,2001.
- [7] Bruce Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *Seventh International Joint Conference on Artificial Intelligence*, pages 674,679, 1981
- [8] Zakaria Belhachmi, Moez Kallel, Maher Moakher and Anis Theljani. Weighted Harmonic and Ginzburg-Landau equations in image inpainting. *Proceeding*.
- [9] Harald Grossauer. Inpainting of movies using optical flow. *Book section: Mathematical Models for Registration and Applications to Medical Imaging*. Springer Berlin Heidelberg, pages 151,162.