Published in Image Processing On Line on 2019–12–23.
Submitted on 2014–06–05, accepted on 2017–03–17.
ISSN 2105–1232 © 2019 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2019.118

Joint TV-L1 Optical Flow and Occlusion Estimation

J.F. Garamendi, Vanel Lazcano, Coloma Ballester

Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra
{jf.garamendi, coloma.ballester}@upf.edu

Abstract

This document describes an implementation of the energy functional minimization proposed by Ballester, Garrido, Lazcano and Caselles for joint optical flow and occlusion estimation. The method build up from the ideas behind the TV-L^1 approach introduced by Zach, Pock and Bischof in 2007 but incorporating information that allows to detect occlusions. This information is based on the divergence of the flow and the proposed energy favors the location of occlusions on regions where this divergence is negative. The implemented variational method uses three consecutive frames. The energy functional is composed of regularization terms using the total variation, a data term using the L^1 norm, and a term dealing with the occlusions. In the present implementation, we solve the stationary system of partial differential equations arising from the dual minimization problem associated with the TV operator by a variation of the box relaxation numerical scheme proposed by Garamendi, Gaspar, Malpica and Schiavi. This makes the overall algorithm faster than previous implementations based on a gradient descent method.

Source Code

C source code of the described algorithms is provided and accessible at the IPOL web page of this article[1].

Keywords: optical flow; occlusions; total variation; TV-L1; staggered; box cell centered

1 Introduction

A key problem in computer vision is the estimation of the movement from a sequence of images. The purpose of motion estimation methods is to compute a motion field representing the displacement of points in consecutive images. This problem can be addressed using a variational formulation by modeling the problem as the minimization of an energy functional. Generally, the functional is composed of a data fidelity term and a regularization term. The data fidelity term is usually based on the conservation of some property during motion. A common data term is based on the brightness constancy assumption, which assumes that the observed object brightness -in particular, the observed point brightness- does not change along its motion trajectory. The regularization term

[1] https://doi.org/10.5201/ipol.2019.118

allows to define the structure of the motion field and ensures that the optical flow problem is well posed; it can be seen as an "a priori" requirement.

However, these methods may fail in occlusion areas due to forced, but unreliable, intensity matching. The problem can be further accentuated if the optical flow is smoothed across object boundaries adjacent to occlusion areas. In this work, we propose an implementation of the method proposed by Ballester et al. [1] for joint estimation of occlusion areas and optical flow. The method builds up from the ideas behind the Zach et al. approach [6] and incorporates information that allows to detect occlusions. An important difference between the present method [1] and the previous ones is that it requires three consecutive frames (instead of two) because it is based on the assumption that pixels that are not visible in the next frame are visible in the previous one.

Let $\Omega \subset \mathbb{R}^2$ be the image domain, usually a rectangle, and let $I_0, I_1 : \Omega \to \mathbb{R}$ be two consecutive image frames of a video secuence. Let $\mathbf{u} = (u_1, u_2) : \Omega \to \mathbb{R}^2$ be the optical flow between I_0 and I_1, and $\mathbf{x} \in \Omega$ be the spatial variable that denotes pixel location. To estimate $\mathbf{u}$, Zach et al. [6], propose to minimize the energy

$$\widetilde{J}(\mathbf{u}) = |Du_1|(\Omega) + |Du_2|(\Omega) + \lambda \int_\Omega |I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| d\mathbf{x}, \tag{1}$$

where the last term, $\int_\Omega |I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| d\mathbf{x}$, has a data fidelity role favoring the well-known brightness constancy assumption (that is, for a point $\mathbf{x}$ which moves with optical flow $\mathbf{u}(\mathbf{x})$, its brightness $I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))$ in the following frame should be similar to $I_0(\mathbf{x})$), the regularization term is based on the total variation of u_1 and u_2 (denoted here by $|Du_1|(\Omega)$ and $|Du_2|(\Omega)$, respectively) and λ is a parameter that balances the regularization term and the data fidelity term. Let us briefly recall that the total variation of a scalar function w belonging to $L^1(\Omega)$ in Ω is defined by

$$|Dw|(\Omega) = \sup_{\mathbf{p} \in [C_c^1(\Omega)]^2} \left\{ \int_\Omega w \operatorname{div}(\mathbf{p})\ d\mathbf{x}; \quad |\mathbf{p}|_{[L^\infty(\Omega)]^2} \leq 1 \right\},$$

where $[C_c^1(\Omega)]^2$ stands for the space of continuously differentiable vectorial functions with compact support in Ω and $|\cdot|_{[L^\infty(\Omega)]^2}$ is the essential supremum norm. With the previous definition, the space of functions in Ω with bounded variation, $BV(\Omega)$, is defined as

$$BV(\Omega) = \left\{ u \in L^1(\Omega);\ |Du|(\Omega) < \infty \right\},$$

Notice that if $w \in C^1(\Omega)$, then $\int_\Omega w \operatorname{div}(\mathbf{p})\ d\mathbf{x} = -\int_\Omega \nabla w \cdot \mathbf{p}\ d\mathbf{x}$ and $|Dw|(\Omega) = \int_\Omega |\nabla w(\mathbf{x})|\ d\mathbf{x}$. In the optical flow literature, the TV-L1 variational method given by (1) is usually written as

$$\widetilde{J}(\mathbf{u}) = \int_\Omega |\nabla u_1| + \int_\Omega |\nabla u_2| + \lambda \int_\Omega |I_0(\mathbf{x}) - I_1(\mathbf{x} + \mathbf{u}(\mathbf{x}))| d\mathbf{x}, \tag{2}$$

We will follow this notation in the following.

This method does not take into account the occlusions between two consecutive frames. Ballester et al. [1] added some terms to the functional in order to jointly compute occlusions and optical flow. Their model is based on the assumption that pixels of I_0 that are not visible in frame I_1 are visible in the previous frame to I_0, that is, in I_{-1}. Let $\chi : \Omega \to \{0, 1\}$ be a characteristic function representing the occlusion map, so that $\chi(\mathbf{x}) = 1$ means that $\mathbf{x}$ is visible in I_0 but not in I_1. That is, the set $\{\mathbf{x} \in \Omega;\ \chi(\mathbf{x}) = 1\}$ represents the occluded region due to motion. Let $I_{-1} : \Omega \to \mathbb{R}$ be the previous frame to I_0. It is assumed that the occluded pixels in I_1 are visible in I_{-1}. Thus, if $\chi(\mathbf{x}) = 0$ (the pixel is not occluded), we compare $I_0(\mathbf{x})$ with $I_1(\mathbf{x}+\mathbf{u}(\mathbf{x}))$. If $\chi(\mathbf{x}) = 1$ (the pixel is occluded), we compare $I_0(\mathbf{x})$ with $I_{-1}(\mathbf{x} - \mathbf{u}(\mathbf{x}))$. On the other hand, the occluded region should be correlated with the region where the divergence of the optical flow, $\operatorname{div}(\mathbf{u})$, is negative (for infinitesimal displacement).

This was pointed out by Sand and Teller [5], who noticed that the divergence of the motion field may be used to distinguish between different types of motion areas. Schematically, the divergence of a flow field is negative for occluded areas, positive for disoccluded, and near zero for the matched areas. Taking this into account, in [1] it is proposed to jointly compute the optical flow and occlusions by minimizing the energy

$$J(\mathbf{u},\chi) = J_d(\mathbf{u},\chi) + J_r(\mathbf{u},\chi) + \beta \int_\Omega \chi \operatorname{div}(\mathbf{u}) d\mathbf{x} + \frac{\alpha}{2}\int_\Omega \chi |\mathbf{u}|^2 d\mathbf{x}, \tag{3}$$

with $\alpha \geq 0$ and $\beta > 0$, $|\cdot|$ refers to the Euclidean norm and where

$$J_d(\mathbf{u},\chi) = \lambda \int_\Omega ((1-\chi)|I_0(\mathbf{x}) - I_1(\mathbf{x}+\mathbf{u}(\mathbf{x}))| + \chi |I_0(\mathbf{x}) - I_{-1}(\mathbf{x}-\mathbf{u}(\mathbf{x}))|) d\mathbf{x} \tag{4}$$

is the data fidelity term and

$$J_r(\mathbf{u},\chi) = \int_\Omega g(\mathbf{x}) \left(|\nabla u_1| + |\nabla u_2| + |\nabla \chi|\right) d\mathbf{x}$$

is the regularization term. The weighting function g is used to penalize regularization through an intensity edge. In the actual implementation, g is simply chosen as

$$g(\mathbf{x}) = (1 + \gamma |\nabla \tilde{I}_0(\mathbf{x})|)^{-1}, \qquad \mathbf{x} \in \Omega \tag{5}$$

where $\tilde{I}_0$ is either I_0 or any smoothed version of I_0. Notice that there is a new term in (3), weighted by $\alpha \geq 0$ (taken small relative to λ) to penalize large displacements where $\chi = 1$. In [1] this term is justified by two observations. On the one hand, it is assumed that the occluded background area is moving slower than the occluding foreground. On the other hand, since images have usually self-similarities, a pixel may have several possibilities to match. Based on this, taking $\alpha > 0$ and small, the model would encourage choosing the smallest displacement. As usual, in order to cope with the L^1 data term, an auxiliary variable $\mathbf{v} = (v_1, v_2)$ with $v_i \in BV(\Omega)$, that represents the optical flow, is introduced and penalized its deviation from $\mathbf{u}$ [6]. Thus, we minimize the energy

$$J_\theta(\mathbf{v},\mathbf{u},\chi) = J_d(\mathbf{v},\chi) + J_r(\mathbf{u},\chi) + \beta \int_\Omega \chi \operatorname{div}(\mathbf{u}) d\mathbf{x} + \frac{\alpha}{2}\int_\Omega \chi |\mathbf{v}|^2 d\mathbf{x} + \frac{1}{2\theta}\int_\Omega |\mathbf{u}-\mathbf{v}|^2 d\mathbf{x}, \tag{6}$$

depending on the three variables $(\mathbf{v},\mathbf{u},\chi)$, where $\theta > 0$. This energy can be minimized by alternatively fixing two variables and minimizing with respect to the third one.

The non-linear terms $I_1(\mathbf{x}+\mathbf{v}(\mathbf{x}))$ and $I_{-1}(\mathbf{x}-\mathbf{v}(\mathbf{x}))$ can be linearized using Taylor expansions around a given vector field $\mathbf{u}_0$. Let us define the residual

$$\rho_{\pm 1}(\mathbf{v}) := I_{\pm 1}(\mathbf{x} \pm \mathbf{u}_0(\mathbf{x})) \pm \nabla I_{\pm 1}\left(\mathbf{x} \pm \mathbf{u}_0(\mathbf{x})\right) \cdot (\mathbf{v}(\mathbf{x}) - \mathbf{u}_0(\mathbf{x})) - I_0(\mathbf{x}).$$

The procedure is applied by iteratively minimizing the energy

$$\tilde{J}_\theta(\mathbf{v},\mathbf{u},\chi) = \tilde{J}_d(\mathbf{v},\chi) + J_r(\mathbf{u},\chi) + \beta \int_\Omega \chi \operatorname{div}(\mathbf{u}) d\mathbf{x} + \frac{\alpha}{2}\int_\Omega \chi |\mathbf{v}|^2 d\mathbf{x} + \frac{1}{2\theta}\int_\Omega |\mathbf{u}-\mathbf{v}|^2 d\mathbf{x}, \tag{7}$$

where

$$\tilde{J}_d(\mathbf{v},\chi) = \lambda \int_\Omega \left((1-\chi)|\rho_1(\mathbf{v})| + \chi |\rho_{-1}(\mathbf{v})|\right) d\mathbf{x}.$$

To minimize $\tilde{J}_\theta$ we alternate between the minimization with respect to each variable keeping the other two fixed. More precisely, the energy functionals associated to each minimization are the following and in the next section we will detail the numerical algorithm used to solve each of them.

1. Minimization with respect to $\mathbf{v}$

$$\lambda \int_\Omega ((1-\chi)|\rho_1(\mathbf{v})| + \chi|\rho_{-1}(\mathbf{v})|)\, d\mathbf{x} + \frac{\alpha}{2}\int_\Omega \chi|\mathbf{v}|^2 d\mathbf{x} + \frac{1}{2\theta}\int_\Omega |\mathbf{u}-\mathbf{v}|^2 d\mathbf{x}. \tag{8}$$

2. Minimization with respect to $\mathbf{u}$

$$\int_\Omega g(\mathbf{x})\,(|\nabla u_1| + |\nabla u_2|)\, d\mathbf{x} + \beta\int_\Omega \chi \operatorname{div}(\mathbf{u}) d\mathbf{x} + \frac{1}{2\theta}\int_\Omega |\mathbf{u}-\mathbf{v}|^2 d\mathbf{x}. \tag{9}$$

3. Minimization with respect to χ

$$\int_\Omega g(\mathbf{x})|\nabla\chi| d\mathbf{x} + \beta\int_\Omega \chi \operatorname{div}(\mathbf{u}) d\mathbf{x} + \lambda\int_\Omega ((1-\chi)|\rho_1(\mathbf{v})| + \chi|\rho_{-1}(\mathbf{v})|)\, d\mathbf{x} + \frac{\alpha}{2}\int_\Omega \chi|\mathbf{v}|^2 d\mathbf{x}. \tag{10}$$

2 Numerical Scheme and Implementation Details

The minimization of (7) is done by alternating between the minimization with respect to each variable while keeping the other two fixed, i.e, minimizing energy functionals (8), (9) and (10).

2.1 Minimization with respect to v

Let us define

$$\begin{array}{lcl}
\delta_1 & = & 1,\\
\delta_{-1} & = & 1/(1+\alpha\theta),\\
\mu_1 & = & \lambda\theta,\\
\mu_{-1} & = & \mu_1/(1+\alpha\theta),\\
\Lambda_1(\mathbf{u}) & = & \rho_1(\mathbf{u}),\\
\Lambda_{-1}(\mathbf{u}) & = & \rho_{-1}(\mathbf{u}) + \mu_{-1}\mathbf{u}\cdot\nabla I_{-1}(\mathbf{x}-\mathbf{u}_0),\\
i & = & \{-1,1\}.
\end{array}$$

Notice that we omitted the arguments $\mathbf{x}$ in $\mathbf{u}, \mathbf{u}_0$. Fixed $\mathbf{u}$ and χ, the minimum of $\tilde{J}_\theta$ with respect to $\mathbf{v} = (v_1, v_2)$ is

$$\mathbf{v} = TH(\mathbf{u},\mathbf{u}_0)) := \begin{cases} \delta_i\mathbf{u} - i\mu_i\nabla I_i(\mathbf{x}+i\mathbf{u}_0) & \text{if} \quad \Lambda_i(\mathbf{u}) > \mu_i|\nabla I_i(\mathbf{x}+i\mathbf{u}_0)|^2,\\ \delta_i\mathbf{u} + i\mu_i\nabla I_i(\mathbf{x}+i\mathbf{u}_0) & \text{if} \quad \Lambda_i(\mathbf{u}) < -\mu_i|\nabla I_i(\mathbf{x}+i\mathbf{u}_0)|^2,\\ \mathbf{u} - i\rho_i(\mathbf{u})\frac{\nabla I_i(\mathbf{x}+i\mathbf{u}_0)}{|\nabla I_i(\mathbf{x}+i\mathbf{u}_0)|^2} & \text{if} \quad |\Lambda_i(\mathbf{u})| \le \mu_i|\nabla I_i(\mathbf{x}+i\mathbf{u}_0)|^2. \end{cases} \tag{11}$$

with $i = 1$ for those points where $\chi = 0$ and $i = -1$ for those points where $\chi = 1$. The ∇ operator applied to I_i is discretized using central differences.

2.2 Minimization with respect to u

Assume that $\mathbf{v}$ and χ are fixed. Then the minimum of $\tilde{J}_\theta$ with respect to $\mathbf{u} = (u_1, u_2)$ is given by

$$u_i = v_i + \theta\operatorname{div}(g\boldsymbol{\xi_i}) + \theta\beta\frac{\partial\chi}{\partial x_i}, \quad i = 1,2$$

where $\boldsymbol{\xi_i} = (\xi_{i,1}, \xi_{i,2}) : \Omega \to \mathbb{R}^2$ is the dual vector field for u_i computed as the solution of the following system of partial differential equations

$$g\nabla\left(v_i + \theta\operatorname{div}(g\boldsymbol{\xi_i}) + \theta\beta\frac{\partial\chi}{\partial x_i}\right) - \left|g\nabla\left(v_i + \theta\operatorname{div}(g\boldsymbol{\xi_i}) + \theta\beta\frac{\partial\chi}{\partial x_i}\right)\right|\boldsymbol{\xi_i} = 0, \tag{12}$$

completed with homogeneous Dirichlet boundary conditions.

In Ballester et al., the system of equations (12) is solved using the following fixed point (gradient descent) iterative scheme

$$\boldsymbol{\xi_i}^{k+1} = \frac{\boldsymbol{\xi_i}^k + \frac{\tau}{\theta} g\nabla\left(v_i + \theta\,\mathrm{div}(g\boldsymbol{\xi_i}^k) + \theta\beta\frac{\partial\chi}{\partial x_i}\right)}{1+\frac{\tau}{\theta}\left|g\nabla\left(v_i+\theta\,\mathrm{div}(g\boldsymbol{\xi_i}^k)+\theta\beta\frac{\partial\chi}{\partial x_i}\right)\right|}, \tag{13}$$

where τ is an artificial time step. Althought easy to implement, scheme (13) inherits the disadvantages of descent methods like the large number of iterations necessary for the stabilization. In this work, we adapt one of the numerical schemes proposed by Garamendi et al. [3] for solving the dual formulation of the Rudin-Osher-Fatemi model [4, 2]. We rewrite (12) as

$$\nabla\left(f_i + \mathrm{div}(\mathbf{p_i})\right) - \frac{1}{g}\left|\nabla\left(f_i + \mathrm{div}(\mathbf{p_i})\right)\right|\mathbf{p_i} = 0, \tag{14}$$

where the new varible $\mathbf{p_i} = (p_{i,1}, p_{i,2})$ is defined as $\mathbf{p_i} = g\boldsymbol{\xi_i}$ and $f_i = \theta^{-1}v_i + \beta\frac{\partial\chi}{\partial x_i}$. Notice that the θ parameter has been embedded in f_i. For the practical numerical implementation, we rewrite the equations as

$$u_i = \theta\left(f_i + \mathrm{div}(\mathbf{p_i})\right), \tag{15}$$

$$\nabla\left(f_i + \mathrm{div}(\mathbf{p_i})\right) - \frac{1}{g\theta}\left|\nabla u_i\right|\mathbf{p_i} = 0. \tag{16}$$

Notice that when $g(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$, we have exactly (for a given image u_0 such as $f_i = u_0/\theta$) the first order optimality condition of the Rudin-Osher-Fatemi denoising energy functional in its dual definition [4, 2].

We implemented an improved version of the box cell centered (BCC) scheme, that was the fastest of the proposed methods in [3]. The BCC method is an iterative box relaxation numerical scheme defined on staggered grids. In a staggered grid, the nodes corresponding to different unknowns are placed in alternated positions on the grid. In our case, the initial data f_i and the computed field flow values are located at cell centers, and the discrete values of the vectorial function $\mathbf{p_i}$ are located at the grid cell edges (see Figure 1).

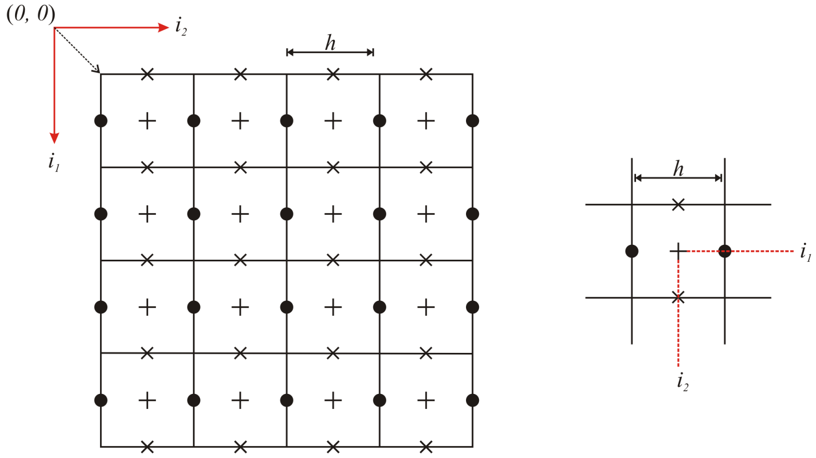

Figure 1: Staggered location of unknowns: u and f are described at + nodes, p^{x_1} is described at × nodes and p^{x_2} is described at • nodes. Adapted from [3].

To simplify the notation in the discretized expressions, we will omit the index i of $\mathbf{p_i}$ and f_i corresponding to the first and second component of the flow $\mathbf{u}$. Also, from now on, the subscripts will represent relative location to a specific pixel in the discretized staggered grid. Superscripts will denote vector components of the dual variable $\mathbf{p}$ (x_1 for the first component and x_2 for the second component); also, superscripts will denote iteration. For instance, for the pixel located at the (j,k) grid position, $p^{x_1,k+1}_{1/2,0}$ denotes the value of $p_{i,1}$ at position $(j+\frac{1}{2}h, k+0h)$ in the iteration $k+1$, where h is the pixel size, usually $h=1$ (see Figures 1 and 2).

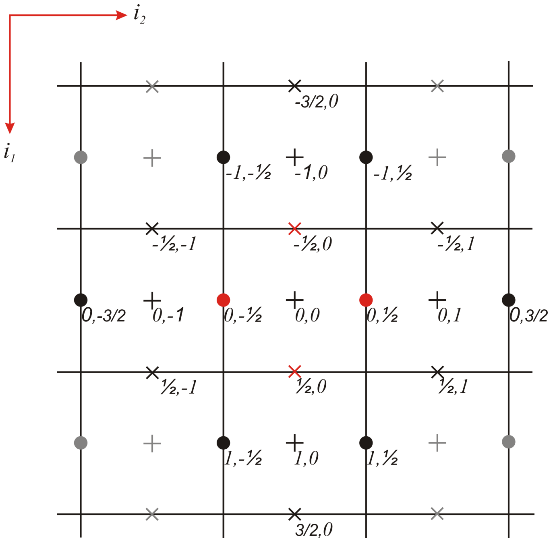

Figure 2: In red, nodes updated simultaneously at each pixel. In black, nodes involved in the independent term $(W,N,S,E)^T$ of the system. Adapted from [3].

At each iteration k of the BCC scheme (one iteration is finished when all pixels are visited) and for each pixel in the image we have to solve a 4×4 algebraic system $\mathbf{A}\bar{x}=\bar{b}$, with $\bar{x}=\left(p^{x_2,k+1}_{0,-1/2},\, p^{x_1,k+1}_{-1/2,0},\, p^{x_1,k+1}_{1/2,0},\, p^{x_2,k+1}_{0,1/2}\right)^T$. The improvement introduced in the present work to the BCC scheme is the inclusion of a parameter, ω, that balances the solution computed in the current iteration $(k+1)$ and the solution computed in the previous iteration (k), in the following way,

1. Let $\bar{x}^{k+1}_0$ be a solution of the 4×4 algebraic system of equations $\mathbf{A}\bar{x}=\bar{b}$ found by an exact method, e.g. Cramer's rule or Gauss elimination.

2. Compute $\bar{x}^{k+1}$ as
$$\bar{x}^{k+1}=(1-\omega)\bar{x}^k+\omega\bar{x}^{k+1}_0.$$

This parameter plays the same role as the ω parameter in a successive over-relaxation method for solving a linear system of algebraic equations. Notice that if $\omega=1$ we have exactly the original BCC scheme. We refer to the original work of Garamendi et al. [3] for a detailed explanation of the staggered grid, the operators defined on it and the BCC scheme.

Let us now explain in detail the matrix $\mathbf{A}$ and the independent term $\bar{b}$ (in the form of augmented matrix $(\mathbf{A}|\bar{b})$).

2.2.1 Inner Cells

For each cell that does not belong to the boundary of the image domain, the augmented matrix is as follows

$$(\mathbf{A}|\bar{b})=\left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 1 & 1 & B_{1/2,0} & -1 & S \\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W &=& -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}),\\ N &=& -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}),\\ S &=& -p^{x_1,k}_{3/2,0} - p^{x_2,k}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}),\\ E &=& -p^{x_2,k}_{0,3/2} - p^{x_1,k}_{1/2,1} + p^{x_1,k}_{-1/2,1} - (f_{0,1} - f_{0,0}), \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& (1-\omega)p^{x_2,k}_{0,-1/2} + \omega\frac{|A_1|}{|A|},\\ p^{x_1,k+1}_{-1/2,0} &=& (1-\omega)p^{x_1,k}_{-1/2,0} + \omega\frac{|A_2|}{|A|},\\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{|A_3|}{|A|},\\ p^{x_2,k+1}_{0,1/2} &=& (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{|A_4|}{|A|}, \end{array} \tag{17}$$

with

$$\begin{array}{rcl} |A_1| &=& W[B_{-1/2,0}B_{1/2,0}B_{0,1/2} - B_{-1/2,0} - B_{1/2,0} - B_{0,1/2} - 2]\\ &+& [NB_{1/2,0}B_{0,1/2} - EB_{1/2,0} - SB_{0,1/2} - N - E - S]\\ &+& [SB_{-1/2,0}B_{0,1/2} - EB_{-1/2,0} + NB_{0,1/2} - E + S + N]\\ &-& [EB_{-1/2,0}B_{1/2,0} + SB_{-1/2,0} - NB_{1/2,0} - E + S - N],\\ |A_2| &=& B_{0,-1/2}[NB_{1/2,0}B_{0,1/2} - EB_{1/2,0} - SB_{0,1/2} - N - E - S]\\ &+& [-B_{1/2,0}B_{0,1/2} - B_{1/2,0} - B_{0,1/2} - 1]\\ &+& [-SB_{0,1/2} - NB_{0,1/2} - N - S]\\ &-& [NB_{1/2,0} + EB_{1/2,0} + N + E],\\ |A_3| &=& -B_{0,-1/2}[SB_{-1/2,0}B_{0,1/2} - EB_{-1/2,0} + NB_{0,1/2} - E + S + N]\\ &+& [-SB_{0,1/2} - NB_{0,1/2} - N - S]\\ &+& W[-B_{1/2,0}B_{0,1/2} - B_{1/2,0} - B_{0,1/2} - 1]\\ &-& [EB_{-1/2,0} + SB_{-1/2,0} - E + S],\\ |A_4| &=& B_{0,-1/2}[EB_{-1/2,0}B_{1/2,0} + SB_{-1/2,0} - NB_{1/2,0} - E + S - N]\\ &-& [NB_{1/2,0} + EB_{1/2,0} + N + E]\\ &+& [EB_{-1/2,0} + SB_{-1/2,0} - E + S]\\ &-& W[B_{-1/2,0}B_{1/2,0} + B_{-1/2,0} + B_{1/2,0} + 1],\\ |A| &=& B_{0,-1/2}[B_{-1/2,0}B_{1/2,0}B_{0,1/2} - B_{-1/2,0} - B_{1/2,0} - B_{0,1/2} - 2]\\ &+& [-B_{1/2,0}B_{0,1/2} - B_{1/2,0} - B_{0,1/2} - 1]\\ &+& [-B_{1/2,0}B_{0,1/2} - B_{1/2,0} - B_{0,1/2} - 1]\\ &-& [B_{-1/2,0}B_{1/2,0} + B_{-1/2,0} + B_{1/2,0} + 1]. \end{array}$$

2.2.2 Boundary Cells

Due to the boundary conditions $\bar{p} \cdot \bar{n} = 0$, at pixels in the boundary cells of the image domain, the augmented matrix $(\mathbf{A}|\bar{b})$ is as follows

North-West Corner

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 1 & B_{1/2,0} & -1 & S\\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$ and

$$\begin{array}{rcl} S &=& -p^{x_1,m}_{3/2,0} - p^{x_2,m}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}),\\ E &=& -p^{x_2,m}_{0,3/2} - p^{x_1,m}_{1/2,1} + p^{x_1,m}_{-1/2,1} - (f_{0,1} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& 0, \\ p^{x_1,k+1}_{-1/2,0} &=& 0, \\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{SB_{0,1/2}+E}{|A|}, \\ p^{x_2,k+1}_{0,1/2} &=& (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{EB_{1/2,0}+S}{|A|}, \end{array} \tag{18}$$

with

$$|A| = B_{1/2,0}B_{0,1/2} - 1.$$

North Side

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & B_{1/2,0} & -1 & S \\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W &=& -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}), \\ S &=& -p^{x_1,m}_{3/2,0} - p^{x_2,m}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}), \\ E &=& -p^{x_2,m}_{0,3/2} - p^{x_1,m}_{1/2,1} + p^{x_1,m}_{-1/2,1} - (f_{0,1} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& (1-\omega)p^{x_2,k}_{0,-1/2} + \omega\frac{WB_{1/2,0}B_{0,1/2}-E-S-EB_{1/2,0}-SB_{0,1/2}-W}{|A|}, \\ p^{x_1,k+1}_{-1/2,0} &=& 0, \\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{SB_{0,-1/2}B_{0,1/2}-W+E-S-WB_{0,1/2}+EB_{0,-1/2}}{|A|}, \\ p^{x_2,k+1}_{0,1/2} &=& (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{EB_{0,-1/2}B_{1/2,0}+S-W-WB_{1/20}+SB_{0,-1/2}-E}{|A|}, \end{array} \tag{19}$$

with

$$|A| = B_{0,-1/2}B_{1/2,0}B_{0,1/2} - B_{1/2,0} - B_{0,1/2} - B_{0,-1/2} - 2.$$

North-East Corner

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & B_{1/2,0} & -1 & S \\ 0 & 0 & 0 & 1 & 0 \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W &=& -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}), \\ S &=& -p^{x_1,m}_{3/2,0} - p^{x_2,m}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& (1-\omega)p^{x_2,k}_{0,-1/2} + \omega\frac{WB_{1/2,0}-S}{|A|}, \\ p^{x_1,k+1}_{-1/2,0} &=& 0, \\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{SB_{0,-1/2}-W}{|A|}, \\ p^{x_2,k+1}_{0,1/2} &=& 0, \end{array} \tag{20}$$

with

$$|A| = B_{0,-1/2}B_{1/2,0} - 1.$$

West Side

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 1 & 1 & B_{1/2,0} & -1 & S \\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} N &=& -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}), \\ S &=& -p^{x_1,m}_{3/2,0} - p^{x_2,m}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}), \\ E &=& -p^{x_2,m}_{0,3/2} - p^{x_1,m}_{1/2,1} + p^{x_1,m}_{-1/2,1} - (f_{0,1} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& 0, \\ p^{x_1,k+1}_{-1/2,0} &=& (1-\omega)p^{x_1,k}_{-1/2,0} + \omega\frac{NB_{1/2,0}B_{0,1/2}-EB_{1/2,0}-SB_{0,1/2}-E-S-N}{|A|}, \\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{SB_{-1/2,0}B_{0,1/2}+EB_{-1/2,0}-NB_{0,1/2}-N+E-S}{|A|}, \\ p^{x_2,k+1}_{0,1/2} &=& (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{EB_{-1/2,0}B_{1/2,0}-NB_{1/2,0}+SB_{-1/2,0}-N+S-E}{|A|}, \end{array} \tag{21}$$

with

$$|A| = B_{-1/2,0}B_{1/2,0}B_{0,1/2} - B_{-1/2,0} - B_{0,1/2} - B_{1/2,0} - 2.$$

East Side

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 1 & 1 & B_{1/2,0} & -1 & S \\ 0 & 0 & 0 & 1 & 0 \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W &=& -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}), \\ N &=& -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}), \\ S &=& -p^{x_1,m}_{3/2,0} - p^{x_2,m}_{1,1/2} + p^{x_2,k+1}_{1,-1/2} - (f_{1,0} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} &=& (1-\omega)p^{x_2,k}_{0,-1/2} + \omega\frac{WB_{-1/2,0}B_{1/2,0}-SB_{-1/2,0}+NB_{1/2,0}-W+N-S}{|A|}, \\ p^{x_1,k+1}_{-1/2,0} &=& (1-\omega)p^{x_1,k}_{-1/2,0} + \omega\frac{NB_{0,-1/2}B_{1/2,0}-SB_{0,-1/2}+WB_{1/2,0}+W-N-S}{|A|}, \\ p^{x_1,k+1}_{1/2,0} &=& (1-\omega)p^{x_1,k}_{1/2,0} + \omega\frac{SB_{0,-1/2}B_{-1/2,0}-NB_{0,-1/2}-WB_{-1/2,0}-W-N-S}{|A|}, \\ p^{x_2,k+1}_{0,1/2} &=& 0, \end{array} \tag{22}$$

with

$$|A| = B_{0,-1/2}B_{-1/2,0}B_{1/2,0} - B_{0,-1/2} - B_{-1/2,0} - B_{1/2,0} - 2.$$

South-West Corner

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} N & = & -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}), \\ E & = & -p^{x_2,m}_{0,3/2} - p^{x_1,m}_{1/2,1} + p^{x_1,m}_{-1/2,1} - (f_{0,1} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} & = & 0, \\ p^{x_1,k+1}_{-1/2,0} & = & (1-\omega)p^{x_1,k}_{-1/2,0} + \omega\frac{NB_{0,1/2}-E}{|A|}, \\ p^{x_1,k+1}_{1/2,0} & = & 0, \\ p^{x_2,k+1}_{0,1/2} & = & (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{EB_{-1/2,0}-N}{|A|}, \end{array} \tag{23}$$

with

$$|A| = B_{-1/2,0}B_{0,1/2} - 1.$$

South Side

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & B_{0,1/2} & E \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W & = & -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}), \\ N & = & -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}), \\ E & = & -p^{x_2,m}_{0,3/2} - p^{x_1,m}_{1/2,1} + p^{x_1,m}_{-1/2,1} - (f_{0,1} - f_{0,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl} p^{x_2,k+1}_{0,-1/2} & = & (1-\omega)p^{x_2,k}_{0,-1/2} + \omega\frac{WB_{-1/2,0}B_{0,1/2}-EB_{-1/2,0}-NB_{0,1/2}-W+N-E}{|A|}, \\ p^{x_1,k+1}_{-1/2,0} & = & (1-\omega)p^{x_1,k}_{-1/2,0} + \omega\frac{NB_{0,-1/2}B_{0,1/2}-EB_{0,-1/2}+WB_{0,1/2}+W-N-E}{|A|}, \\ p^{x_1,k+1}_{1/2,0} & = & 0, \\ p^{x_2,k+1}_{0,1/2} & = & (1-\omega)p^{x_2,k}_{0,1/2} + \omega\frac{EB_{0,-1/2}B_{-1/2,0}-WB_{-1/2,0}-NB_{0,-1/2}-W-N-E}{|A|}, \end{array} \tag{24}$$

with

$$|A| = B_{0,-1/2}B_{-1/2,0}B_{0,1/2} - B_{0,-1/2} - B_{-1/2,0} - B_{0,1/2} - 2.$$

South-East Corner

$$(A|\bar{b}) = \left(\begin{array}{cccc|c} B_{0,-1/2} & -1 & 1 & 1 & W \\ -1 & B_{-1/2,0} & 1 & 1 & N \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{array}\right),$$

where $B_{i,j} = -2 - (g_{i,j}\theta)^{-1}|\nabla u^k_{i,j}|$, and

$$\begin{array}{rcl} W & = & -p^{x_2,k+1}_{0,-3/2} + p^{x_1,k+1}_{1/2,-1} - p^{x_1,k+1}_{-1/2,-1} - (f_{0,0} - f_{0,-1}), \\ N & = & -p^{x_1,k+1}_{-3/2,0} + p^{x_2,k+1}_{-1,1/2} - p^{x_2,k+1}_{-1,-1/2} - (f_{0,0} - f_{-1,0}). \end{array}$$

Using Cramer's rule, the solution of the system is as follows

$$\begin{array}{rcl}
p_{0,-1/2}^{x_2,k+1} & = & (1-\omega)p_{0,-1/2}^{x_2,k} + \omega\frac{WB_{-1/2,0}+N}{|A|}, \\
p_{-1/2,0}^{x_1,k+1} & = & (1-\omega)p_{-1/2,0}^{x_1,k} + \omega\frac{NB_{0,-1/2}+W}{|A|}, \\
p_{1/2,0}^{x_1,k+1} & = & 0, \\
p_{0,1/2}^{x_2,k+1} & = & 0,
\end{array} \tag{25}$$

with

$$|A| = B_{0,-1/2}B_{-1/2,0} - 1.$$

2.3 Minimization with respect to χ

Let $F = \lambda(|\rho_{-1}(\mathbf{v})| - |\rho_1(\mathbf{v})|)$ and $G = \frac{\alpha}{2}|\mathbf{v}|^2$. Let $0 < \tau_\eta, \tau_\chi < 1/8$. Given $\mathbf{u}$ and $\mathbf{v}$, the minimum $\widetilde{\chi}$ of $\tilde{J}_\theta$ with respect to χ can be obtained by the following primal-dual algorithm

$$\begin{array}{rcl}
\eta^{n+1} & = & P_B(\eta^n + \tau_\eta g \nabla \chi^n), \\
\chi^{n+1} & = & P_{[0,1]}(\chi^n + \tau_\chi(\mathrm{div}(g\eta^{n+1}) - \beta\, \mathrm{div}\mathbf{u} - F - G)),
\end{array} \tag{26}$$

where $P_B(\eta)$ denotes the projection of η on the unit ball of $\mathbb{R}^2$ and $P_{[0,1]}(r) = \max(\min(r,1),0)$, $r \in \mathbb{R}$. Notice that, by the co-area formula, the level sets of $\widetilde{\chi}$ are also minimizers of $\widetilde{J}_\theta$ ($\mathbf{u}, \mathbf{v}$ being fixed). Thus, before going to the next minimization of $\widetilde{J}_\theta$ with respect to $\mathbf{u}$, we redefine $\chi(\mathbf{x}) = T_\delta(\widetilde{\chi}(\mathbf{x}))$, where $T_\delta(r) = 1$ (resp. 0) if $r \geq \delta$ (resp. $< \delta$).

The ∇ and div operators applied on $\mathbf{v}$ and χ computations correspond to forward and backward finite differences (respectively).

3 Implemented Algorithm

The optical flow computation has been embedded in a pyramidal set of scales in order to cope with large displacements. At a certain scale and starting from downsampled versions of the input images, an optical flow problem is solved. The resulting optical flow is interpolated into the following finer scale and it is used as starting point for the optical flow problem at this finer scale. This procedure is iterated recursively from the coarsest version of the images to the original ones.

The algorithm is summarized in Algorithms 1, 2, 3 and 4. The inputs are three consecutive frames I_{-1}, I_0, I_1 of a given video sequence, a smoothed version of I_0 (which is only used to compute the function g multiplying the regularization terms in order to penalize regularization through an intensity edge) and the parameters of the model. The output of the algorithm are the optical flow field $\mathbf{u}$ and the occlusion layer χ. The occlusion layer χ is generated by thresholding χ^1 using function T so as to obtain a binary image, see Algorithm 1. Algorithm 2 includes a median filtering of the optical flow $\mathbf{u}$ before computing χ. This step allows to improve robustness for the computation of χ.

The implementation of the C code has been performed in the following main modules,

- Warp module: This module implements the necessary functions for the warping process.
- Pyramid module: This module implements the necessary functions for the pyramidal scale process.
- Solvers module: This module implements the procedures for minimizing the energy functional with respect to the variables $\mathbf{u}$, $\mathbf{v}$ and χ.

Algorithm 1: Pyramidal procedure

```
input  : Three consecutive frames $I_{-1}$, $I_0$, $I_1$
         A smoothed version of $I_0$, named $filtI_0$ (which can be chosen equal to $I_0$)
         The parameters for the optical flow computation at each scale
output: Flow field $\mathbf{u}$ and occlusion map $\chi$ for $I_0$

Normalize images in the range [0,255]
Pre-smooth the input images with Gaussian of standard deviation $\sigma$
Compute downscaled images $I^s_{-1}$,$I^s_0$, $I^s_1$, $filtI^s_0$ for $s = 1, \dots, N_{\text{scales}}$, (1 finest, $N_{\text{scales}}$ coarsest)
Initialize coarsest $\mathbf{u}^{N_{\text{scales}}} = 0$ and $\chi^{N_{\text{nscales}}} = 0$
for $s \leftarrow N_{\text{scales}}$ to 2 do
    $\mathbf{u}^s, \chi^s \leftarrow$ Dual_TVL1_optic_flow (Algorithm 2)
    $[\mathbf{u}^{s-1}, \chi^{s-1}] \leftarrow$ UpSampling($\mathbf{u}^s, \chi^s$)
$[\mathbf{u}^1, \chi^1] \leftarrow$ Dual_TVL1_optic_flow (Algorithm 2)
$\chi \leftarrow T(\chi^1)$
```

Algorithm 2: Dual_TVL1_optic_flow

```
input  : Three consecutive frames $I_{-1}$, $I_0$, $I_1$
         A smoothed version of $I_0$ named $filtI_0$
         $\lambda$, $\alpha$, $\beta$, $\gamma$ and $\theta$ parameters
         An initial optical flow $\mathbf{u}_0$ (usually coming from the previous scale)
output: Flow field $\mathbf{u}$ and occlusion layer $\chi$ for $I_0$

Compute function $g(\mathbf{x})$ using $filtI_0$ and Equation (5)
Initialize $\mathbf{v} = 0$ for the current scale
for $s \leftarrow 1$ to $N_{\text{warps}}$ do
    Compute the warped images $I_i(\mathbf{x} + i\mathbf{u}_0(\mathbf{x}))$, $\nabla I_i(\mathbf{x} + i\mathbf{u}_0(\mathbf{x}))$, and $\rho_i$, $i = \{-1, 1\}$
    while $n <$ ext_max_iterations and stopping_criterion $> \epsilon$ do
        $\mathbf{v} \leftarrow TH(\mathbf{u}, \mathbf{u}_0)$ (threshold using Equation (11)) // Minimization w.r.t. $\mathbf{v}$ (Section 2.1)
        $\mathbf{u} \leftarrow$ Solver_wrt_u (Algorithm 3) // Minimization w.r.t. $\mathbf{u}$ (Section 2.2)
        $\mathbf{u} \leftarrow$ if it is requiered, median filtering of $\mathbf{u}$ using a $3 \times 3$ window size.
        $\chi \leftarrow$ Solver_wrt_chi (primal-dual scheme (26)) //Minimization w.r.t. $\chi$ (Section 2.3)
```

Algorithm 3: Solver_wrt_u

```
input  : $\mathbf{v} = (\mathrm{v}_1, \mathrm{v}_2)$, the occlusion map $\chi$ and $g(\mathbf{x})$
         $\beta$, and $\theta$ parameters
output: Flow field $\mathbf{u} = (u_1, u_2)$
$(\chi_x, \chi_y) \leftarrow \nabla\chi$
$\mathbf{f} \leftarrow \mathbf{v}/\theta + \beta(\chi_x, \chi_y)$
$u_1 \leftarrow Scalar\_ROF\_BoxCellCentered(f_1, g, \theta)$ (Algorithm 4)
$u_2 \leftarrow Scalar\_ROF\_BoxCellCentered(f_2, g, \theta)$ (Algorithm 4)
```

```
Algorithm 4: Scalar_ROF_BoxCellCentered
  input  : f and $g(\mathbf{x})$
           $\omega$ successive over-relaxation parameter, $\theta$ parameter
  output: Denoised image u
  Initialize the dual variable $\mathbf{p} = (\mathrm{p}^{\mathrm{x}_1}, \mathrm{p}^{\mathrm{x}_2}) = \bar{0}$
  Initialize the solution u = f
  for $iter \leftarrow 1$ to MAX_ITERATIONS_U do
      $(u_x, u_y) \leftarrow ForwardGradient(u)$
      $\alpha \leftarrow \frac{\sqrt{u_x^2+u_y^2}}{g\theta}$
      for Each cell on the staggered grid do
          $B_{i,j} \leftarrow -2 - \alpha_{i,j}$
          $\begin{pmatrix} p^{x_2,k+1}_{0,-1/2} \\ p^{x_1,k+1}_{-1/2,0} \\ p^{x_1,k+1}_{1/2,0} \\ p^{x_2,k+1}_{0,1/2} \end{pmatrix} = \begin{cases} \text{Equation (17)} & if \quad \text{cell is an inner cell} \\ \text{Equation (18)} & if \quad \text{cell is the North-West corner cell} \\ \text{Equation (19)} & if \quad \text{cell is on the North line} \\ \text{Equation (20)} & if \quad \text{cell is the North-East corner cell} \\ \text{Equation (21)} & if \quad \text{cell is on the West column} \\ \text{Equation (22)} & if \quad \text{cell is on the East column} \\ \text{Equation (23)} & if \quad \text{cell is the South-West corner cell} \\ \text{Equation (24)} & if \quad \text{cell is on the South line} \\ \text{Equation (25)} & if \quad \text{cell is the South-East corner cell} \end{cases}$
      $u \leftarrow \theta(f + \mathrm{div}(\mathbf{p}))$
```

- tvl1OccFlow module: This module performs the iterations of the overall relaxation process for a given scale.

Let us add some comments on the implementation of the minimization with respect to $\mathbf{v}$, which is given by (3). In practice, these equations are implemented in the code by using two auxiliary variables $\mathbf{v}_{fwd}$ and $\mathbf{v}_{bck}$, satisfying

$$\mathbf{v}(\mathbf{x}) = \begin{cases} \mathbf{v}_{fwd}(\mathbf{x}) & \text{if} \quad \chi(\mathbf{x}) = 0, \\ \mathbf{v}_{bck}(\mathbf{x}) & \text{if} \quad \chi(\mathbf{x}) = 1. \end{cases} \tag{27}$$

In the same way, in the implementation of the minimization with respect to χ, the F and G variables are calculated depending whether the continuous *chi* is less or greater than 0.5.

The whole algorithm has some parameters. In our implementation there are parameters for the model and parameters to control the behavior of the numerical implementation. The parameters for the model are given in command line and the parameters for the numerical implementation can be controlled in a header file before compiling. This fact, and the modular organization of the algorithms, allow the possibility of changing the numerical aspects of the algorithms related to the minimization w.r.t the variables $\mathbf{u}$, $\mathbf{v}$, and χ without affecting the rest of the code, even without modifying the program call. The meaning of the model parameters is as follows:

Model parameters. These parameters allow to control the parameters for the model. Default values are given in parentheses.

- `lambda` ($\lambda = 0.15$): This parameter, see Equation (4), controls the trade-off between the data term and the regularization term. The smaller the parameter is, the smoother the solution is.

- alpha ($\alpha = 0.01$): This parameter, $\alpha \geq 0$ in Equation (6), is used to penalize large displacements of the motion field u in the occluded areas. The smaller the parameter is, the smaller the computed displacement in the occluded areas.

- beta ($\beta = 0.15$): The parameter, $\beta > 0$ in Equation (6), allows to control the sensibility of the algorithm to the detection of occluded areas.

- theta ($\theta = 0.3$): This parameter, see Equation (6), controls the attachment between the auxiliary variable $\mathbf{v}$ and the final solution $\mathbf{u}$. When θ is large there is no effective transfer of information between $\mathbf{u}$ and $\mathbf{v}$.

- nwarps ($N_{\text{warps}} = 2$): This parameter, see Algorithm 2, sets the number of warping iterations to be performed at each level. In order to obtain more accurate results, this parameter can be increased for instance to 10.

- epsilon ($\epsilon = 0.01$): This parameter, see Algorithm 2, is the stopping criterion for the iterative process.

- nscales ($N_{\text{nscales}} = \mathit{maximum}$): It corresponds to the number of levels for creating the image pyramid and is used jointly with the zfactor parameter. See Algorithm 1. The maximum number of scales is computed to assure that the smallest images of the pyramid don't have a size smaller than 16×16.

  If the displacements inside the video sequence are small, it can be set to 1.

- zfactor (0.5): This parameter, with values between 0 and 1, sets the downsampling factor to construct the image pyramid. A value of 0.5 is associated to the construction of a pyramid in which images are downsampled by a factor of 2. This parameter is also used for the UpSampling function that can be seen in Algorithm 1.

- medianFilter (1): This parameter switchs on (medianFilter=1) or off (medianFilter=0) the median filter (see Algorithm 2). When medianFilter=1 the final results of the test with the ground truth have less error, but we are not minimizing exactly the proposed energy functional.

Numerical parameters. The parameters below can be modified by editing the header source file constants.h. In any case, it is not needed/recommended.

- OMEGA ($\omega = 1.25$): It corresponds to the ω relaxation value in the minimization w.r.t. $\mathbf{u}$.

- TAU_ETA ($\tau_\eta = 0.15$): It corresponds to the time step for η in the minimization w.r.t. χ.

- TAU_CHI: ($\tau_\chi = 0.15$): It corresponds to the time step for χ in the minimization w.r.t. χ.

- THR_CHI: ($\delta = 0.75$): It corresponds to the threshold for χ.

- MAX_ITERATIONS_CHI (100): It corresponds to the number of iterations of the algorithm for minimizing w.r.t. χ.

- MAX_ITERATIONS_U (10): It corresponds to the number of iterations of the algorithm for minimizing w.r.t. $\mathbf{u}$.

- EXT_MAX_ITERATIONS (20): It corresponds to the maximum number of iterations for the second loop of Algorithm 2.

- G_FACTOR ($\gamma = 0.05$): It corresponds to the weight γ in Equation (5). The g function can be set to identity (so, without effect) setting the variable G_CHOICE to 1.

Output Optical Flow `.flo` file format. The program output for the optical flow is a file in format `.flo`. This format is defined in the Middlebury optical flow database[2], where the readers can find information and source code (in C++ and Matlab) for reading, writing and visualizing optical flow files.

4 Experimental Results

We present some experimental results of the implemented algorithm. The experiments were performed on sequences from the Middlebury Benchmark Database. This database contains data with public ground truth, so we can evaluate the error of the solution. As error metric we show the Average End-Point Error (EPE) and the Average Angular-Point Error (AAE), which can be computed as

$$EPE := (1/N)\sum_{i=1}^{N}\sqrt{\left(u_{1,i}-u_{1,i}^{gt}\right)^2+\left(u_{2,i}-u_{2,i}^{gt}\right)^2},$$

$$AAE := (1/N)\sum_{i=1}^{N}\arccos\left(\frac{u_{1,i}u_{1,i}^{gt}+u_{2,i}u_{2,i}^{gt}}{\sqrt{u_{1,i}^2-u_{2,i}^2+1}\sqrt{(u_{1,i}^{gt})^2-(u_{2,i}^{gt})^2+1}}\right),$$

with $\mathbf{u}^{gt} = (u_1^{gt}, u_2^{gt})$ being the ground truth solution (provided by the Middlebury Benchmark Database).

4.1 Evaluation on Middlebury

We did a parameter study and in Table 1 are shown the EPE and AAE for the test sequences with the best model parameters. Parameter β was set to 1 and ϵ was set to 10^{-6}. The other parameters were set to the default value except the number of warps, that we set to 10. The size of the images and the default scale factor of 0.5 gives a maximum number of scales of 5.

	Urban2	Urban3	Grove2	Grove3	Rubberwhale	Hydrangea
EPE	0.31488	0.34046	0.13573	0.56019	0.16501	0.21941
λ	0.6	1	0.4	0.8	0.3	0.1
θ	0.2	0.5	0.3	0.4	0.2	0.8
AAE	2.39342	2.97392	2.01494	5.3	5.33447	2.4380
λ	0.6	1	0.4	0.9	0.2	0.1
θ	0.2	0.5	0.3	0.5	0.2	0.8

Table 1: EPE, AAE, and best parameters for the Middlebury test sequences.

In Figure 3 the optical flow and the occlusion layer χ are shown for the Urban2 sequence. The optical flow u is shown using the color coding scheme at the right of the image. The occlusion layer is shown superimposed on the central frame of the sequence. As it can be seen, the method is able to detect the regions of I_0 that get occluded at frame I_1 due to the apparent movement of the buildings.

[2] http://vision.middlebury.edu/flow/

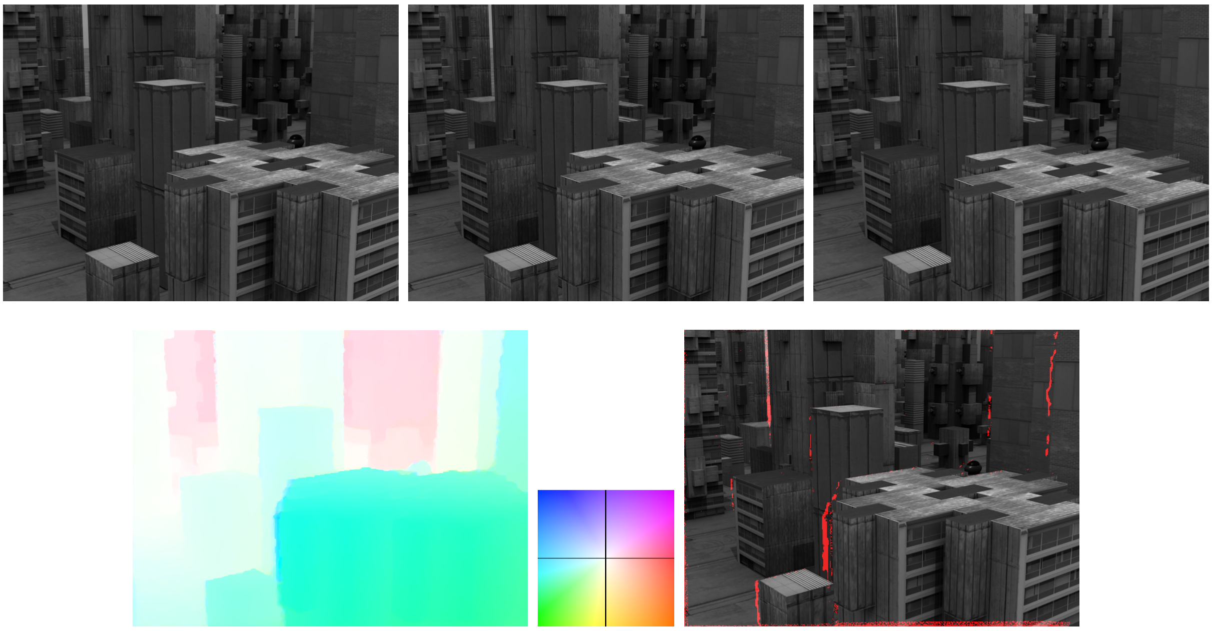

Figure 3: *First row:* Three consecutive frames I_{-1}, I_0, I_1 of the Urban2 Middlebury video sequence. *Second row:* The optical flow $\mathbf{u}$ in color coded (the color legend is besides) and the occlusion layer in red overimposed on I_0. The values of the parameters are given in the text.

4.2 Evaluation on MPI-Sintel

The MPI-Sintel database presents long synthetic video sequences containing large displacements and image degradation as blur, reflections, shadows and fog. There are two versions of the MPI-Sintel database: clean and final. The final version includes large displacements and image degradation. For our evaluation we took the final version of the video sequences. In Figure 4 we present some examples of sequences in the MPI-Sintel database.

In Figure 4, in the first row, is presented an image sequence with large displacement, around 170 pixels. In the second row is shown a girl trying to catch the small dragon. We observe large displacements in the girl's arms and also in the wings of the small dragon. In the third row we observe fog and deformation of the hand.

Our experiment is shown in Figure 5. We present an optical flow estimation and also the occlusion layer estimation. In the first row we show frame 20 and frame 21 of the sequence alley_1. The girl moves her arm downwards. In the second row we show the optical flow ground truth and the estimated optical flow. We observe that the optical flow is estimated correctly. In the third row we show the occlusion ground truth and the estimated occlusion layer. We observe that the occlusion is correctly estimated.

In the first row of Figure 6 we present frames of the sequence cave_4. This sequence presents large displacements (around 170 pixels). In the second row we present the optical flow ground truth and the estimated optical flow. We observe that the optical flow is not well estimated due to the large displacements. In the third row we present the occlusion layer ground truth and the estimated occlusion layer.

We have evaluated our method using the final version of the MPI database. We divided the database in three subsets: large displacements (displacements larger that 40 pixels), medium displacements (less than 40 pixels) and small displacements. We set $\lambda = 0.60$ and $\theta = 0.1$, $N_{\text{warps}} = 10$, $\alpha = 0.01$. We computed the average End Point Error for each video sequence in subsets of the database and we present the results in Table 2.

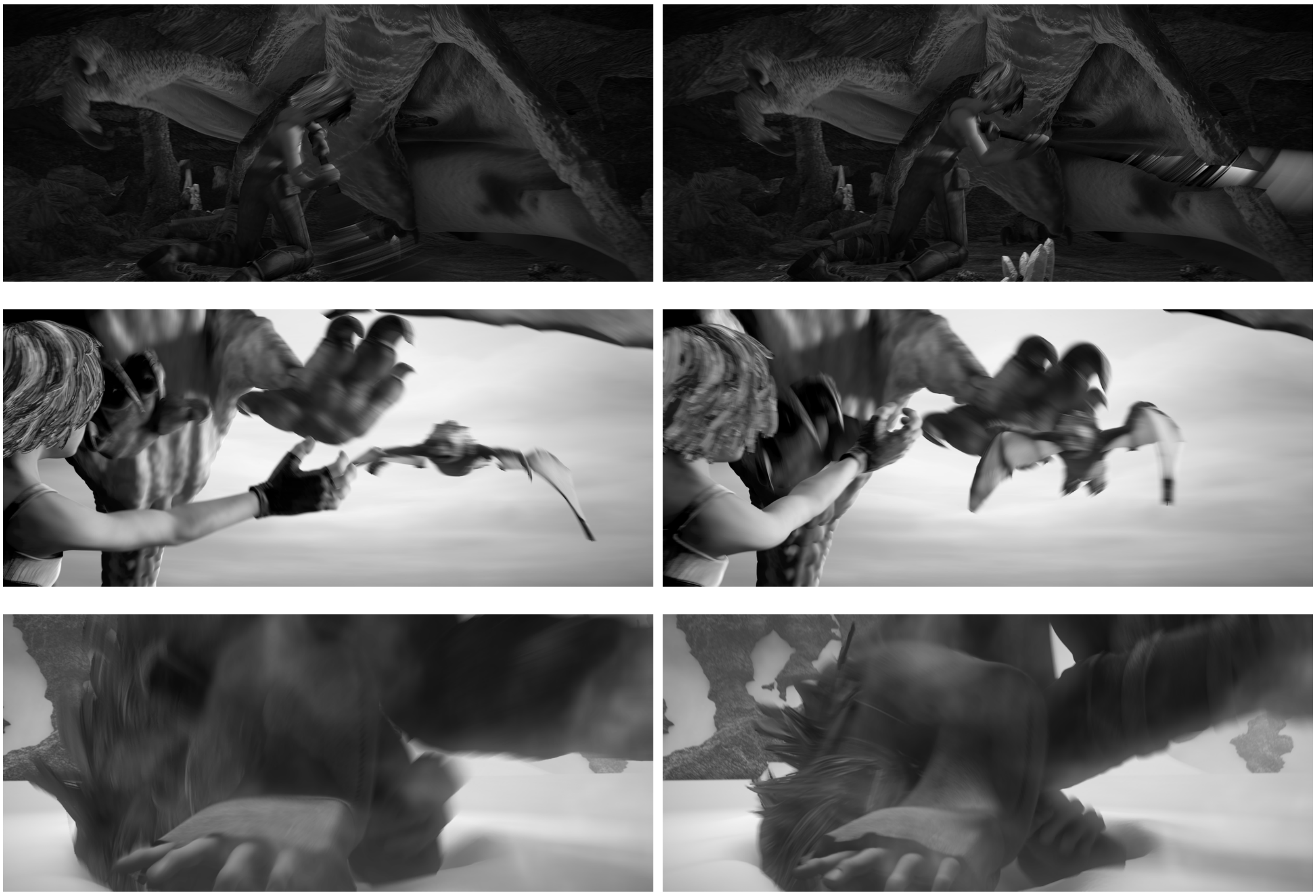

Figure 4: *First row:* Frames 10 and 11 of the sequence cave_4. *Second row:* Frames 30 and 31 of temple_3 video sequence. *Third row:* Frames 06 and 07 of ambush_4 video sequence.

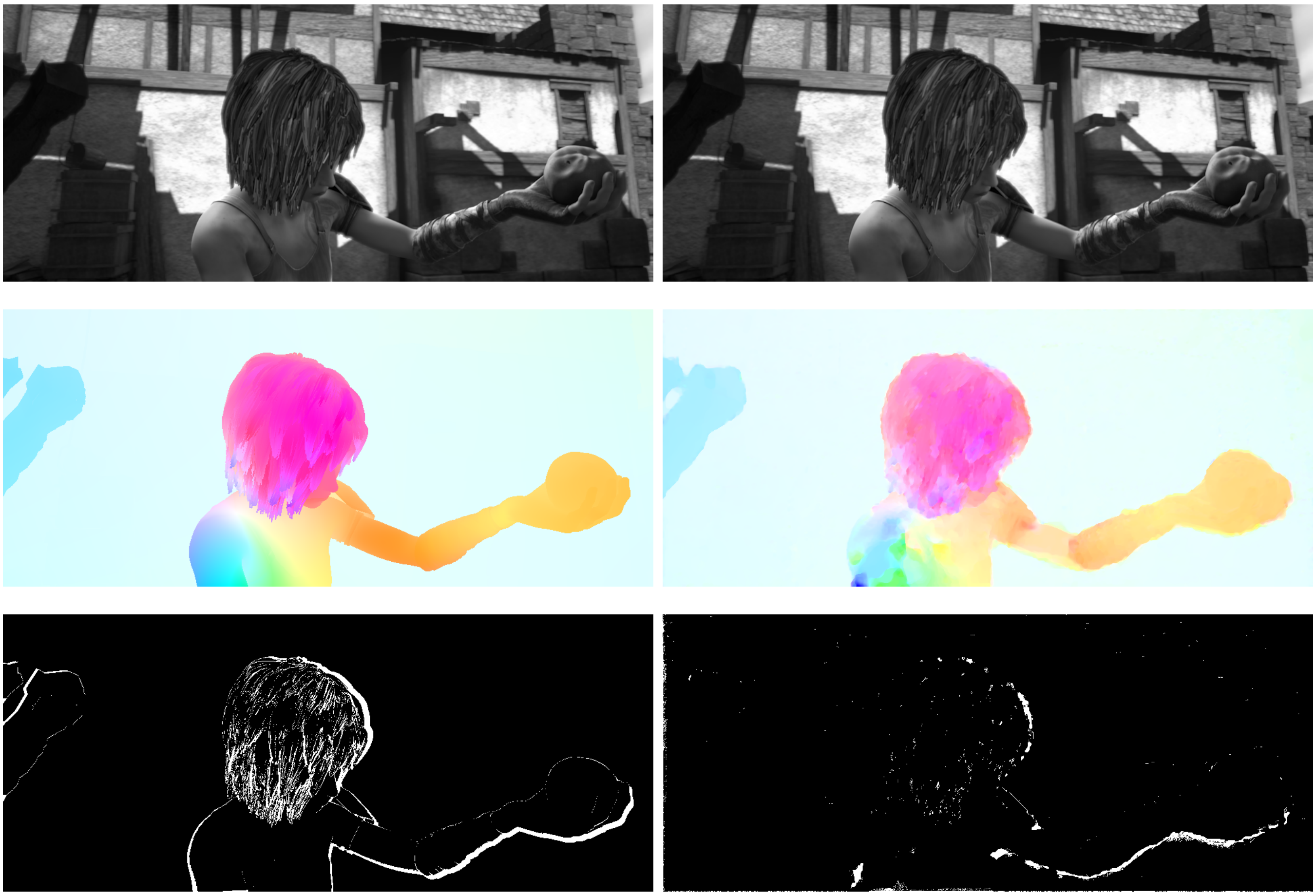

Figure 5: *First row:* Frames 19 and 20 of the sequence alley_1. *Second row:* Estimated optical flow and optical flow ground truth. *Third row:* Occlusion ground truth and estimated occlusion layer.

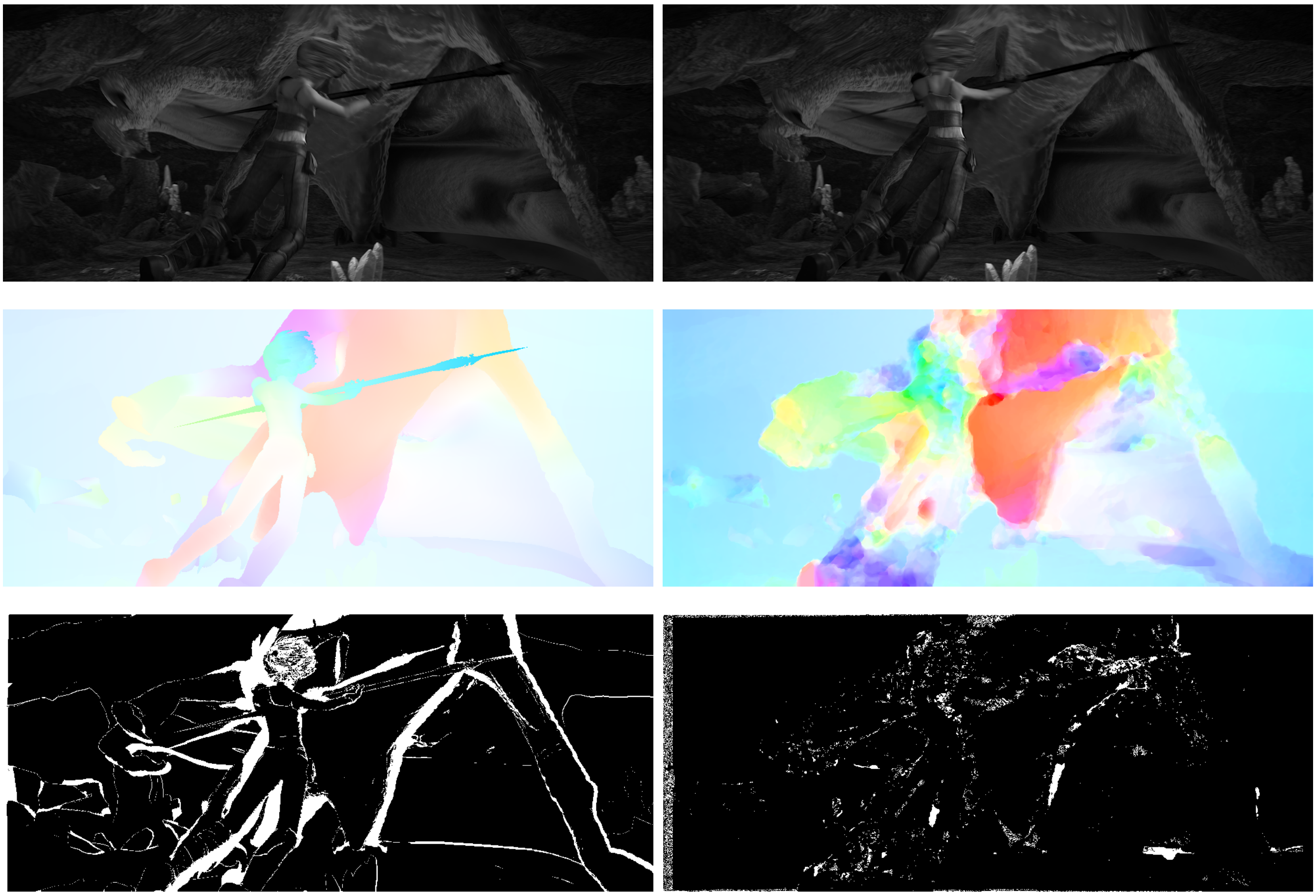

Figure 6: *First row:* Frames 13 and 14 of the sequence cave_4. *Second row:* Optical flow ground truth and estimated optical flow. *Third row* Ground truth layer and estimated occlusion layer.

Table 2: End Point Error obtained by our model in subset: large displacement, medium displacement and small displacement of MPI.

Large Displacement		Medium Displacement		Small Displacement	
Sequence	EPE	Sequence	EPE	Sequence	EPE
Ambush_2	52.16	Alley_2	0.62	Alley_1	0.46
Ambush_4	30.15	Bamboo_1	0.41	Ambush_7	4.76
Ambush_5	15.52	Bamboo_2	1.27	Bandage_1	1.12
Ambush_6	23.39	Market_2	1.24	Bandage_2	0.69
Market_5	21.84	Temple_2	4.28	Mountain_1	2.46
Market_6	9.23			Shaman_2	0.33
Cave_2	22.35			Shaman_3	0.95
Cave_4	5.78			Sleeping_1	0.11
Temple_3	26.33			Sleeping_2	0.18
Average EPE	21.08	Average EPE	1.56	Average EPE	1.37
Total Average EPE	8.34				

We observe in Table 2 that the average EPE in all sequences is 8.34. We observe that for small and medium displacements our method presents an EPE smaller than 1.56. For large displacements our method presents an EPE of 21.08. We conclude that the performance of our method is acceptable for small and medium displacements and that the method cannot handle large displacements.

5 Conclusions

This work proposes an implementation of the variational model for joint optical flow and occlusion estimation proposed by Ballester et al. The problem is modeled as an energy functional minimization with a term based on the divergence of the flow that favors the location of occlusions on regions where this divergence is negative. Our method requires three consecutive frames. Assuming that occluded pixels are visible in the previous frame, the optical flow on non-occluded pixels is forward estimated whereas is backwards estimated on the occluded ones. In the minimization process, there are three different energy functionals to minimize. The first one leads to a point-wise threshold, the second one is a problem very similar to the Rudin-Osher-Fatemi denoising model. The problem is solved following a Box Cell Centered on a staggered grid that gives a faster convergence on the ROF minimization. The third one is minimized using a primal-dual scheme. Finally, some test were done using the well known Middlebury database, with good quantitative results on the error with respect to the ground truth.

Acknowledgements

All three authors would like to thank Lluís Garrido and Vicent Caselles for their comments as well as to the anonymous reviewers that really helped us improve the quality of the paper and code.

The first author was partially supported by the Agència de Gestió d'Ajuts Universitaris i de Recerca de la Generalitat de Catalunya within the project PDJ-2014-00038. The first and third authors acknowledge partial support by the MINECO/FEDER project with reference TIN2015-70410-C2-1-R and by the MICINN project with reference MTM2012-30772.

Image Credits

The original images in Figure 3 are from the Middlebury dataset http://vision.middlebury.edu/flow/. The original images in Figures 4 to 6 have been extracted from the MPI-Sintel Dataset at http://sintel.is.tue.mpg.de/downloads.

References

[1] C. Ballester, L. Garrido, V. Lazcano, and V. Caselles, *A TV-L1 Optical Flow Method with Occlusion Detection*, in Pattern Recognition, vol. 7476, Springer Berlin Heidelberg, 2012, pp. 31–40. http://dx.doi.org/10.1007/978-3-642-32717-9_4.

[2] A. Chambolle, *An algorithm for total variation minimization and applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97. http://dx.doi.org/10.1023/B:JMIV.0000011325.36760.1e.

[3] J.F. Garamendi, F.J. Gaspar, N. Malpica, and E. Schiavi, *Box relaxation schemes in staggered discretizations for the dual formulation of total variation minimization.*, IEEE Transactions on Image Processing, 22 (2013), pp. 2030–43. http://dx.doi.org/10.1109/TIP.2013.2244220.

[4] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D Nonlinear Phenomena, 60 (1992), pp. 259–268. http://dx.doi.org/10.1016/0167-2789(92)90242-F.

[5] P. Sand and S. Teller, *Particle video: Long-range motion estimation using point trajectories*, International Journal of Computer Vision, 80 (2008), pp. 72–91. https://doi.org/10.1007/s11263-008-0136-6.

[6] C Zach, T Pock, and H Bischof, *A duality based approach for realtime TV-L 1 optical flow*, Pattern Recognition, 1 (2007). https://doi.org/10.1007/978-3-540-74936-3_22.