International Journal of Image and Graphics
Vol. 20, No. 4 (2020) 2050027 (35 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0219467820500278

Sparse Optical Flow Computation Using Wave Equation-Based Energy

Luiz Maurílio da Silva Maciel* and Marcelo Bernardes Vieira†

Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora, Rua Lourenço Kelmer
Juiz de Fora, Minas Gerais 36036-330, Brazil
**luiz.maurilio@ice.ufjf.br*
†marcelo.bernardes@ufjf.edu.br

Received 21 September 2018
Revised 16 February 2019
Accepted 18 February 2019
Published 9 October 2020

Identification of motion in videos is a fundamental task for several computer vision problems. One of the main tools for motion identification is optical flow, which estimates the projection of the 3D velocity of the objects onto the plane of the camera. In this work, we propose a differential optical flow method based on the wave equation. The optical flow is computed by minimizing a functional energy composed by two terms: a data term based on brightness constancy and a regularization term based on energy of the wave. Flow is determined by solving a system of linear equations. The decoupling of the pixels in the solution allows solving the system by a direct or iterative approach and makes the method suitable for parallelization. We present the convergence conditions for our method since it does not converge for all the image points. For comparison purposes, we create a global video descriptor based on histograms of optical flow for the problem of action recognition. Despite its sparsity, results show that our method improves the average motion estimation, compared with classical methods. We also evaluate optical flow error measures in image sequences of a classical dataset for method comparison.

Keywords: Optical flow; differential methods; wave equation; points-of-interest extraction; edge characterization.

1. Introduction

The study of movement in image sequences is an important field in computer vision for many years. Identifying movement in a video is a fundamental task in order to analyze its semantic information. This kind of information is useful in several applications, for example time-to-collision, motion compensated encoding, stereo disparity measurement, action recognition, and motion detection. However, extracting features that represent movement in a video is a challenge and not a fully exploited problem.

[*] Corresponding author.

Optical flow is a motion representation widely used in computer vision. It consists in estimating the projection of the 3D velocity of the objects onto the plane of the camera. In general, optical flow computation is based on estimation of the brightness variation from a sequence of images. The occlusion of objects is a major problem. Dealing with homogeneous regions and determining large displacements are other problems for optical flow methods.[1,2]

Several methods have been developed since Horn and Schunck[3] proposed their differential optical flow method.[4–7] Also, in the last years, new datasets for optical flow evaluation were proposed.[2,8,9] These are useful to evaluate the new optical flow algorithms. The set of benchmarks of these datasets contains sequences for several scene configurations. This shows that the computation of optical flow is still an open problem.

There is a large variety of optical flow methods and the most suitable for any problem is application dependent. An example of problem where movement extraction is useful is the human action recognition,[10,11] which consists of three stages: feature extraction, video descriptors creation and classification. Optical flow can be useful in the stage of motion features extraction, since it is an estimative of motion in image sequences.

Differential optical flow methods use the brightness of the images to extract movement information. Thus, making a physical analogy, we can assume the brightness as mass elements and analyses its temporal variation. For example, Lucas and Kanade[12] and Horn and Schunck[3] methods start from a movement equation which is just an equation with advective terms.

Another example of transport phenomenon is modeled by the wave equation.[13] This equation describes the propagation of waves in a continuous medium and was studied by many famous mathematicians including Euler, Bernoulli, d'Alembert and Lagrange. Several physical phenomena are based on this equation, for example, water waves and vibration of an elastic string.

In this work, we propose a differential optical flow method based on the wave equation. We start by modeling the brightness variation as waves propagating in a medium and by showing how the wave equation can contribute to estimate motion from image sequences. This study inspires us to propose an energy function that must be minimized similarly to Horn and Schunck approach.[3] Our energy function and its computational method are the main contributions of this work, which is based on the hypothesis that the wave equation is useful for motion estimation in videos. Our method provides a sparse optical flow, mainly concentrated on high brightness variation points, useful for Computer Vision problems that need good flow estimations on these regions. Our convergence condition, for instance, is useful for detecting points of interest on edges. We apply our method on the problem of action recognition, that needs meaningful motion flows, to compare with classical works.[12,3] Considering only the valid points, the results showed that our method overcomes the classical ones. The performance of the flow reconstructed by using the wave equation indicates that our approach is suitable whenever differential methods are needed. It is important to notice that our goal is not to overcome the state-of-the-art methods for optical flow. Differential methods lack non-local information and tend to be sparse due to high frequency dependency. We instead look for a new constraint for differential methods using the wave equation and showing that it is useful to extract reliable brightness variation spots.

Consider $I_1(x, y)$ and $I_2(x, y)$ two consecutive images in time. Formally, for each point (x, y), the optical flow is represented by a vector $[u(x, y), v(x, y)]$ that describes the movement between I_1 and I_2. In other words, let (x, y) be a point in the image I_1, then the optical flow vector associated to this point is $[u(x, y), v(x, y)]$ such that $I_1(x, y) = I_2(x + u(x, y), y + v(x, y))$. The following sections present the related works, fundamentals, the proposed method, and the experiments in order to:

- show the convergence conditions of the proposed method;
- find the best values of the method parameters;
- apply our optical flow to the human action recognition problem, using Histograms of Optical Flow, as an evaluation application;
- compare the performance of our method to the classical optical flow methods, although it is naturally sparse.

1.1. *Related works*

This section describes the related approaches to calculate optical flow, including differential (variational), hierarchical, physical model based and other methods.

1.1.1. *Differential methods*

Differential methods are based on the spatio-temporal image gradients. These methods are classified into local and global groups. Local methods assume that the flow is uniform in a pixel neighborhood. The first local optical flow method was proposed by Lucas and Kanade.[12] On the other hand, global methods, for example the classical Horn and Schunck[3] method assume that the flow is smooth over the whole image. In global methods, the flow is propagated to homogeneous regions, where the derivatives are null, therefore making it impossible to estimate the flow. On the other hand, the flow field estimated by local method tends to be more robust against noise.

Bruhn *et al.*[4] propose to combine the local Lucas and Kanade method and the global method of Horn and Schunck. The objective is to generate a dense flow robust to noise, combining the main advantages of the local and global differential methods. The method, known as *combined local-global (CLG)*, permits to gradually scatter a dense flow field. Additionally, non-quadratic and multiresolution approaches are presented. In this work, we propose an energy minimization similar to Horn and Schunck. However, our energy depends only on a small neighborhood of each point used to calculate image derivatives. Our method is thus classified as local. Furthermore, our method is sparse, but the resulting flow has high quality for the selected points.

Brox *et al.*[5] presented a differential method that computes optical flow by minimizing a functional combining a brightness constancy assumption (Sec. 2.1), a gradient constancy assumption, and a discontinuity-preserving spatio-temporal smoothness constraint. The first two constraints correspond to the data term while the third penalizes the total variation of the flow field. This approach still justifies theoretically how warping methods can be used in order to improve the performance. In our method, we proposed to minimize a functional that combines the data term of brightness constancy and a term using the wave equation energy. Our method converges only for some image pixels, which can be useful as a feature extractor.

Girosi *et al.*[14] observe that when the image brightness changes over time, its changes can be described in terms of infinitesimal deformations. Based on the Helmoltz theorem on deformable objects, they proposed four constraints to calculate optical flow. These constraints are associated to elementary deformations: rotation over the image plane, uniform expansion, and two components of shear. Combining these constraints, they obtain a general method for optical flow computation. This method uses second-order differential operators. Second-order differentials add extra smoothness constraints to the resulting optical flow and this is another motivation for using the wave equation in our work.

Recently, Rashwan *et al.*[6] proposed a differential method that adapts the data term using anisotropic stick tensor voting. This term still uses a sub-quadratic penalization function in order to make the method robust against outliers. They define a regularization tensor to complete the functional to be minimized. The directional information of this tensor, represented by its eigenvectors, is used to define the additional energy term. Additionally, they propose to introduce a weighted non-local term in order to reduce the impact on flow discontinuities. Their work shows that the energy minimization proposed by Horn and Schunck[3] is still the basis of several differential methods. Our method is based on the differential approaches, solving a simple system for each image pixel.

1.1.2. *Hierarchical (pyramidal) methods*

Bouguet[15] proposes a pyramidal implementation of Lucas and Kanade method. Initially, the pyramid is computed for each image. The flow computation starts from the coarsest pyramid level. For each upper level, the flow of the lower one is used to pre-translate the image. Thus, only a small residual flow is computed in order to adjust the displacement. This residual flow is computed using an iterative Lucas and Kanade algorithm and propagated to the upper level until the highest level is reached. In this work, a simple iterative method to locally calculate the optical flow with few iterations is given. As showed in Sec. 3, however, our energy is local and can be computed directly.

Another hierarchical method was proposed by Hwang and Lee.[16] This method starts calculating a Gaussian pyramid. They compute the flow combining the brightness constancy assumption to a inter-level motion smoothness constraint.

Since the pyramid image consists of low-pass filtered versions of the images, they consider that the flow vector in a pyramid level is the low-pass filtered version of the flow vector at the higher level. Based on this fact, the inter-level motion smoothness constraint is obtained by the difference between the flow vector and its projection from the adjacent level. The flow is calculated by a iterative method and the authors present a convergence analysis. They performed experiments in five datasets, including synthetic and real sequences. The results show their method outperforms the similar methods. In our work, we also presented a convergence criterion, but we solve the flow problem without multi-scale approach.

A problem of these hierarchical approaches was addressed by Brox and Malik:[17] small objects with very fast moving. In their work, they propose to deal with this problem by using rich descriptors in a differential optical flow method. They present a general model of energy minimization containing three terms: a data term, a gradient constraint and a smoothness constraint. In addition, they propose to add two other constraints to the differential model: a term of descriptor matching and a term of point correspondences from descriptor matching. An initial guess of the descriptor matching term is determined separately. The authors perform tests with known descriptors, such as HOG and SIFT. Once the initial guess of the descriptor matching term is determined, the remaining terms are minimized and the flow is obtained. The authors see the possibility of application of their method in the action recognition using HOF. This is also an important application used in our work. Furthermore, our method does not need to extract features from the images, making our pre-processing very simple.

Tu *et al.*[18] proposed a variational method using combined post-filtering to improve the accuracy. They extract flow edges on a structure tensor (ST) and extend the Harris edge detector into a 3D spatial-scale detector to improve the detection. To preserve discontinuities they use a gradient bilateral filter (GBF). However, to improve the computational efficiency, they combine the GBF to a Gaussian filter, using GBF to smooth discontinuity regions and the Gaussian filter is applied to non-discontinuity regions. Finally, they perform a post-filtering using a weighted median filter (WMF) to the detected edges and a bilateral filter (BF) to handle occlusions. As compared to Brox and Malik,[17] our method seek simplicity, estimating the flow with no need to perform several pre- and post-filtering.

Another multi-scale variational method was presented by Tu *et al.*[19] Their proposed to fuse flows achieved by two methods: a nearest neighbor field (NN-field) and a combined post-filtering (CPF) algorithm. The flows are combined by a weighted local intensity fusion. Thus, they estimate the smoothness parameter and treat large displacements. Furthermore, they correct the conflict between the color and motion similarity, using a weighted median filter (WMF) based on the occlusion rate, the color and spatial distances. They performed experiments on two datasets and the achieved results demonstrated the method effectiveness for including large displacements and small objects. They apply several constraints and filtering to refine the flow. As previously mentioned, our method seeks to be as simple as possible.

1.1.3. *Physical model based methods*

Haussecker and Fleet[20] propose to exploit physical models of time-varying brightness to compute optical flow. They consider that the temporal brightness variation can be specified by differential equations for a given physical model. Then, the objective is to estimate the parameters of the optical flow field and a set of parameters of the physical model. They propose formulations for some transport models, for example, diffusion. In our work, we propose the use of a well-known physical model for wave phenomena as an extra restriction to estimate the optical flow. Even if, in general, the overall image brightness does not change as a wave propagation, the convergence conditions for our method naturally reject points whose local brightness variation does not behave as if a wave is passing through them. The points for which our method converges can thus be assumed as feature points, an use that needs further investigation.

The work of Sakaino[21] proposes to estimate fluid flow based on the physical properties of waves. The method exploits three properties of transport present in fluid simulation: convection, diffusion and advection. Based on a wave generation model, the method uses an objective function that estimates two optical flow components and five wave-related parameters that are the two wavenumber components, frequency, amplitude, and orientation. This function is minimized in order to calculate the seven variables. This method is suitable for images of water waves, cloud and smoke. In a different way, we propose to use wave properties in order to calculate the optical flow for general images.

1.1.4. *Other approaches*

Barnard and Thompson[22] propose a method for matching images and computing the differences between them. Initially, some interest points are selected separately from the two images. Then, for each point, a set of labels of possible matches is constructed. The algorithm determines a probability associated with each label based on the sum of the squares of the differences between a small window centered on a point in the first image and the possible correspondent one on the second image. After the probabilities are calculated, the point with highest probability is defined as the correspondent point. The displacement between the two points will be the optical flow. This method presents a high computational cost because it performs many distance calculations and comparisons. We proposed low cost method without extracting correspondences between the images.

In phase-based methods, the phase behavior of band-pass filter outputs defines the velocity. Fleet and Jepson[23] proposed the first phase based method. Initially, they represent the images by a set of shift-invariant filters. These filters are tuned so that their amplitude spectrum concentrates around the appropriate line in frequency space. They argue that the phase component of the filters response is better to approximate the velocities field. The optical flow vectors are expressed by first-order temporal derivative of surfaces of constant phase. The authors emphasize that their method does not depend on previous element detection from the images. This is also a characteristic of our method, but proceeding in space-time domain.

Fortun *et al.*[24] presented a survey of optical flow estimation. They described the main data cost used to complement the constancy brightness constraint, the parametric approaches and the regularized models. Some methods that handle occlusions were presented, including occlusion detection and filling. The survey also presented methods combining feature matching and optical flow. The authors conclude the paper appointing the main challenges of optical flow computation such as handling large displacements and occlusions, and to reduce the computational cost.

Sun *et al.*[25] presented a convolutional neural network (CNN) model for optical flow estimation. They use a pyramidal feature extractor to achieve a feature representation of the images. The features between the images are warped using an upsampled flow. The features are then used to construct a volume cost, which is used by a multi-layer CNN to estimate the optical flow. They performed experiments in two datasets and compare the results to another methods using CNN. The achieved a similar performance to the compared methods using a more compact model. In our work, we do not extract features and not apply machine learning strategies.

Another model based on CNN was presented by Hui *et al.*[26] They use two subnetworks: NetC, which performs the feature extraction and warp them, and the NetE, which estimates the flow by a descriptor matching followed by a sub-pixel refinement. A regularization flow layer remove artifacts and the vague flow boundaries. They evaluated the method performance in five datasets and compare the results state-of-the-art methods, including conventional and CNN based. Their method present similar performance using a smaller model and spending less time. Our work presented a variational method, which does not apply CNN or other learning approaches to flow estimation.

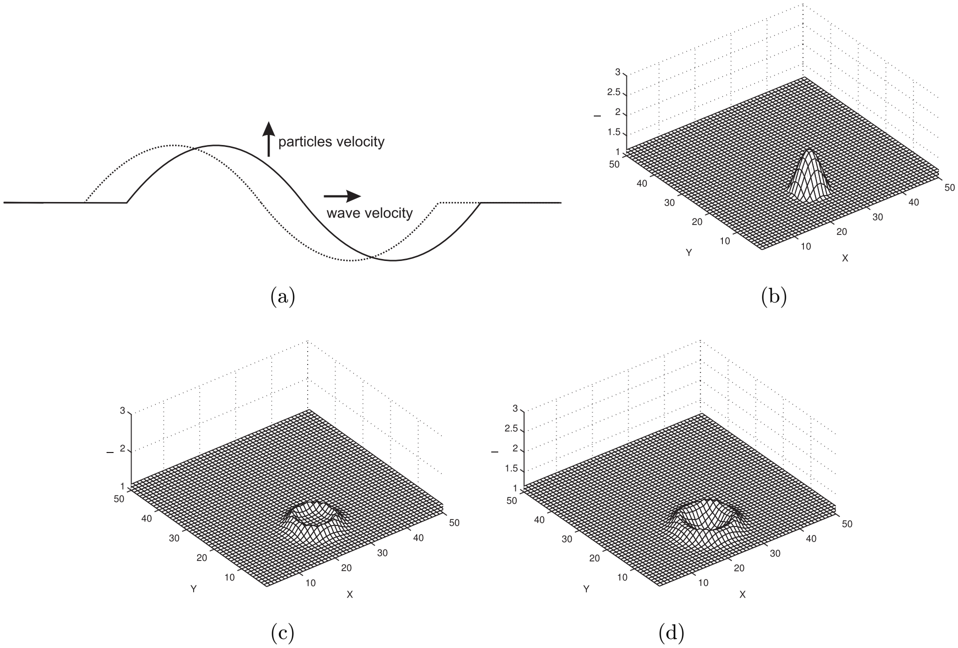
Chen *et al.*[27] proposed a framework to optical flow estimation based on filtering. They derived the model from the variational model using a data term and a regularization term. For the data term, they used a weighted quadratic penalization based on brightness constancy assumption. The regularization was a constrained minimization representing the interrelationship between the flow vectors to the local neighboring. This framework was used by designing new algorithms by employing kernels that represents certain motion constraints. They analyzed alternatives for the regularization term based on general filtering and a 3D filtering. The framework performance was evaluated in three datasets and compared to the state-of-the-art algorithms. The achieved results showed their method outperforms classical state-of-art-methods. Their method presents a strong relationship among pixels. Our approach, on the contrary, tries to decouple the pixels to allow the method parallelization.

Zhang *et al.*[28] proposed a non-local total variation (TV) method using a L^1 norm for smoothing term. They improved the $TV - L^1$ formulation by a self-adaptive weight based on a function of the image gradient along scale parameters. Furthermore, they detected occlusion by a regular triangulation. The occlusion information is used to optimize the computed flow field by weighted median filter that uses an occlusion factor. They evaluated the approach in two datasets and compare with state-of-the-art methods. They claim their results presented a higher accuracy and robustness.

2. Wave Equation

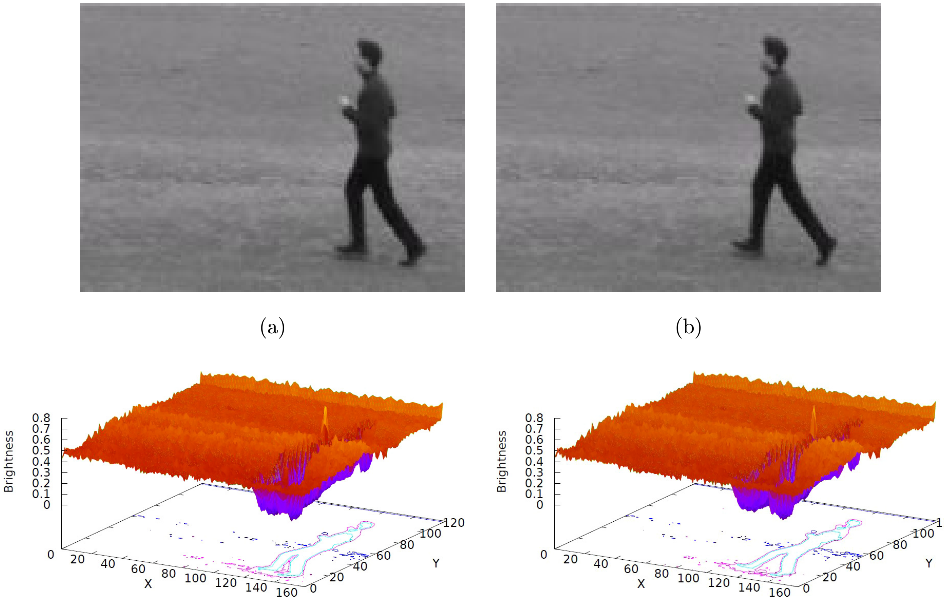
In general, waves are caused by a disturbance in a medium. For example, a pulse traveling on a cord can be formed by a quick up-and-down motion of the hand.[29] This pulse travels along the cord moving its particles vertically. This way, we have two velocities associated to a wave propagation: the velocity of the wave along the cord (or another mean) and the velocity of its forming particles. Figure 1(a) represents a wave propagating and its velocities.

An example of two-dimensional waves in the nature is the tsunamis which are water waves that can be caused by earthquakes, volcanic eruptions and other underwater explosions.[30] These waves travel over the water surface causing elevation. Figure 1 shows a simulation of a 2D water wave. We can see three moments of the



(a) (b) (c) (d)

Fig. 1. Examples of wave propagation. (a) 1D wave propagation. (b)-(d) three moments of a 2D wave propagation.



(a) (b)

(c) (d)

Fig. 2. Images (a) and (b) and their respective surface representations (c) and (d).

wave propagation: the moment which the disturbance happens (Fig. 1(b)) and two moments of the propagation along the water surface (Figs. 1(c) and 1(d)).

In this work, we propose to compare the motion in image sequences to a two-dimensional wave. An image can be represented by a surface where the heights of the points are the brightness intensity. Assuming image brightness constancy, we consider that a wave passing along the image plane causes the temporal brightness variation. Figure 2 shows two successive images and their surface representation. On the horizontal plane we can see the contours of the images from which the inference of propagation is of main interest. The premise of this work is that these contours might be modeled as waves traveling in the image.

Mathematically, the phenomena involving the propagation of waves in a continuous medium are described by the wave equation, which is important in mechanics, acoustics, and fluid dynamics.[13] It has the general form:

$$\frac{\partial^2 I}{\partial t^2} = c^2 \nabla^2 I, \tag{1}$$

where ∇^2 represents the spatial Laplacian and the constant c represents the magnitude of the propagation velocity. I is a scalar function that represents the vertical position of the particles in function of the time.

In the 2D case, the equation has the following shape:

$$\frac{\partial^2 I}{\partial t^2} = c^2\left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}\right). \tag{2}$$

The first solution of (1) in one dimension was proposed by d'Alembert.[31] He solved the problem of initial value, where the initial vertical position of the particles is $I(x,0) = I^{(0)}(x)$ and the initial velocity is $I_t(x,0) = I_t^{(0)}(x)$. The solution obtained was

$$I(x,t) = \frac{1}{2}[I^{(0)}(x-ct) + I^{(0)}(x+ct)] + \frac{1}{2c}\int_{x-ct}^{x+ct} I_t^{(0)}(s)ds.$$

The solution of d'Alembert calculates the position of each medium particle along time. In other words, the function I is the unknown of the differential equation. To obtain the values of the function I it is necessary to know the velocity of propagation c, which is not necessarily constant along the medium, and initial and/or contour conditions are needed.

2.1. *Combining the wave equation with the brightness constancy constraint*

In our problem, two consecutive images represent *two distinct realizations of the medium I in time*. In this case, the unknown is the velocity c of propagation in each point. The solution of this problem is a realization of brightness transport between the two images and might represent an optical flow, also restricted by the aperture problem.

Waves transport energy from one place to another.[29] The energy is transferred from one particle to the neighbor particle in the medium when the wave is traveling. The energy of the two-dimensional wave equation (2) is given by Myint and Debnath[13]:

$$E(t) = \frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I_t^2 + c^2(I_x^2 + I_y^2)dxdy. \tag{3}$$

Generally, the 2D wave equation defines a vertical displacement of particles in the direction perpendicular to the wave plane.[32,33] This is interesting for optical flow computation because most methods try to find a displacement vector to compensate the local brightness variation I_t. Here, the I_t plays the role of how much mass is locally transferred by the passing wave. In an energy minimization model, (3) has also the function of compensating the brightness variation.

The major contribution of this work is to combine the above second-order constraint originated from the wave equation with the classic first-order constraint:

$$I(x,y,t) \approx I(x,y,t) + I_x u + I_y v + I_t,$$

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$. It comes from the Taylor expansion of the *brightness constancy constraint* equation

$$I(x, y, t) = I(x + u(x, y), y + v(x, y), t + \Delta t),$$

where $I(x, y, t)$ is the brightness intensity of the pixel (x, y) at the time t. The vector $\mathbf{v} = [u(x, y), v(x, y)]$ is the optical flow vector and Δt is the time variation. We assume $\Delta t = 1$ since it is the displacement between two consecutive frames. In order to simplify the notation, from now we assume $\mathbf{v} = [u(x, y), v(x, y)]$ as $\mathbf{v} = [u, v]$.

3. Proposed Method

In Sec. 2, we have presented the wave equation and its associated energy. We start by defining the square of the velocity of propagation c in (3). This is a crucial step of our method and the proposed velocity is an important contribution of our work. We need to find velocities that minimize the energy of the wave, yielding vectors whose magnitudes are the lowest possible. Denoting the inner product as $\langle \cdot, \cdot \rangle$, we define:

$$H = \langle [u, v], [u, v] \rangle,$$

which represents the squared norms of all unknown 2D velocities $[u, v]$.

We propose a wave velocity that considers the variation of the square of the norm of the image velocity vectors. The reason is that minimizing the variation of the squared norm favors constant actual velocities. We also propose this to decouple the contributions of the x and y velocities. The velocity of propagation of the wave is then defined as the spatial gradient of H, resulting in the energy

$$E_w = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{2}\left[I_t^2 + \nabla H \cdot (I_x^2 + I_y^2)\right]\, dxdy, \tag{4}$$

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$. Note that this empirical approach implies separated energies for x and y.

Horn and Shunck used the optical flow constraint $I_x u + I_y v + I_t = 0$ as a first-order term of the energy. This term is defined as:

$$E_d = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2\, dxdy. \tag{5}$$

Combining (5) and (4), the complete proposed energy is:

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(I_x u + I_y v + I_t\right)^2 + \alpha\left[\frac{1}{2}\left(I_t^2 + \nabla H \cdot \left(I_x^2 + I_y^2\right)\right)\right] dxdy, \tag{6}$$

where α is a weight to control the influence of the energy of the wave equation. This is similar to what Horn and Shunck proposed to control the smoothness constraint.[3] As shown below, however, the effect of α in our method is not related to a global smoothing process since our solution is locally defined.

Calculating the partial derivatives of H, we have

$$\frac{\partial H}{\partial x} = 2\left\langle [u,v], \frac{\partial [u,v]}{\partial x} \right\rangle = 2(uu_x + vv_x),$$
$$\frac{\partial H}{\partial y} = 2\left\langle [u,v], \frac{\partial [u,v]}{\partial y} \right\rangle = 2(uu_y + vv_y),$$

where $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $v_x = \frac{\partial v}{\partial x}$ and $v_y = \frac{\partial v}{\partial y}$.

Introducing the 2D velocity in the energy, we obtain naturally decoupled energy equations for x and y. Separating x and y components of (6), we have two energy terms:

$$\begin{aligned} E_x = &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 \\ &+ \alpha\left[\frac{1}{2}(I_t^2 + 2(uu_x + vv_x)(I_x^2 + I_y^2))\right] dxdy, \end{aligned} \tag{7}$$

$$\begin{aligned} E_y = &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 \\ &+ \alpha\left[\frac{1}{2}(I_t^2 + 2(uu_y + vv_y)(I_x^2 + I_y^2))\right] dxdy. \end{aligned} \tag{8}$$

The total energy to be minimized is the sum of (7) and (8):

$$E_{\text{total}} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} 2(I_x u + I_y v + I_t)^2 + \alpha h[I_t^2 + (I_x^2 + I_y^2)]\ dxdy, \tag{9}$$

where $h = uu_x + vv_x + uu_y + vv_y$. This equation defines the energy of waves traveling along the image sequence and obeying the constraint of brightness constancy. We need to find the optical flow vector $[u,v]$ that minimizes this equation.

In order to minimize (9), we apply the Euler–Lagrange equations[34]:

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial u_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial L}{\partial u_y}\right) = 0, \tag{10}$$

$$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial v_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial L}{\partial v_y}\right) = 0, \tag{11}$$

where L is the functional to be minimized.

Applying the Euler–Lagrange equation (11) to (9), we have:

$$(2I_x^2 - \alpha(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y))u + 2I_xI_yv + 2I_xI_t = 0, \tag{12}$$

where $I_{xx} = \frac{\partial^2 I}{\partial x^2}$, $I_{yy} = \frac{\partial^2 I}{\partial y^2}$ and $I_{xy} = \frac{\partial^2 I}{\partial x \partial y} = \frac{\partial^2 I}{\partial y \partial x}$.

Similarly, applying the Euler–Lagrange equation (11) to (9), we also have:

$$(2I_y^2 - \alpha(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y))v + 2I_xI_yu + 2I_xI_t = 0. \tag{13}$$

Therefore, we find the linear system whose unique solution finds the flow vectors $[u, v]$ for each pixel that minimizes our energy (9):

$$\begin{cases}(2I_x^2 - \alpha\bar{I})u + 2I_xI_yv + 2I_xI_t = 0,\\ 2I_xI_yu + (2I_y^2 - \alpha\bar{I})v + 2I_yI_t = 0,\end{cases}$$

where $\bar{I} = I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y$.

Notice that, at each pixel, the flow does not depend on the neighbors. Thus, for each image point, we have the following system:

$$\begin{bmatrix} D_u & 2I_xI_y \\ 2I_xI_y & D_v \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -2I_xI_t \\ -2I_yI_t \end{bmatrix}, \tag{14}$$

where

$$D_u = 2I_x^2 - \alpha\bar{I},$$
$$D_v = 2I_y^2 - \alpha\bar{I}.$$

The system (14) can be solved by a direct or an iterative method. Using the iterative Jacobi method, we have:

$$u^{(k+1)} = \frac{-2I_xI_yv^{(k)} - 2I_xI_t}{D_u}, \tag{15}$$

$$v^{(k+1)} = \frac{-2I_xI_yu^{(k)} - 2I_yI_t}{D_v}, \tag{16}$$

where $[u^{(k)}, v^{(k)}]$ is the optical flow at the iteration k.

One may note in the iteration equations that the flow vector at each pixel depends just on itself at the previous iteration. Considering one iteration at position (x, y), the component $u^{(k+1)}$ depends only on $v^{(k)}$ and $v^{(k+1)}$ depends only on $u^{(k)}$. In other words, the flow at a point does not depend immediately on its neighbors. This ensures quick convergence and makes the method suitable for parallelization. This is especially interesting for use in modern GPUs. The individual pixels can be easily grouped to take advantage of stream processors.

However, Jacobi method does not converge for all the image points. A convergence condition is necessary as exposed in Sec. 3.3.

3.1. *Direct method*

In addition to the iterative solution, we can solve the system (14) by a direct method. For each point, we have the matrix:

$$\mathrm{A} = \begin{bmatrix} D_u & 2I_xI_y \\ 2I_xI_y & D_v \end{bmatrix},$$

whose inverse is

$$\mathrm{A}^{-1} = \frac{1}{\det(\mathrm{A})} \begin{bmatrix} D_v & -2I_xI_y \\ -2I_xI_y & D_u \end{bmatrix}.$$

Therefore, one may calculate the flow directly by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathrm{A}^{-1} \begin{bmatrix} -2I_xI_t \\ -2I_yI_t \end{bmatrix} \tag{17}$$

It is important to note that (17) can be solved just at the points where $\det(\mathrm{A}) \neq 0$. Similarly to the iterative method, in the direct solution some calculated vectors present high magnitudes and all vectors bigger than seven pixels are set to null. With the direct approach is easier to see how the method can be parallelized in a straight forward way. It requires only an order 2 matrix inversion followed by a matrix vector multiplication. Since the derivatives are computed by constant masks through convolution, our method is suitable for GPU parallelization, for instance.

3.2. *Discretization*

Differential methods extract motion information based on instantaneous variations in the image. An important issue for these methods is the computation of these variations. Since the image domain is not continuous, a discretization is necessary.

The finite difference method is very useful to solve differential equations numerically. In order to reduce time costs, the derivative elements of images are often estimated by the convolution of high-pass linear and shift-invariant filters. It is important to choose appropriate filters to reduce the effect of noise. In order to make our work reproducible, we present here the steps to calculate the derivatives:

- Gaussian filtering of each input image using the impulse response [0.006 0.061 0.242 0.383 0.242 0.061 0.006] in the directions X and Y;
- $\frac{\partial I}{\partial x}$:

— convolution of the filtered input image by the low-pass impulse response [0.5 0.5] in the Y-direction;

— convolution of the resulting image by the high-pass filter [−0.5 0.5] in the X-direction.

- $\frac{\partial I}{\partial y}$:

— convolution of the filtered input image by the low-pass impulse response [0.5 0.5] in the X-direction;

— convolution of the resulting image by the high-pass filter [−0.5 0.5] in the Y-direction.

- $\frac{\partial I}{\partial t}$:

— convolution of the filtered input image by the low-pass impulse response [0.5 0.5] in the X and Y-directions;
— convolution of the resulting image by the high-pass filter [-0.5 0.5] in the T-direction.

- $\frac{\partial^2 I}{\partial x^2}$:

— convolution of the filtered input image by the low-pass impulse response [0.25 0. 5 0.25] in the Y-direction;
— convolution of the resulting image by the high-pass filter [0.25 -0.5 0.25] in the X-direction.

- $\frac{\partial^2 I}{\partial y^2}$:

— convolution of the filtered input image by the low-pass impulse response [0.25 0.5 0.25] in the X-direction;
– convolution of the resulting image by the high-pass filter [0.25 -0.5 0.25] in the Y-direction.

- $\frac{\partial^2 I}{\partial x \partial y}$:

— convolution of the filtered input image by the high-pass filter [0.25 -0.5 0.25] in the X- and Y-directions.

The discretization above presented good qualitative and quantitative results as shown in Sec. 4. Note that the calculation of $\frac{\partial I}{\partial t}$ is the unique having filters applied in time. Low-pass filters are applied in X- and Y-directions which are both orthogonal to T. This means that the derivative filter for $\frac{\partial I}{\partial t}$ has a 3D impulse response. For all other derivatives, however, there is no low-pass filtering in T direction, resulting in 2D filter masks. The unity of the independent variable T (seconds) is different from the unity of X and Y (meters), making it hard to match their magnitudes. Furthermore, 3D masks would result in more computational cost.

With the derivatives computed, the convergence conditions (18), (19) and (22) are tested for each pixel. For the points where the convergence conditions are satisfied, the iterative system (15) and (16) is solved. The proposed method is thus classified as local and iterative. It can be calculated quickly because of the decoupling of the pixels in the system solution. In some pixels, the vectors converge to high magnitudes. This is due to noise and other inconsistencies and can be safely discarded since differential methods by definition cannot detect long displacements. In this work, all vectors bigger than seven pixels are just set to null. The value of seven pixels was defined based on the assumption that such a displacement is unlike to be captured by differential methods.

3.3. *Convergence condition*

Motion behavior of wave propagation is locally smooth and nonrigid. Such nature of the energy leads to the fact that the minimization process is sensitive to both the motion boundaries and purely flat region. Thus, some conditions are necessary in order to ensure convergence of (15) and (16). A simple condition is that the denominator of the equations must not be 0. Therefore, we must have:

$$D_u \neq 0, \tag{18}$$

$$D_v \neq 0. \tag{19}$$

Except above conditions, the Jacobi method (and any iterative method) converges if and only if the spectral radius of the iteration matrix is strictly less than one. In other words, if the absolute value of each eigenvalue of the iteration matrix is smaller than the unity. As each point does not depend on the neighbors, we have an iterative system for each point defined by:

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} = \begin{bmatrix} 0 & -\dfrac{2I_xI_y}{D_u} \\ -\dfrac{2I_xI_y}{D_v} & 0 \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} + \begin{bmatrix} -\dfrac{2I_xI_t}{D_u} \\ -\dfrac{2I_yI_t}{D_v} \end{bmatrix}, \tag{20}$$

from which we obtain the iteration matrix:

$$\mathrm{J} = \begin{bmatrix} 0 & -\dfrac{2I_xI_y}{D_u} \\ -\dfrac{2I_xI_y}{D_v} & 0 \end{bmatrix}.$$

We can obtain the eigenvalues of J by finding the roots of the equation:

$$\det(\mathrm{J} - \lambda \mathrm{I}_2) = 0, \tag{21}$$

where $\det(\cdot)$ denotes the determinant and I_2 is the 2×2 identity matrix.

Expanding (21), we have

$$\det(\mathrm{J} - \lambda \mathrm{I}_2) = \lambda^2 - \frac{4I_x^2I_y^2}{D_uD_v} = 0,$$

whose roots are

$$\lambda = \pm \frac{2I_xI_y}{\sqrt{D_uD_v}}.$$

We can observe that when $D_uD_v < 0$, we have complex values for λ. However, for our convergence condition, we desire λ values such that their absolute values are less than one. Thus, we have the following convergence condition:

$$\frac{|2I_x I_y|}{\sqrt{|D_u D_v|}} < 1, \tag{22}$$

where $|\cdot|$ denotes the absolute value.

We can compute the optical flow vector $[u, v]$ only at points where the conditions expressed by (18), (19) and (22) are satisfied. It is important to note that solving the system (14) by a direct method, we also need to exclude some points in order to avoid division by 0.

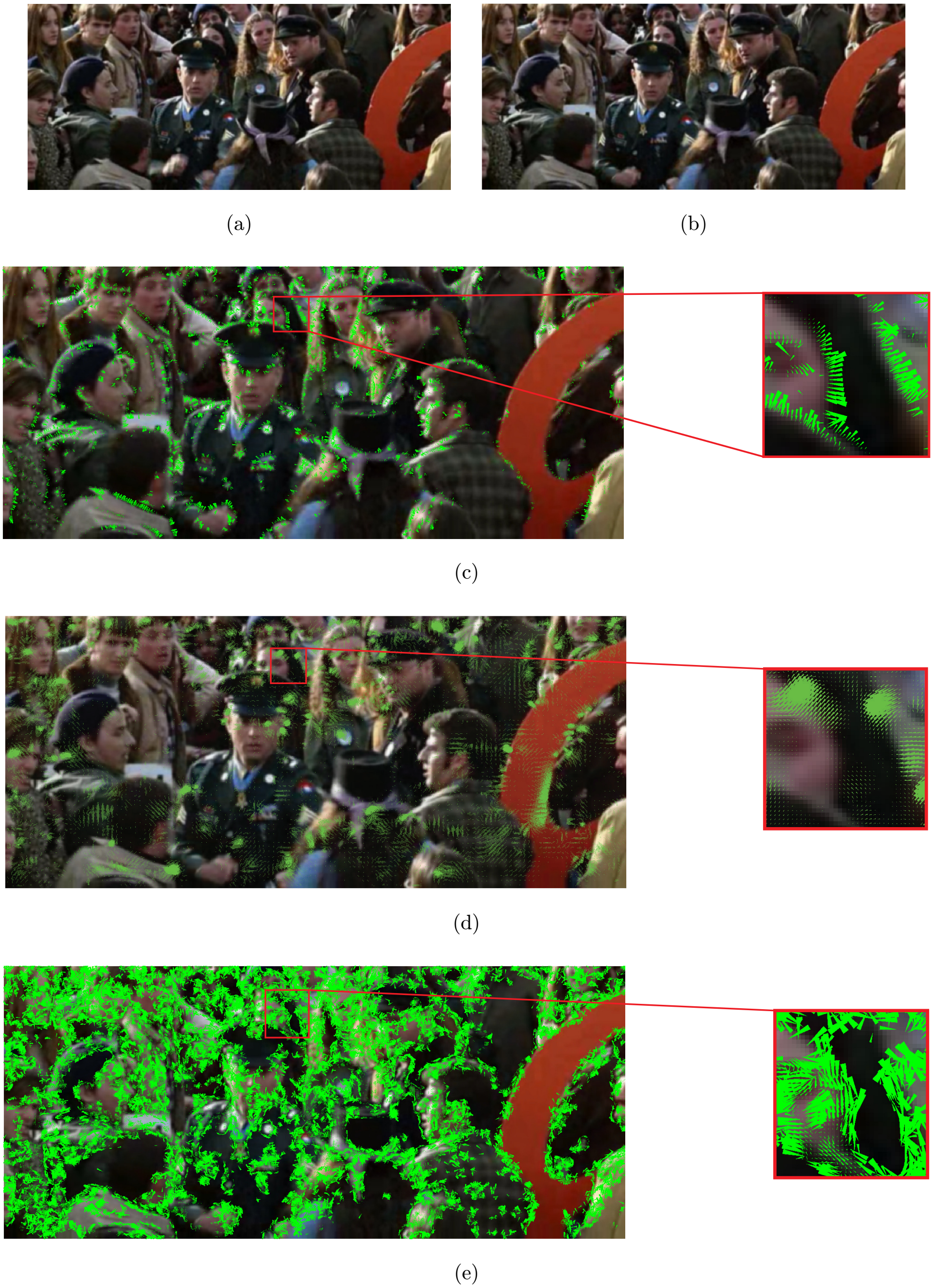
4. Experimental Results

Some parameters were selected by using the human action recognition problem. The video classification protocol for the KTH dataset is described in Sec. 4.4. In Sec. 3.2, we described our discretization to compute the derivatives of the images. The first-order derivatives in this scheme are computed by non-centered finite difference given by the low-pass and high-pass filters $([0.5\ 0.5], [-0.5\ 0.5])$, whose the correspondent second-order filters are given by $([0.25\ 0.5\ 0.25], [0.25\ -0.5\ 0.25])$. In order to investigate if a centered finite difference is better than the non-centered one to estimate the image gradient, we performed tests using the masks $([0.25\ 0.5\ 0.25]$, $[-0.5\ 0\ 0.5])$, whose respective second-order centered filters are given by $([0.0625\ 0.25\ 0.375\ 0.25\ 0.0625], [0.25\ 0.0\ -0.5\ 0.0\ 0.25])$. The centered version, however, achieved only 84.1% of recognition, against 87.8% of the non-centered version, and we adopted the discretization described in Sec. 3.2, with the non-centered masks.

In Sec. 3, we presented the direct and the iterative version of our optical flow differential method. In our experiments, the best results for human action recognition were achieved using the iterative algorithm, which reached 87.8% of recognition against 85.2% of the direct approach. Perhaps this is due to numerical errors of the direct solution but it needs further investigation. Despite the small gain due to the use of the iterative method, we chose to use this version in all experiments.

Furthermore, we performed tests without the initial Gaussian filtering of the images and without the low-pass filtering in the orthogonal directions before computing the derivatives. Our best result without the orthogonal low-pass filtering was 85.1%. The tests without the initial Gaussian filtering resulted only in 81.6% of recognition. Consequently, we apply both filterings in all experiments.

The main parameter of our optical flow method is the weight α used to control the influence of the wave energy term in our functional. The brightness constancy constraint will be the dominant term if α is low. On the other hand, a high α leads to local brightness being less conserved by the resulting flow. We performed tests varying α values in the range of $[0.1,\ 10]$ and the best results for the human action recognition problem using our setup is $\alpha = 1.0$.



(a)

(b)

(c)

(d)

(e)

Fig. 3. Optical flow detected by our method. (a) First image. (b) Second image. Optical flow calculated using our method (c), with Horn and Schunck method (d), and using Lucas and Kanade method (e).

Our iterative method presents a quick convergence and, consequently, only a few iterations are needed per pixel to obtain a flow estimation. The decoupling of the pixels in the system solution also helps a fast overall flow computation. We used 10 iterations in all experiments. The same number of iterations was used for the Horn and Schunk method.

4.1. *Qualitative comparison*

A simple representation of optical flow is a vector field over the image plane. Figure 3(c) shows this representation for a field flow computed by our method. By visually inspecting the scene (Figs. 3(a) and 3(b)), one may see that the person moves to the right side. The whole scene presents objects moving in several directions. Due to the convergence constraints, the method only computes the vectors at the points where the image derivatives are not null and (22) is satisfied. The flow was computed with $\alpha = 1.0$ for the wave energy. Despite the result is sparse, our flow is fairly aligned with the moving edges.

Figure 3(d) shows the flow computed by the Horn and Schunck's method. This method is global and generates a flow denser than our approach. Note that the flow is smoother, spreading the motion information detected at strong gradient areas. In other words, the motion detected on borders tends to influence the surrounding pixels, depending on the smoothing factor. The flow shown in Fig. 3(d) was computed with the smoothness weight $\alpha = 2.0$. This value is the best in our tests on human action recognition (Sec. 4.4).

In Fig. 3(e), we show the result of Lucas and Kanade's method. The flow vectors estimated are strongly aligned to the image edges. The flow is denser than ours in this example. Note, however, that the displacement vectors have somewhat conflicting directions. Some of them are following the face to hair contour. The density and the

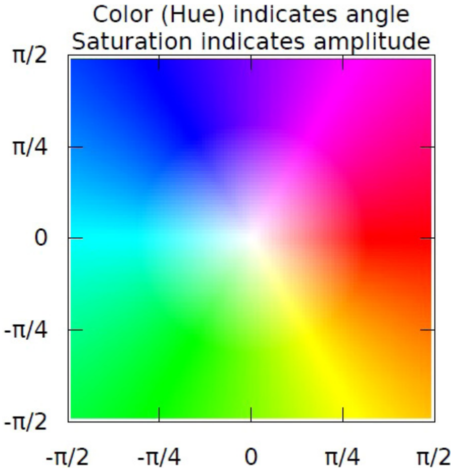
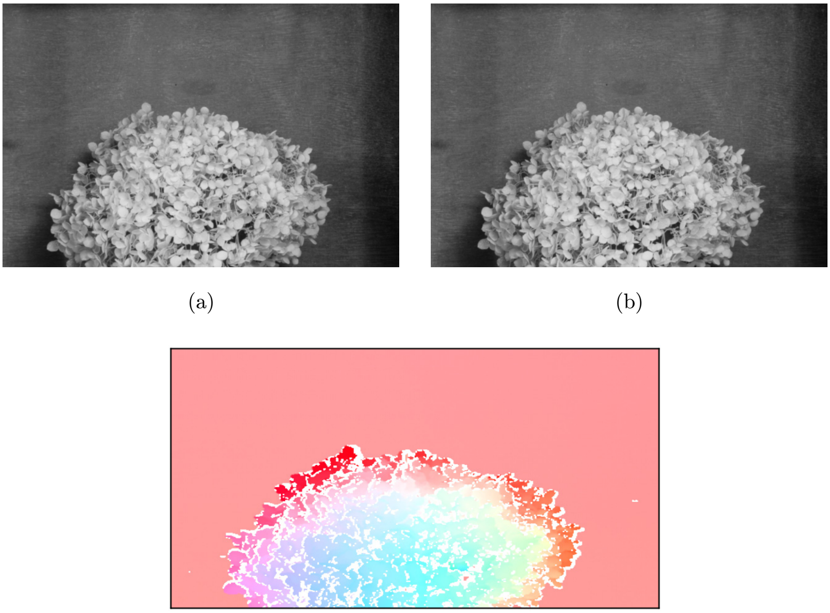


Fig. 4. Color coding for optical flow visualization.

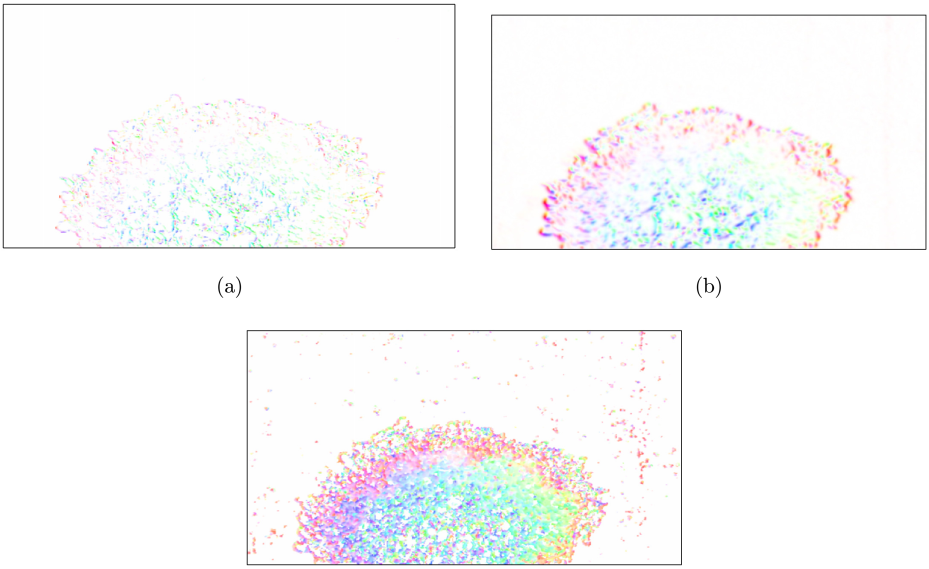


(a)

(b)

(c)

Fig. 5. Example of ground truth in color coding. (a) First image. (b) Second image. (c) Ground truth.



(a)

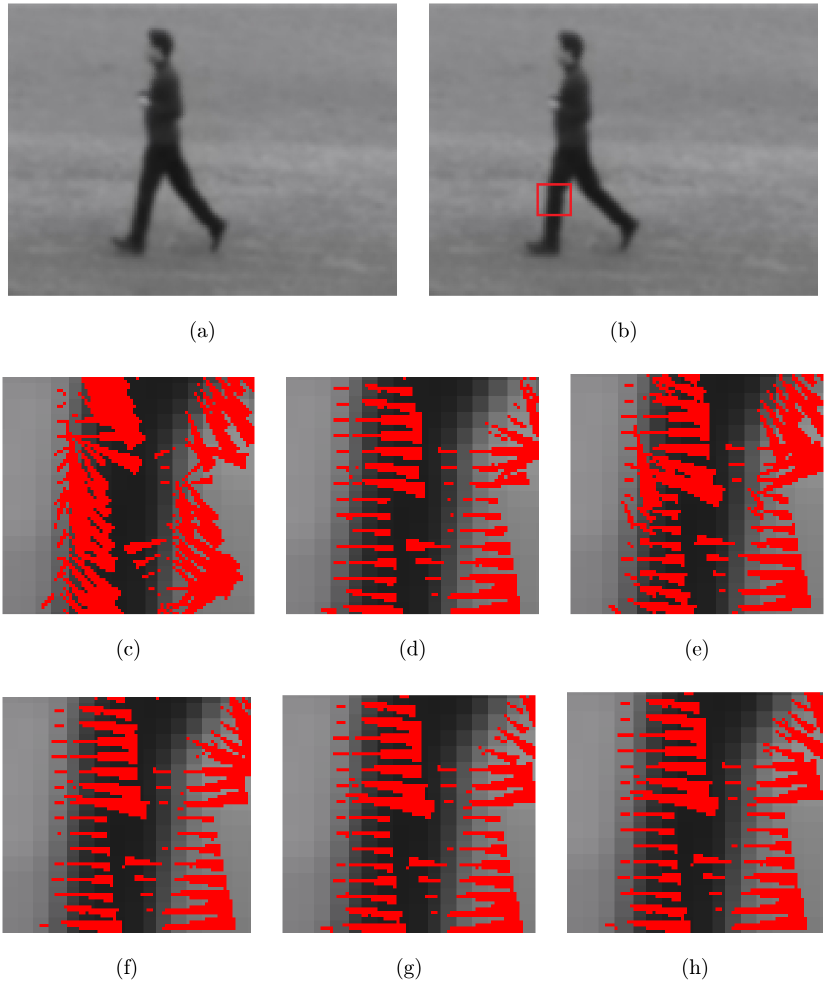
(b)

(c)

Fig. 6. Optical flow computed for Hydrangea sequence. (a) Our method. Note that the method does not converge for the homogeneous regions. (b) Horn and Schunk. (c) Lucas and Kanade.

flow quality depend on the window used. We used a window of 5 × 5 elements, the same used by Lucas and Kanade. The Lucas and Kanade's method can give vectors with high magnitudes due to discontinuities in the image and its partial derivatives. The vectors greater than seven pixels are set to null, in order to be fairly compared with our results.

Baker *et al.*[1] propose a color coding manner to represent optical flow. In their scheme, each direction is represented by a color hue and the flow norm is represented by the saturation. Figure 4 shows this color coding, which is used in several recent works in optical flow.



(a) (b)

(c) (d) (e)

(f) (g) (h)

Fig. 7. Evolution of our flow for a region of an image sequence. (a) First image. (b) Second image. (c) 1 iteration. (d) 2 iterations. (e) 3 iterations. (f) 4 iterations. (g) 5 iterations. (h) 10 iterations.

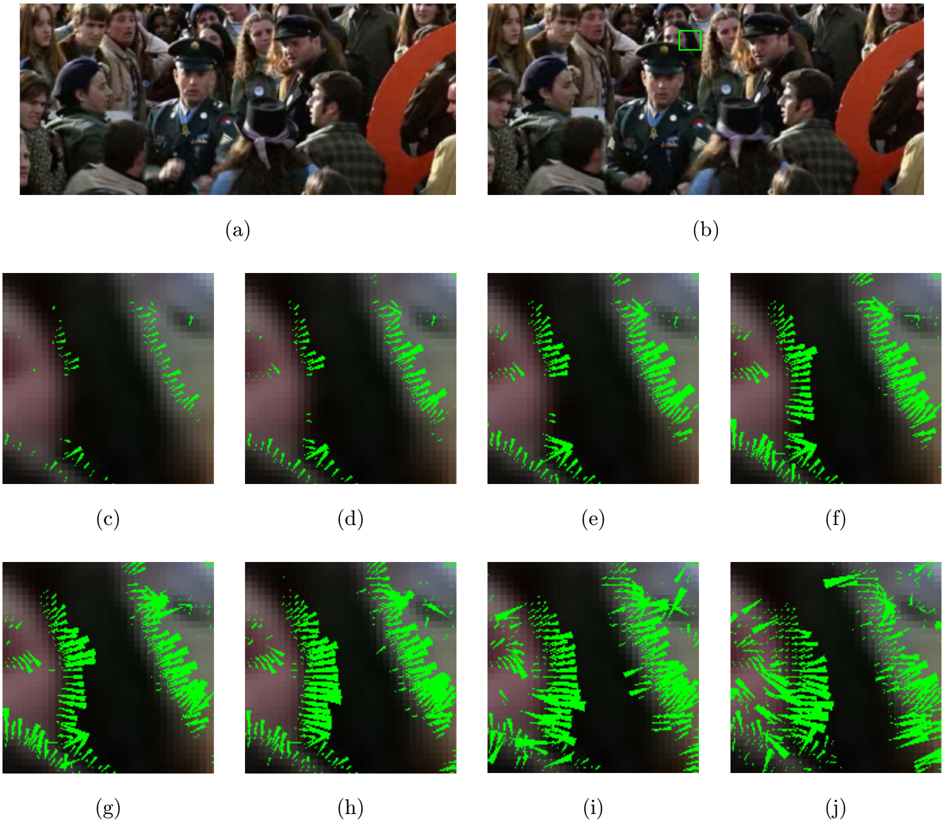


Fig. 8. Flow computed for different α values. (a) First image. (b) Second image. (c) $\alpha = 0.1$. (d) $\alpha = 0.2$. (e) $\alpha = 0.4$. (f) $\alpha = 0.8$. (g) $\alpha = 1.0$. (h) $\alpha = 2.0$. (i) $\alpha = 4.0$. (j) $\alpha = 10.0$.

They also provide a dataset to evaluate optical flow algorithms, including ground truths for some image sequences. As an example, the image sequence *Hydrangea* and the respective ground truth are presented in Fig. 5.

The flow for Hydrangea sequence using our method is showen in Fig. 6(a). Although our flow is not dense, one may note color similarity between our flow and the ground truth (Fig. 5). Our method was capable to extract some directional information from the regions having high brightness variation. Note that the homogeneous background is completely discarded by the convergence constraints.

Figures 6(b) and 6(c) show the flow of the Hydrangea sequence by using the classical methods. Observe that they compute a denser flow field. The background is not detected since the image derivatives are null in this region. Lucas and Kanade's approach presents a more intense flow for this example. It is possible to note color similarity among the results of the classical methods and our flow.

4.2. *Influence of the method parameters on the results*

In this section, we analyze the influence of the number of iterations and the weight α on our method performance. In Sec. 3, we showed that the flow in an iteration only depends on its own value in the previous iteration. Figure 7 shows an image sequence and the flow computed for a region with 1, 2, 3, 4, 5, and 10 iterations. One may observe that the flow is stable after four iterations. Due to the decoupling of the pixels in the system solution, our method converges quickly. The flow at a point does not depend on its neighbors.

Similar to Horn and Schunck,[3] our method has a parameter α to control the influence of the wave equation energy. We observed that low values of α (around 0.1) hinders the convergence for several points, with arbitrary outcomes. On the other hand, a high α reduces the energy of the data term and the resulting optical flow tend to be inconsistent in some regions, notably far from the edges (Fig. 8).

4.3. *Quantitative comparison*

Baker *et al.*[2] propose performance measures for optical flow. In recent years, these measures have been used to evaluate the results of new optical flow methods.[7,6] In our experiments, we used two of them.

The first measure is the *angular error*, which is the angle between the flow vector $[u, v]$ and the ground truth vector $[u_{GT}, v_{GT}]$ in the three-dimensional space. The third coordinate is set to 1.0. Therefore, the angular error is defined as

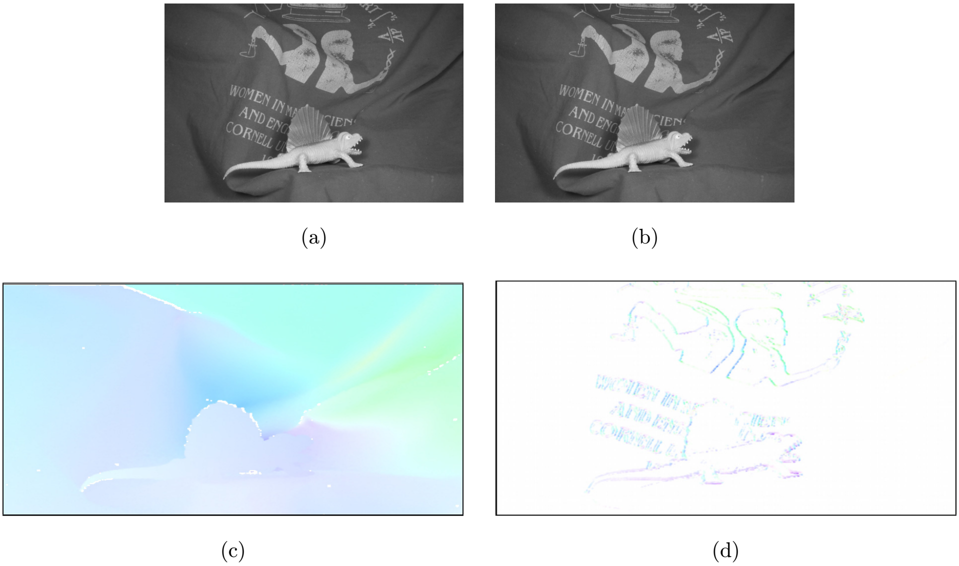

(a)
(b)
(c)
(d)

Fig. 9. Flow computed for Dimetrodon sequence by our method. (a) First frame. (b) Second frame. (c) Ground truth. (d) Computed flow.

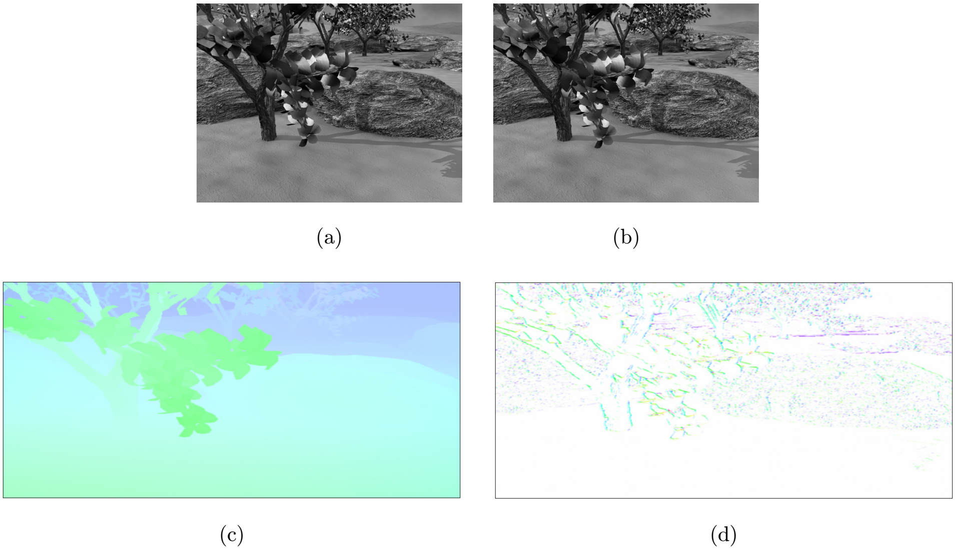


Fig. 10. Flow computed for Grove2 sequence by our method. (a) First frame. (b) Second frame. (c) Ground truth. (d) Computed flow.

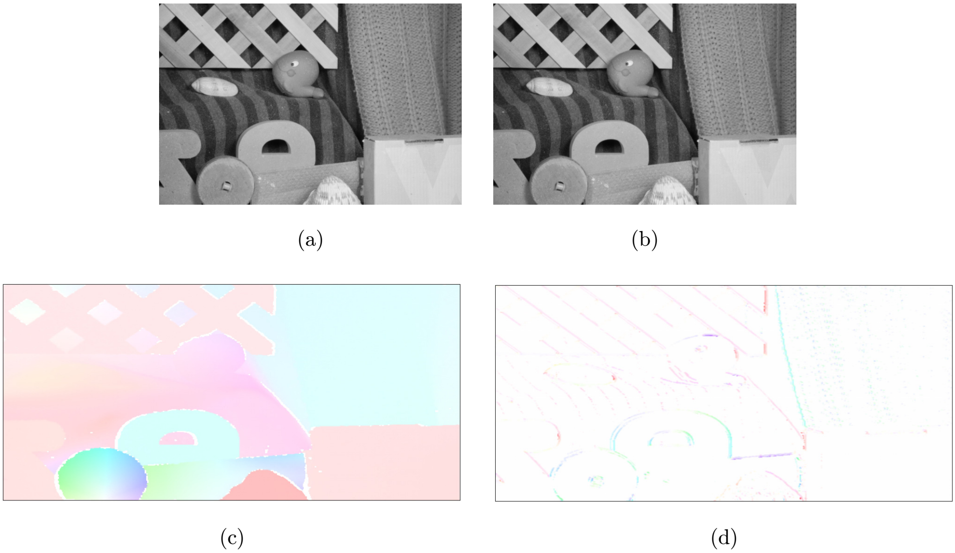
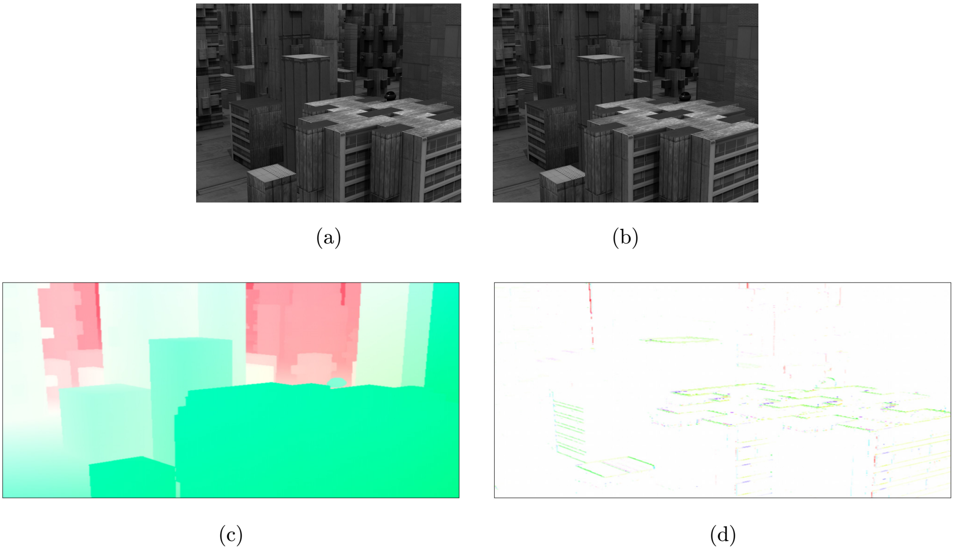


Fig. 11. Flow computed for RubberWhale sequence by our method. (a) First frame. (b) Second frame. (c) Ground truth. (d) Computed flow.



(a) (b)

(c) (d)

Fig. 12. Flow computed for Urban2 sequence by our method. (a) First frame. (b) Second frame. (c) Ground truth. (d) Computed flow.

Table 1. Error measures for Dimetrodon sequence.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	45.156	59.472	36.860
STD AE	19.848	10.565	25.896
AVG EE	1.696	2.007	2.174
STD EE	0.808	0.703	1.468

Table 2. Error measures for Grove2 sequence.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	60.911	66.345	48.038
STD AE	23.892	11.872	33.932
AVG EE	2.862	2.980	2.931
STD EE	0.9282	0.606	1.712

Table 3. Error measures for Hydrangea sequence.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	81.749	74.373	88.012
STD AE	29.718	14.853	44.673
AVG EE	4.070	3.766	5.108
STD EE	1.634	1.205	2.499

Table 4. Error measures for RubberWhale sequence.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	53.082	50.072	64.898
STD AE	22.133	21.335	40.752
AVG EE	1.337	1.267	2.392
STD EE	0.635	0.513	1.407

Table 5. Error measures for Urban2 sequence.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	67.210	69.117	75.157
STD AE	25.311	21.335	44.257
AVG EE	7.822	8.399	10.141
STD EE	7.780	8.079	8.128

$$AE = \cos^{-1}\left(\frac{1.0 + uu_{GT} + vv_{GT}}{\sqrt{1.0 + u^2 + v^2}\sqrt{1.0 + u_{GT}^2 + v_{GT}^2}}\right).$$

The second measure proposed by Baker *et al.*[2] is the absolute error computed in the flow *endpoint*. This error is defined by the L_2 norm of the difference between the flow vector $[u, v]$ and the ground truth vector $[u_{GT}, v_{GT}]$:

$$EE = \sqrt{(u - u_{GT})^2 + (v - v_{GT})^2}.$$

In order to compare our method to the classical approaches, we compute the optical flow of some sequences in dataset from Baker *et al.*[2] Figures 9–12 show the used sequences, the respective ground truth and the flow computed by our method.

We compute the angular and the endpoint errors for the sequences Dimetrodon, Grove2, Hydrangea, RubberWhale and Urban2. Tables 1–5 show the results for each image sequence. AVG AE and AVG EE are the average values for angular error (in degrees) and endpoint error (in pixels). SD AE and SD EE indicate the standard deviation for angular and endpoint errors, respectively.

Based on the quantitative errors, it is not possible to state which method is the best for all sequences. Note, however, that our method performs better in average than the classical methods, as it is not the worst in any sequence and is close to the best one. The measures proposed by Baker *et al.*[2] are in fact specific for dense optical flow methods. Our method and the classical differential approaches are actually an estimation of *brightness variation* and cannot be fairly compared to state-of-the-art optical flow methods. Thus, our goal is not to overcome the state-of-the-art methods for optical flow. Differential methods tend to be sparse due to high frequency regions dependency. Considering only the valid points, the results showed that our method overcomes the classical ones. The resulting flow for the ground truth pairs shows that our results are competitive, despite the sparseness. In fact, we observed that our method find less points but with higher confidence. This is particularly valuable for finding points of interest in the image sequence.

4.4. *Comparison using the human action recognition problem*

In recent years, several works focused on the problem of recognizing human actions in videos. This problem is one of the key prerequisites for video analysis and understanding. Some works have used optical flow in order to extract motion information from videos.[11,35–37] We propose to use our optical flow to form a global video descriptor based on Histograms of Optical Flow (HOF).[11] The approach presented in this section provides a different way for comparing optical flow methods. For video classification problem, we assume that a more precise and less noisy optical flow tends to give better accuracy results. Nevertheless, we remark that our goal is to compare optical flow methods using this problem and not to propose a new breakthrough for this application.

4.4.1. *Video descriptor using histograms of optical flow*

Our video descriptor is based on the scheme of Perez *et al.*[38,36] for histograms of gradients. Since we have computed the optical flow vector $\mathbf{v}_p = [u, v]$ in a point p, we can represent this vector in polar coordinates $\mathbf{s}_p = [r, \theta]$ with $\theta \in [0, \pi]$ and r is the magnitude of $\mathbf{v}_p$. The optical flow field for an image I_j can be compactly represented by a one-dimensional histogram of optical flow $\mathbf{h}_j = h_k,\ k \in [1, b_\theta]$, where b_θ is the number of cells for θ coordinate. For simplicity, we use a uniform subdivision of the angle intervals to populate the b_θ bins of the histogram:

$$h_k = \sum_p r_p w_p,$$

where $\{p \in I_j | k = 1 + \lfloor \frac{b_\theta \theta_p}{\pi} \rfloor\}$ are all points whose angle maps to the k bin and w_p is a per pixel Gaussian weighting factor. The complete optical flow field is represented by a vector with b_θ elements.

Since we have computed the histogram of optical flow for an image I_j, we compute an orientation tensor from the histogram $\mathbf{h}_j$:

$$T_j = \mathbf{h}_j \mathbf{h}_j^T. \tag{23}$$

The orientation tensor is a symmetric $k \times k$ matrix that carries the information of the optical flow distribution of the image I_j. It can be combined with other tensors in order to find component covariances.

In order to express the motion average of consecutive images, we use a series of tensors. The average motion of a video can be computed by $T = \sum T_j$ with $j = [1, n]$, where n is the length of the video. Each T_j is normalized by L_2-norm before summing and the final tensor T also is normalized by the L_2-norm. The normalization allows comparing descriptors of videos regardless their length or image resolution.

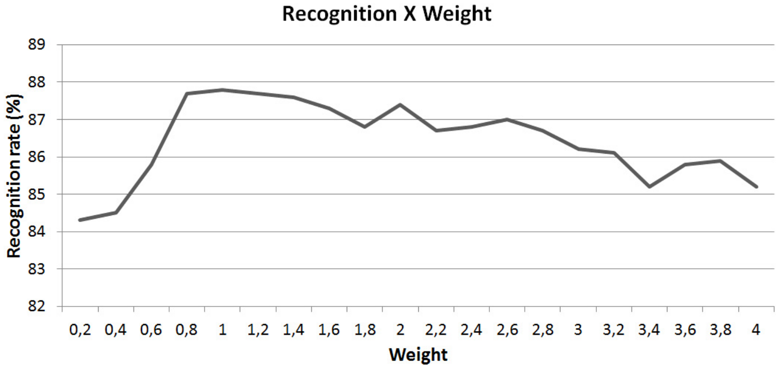


Fig. 13. Accuracies for different α weights. X-axis indicates α.

Table 6. Results of our method for different window size.

		Window height					
		5 (%)	8 (%)	10 (%)	15 (%)	20 (%)	30 (%)
Window Width	4	83.4	83.7	84.4	84.7	85.5	87.6
	5	84.3	84.5	84.4	85.4	85.6	87.8
	8	83.4	84.4	84.6	85.2	86.0	87.8
	10	84.1	84.6	83.8	85.2	85.8	86.6
	16	84.0	84.6	85.1	85.2	85.5	86.0

Table 7. Best configuration for each method.

Method	Configuration	Recognition
Our method	Window 5 × 30, 60 bins, $\alpha = 1.0$	87.8%
Lucas and Kanade	Window 4 × 30, 33 bins	86.1%
Horn and Schunck	Window 5 × 30, 36 bins, $\alpha = 2.0$	83.9%

In order to maintain the spatial correlation, the video frames are subdivided in windows of $a \times b$ pixels. We compute the histograms separately for each window. The final descriptor for each image I_j is then computed by the sum of the tensors of each window.

4.4.2. *Classification results*

We perform most our tests in the KTH dataset,[39] since it has been widely used in literature and has well controlled movements. We also perform some experiments in the Hollywood2 dataset.[40] The classification was performed using a two-fold strategy on a nonlinear SVM classifier.

For our method, we tested the influence of three parameters in the recognition rate: the weight α of our wave energy term, the size of the window in the image subdivision and the number of bins of the histogram. Several combinations of these

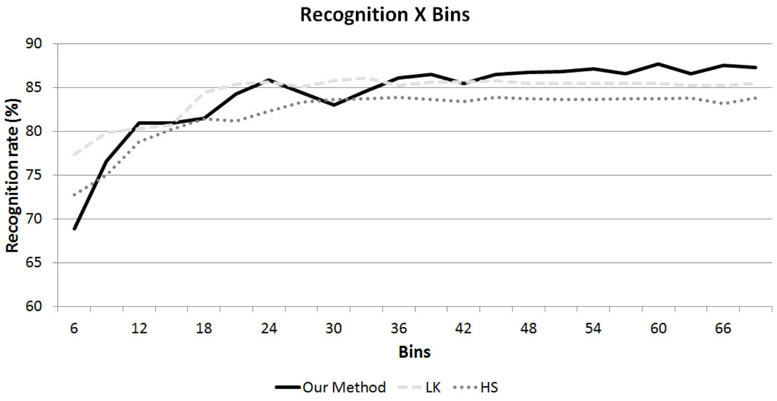


Fig. 14. Accuracies for different number of bins.

Table 8. Confusion matrix for our method using our best configuration (Window 5×30, 60 bins, $\alpha = 1.0$).

	Box (%)	HClap (%)	HWave (%)	Jog (%)	Run (%)	Walk (%)
Box	95.8	4.2	0.0	0.0	0.0	0.0
HClap	6.9	92.4	0.7	0.0	0.0	0.0
HWave	6.9	2.1	91.0	0.0	0.0	0.0
Jog	0.0	0.0	0.0	87.5	4.2	8.3
Run	0.0	0.0	0.0	27.8	69.4	2.8
Walk	0.0	0.0	0.0	6.9	2.1	91.0

parameters were tested and we have found the best configuration with $\alpha = 1.0$, window of 5×30 or 8×30 pixels and histogram of 60 bins, reaching 87.8% of accuracy.

Setting the window to 5×30 pixels and the histogram size to 60 bins, we analyze the influence of the weight α of the wave energy on the accuracy for our method. Figure 13 shows the accuracy in function of the α values. The best recognition rates are in the interval $0.6 < \alpha < 2.6$.

Table 6 shows the accuracy in function of the window size. The weight α was set to 1.0 and we use a histogram with 60 bins. The best accuracies are obtained by using

Table 9. Confusion matrix for Lucas and Kanade method using the best configuration (window 4×30, 33 bins).

	Box (%)	HClap (%)	HWave (%)	Jog (%)	Run (%)	Walk (%)
Box	91.6	8.4	0.0	0.0	0.0	0.0
HClap	8.3	89.6	2.1	0.0	0.0	0.0
HWave	3.5	3.5	90.3	0.0	2.8	0.0
Jog	0.0	0.0	0.0	82.6	8.3	9.0
Run	0.0	0.0	0.0	19.4	70.8	9.7
Walk	0.0	0.0	0.0	3.5	4.9	91.7

Table 10. Confusion matrix for Horn and Schunck method using the best configuration (Window 5 × 30, 36 bins, $\alpha = 2.0$).

	Box (%)	HClap (%)	HWave (%)	Jog (%)	Run (%)	Walk (%)
Box	95.1	4.2	0.7	0.0	0.0	0.0
HClap	1.4	81.2	17.4	0.0	0.0	0.0
HWave	6.2	7.6	85.4	0.0	0.7	0.0
Jog	0.0	0.0	0.0	80.6	12.5	6.9
Run	0.0	0.0	0.0	18.1	79.2	2.8
Walk	0.0	0.0	0.0	14.6	3.5	81.9

Table 11. Results in the Hollywood2 dataset.

Method	Recognition (%)
Our method	31.9
Horn and Schunck	28.8
Lucas and Kanade	27.4

windows with height 20 and 30 pixels, but small width windows (4, 5 or 8 pixels) also present good results.

Analogously, we performed tests with the classical methods using several window sizes, number of bins and, for the Horn and Schunck method, weight α. The best result for Lucas Kanade was 86.1% with window of 4 × 30 pixels and histogram of 33 bins. For Horn and Schunck, we achieved 83.9% with a 5 × 30 window, 36 or 52 bins in the histogram and $\alpha = 2.0$. Table 7 summarizes the results for the three methods.

We also tested the influence of the number of bins of the histogram for each method. Figure 14 shows the results where our method outperforms the classical approaches for histograms greater than 45 bins.

The confusion matrix for the best configuration of our method is showed in Table 8. One may observe that the major mislabeling is for jogging, walking and running actions. The speed of the motion is the main difference among these actions. Since our method is differential, it is difficult to identify long displacements and this mislabeling is expected. With our method, there is no confusion among arm movements (boxing, handclapping and handwaving) and the other movements.

Tables 9 and 10 show the confusion matrix for Lucas and Kanade and Horn and Schunck methods, respectively. One may see that our method identify better the movement jogging than the other approaches. Compared to Horn and Schunck, our method is better for the movement walking but worst for running. The low recognition of the class running by our method is justified by the fact that differential methods by definition cannot detect long displacements and this class presents a fast movement. The classical approaches yield more confusion between handwaving and running.

We also present results with the challenging dataset Hollywood2. The computations were performed using the best configuration found for each method in the KTH dataset. Table 11 shows the achieved results, where one may note that our method outperforms the classical methods. More tests in Hollywood2 dataset showed prohibitive due to its much higher size and complexity.

By the results presented in this section, we observed that our method achieve recognition rates slightly higher than the classical ones. These results show that our method is useful for applications evolving motion estimation. Despite the sparsity, the valid points are a good representation of the motion scene. Our comparison was performed against classical differential methods, since they tend to be sparse due to high frequency dependency. Our objective is to show that the wave equation is a useful constraint to extract reliable brightness variation spots and motion information from image sequences.

In terms of time complexity, the descriptors using HOF and our iterative version were computed with an average of 19 frames per second for the whole KTH dataset in an Intel® Xeon® 2.20 GHz processor with 32GB of memory. For the classical methods, we computed for the same dataset with 21 frames per second for the Horn and Schunck and 25 for the Lucas and Kanade methods. Using 35 videos from Hollywood dataset the descriptors were computed with 2.12 frames per second using our method, 1.62 using Horn and Schunck and 2.73 using Lucas and Kanade. Using the direct version, the descriptors were computed with 20 and 2.28 frames per second for the KTH and Hollywood2 dataset, respectively.

5. Conclusion

We proposed a differential optical flow method based on the energy of wave equation. A sparse optical flow is computed by minimizing a functional energy composed by two terms: A term of brightness constancy and a term of energy of the wave. This second term is the main contribution of our work.

Minimizing the functional energy through Euler–Lagrange equations, we obtain a system of linear equations. Due to the decoupling of the pixels, the system can be quickly solved by an iterative or a direct method. Besides, the decoupling strategy makes our approach local and suitable for parallelization. The derivatives computation using convolution can be easily mapped into GPU architectures.

Differential methods lack non-local information and tend to be sparse due to high frequency dependency. Therefore, the resulting flow field is sparse. Using a direct solution, the velocities can be estimated only where the matrix of the system is invertible. For the iterative solution, we presented the convergence conditions and the flow is computed only at the points that satisfies them. In spite of the sparsity, we have high quality flow in the detected points.

We performed comparative tests with the classical Horn and Schunck[3] and Lucas and Kanade[12] methods. This choice is justified by the fact that these classical approaches are the basis of most differential methods.[6,4] For the performance measures proposed by Baker *et al.*,[2] we verified that our method performs similar results to the classical ones with slight improvement on the flow precision. Our goal is to present a new constraint for differential methods using the wave equation and show that it is useful to extract reliable brightness variation spots.

Our flow was applied into the problem of human action recognition using histograms of optical flow. The best result of our method in the KTH dataset was 87.8% of recognition, outperforming the best results for Horn and Schunck (83.9%) and Lucas and Kanade (86.1%) methods, using the same recognition protocol.[38,36] These accuracy rates were computed using the best parameters for each method, obtained through extensive experimentation. Our method has a parameter α related to the weight of the energy of the wave equation in the total energy. The experiments showed that the resulting flow presents a better quality with $0.6 < \alpha < 2.6$ for the action recognition problem. The results achieved for this problem show that our method is useful for applications evolving motion estimation. Despite the sparsity, the valid points are a good representation of the motion scene.

This work presents a first model for the use of the wave equation as a point based optical flow detector. Due to the local constraints imposed by the second order differentials, our method is sparse. However, the quality of the velocity vectors obtained in such regions show that the use of the wave equation is promising. In the future, we intend to model the motion as a multiple wave propagation in order to find displacement vectors closely linked to their neighborhood. Such approach can lead to a less sparse flow field. Furthermore, we intend to include global elements in the flow computation, which could also reduce the sparsity. The first attempt was to constrain the flow rotational in the energy functional. However, this approach did not converge easily, since its overall conditions depended on all the points of the image. Another possibility is to use our convergence conditions as a feature detector, defining points with second order smoothness that might be regarded as salient points.

References

1. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black and R. Szeliski, “A database and evaluation methodology for optical flow,” *Proc. IEEE International Conference on Computer Vision* (Rio de Janeiro, Brazil, 2007).
2. S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black and R. Szeliski, “A database and evaluation methodology for optical flow,” *International Journal of Computer Vision*, **92**(1), 1 (2011).
3. B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, **17**(1–3), 185 (1981).
4. A. Bruhn, J. Weickert and C. Schnörr, “Lucas/kanade meets horn/schunck: Combining local and global optic flow methods,” *International Journal of Computer Vision*, **61**, 211 (2005).
5. T. Brox, A. Bruhn, N. Papenberg and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in *European Conf. Computer Vision*, Lecture Notes in Computer Science, Vol. 3024 (Springer, 2004).
6. H. A. Rashwan, M. A. García and D. Puig, “Variational optical flow estimation based on stick tensor voting,” *IEEE Transaction on Image Processing* **22**(7), 2589 (2013).

7. D. D. Nguyen and J. W. Jeon, "Tuning optical flow estimation with image-driven functions," *IEEE International Conf. Robotics and Automation* (Shanghai, China, 2011).
8. D. J. Butler, J. Wulff, G. B. Stanley and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision*, Part IV, Lecture Notes in Computer Science, Vol. 7577 (Springer-Verlag, 2012).
9. M. Menze, C. Heipke and A. Geiger, "Joint 3d estimation of vehicles and scene flow," *ISPRS Workshop on Image Sequence Analysis* (Santiago, Chile, 2015).
10. H. Wang, A. Klaser, C. Schmid and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11*, IEEE Computer Society, Washington, DC, USA (2011).
11. I. Laptev, M. Marszałek, C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies," in *Conf. Computer Vision and Pattern Recognition* (2008).
12. B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artificial Intelligence, IJCAI'81*, Vol. 2 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1981).
13. T. Myint-U and L. Debnath, *Linear Partial Differential Equations for Scientists and Engineers* (Birkhäuser, 2007).
14. F. Girosi, A. Verri and V. Torre, "Constraints for the computation of optical flow," in *Proc. Workshop on Visual Motion* (1989).
15. J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: Description of the algorithm," Technical report, Intel Corporation Microprocessor Research Labs (2000).
16. S. H. Hwang and S. U. Lee, "A hierarchical optical flow estimation algorithm based on the interlevel motion smoothness constraint," *Pattern Recognit.* **26**(6), 939 (1993).
17. T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(3), 500 (2011).
18. Z. Tu, N. Van Der Aa, C. Van Gemeren and R. C. Veltkamp, "A combined post-filtering method to improve accuracy of variational optical flow estimation," *Pattern Recogn.* **47** (5), 1926 (2014).
19. Z. Tu, R. Poppe and R. C. Veltkamp, "Weighted local intensity fusion method for variational optical flow estimation," *Pattern Recognit.* **50**, 223 (2016).
20. H. W. Haussecker and D. J. Fleet, "Computing optical flow with physical models of brightness variation," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(6), 661 (2001).
21. H. Sakaino, "Fluid motion estimation method based on physical properties of waves," in *Conf. Computer Vision and Pattern Recognition* (Anchorage, Alaska, USA, 2008).
22. S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Mach. Intell.* **2**(4), 333 (1980).
23. D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International Journal of Computer Vision*, **5**(1), 77 (1990).
24. D. Fortun, P. Bouthemy and C. Kervrann, "Optical flow modeling and computation: A survey," *Computer Vision and Image Understanding*, **134**, 1 (2015).
25. D. Sun, X. Yang, M.-Y. Liu and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2018).
26. T.-W. Hui, X. Tang and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (2018).

27. J. Chen, Z. Cai, J. Lai and X. Xie, "A filtering based framework for optical flow estimation," *IEEE Transactions on Circuit and Systems for Video Technology*, **29**(5), 1350 (2018).
28. C. Zhang, Z. Chen, M. Wang, M. Li and S. Jiang, "Robust non-local tv-l1 optical flow estimation with occlusion detection," *IEEE Transactions on Image Processing*, **26**(8), 4055 (2017).
29. D. Giancoli, *Physics for Scientists and Engineers with Modern Physics* (Prentice Hall, 1989).
30. J. Latter, "Tsunamis of volcanic origin: Summary of causes, with particular reference to Krakatoa, 1883," *Bulletin of Volcanology*, **44**(3), 467 (1981).
31. J. l. R. d'Alembert, "Recherches sur la courbe que forme une corde tendue mise en vibration" (1747).
32. S. A. Van Duyne and J. O. Smith, "The 2-D digital waveguide mesh," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (New York, USA, 1993).
33. F. Fontana and D. Rocchesso, "A new formulation of the 2D-waveguide mesh for percussion instruments," in *Proc. XI Colloquium on Musical Informatics* (Bologna, Italy, 1995).
34. C. Lanczos, *The Variational Principles of Mechanics*, Dover Books on Physics and chemistry (Dover Publications, 2012).
35. V. F. Mota, E. A. Perez, M. B. Vieira, L. M. Maciel, F. Precioso and P. H. Gosselin, "A tensor based on optical flow for global description of motion in videos," *SIBGRAPI - Conf. Graphics, Patterns and Images* (Ouro Preto, Brazil, 2012).
36. V. Mota, E. Perez, L. Maciel, M. Vieira and P. Gosselin, "A tensor motion descriptor based on histograms of gradients and optical flow," *Pattern Recognition Letters* **39**, 85 (2014).
37. A. A. Efros, A. C. Berg, E. C. Berg, G. Mori and J. Malik, "Recognizing action at a distance," in *Int. Conf. Computer Vision* (Nice, France, 2003).
38. E. A. Perez, V. F. Mota, L. M. Maciel, D. O. Sad and M. B. Vieira, "Combining gradient histograms using orientation tensors for human action recognition," in *Proc. Int. Conf. Pattern Recognition* (Tsukuba, Japan, 2012).
39. C. Schuldt, I. Laptev and B. Caputo, "Recognizing human actions: A local svm approach," in *Proc. Int. Conf. Pattern Recognition* (Cambridge, UK, 2004).
40. M. Marszałek, I. Laptev and C. Schmid, "Actions in context," in *Conf. Computer Vision & Pattern Recognition* (Miami, USA, 2009).

Luiz Maurlio da Silva Maciel is Ph.D. in Systems Engineering and Computer Science at UFRJ (2018). He hold M.Sc. degree in Computer Science from UFJF (2014) and graduated in Computer Science at the same university (2011). He is professor at UFJF since 2019. His main knowledge area is Computer Science focusing on Image Proccessing and Computer Vision.

Marcelo Bernardes Vieira is Ph.D. in Image & Signal Processing (2002) from ENSEA/UCP in France and Computer Science (2002) from UFMG. He has M.Sc. in Computer Science (1998) from UFMG and graduated in Computer Science in PUC-MG (1995). He is associate professor at UFJF with interests in computer vision, computer geometry and fluid simulation.