UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Luiz Maurílio da Silva Maciel

Optical Flow Computation Using Wave Equation Based Energy

Juiz de Fora

2014

Luiz Maurílio da Silva Maciel

Optical Flow Computation Using Wave Equation Based Energy

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 27 de Fevereiro de 2014.

BANCA EXAMINADORA

Prof. D.Sc. Marcelo Bernardes Vieira - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora

Prof. D.Sc. William Robson Schwartz
Universidade Federal de Minas Gerais

For my parents, my girlfriend,
my brother, my grandparents and
my friends.

AGRADECIMENTOS

Firstly I thank God for giving me the opportunity to be here. I thank my parents, Maurílio and Maria Aparecida, for the immense support and encouragement along this trajectory. I thank my girlfriend, Rosa, for the support, encouragement and love in all the moments.

I thank my advisor, Marcelo Bernardes Vieira, for their dedication and patience throughout this work. Thanks to Virginia Mota and the other GCG members for the support in this work.

I also thank all my friends for companionship, support and encouragement during these two years. Anyway, I thank everyone who contributed in some way for my trajectory.

"If one day you have to choose
between the world and love,
remember: If you choose the
world you'll be left without love,
but if you choose love, with it
you will conquer the world."
Eistein, A.

RESUMO

Identificar movimento em vídeos é uma tarefa fundamental a fim de analisar a sua informação semântica. Uma das principais ferramentas para a identificação de movimento é o fluxo óptico, o qual estima a projeção da velocidade 3D dos objetos sobre o plano da câmera.

Neste trabalho é proposto um método diferencial de fluxo óptico baseado na equação da onda. O fluxo óptico é calculado através da minimização de funcional de energia composto por dois termos: um termo de constância de brilho e um termo de energia da onda. O fluxo é então determinado através da resolução iterativa de um sistema de equações lineares. O desacoplamento entre os pixels na solução garante convergência rápida e torna o método adequado para a paralelização. No entanto, nossa abordagem não converge para todos os pontos de imagem, sendo apresentadas as suas condições de convergência.

O fluxo proposto é aplicado no problema de reconhecimento de ação através da criação de um descritor global de vídeo baseado em histogramas de fluxo óptico (HOF). Apesar da sua esparsidade, o método proposto supera as abordagens clássicas. Também são avaliadas medidas de erro de fluxo óptico para algumas sequências de imagens conhecidas. Os erros encontrados são similares para o nosso método e as abordagens clássicas de fluxo óptico.

Palavras-chave: Fluxo óptico. Método diferencial. Equação da onda.

ABSTRACT

Identification of motion in videos is a fundamental task to analyse their semantic information. One of the main tools for motion identification is the 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: brightness constancy and energy of the wave. The flow is then determined by solving iteratively a system of linear equations. The decoupling of the pixels in the solution ensures quick convergence and makes the method suitable for parallelization. However, our approach does not converge for all the image points and we present its convergence conditions.

We apply our optical flow in the action recognition problem by creating a global video descriptor based on histograms of optical flow (HOF). Despite its sparsity, our method outperforms the classical approaches. We also evaluate optical flow error measures for some known image sequences. The errors found are similar for our method and the classical optical flow approaches.

Keywords: Optical flow. Differential method. Wave equation.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUCTION

The study of movement in image sequences is an important field in computer vision for many years. Identify movement in a video is a fundamental task in order to analyse its semantic information. This information extracted 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.

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 the first image to the second image. As such, several problems are associated to optical flow estimation. The occlusion of objects, for example, is a major problem. Dealing with homogeneous regions and determining large displacements are other problems for optical flow methods.

In order to deal with these problems and improve the computed optical flow, several methods have been proposed since Horn and Schunck (HORN; SCHUNCK, 1981) proposed their differential optical flow method. Recently, a new database was proposed by Baker et al. (2011) in order to evaluate the new optical flow algorithms. The set of benchmarks of this database contains sequences for several scene configurations. This shows that optical flow is still an open problem.

Due to the variety of applications of optical flow, each method tends to be better for a specific application. An example of problem where movement extraction is useful is the human action recognition problem (WANG et al., 2011; LAPTEV et al., 2008). This problem consists in three stages: extraction of features, video descriptors creation and classification. In the first stage are extracted motion features from the videos. Since optical flow is a motion representation, it can be used to compose features.

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 analyse its temporal variation. For example, Lucas and Kanade (LUCAS; KANADE, 1981) and Horn and Schunck (HORN; SCHUNCK, 1981) methods start from

a movement equation which is just an equation with advective terms.

Another example of transport phenomenon is modelled by the wave equation (MYINT-U; DEBNATH, 2007). 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 modelling the brightness variation as waves propagating in a medium. We study how the wave equation can contribute to estimate motion from image sequences. This study led us to propose an energy function that must be minimized similarly to Horn and Schunck' approach (HORN; SCHUNCK, 1981). This energy and its computational model are the main contributions of this work.

1.1 PROBLEM DEFINITION

This work discuss the problem of optical flow computation in an image sequence. Consider $I_1(x, y)$ and $I_2(x, y)$ two sequential images. 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) 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 $I_1(x, y) = I_2(x + u(x, y), y + v(x, y))$. Our problem is how to integrate the wave equation to the model of optical flow computation.

This work is based on the hypothesis that the wave equation is useful for motion estimation. As such, the modeling and evaluation of the wave equation for optical flow computation is an important part of the problem in this work.

1.2 OBJECTIVES

The objective of this work is to propose a new differential optical flow computation method based on wave equation energy. The secondary objectives are:

- To show the convergence conditions of the proposed method,
- To find the best values of the method parameters,

- To apply the optical flow calculated in the human action recognition problem using histograms of optical flow,
- To compare the performance of our method to the classical optical flow methods.

1.3 RELATED WORKS

This section describes several approaches to calculate optical flow. These approaches include differential (variational), hierarchical, physical model based and other methods.

1.3.1 DIFFERENTIAL METHODS

Differential methods are based on the spatiotemporal image gradients. These methods are classified into local and global methods. Local methods assume that the flow is uniform in a pixel neighborhood. The first local optical flow method was proposed by Lucas e Kanade (1981) and it is explained in the Section 2.3. On the other hand, global methods, for example the classical Horn and Schunck (HORN; SCHUNCK, 1981) method that is explained in the Section 2.2, 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. On the other hand, the flow field estimated by local method tend to be more robust against noise.

Bruhn et al. (2005) 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, joining the main advantages of the local and global differential methods. They reformulate the approaches of the classical methods in order to join the quadratic form minimized by Lucas e Kanade (1981) with the global energy functional of Horn e Schunck (1981). The method, known as *combined local-global (CLG)*, also permits to sparsify gradually a dense flow field. Additionally, nonquadratic and multiresolution approaches are presented. In the present work, we propose an energy minimization similarly to Horn and Schunck. However, our energy depends only on a small neighborhood of each point used to calculate image derivatives. This makes our method local.

Most of differential methods minimize a functional energy. This energy is composed by two or more terms including the term of brightness constancy. This term is known by *data term*. Section 2.1 shows more details about this term. Brox et al. (2004) presented

a differential method that computes optical flow by minimizing a functional combining a brightness constancy assumption, a gradient constancy assumption, and a discontinuity-preserving spatiotemporal smoothness constraint. The first two constraints correspond to the data term while the third penalize 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.

Girosi et al. (1989) observe that when the image brightness changes over time, its change 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 operators also are used in our method due to the wave equation.

The approach of Nguyen e Jeon (2011) uses image-driven functions to compute optical flow. Similarly to other differential methods, they also propose an energy functional that considers the data term, gradient term and a smoothness term. The major contribution proposed by this work is to weight the terms of energy by image-driven functions. These functions determine the constraints that must influence each image region. The energy proposed is minimized using the Euler Lagrange equation in a similar manner we do in the present work.

Recently, Rashwan et al. (2013) proposed a differential method that adapts the data term using anisotropic stick tensor voting. This term still uses a subquadratic 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 e Schunck (1981) is still the basis of several differential methods.

1.3.2 HIERARCHICAL METHODS

In order to improve the detection of large displacements, some optical flow methods use a hierarchical (pyramidal) strategy. This strategy consists in calculating the flow in lower resolution images and then projecting this flow in the higher resolution images as an initial guess. Bouguet (2000) proposes a pyramidal implementation of Lucas and Kanade method. Initially, the pyramid is computed for each image. The flow computation starts from the deepest 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 the present work, we also use an iterative algorithm to calculate the optical flow with few iterations but we do not use a hierarchical scheme. As showed in the Chapter 3, however, our energy is local and can be computed directly, without any iteration.

Another hierarchical method was proposed by Hwang e Lee (1993). This method starts calculating a Gaussian pyramid applying repetitively the Gaussian filtering. They calculate the flow combining the brightness constancy assumption to a interlevel 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 interlevel 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. In the Section 3.1, we also present the convergence condition for our method.

A problem of these hierarchical approaches was identified by Brox e Malik (2011): small objects can move very fast. 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 remain 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.

1.3.3 PHYSICAL MODEL BASED METHODS

Haussecker e Fleet (2001) 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 models, for example diffusion. At the end, they show a generalized formulation to extract parameters of general physical models. Instead of determining the parameters of a physical model, in our work we propose to use a physical model in order to estimate optical flow.

Similarly, the work of Sakaino (2008) proposes to estimate fluid flow based on the physical properties of waves. The method exploits three properties of waves present in fluid-like images: 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 which are the two wavenumber components, frequency, amplitude, and orientation. This function is minimized in order to calculate the seven variables. This method is very useful 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.

1.3.4 OTHER APPROACHES

Region correlation based methods estimate optical flow by calculating a similarity measure between the neighborhood of a pixel in an image and the neighborhood of a candidate pixel in the following image. The pixel in the second image that maximizes the similarity measure in relation to the pixel in the first image will be the displaced pixel. The distance between the two pixels will be the optical flow.

For example, Barnard e Thompson (1980) 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 is constructed a set of labels which

correspond to possible matches. 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. Once the probabilities are calculated, the point with highest probability will be the correspondent point. This method presents a high computational cost because it performs many distance calculations and comparisons.

Farnebäck (2001) proposes a method using orientation tensors and parametric motion. Assuming that the motion in a region is coherent with respect to the affine motion model, his model proposed to extract the motion parameters directly from the orientation tensors of the images. These orientation tensors are obtained from the video volume. A segmentation of the images is performed in order to divide them into a set of disjoint regions. Each region is characterized by a coherent motion. This segmentation improves the algorithm accuracy but raises the computational cost.

Energy based methods exploit output velocity-tuned filters to analyze the movement in the Fourier domain. They are also called frequency based methods. In their work, Adelson e Bergen (1985) start representing motion in a three-dimensional space in which X and Y are the two spatial dimensions and T is the temporal dimension. The motion is identified by the energy of the orientation in this space. They propose to extract spatiotemporal energy by oriented linear filters and combine their outputs to give a measure of motion. An interesting fact is that some energy based methods are equivalent to correlation based methods (SANTEN; SPERLING, 1985).

In phase based methods, the phase behavior of band-pass filter outputs defines the velocity. Fleet e Jepson (1990) 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 concentrate 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. Several experiments were performed in order to evaluate method robustness under some scene conditions such as translational camera motion, image rotation, additive noise and transparency. The authors emphasize that their method does not depend on previous element detection from the images. This is also a feature of our proposed method.

2 FUNDAMENTALS

In this chapter we present the basic fundamentals of this work. These fundamentals include the brightness constancy constraint, the classical optical flow methods and the mathematical model of wave propagation.

2.1 BRIGHTNESS CONSTANCY CONSTRAINT

Image brightness constancy is the constraint used by most optical flow algorithms. This assumption states that the brightness intensity remains from an image to the next one. This constancy can be expressed as:

$$I(x, y, t) = I(x + u(x, y), y + v(x, y), t + \Delta t), \tag{2.1}$$

where $I(x, y, t)$ is the brightness intensity of the pixel (x, y) at the time t. The vector $\vec{v} = [u(x, y), v(x, y)]$ 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 on we will denote $\vec{v} = [u(x, y), v(x, y)]$ by $\vec{v} = [u, v]$.

Applying a first-order Taylor expansion to the right-hand side, we have the approximation:

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

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$. The Optical Flow Constraint is then defined as:

$$I_x u + I_y v + I_t = 0. \tag{2.2}$$

Equation 2.2 is not enough to determine $[u, v]$ since there are two unknowns and just one constraint. This problem is called aperture problem. Due to aperture problem, only the vector component parallel to the spatial gradient is determined. To solve this problem, it is necessary another constraint to complete the system. In our work, we propose the use of the energy of the wave equation to provide additional constraints to the system. Since the wave equation defines the local motion in a point induced by a wave passing through that point, these constraints depend solely on the local first and second-order brightness

variation. As such, the resulting method is straightforward to implement.

2.2 HORN AND SCHUNCK METHOD

The first differential method for optical flow computation was proposed by Horn e Schunck (1981). Due to its revolutionary idea, this propose is the basis for the most differential methods for optical flow computation.

This global method uses the Equation 2.2 of brightness constancy and tries to solve the aperture problem by imposing a smoothness constraint to the flow field. This constraint is expressed by the square of the magnitude of gradient of the velocity:

$$E_c^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2. \tag{2.3}$$

Combining the above constraint to the Equation 2.2, we have the total energy of the Horn and Schunck method:

$$E^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 + \alpha^2 \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] dxdy,$$

where α is a suitable weighting factor. The optical flow vector $\vec{v} = [u, v]$ is calculated by minimizing the above functional.

Our proposed method, which will be exposed in the Chapter 3, also use an energy functional. The smoothness constraint of the Horn and Schunck method is replaced by a wave equation based constraint.

2.3 LUCAS AND KANADE METHOD

Another classical optical flow estimation method was proposed by Lucas e Kanade (1981). This method is local, assuming the flow is constant in the neighborhood of each pixel. Using the brightness constancy constraint, the optical flow is computed by minimizing:

$$E = \sum_{(x,y)\in\Omega} W^2(x,y)(I_x u + I_y v + I_t)^2 \tag{2.4}$$

where $W(x, y)$ is a weight associated to the point (x, y) in the window Ω. Central pixels have more influence than those in the window edges. For each pixel in the image, Equation

2.4 generates a overdetermined system that can be solved using least squares fit.

Local methods generates a non-dense flow field but they tend to be more robust against noise. Our wave equation based method is local but, differently than Lucas and Kanade´s, we use first and second-order derivative elements. In our experimental evaluation, we use Horn and Schunck and Lucas and Kanade methods to compare our performance.

2.4 WAVE EQUATION

In general, waves are caused by a disturbance in a medium. For example, a pulse travelling on a cord can be formed by a quick up-and-down motion of the hand (GIANCOLI, 1989). This pulse travels along the cord moving the 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. The Figure 2.1 represents a wave propagating and its velocities.

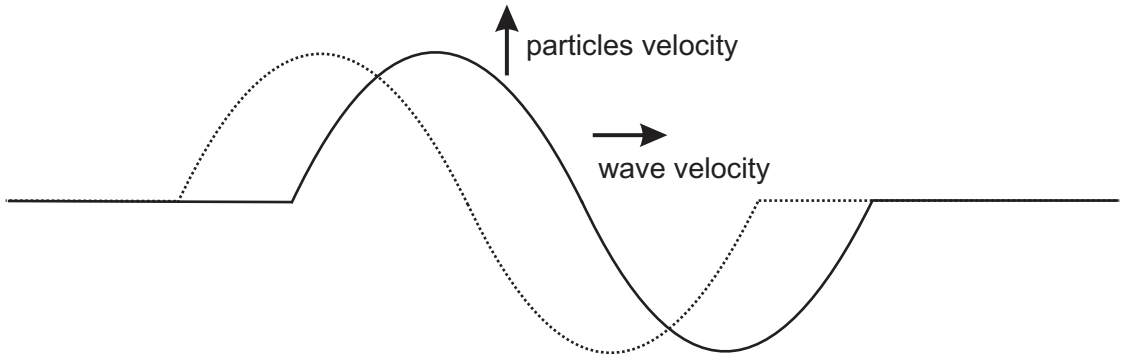


Figura 2.1: 1D wave propagation example. The dashed line represents the wave in a previous moment.

An example of two-dimensional waves in the nature is the tsunami. Tsunamis are water waves that can be caused by earthquakes, volcanic eruptions and other underwater explosions (LATTER, 1981). These waves travel over the water surface causing elevation. The Figure 2.2 shows a simulation of a 2D water wave. We can see three moments of the wave propagation: the moment which the disturbance happens and two moments of the propagation along the water surface.

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 the temporal brightness variation is caused by a wave passing along the image plane. The Figure 2.3

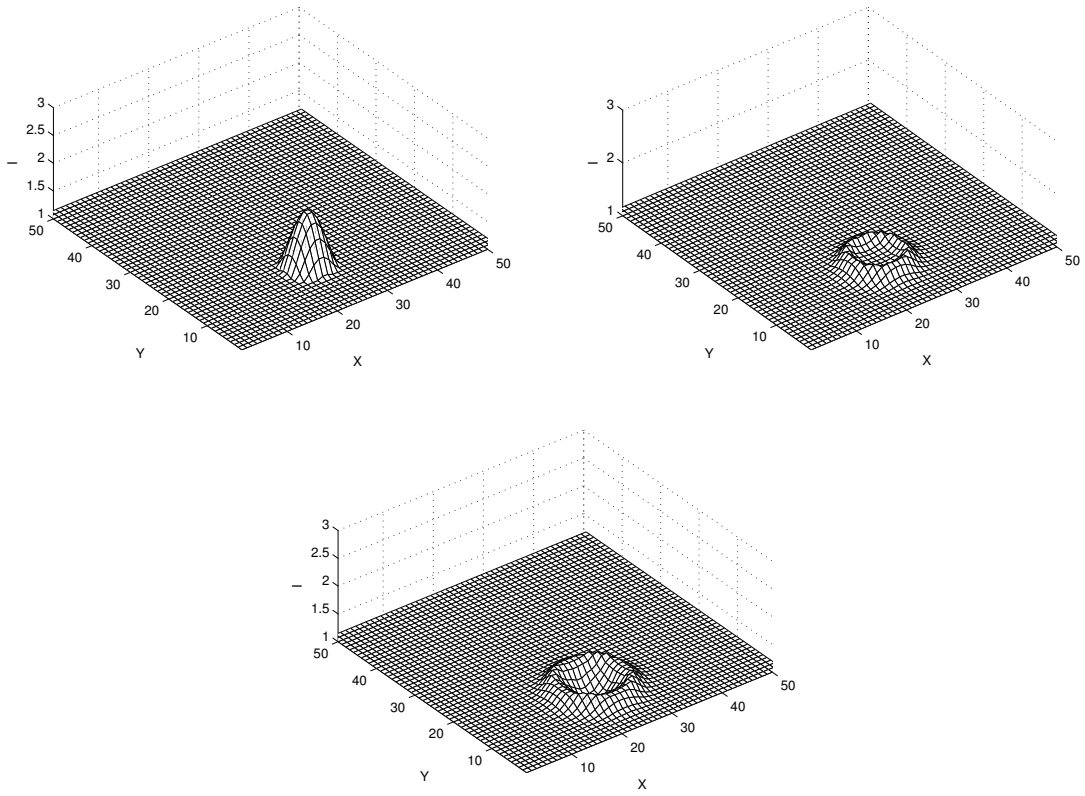


Figura 2.2: 2D wave propagation example.

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 travelling 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 (MYINT-U; DEBNATH, 2007). It has the general form:

$$\frac{\partial^2 I}{\partial t^2} = c^2 \nabla^2 I, \tag{2.5}$$

where ∇^2 represents the spatial Laplacian and the constant c represents the velocity of propagation. 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:

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

The first solution of the Equation 2.5 in one dimension was proposed by d'Alembert. 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 general solution is obtained by introducing new variables:

$$\xi = x - ct,\ \eta = x + ct,$$

changing the wave equation into:

$$\frac{\partial^2 I}{\partial \xi \partial \eta} = 0. \tag{2.6}$$

The general solution of the Equation 2.6 is:

$$I(\xi,\eta) = F(\xi) + G(\eta) = F(x-ct) + G(x+ct).$$

Thus, the solutions of the one dimensional wave equation is the sum of a right travelling function F and a left travelling function G. For the initial values, we have the solution:

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

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 some initial and/or contour conditions of the problem.

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

Waves transport energy from one place to another one (GIANCOLI, 1989). The energy is transferred from one particle to the neighbor particle in the medium where the wave is travelling. The energy of the general wave equation 2.5 is given by (MYINT-U; DEBNATH, 2007):

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

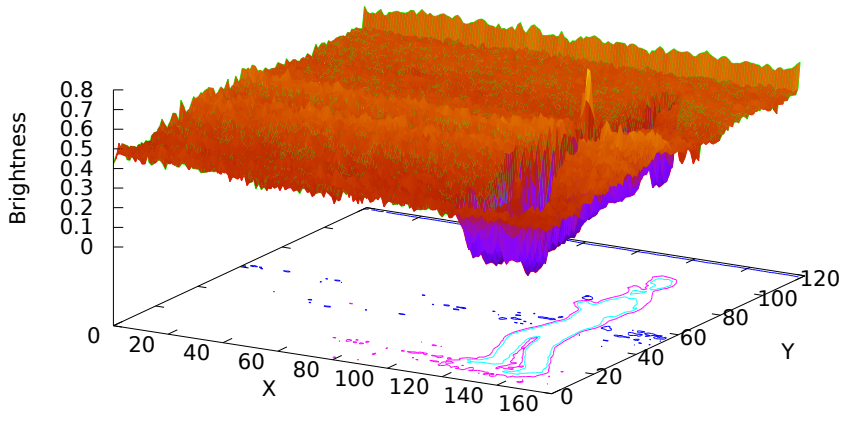
Extending to the two-dimensional case we have:

$$E(t) = \frac{1}{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_t^2 + c^2 (I_x^2 + I_y^2) \; dxdy. \tag{2.7}$$

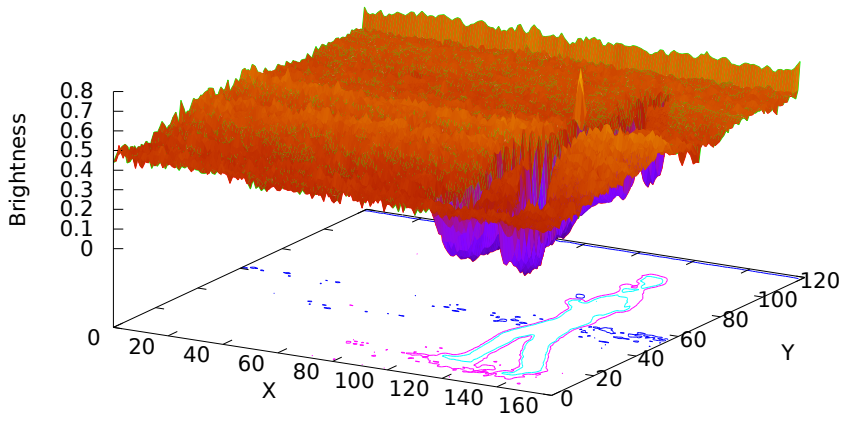
Generally, the 2D wave equation defines a vertical displacement of the moving wave in the 2D plane (FONTANA; ROCCHESSO, 1995; DUYNE et al., 1993). This is interesting for optical flow computation because most methods try to find a displacement vector to compensate the local brightness variation I_t. Here, the I_t plays the role of how much mass is locally transferred by the passing wave. In an energy minimization model, the Equation 2.7 has also the function of compensating the brightness variation. This complementary second-order constraint originated from the wave equation combined with the classic first-order Equation 2.2 is a major contribution of this work.

(a) First image

(b) Second Image



(c) Surface representation of the first image



(d) Surface representation of the second image

Figura 2.3: Two Images and their surface representation.

3 PROPOSED METHOD

In the Section 2.4, we have presented the wave equation and its associated energy. As the classical Horn and Schunck method (HORN; SCHUNCK, 1981) presented in the Section 2.2 and several other differential optical flow methods (NGUYEN; JEON, 2011; BROX et al., 2004; RASHWAN et al., 2013), we propose to minimize an energy functional involving the wave equation energy 2.7.

We need to define the square of the velocity of propagation in the wave energy (Eq. 2.7). This is a crucial step of our method and the proposed velocity is an important contribution of our work. We desire to find the velocities that minimize the energy of the wave yielding vectors whose magnitudes are the lowest possible. We then propose a wave velocity that consider the variation of the square of the norm of the image velocity vectors. Denoting the inner product as $\langle \cdot, \cdot \rangle$, we define:

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

Calculating the partial derivatives of H we have:

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

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

The velocity of propagation of the wave is defined as the spatial gradient of H:

$$c^2 = \nabla H = \left[\frac{\partial H}{\partial x}, \frac{\partial H}{\partial y} \right] = [2(uu_x + vv_x), 2(uu_y + vv_y)]. \tag{3.1}$$

Replacing 3.1 in the Equation 2.7 results in the wave energy

$$E_w = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2} \left(I_t^2 + [2(uu_x + vv_x), 2(uu_y + vv_y)](I_x^2 + I_y^2) \right) \, dxdy, \tag{3.2}$$

where $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$ and $I_t = \frac{\partial I}{\partial t}$.

In the Section 2.2, we have seen that Horn and Shunck use the Optical Flow Constraint

as a first-order term of the energy. This term is defined as:

$$E_d = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 \, dxdy \tag{3.3}$$

Combining the Equation 3.3 and the Equation 3.2, the complete proposed energy is:

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 + \alpha \left[\frac{1}{2}\left(I_t^2 + \nabla H \cdot \left(I_x^2 + I_y^2\right)\right)\right] \, dxdy, \tag{3.4}$$

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 (HORN; SCHUNCK, 1981). 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.

Separating x and y components of 3.4, we have two energy terms:

$$E_x = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 + \alpha^2 \left[\frac{1}{2}\left(I_t^2 + 2\left(uu_x + vv_x\right)\left(I_x^2 + I_y^2\right)\right)\right] \, dxdy, \tag{3.5}$$

$$E_y = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (I_x u + I_y v + I_t)^2 + \alpha^2 \left[\frac{1}{2}\left(I_t^2 + 2\left(uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right] \, dxdy. \tag{3.6}$$

The total energy to be minimized is the sum of the Equations 3.5 and 3.6:

$$\begin{aligned} E_{total} = & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} 2\left(I_x u + I_y v + I_t\right)^2 \\ & + \alpha^2 \left[I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right] \, dxdy, \end{aligned} \tag{3.7}$$

which defines the energy of waves travelling 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 the Equation 3.7, we apply the Euler-Lagrange equations (LANCZOS, 2012):

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial u_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial L}{\partial u_y}\right) = 0, \tag{3.8}$$

$$\frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\left(\frac{\partial L}{\partial v_x}\right) - \frac{\partial}{\partial y}\left(\frac{\partial L}{\partial v_y}\right) = 0, \tag{3.9}$$

where L is the functional to be minimized.

The partial derivatives of our functional 3.7 are

$$\begin{aligned}
\frac{\partial L}{\partial u} &= \frac{\partial}{\partial u}\left(2\left(I_x u + I_y v + I_t\right)^2\right) + \frac{\partial}{\partial u}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) \\
&= 4I_x\left(I_x u + I_y v + I_t\right) + \alpha^2\left(u_x + u_y\right)\left(I_x^2 + I_y^2\right), \\
\frac{\partial L}{\partial v} &= \frac{\partial}{\partial v}\left(2\left(I_x u + I_y v + I_t\right)^2\right) + \frac{\partial}{\partial v}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) \\
&= 4I_y\left(I_x u + I_y v + I_t\right) + \alpha^2\left(v_x + v_y\right)\left(I_x^2 + I_y^2\right), \\
\frac{\partial L}{\partial u_x} &= \frac{\partial}{\partial u_x}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) = \alpha^2 u\left(I_x^2 + I_y^2\right), \\
\frac{\partial L}{\partial u_y} &= \frac{\partial}{\partial u_y}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) = \alpha^2 u\left(I_x^2 + I_y^2\right), \\
\frac{\partial L}{\partial v_x} &= \frac{\partial}{\partial v_x}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) = \alpha^2 v\left(I_x^2 + I_y^2\right), \\
\frac{\partial L}{\partial v_y} &= \frac{\partial}{\partial v_y}\left(\alpha^2\left(I_t^2 + \left(uu_x + vv_x + uu_y + vv_y\right)\left(I_x^2 + I_y^2\right)\right)\right) = \alpha^2 v\left(I_x^2 + I_y^2\right).
\end{aligned}$$

Applying the Euler-Lagrange Equation 3.8 to the Equation 3.7, we thus have:

$$\begin{aligned}
4I_x\left(I_x u + I_y v + I_t\right) + \alpha^2\left(u_x + u_y\right)\left(I_x^2 + I_y^2\right) - & \\
\frac{\partial}{\partial x}\left(\alpha^2 u\left(I_x^2 + I_y^2\right)\right) - \frac{\partial}{\partial y}\left(\alpha^2 u\left(I_x^2 + I_y^2\right)\right) &= 0 \Rightarrow \\
4I_x^2 u + 4I_x I_y v + 4I_x I_t + \alpha^2\left(I_x^2 u_x + I_x^2 u_y + I_y^2 u_x + I_y^2 u_y\right. & \\
\left. -2I_{xx} I_x u - I_x^2 u_x - 2I_{xy} I_y u - I_y^2 u_x - 2I_{xy} I_x u - I_x^2 u_y - 2I_{yy} I_y u - I_y^2 u_y\right) &= 0 \Rightarrow \\
\left(2I_x^2 - \alpha^2\left(I_{xx} I_x + I_{xy} I_y + I_{xy} I_x + I_{yy} I_y\right)\right) u + 2I_x I_y v + 2I_x I_t &= 0,
\end{aligned}$$

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 3.9 to the Equation 3.7, we also have:

$$\begin{aligned}
4I_y\left(I_x u + I_y v + I_t\right) + \alpha^2\left(v_x + v_y\right)\left(I_x^2 + I_y^2\right) - & \\
\frac{\partial}{\partial x}\left(\alpha^2 v\left(I_x^2 + I_y^2\right)\right) - \frac{\partial}{\partial y}\left(\alpha^2 v\left(I_x^2 + I_y^2\right)\right) &= 0 \Rightarrow \\
4I_x I_y u + 4I_y^2 v + 4I_x I_t + \alpha^2\left(I_x^2 v_x + I_x^2 v_y + I_y^2 v_x + I_y^2 v_y\right. & \\
\left. -2I_{xx} I_x v - I_x^2 v_x - 2I_{xy} I_y v - I_y^2 v_x - 2I_{xy} I_x v - I_x^2 v_y - 2I_{yy} I_y v - I_y^2 v_y\right) &= 0 \Rightarrow \\
2I_x I_y u + \left(2I_y^2 - \alpha^2\left(I_{xx} I_x + I_{xy} I_y + I_{xy} I_x + I_{yy} I_y\right)\right) v + 2I_x I_t &= 0.
\end{aligned}$$

Therefore, we find the linear system whose unique solution finds the flow vectors $[u, v]$

for each pixel that minimizes our energy (Eq. 3.7)

$$\begin{cases} \left(2I_x^2 - \alpha^2\left(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y\right)\right)u + 2I_xI_yv + 2I_xI_t = 0, \\ 2I_xI_yu + \left(2I_y^2 - \alpha^2\left(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y\right)\right)v + 2I_yI_t = 0 \end{cases}$$

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

$$\begin{bmatrix} D_u & 2I_xI_y \\ 2I_xI_y & D_v \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -2I_xI_t \\ -2I_yI_t \end{bmatrix} \tag{3.10}$$

where

$$D_u = 2I_x^2 - \alpha^2\left(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y\right),$$
$$D_v = 2I_y^2 - \alpha^2\left(I_{xx}I_x + I_{xy}I_y + I_{xy}I_x + I_{yy}I_y\right).$$

The system of the Equation 3.10 can be solved by a direct or an iterative method. Using the iterative Jacobi method, we have:

$$u^{(k+1)} = \frac{-2I_xI_yv^{(k)} - 2I_xI_t}{D_u}, \tag{3.11}$$

$$v^{(k+1)} = \frac{-2I_xI_yu^{(k)} - 2I_yI_t}{D_v}, \tag{3.12}$$

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. In the 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 on its neighbors. Consequently, the optical flow is local due to the inherently local differential wave equation. This ensures quick convergence and makes the method suitable for parallelization.

However, Jacobi method does not converge for all the image points. For our method be useful, a convergence condition is necessary as exposed in Section 3.1.

3.1 CONVERGENCE CONDITION

Equations 3.11 and 3.12 calculate the optical flow, but they do not converge for all points in an image sequence. A condition for convergence is that the denominator of the equations must not be 0. Therefore, we must have:

$$D_u \neq 0, \tag{3.13}$$

$$D_v \neq 0. \tag{3.14}$$

Besides above conditions, the Jacobi method (and any iterative method) converges if the spectral radius of the iteration matrix is 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

$$\begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{2I_xI_y}{D_u} \\ -\frac{2I_xI_y}{D_v} & 0 \end{bmatrix} \begin{bmatrix} u^{(k)} \\ v^{(k)} \end{bmatrix} + \begin{bmatrix} -\frac{2I_xI_t}{D_u} \\ -\frac{2I_yI_t}{D_v} \end{bmatrix}, \tag{3.15}$$

from which we obtain the iteration matrix:

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

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

$$\det(\mathrm{J} - \mathrm{I}_2\lambda) = 0, \tag{3.16}$$

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

Expanding Equation 3.16 we have:

$$\det\left(\mathrm{J} - \mathrm{I}_2\lambda\right) = \det\begin{pmatrix} -\lambda & -\frac{2I_xI_y}{D_u} \\ -\frac{2I_xI_y}{D_v} & -\lambda \end{pmatrix} = \lambda^2 - \frac{4I_x^2I_y^2}{D_uD_v} = 0,$$

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 1. Thus, we have the following convergence condition:

$$\frac{|2I_xI_y|}{\sqrt{|D_uD_v|}} < 1. \tag{3.17}$$

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

We can compute the optical flow vector $[u, v]$ only at points where the conditions expressed by the Equations 3.13, 3.14 and 3.17 are satisfied. It is important to note that solving the system of the Equation 3.10 by a direct method, we also need to exclude some points in order to avoid division by 0.

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. For example, let $f(x)$ be a real function. We can approximate the first derivative $f'(x)$ of f by the finite difference:

$$f'(x) \approx \frac{f(x) - f(x-h)}{h},$$

where h is the step of discretization. The above equation is the approximation of the first derivative by backward difference.

Let $I(x, y, t)$ be a function of three independent variables. We can approximate the partial derivatives $\frac{\partial I}{\partial x}$ of I by finite backward differences:

$$\frac{\partial I}{\partial x}(x, y, t) = \frac{I(x, y, t) - I(x-h, y, t)}{h}.$$

This can also be done for the second-order derivatives. A simple formulation for a function $f(x)$ is

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}.$$

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 ap-

propriate filters in order to reduce the effect of noise. In this work, we calculate the the partial derivatives by the following steps:

- 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;
- for $\frac{\partial I}{\partial x}$:
  - 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.
- for $\frac{\partial I}{\partial y}$:
  - 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.
- for $\frac{\partial I}{\partial t}$:
  - 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.
- for $\frac{\partial^2 I}{\partial x^2}$:
  - 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.
- for $\frac{\partial^2 I}{\partial y^2}$:
  - 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.

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

  - 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 Chapter 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 than 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 condition of the Equations 3.13, 3.14 and 3.17 are tested for each pixel. For the points where the convergence conditions are satisfied, the iterative system of the Equations 3.11 and 3.12 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 differencial methods by definition cannot detect long displacements. In this work, all vectors bigger than seven pixels are just set to null.

In addition to the iterative solution, we can solve the system of the Equation 3.10 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:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathrm{A}^{-1} \begin{bmatrix} -2I_x I_t \\ -2I_y I_t \end{bmatrix} \tag{3.18}$$

It is important to note that the Equation 3.18 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.

4 EXPERIMENTAL RESULTS

4.1 EXPERIMENTAL SETUP

In this section, we will describe the context of our experiments. In the present work, our main application is the human action recognition using the database KTH. This problem is described in the Section 4.4. Basically, the performance of our optical flow in classifying human actions guided the parameters setup.

In the Chapter 3, we presented the direct and the iterative version of our optical flow differential method. In our experiments, we observe that 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 the use of the iterative method, we chose to use this version in all experiments.

In the Section 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 the Section 3.2, with the non-centered masks.

Futhermore, we performed tests without the previous filtering of the images by the gaussian or without the low-pass filtering in the orthogonal directions before computing the derivatives. We observe that the results without the filtering are worst. 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.

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. Low values of α mean that the constraint of brightness constancy will be the dominant term. On the other hand, high

values of α lead to a loss of information of the brightness conservation. We performed tests varying the alpha values in the range [0.1, 10] and the best results for the human action recognition problem using our setup is $\alpha = 1.0$.

Our iterative method presents a quick convergence and, consequently, only a few iterations are need 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, which is compatible with what is generally used in the literature.

4.2 FLOW VISUALIZATION

A simple representation of optical flow is a vector field over the image plane. Figure 4.1 shows this representation for a field flow computed by our method. By visually inspecting the scene, one may see clearly that the person moves to the right side. We can see that the method identifies movements in several directions in a scene. Due to the convergence constraints, the method only computes the vectors in the points where the image derivatives are not null and the Equation 3.17 is satisfied. In this computation we use the weight $\alpha^2 = 1.0$ for the wave energy.

The Figure 4.2 shows the flow computed by the Horn and Schunck method. This method is global and generates a flow more dense than our approach. For this example, the flow of Horn and Schunck method is more smooth, spreading the motion information detected in strong gradient areas. In other words, the motion detected on borders tend to influence the surrounding pixels, depending on the smoothing factor. This smoothness can be beneficial or not, depending on the application. The flow shown in the Figure 4.2 was computed with the smoothness weight $\alpha = 2.0$. This value was the best in our tests on human action recognition (Sec. 4.4). Despite the sparse result, our flow tends to be aligned to the image edges.

In the Figure 4.3, we show the result of Lucas and Kanade optical flow method. Similarly to our method, Lucas and Kanade approach is local. The flow vectors estimated are more aligned to the image edges. In this example, Lucas and Kanade optical flow is more dense than ours. Note, however, that the displacement vectors have somewhat conflicting directions. Some of them are following the face to hair contour. The density and the flow quality depends on the window used. A common approach with this method,

(a) First image

(b) Second image

(c) Optical flow calculated

Figura 4.1: Optical flow detected by our method.

Figura 4.2: Optical flow detected using Horn and Schunck method (HORN; SCHUNCK, 1981).

used in this work, is to use a window of 5×5 elements. The Lucas and Kanade method can give vectors with high magnitudes, similarly to our method, due to local derivative estimation inconsistencies. In our implementation, we chose to filter the vectors greater than seven pixels, which are set to null, in order to be fairly compared with our results.

Baker et al. (2007) propose a representation of optical flow by color coding. In their scheme, each direction is represented by a color hue and the flow norm is represented by the saturation. In other words, color (hue) indicates the vector angle and the saturation

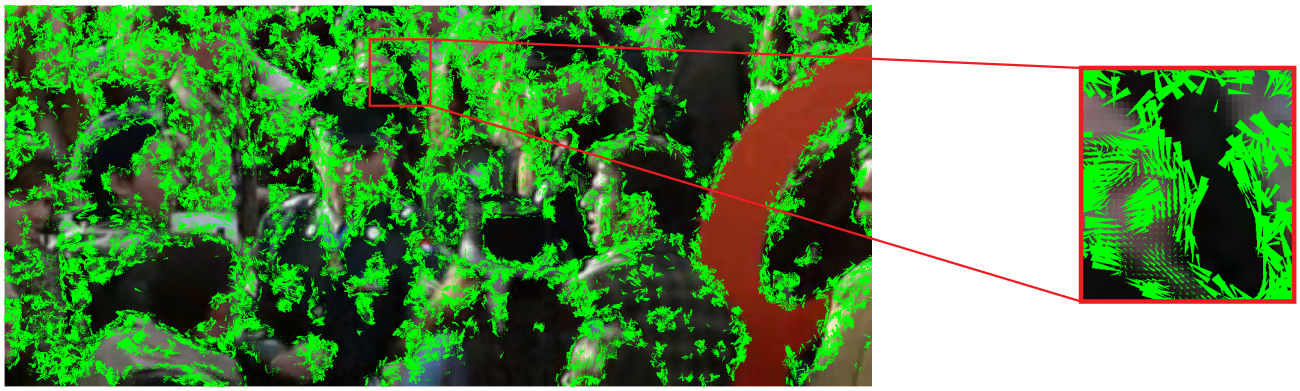


Figura 4.3: Optical flow detected using Lucas and Kanade method (LUCAS; KANADE, 1981).

indicates the amplitude. The Figure 4.4 shows this color coding. Several recent works in optical flow use this optical flow representation.

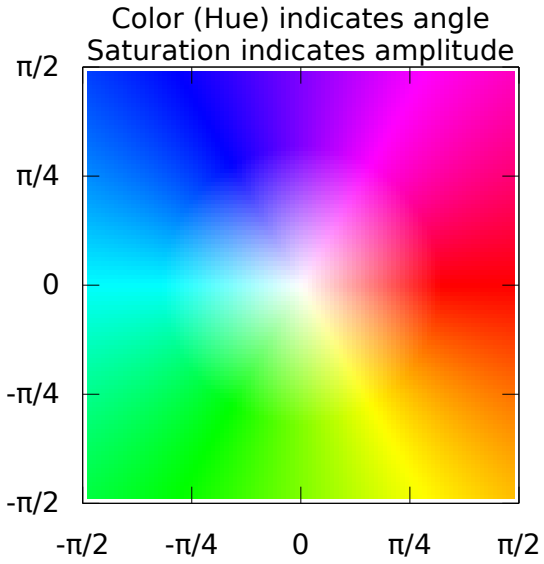
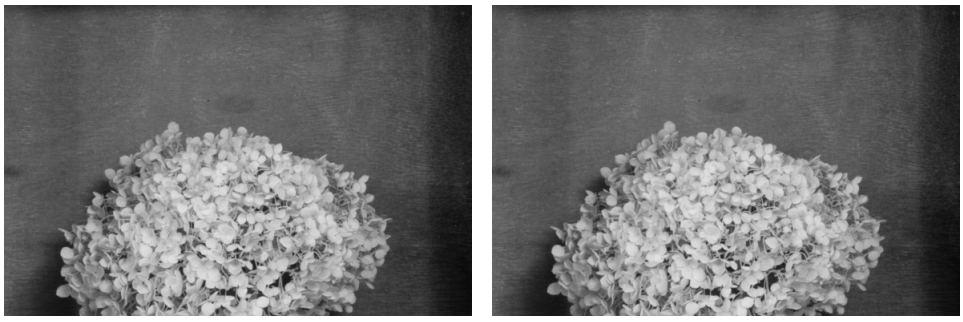


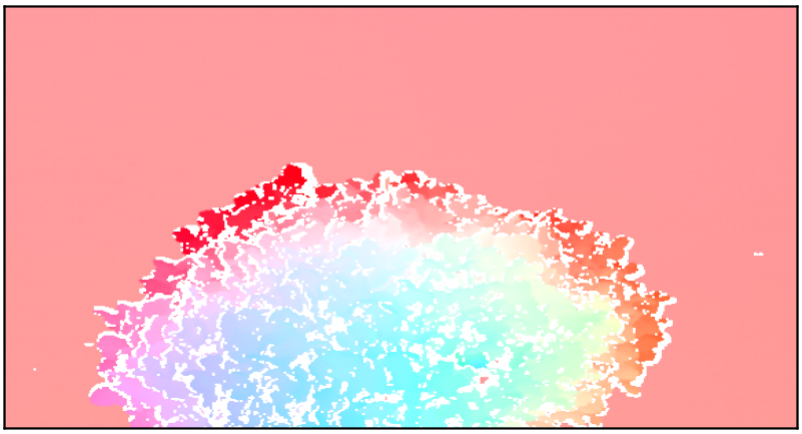
Figura 4.4: Color coding for optical flow visualization (BAKER et al., 2007).

Baker et al. (2007) also propose a new database to evaluate optical flow algorithms. They provide ground truths for some image sequences. In our work, we perform some experiments using these sequences. In the Figure 4.5, we show an example of image sequence *Hydrangea* and the respective ground truth.

The flow for Hydrangea sequence using our method is showed in Figure 4.6. Although our flow is not dense, one may note color similarity between our flow and the ground



(a) First image (b) Second image



(c) Ground truth

Figura 4.5: Example of ground truth in color coding (BAKER et al., 2007).

truth (Fig. 4.5). 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.

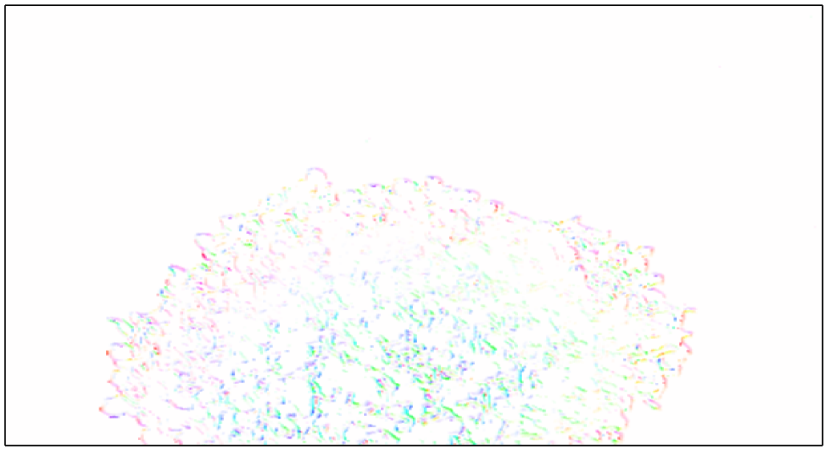
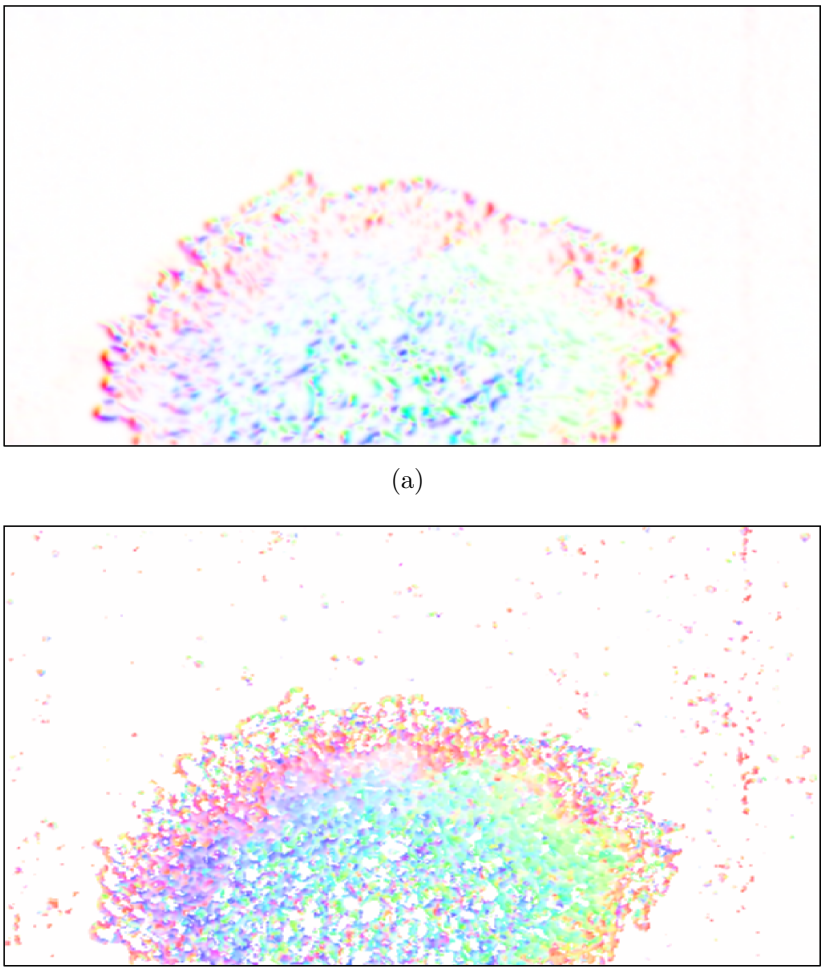


Figura 4.6: Optical flow computed for Hydrangea sequence using our method. Note that the method does not converge for the homogeneous regions.

We also calculate the flow for Hydrangea sequence using the classical methods (Fig.

4.7). We observe that the classical methods compute a more dense flow field. It is possible to note some color similarity among the results of the classical methods and our flow (Fig. 4.6).



(a)

(b)

Figura 4.7: Optical flow computed for Hydrangea sequence using classical methods. (a) Horn and Schunk. (b) Lucas and Kanade.

It is interesting to note that, 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 neighbours. It only depends on its own value in the previous iteration. The Figure 4.8 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 4 iterations.

Similarly to Horn e Schunck (1981), our method has a parameter α to control the influence of the wave equation energy. We observe that low values of α (around 0.1) make difficult the convergence for several points. On other hand, high values of α reduce the energy of the data term and the resulting optical flow tend to be inconsistent in some

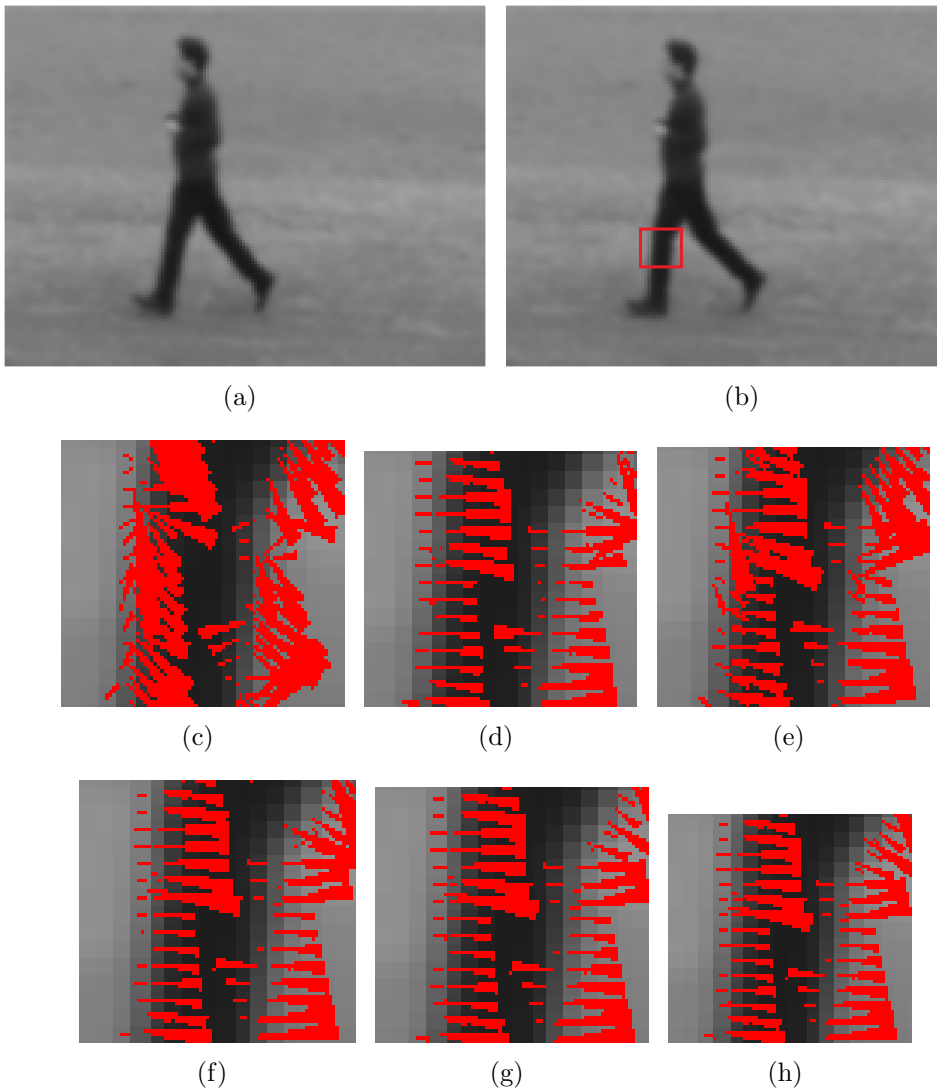


Figura 4.8: Evolution of our flow for a region of a image sequence. (a) First image. (b) Second image. (c) 1 iteration. (d) 2 iterations. (e) 3 iterations. (f) 4 iterations. (g) 5 iterations. (h) 10 iterations.

regions, notably on the borders (Fig. 4.9).

4.3 QUANTITATIVE COMPARISON

Baker et al. (2011) propose performance measures for optical flow. In recent years, these measures have been used in order to evaluate the results of new optical flow methods (NGUYEN; JEON, 2011; RASHWAN et al., 2013). In our experiments, we use two of them.

The first measure is the *angular error* between the flow vector $[u, v]$ and the ground truth vector $[u_{GT}, v_{GT}]$. The error is the angle between the vectors in the three-dimensional

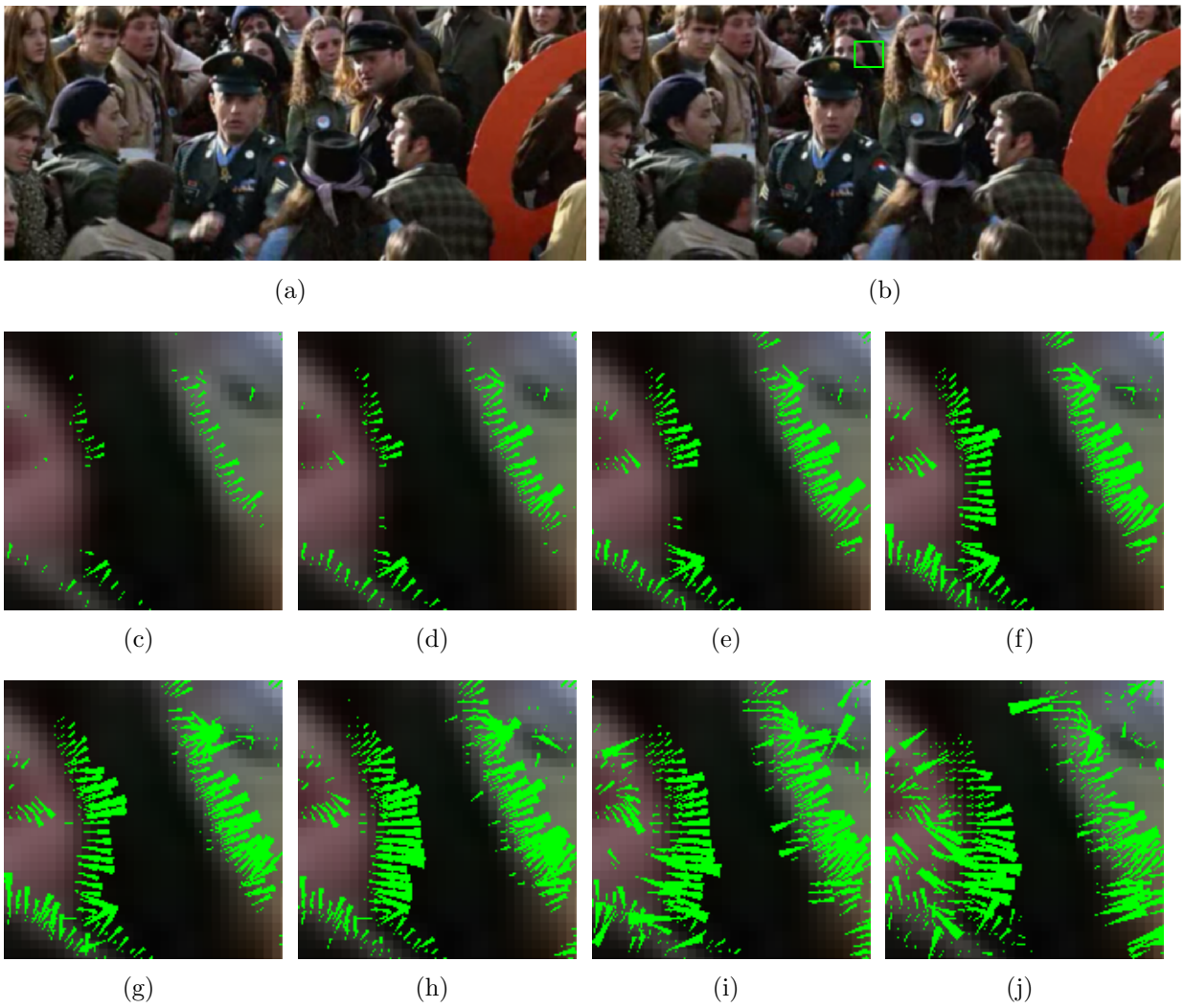


Figura 4.9: Flow computed for different α values. (a)First image. (b) Second image. (c) $\alpha^2 = 0.1$. (d) $\alpha^2 = 0.2$. (e) $\alpha^2 = 0.4$. (f) $\alpha^2 = 0.8$. (g) $\alpha^2 = 1.0$. (h) $\alpha^2 = 2.0$. (i) $\alpha^2 = 4.0$. (j) $\alpha^2 = 10.0$.

space. The third coordinate is set to 1.0. Therefore, we have the error:

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

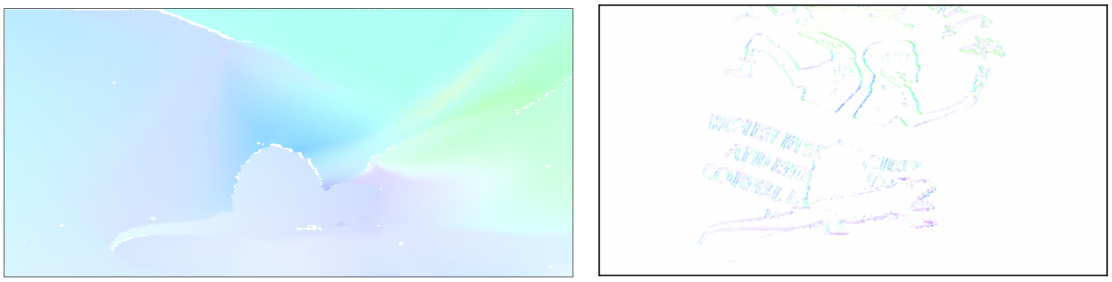
The second measure proposed by Baker et al. (2011) is the absolute error computed in the flow *endpoint*. This error is defined by the L_2 norm of the difference between the flow vector $[u, v]$ and the ground truth vector $[u_{GT}, v_{GT}]$:

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

In their work, (BAKER et al., 2011) propose a new database for evaluation of optical

flow. The database contains several image sequences, including the ground truths for some cases. We use some of these sequences with ground truth in order to compare our method to the classical approaches. The Figures 4.10, 4.11, 4.12 and 4.13 show the sequences used, the respective ground truth and the flow computed by our method.

(a) First frame (b) Second frame



(c) Ground truth (d) Computed flow

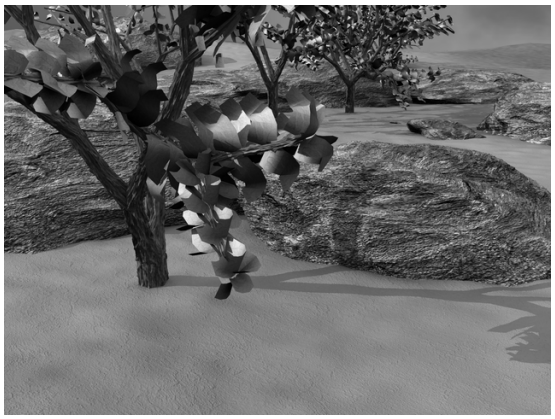
Figura 4.10: Flow computed for Dimetrodon sequence by our method

We compute the angular error and the endpoint error for the sequences Dimetrodon, Grove2, Hydrangea, RubberWhale and Urban2 using our method and the classical Horn e Schunck (1981) and Lucas e Kanade (1981). Tables 4.1, 4.2, 4.3 4.4 and 4.5 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 error and endpoint error respectively.

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	45.156384	59.47282	36.860088
STD AE	19.848906	10.565714	25.896709
AVG EE	1.696999	2.007118	2.174817
STD EE	0.808686	0.703345	1.468324

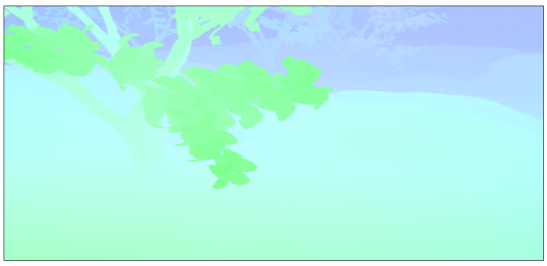
Tabela 4.1: Error measures for Dimetrodon sequence

Analysing the tables is not possible to state if a method is better than the other ones.

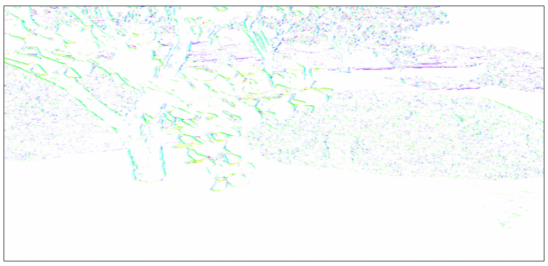


(a) First frame

(b) Second frame



(c) Ground truth

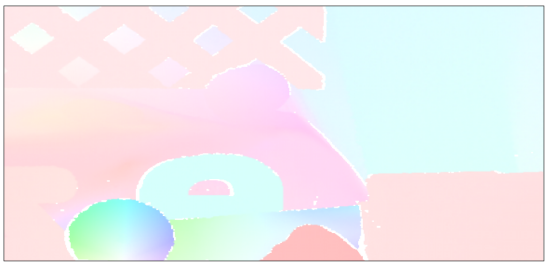


(d) Computed flow

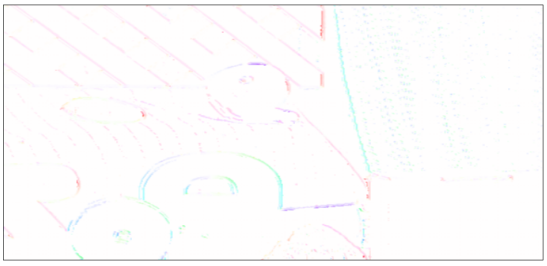
Figura 4.11: Flow computed for Grove2 sequence by our method

(a) First frame

(b) Second frame

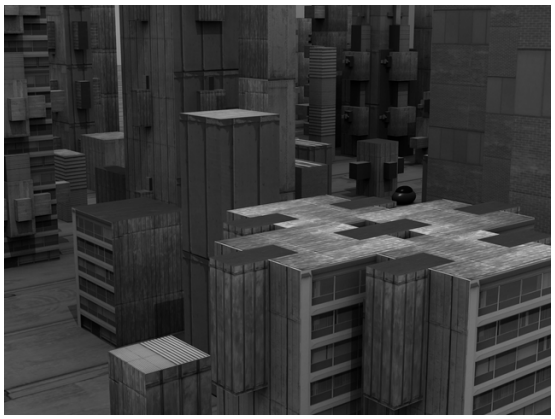


(c) Ground truth

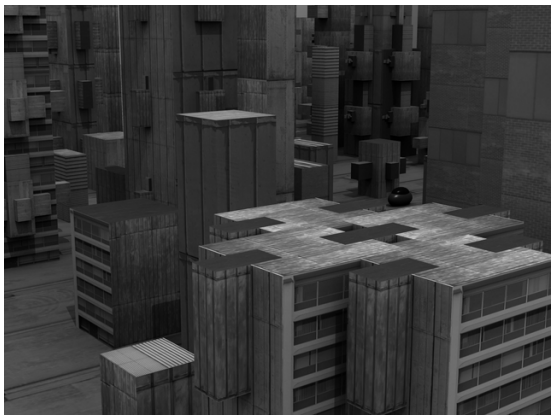


(d) Computed flow

Figura 4.12: Flow computed for RubberWhale sequence by our method

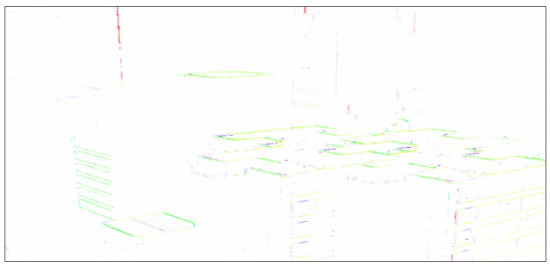


(a) First frame



(b) Second frame

(c) Ground truth



(d) Computed flow

Figura 4.13: Flow computed for Urban2 sequence by our method

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	60.911053	66.34507	48.0382
STD AE	23.892475	11.872857	33.932652
AVG EE	2.862447	2.980207	2.931243
STD EE	0.928243	0.606691	1.71224

Tabela 4.2: Error measures for Grove2 sequence

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	81.749733	74.373672	88.012001
STD AE	29.7183	14.853719	44.673588
AVG EE	4.070399	3.766577	5.10854
STD EE	1.634529	1.205063	2.499768

Tabela 4.3: Error measures for Hydrangea sequence

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	53.082855	50.072758	64.898186
STD AE	22.133101	21.335655	40.752655
AVG EE	1.337554	1.267871	2.392219
STD EE	0.635263	0.513079	1.407653

Tabela 4.4: Error measures for RubberWhale sequence

	Our method	Horn and Schunck	Lucas and Kanade
AVG AE	67.21035	69.117477	75.157982
STD AE	25.311131	21.335655	44.257145
AVG EE	7.822988	8.399518	10.141594
STD EE	7.780985	8.079951	8.128618

Tabela 4.5: Error measures for Urban2 sequence

It is important to note that the measures proposed by Baker et al. (2011) are specific for optical flow methods. Actually, our method and the classical differential approaches are estimation of brightness variation and can not be fairly compared to state-of-the-art optical flow methods. The errors were presented for completeness and to compare our method performance relative to the classical differential approaches. The resulting flow for the ground truth pairs, however, show that our results tend to be plausible, despite the sparseness.

4.4 APPLICATION IN THE HUMAN ACTION RECOGNITION PROBLEM

In recent years, several works focused on the problem of recognition of human actions in videos. This problem is one of the key prerequisites for video analysis and understanding. Some works have been using optical flow in order to extract motion information from videos (LAPTEV et al., 2008; MOTA et al., 2012, 2013; EFROS et al., 2003). In this work, we propose to use our optical flow in a global video descriptor based on histograms of optical flow (HOF) (LAPTEV et al., 2008). The Section 4.4.1 describes our descriptor.

4.4.1 VIDEO DESCRIPTOR USING HISTOGRAMS OF OPTICAL FLOW

Our video descriptor is based on the scheme of Perez et al. (2012); Mota et al. (2013) for histograms of gradients. Since we have computed the optical flow vector $\vec{v}_p = [u, v]$ in a point p, we can represent this vector in polar coordinates $\vec{s}_p = [r, \theta]$ with $\theta \in [0, \pi]$ and r is the magnitude of $\vec{v}_p$. The optical flow field for an image I_j can be compactly represented by a one-dimensional histogram of optical flow $\vec{h}_j = h_k$, $k \in [1, b_\theta]$, 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 k bin and w_p is a per pixel gaussian weighting factor. The whole 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 $\vec{h}_j$:

$$T_j = \vec{h}_j \vec{h}_j^T. \tag{4.1}$$

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. Thus 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 L_2 norm. The normalization allows comparing descriptors of videos regardless their length or image resolution.

In order to maintain the spatial correlation, we propose a uniform subdivision of the video frames 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 (SCHULDT et al., 2004) since it is widely used in literature. This database is composed of six actions: boxing, handclapping, handwaving, jogging running and walking. Each action is performed by 25 subjects several times. Most of the scenes present a homogeneous background. We also perform some experiments in the Hollywood2 database (MARSZAŁEK et al., 2009). The classification was performed using a two-fold strategy on a non-linear SVM classifier. In this section

		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%

Tabela 4.6: Results of our method for different window size.

we will analyse the results of the descriptor using our optical flow and the flow of the classical Lucas e Kanade (1981) and Horn e Schunck (1981) methods.

For our method, we analyse 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 this 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. This configuration reaches 87.8% of recognition.

Setting the window to 5×30 pixels and the histogram size to 60 bins, we analyse the influence of the weight α of the wave energy on the recognition rate for our method. The chart of the Figure 4.14 shows the recognition rate in function of the α values. We can note that the best recognition rates are in the interval $0.6 < \alpha^2 < 2.6$.

Figura 4.14: Recognition rate for different α weights. X-axis indicates α^2.

The Table 4.6 shows the recognition rate in function of the window size. The weight α was set to 1.0 and we use a histogram with 60 bins. One may note that the best recognition rates are achieved using windows with height 20 and 30 pixels. On other hand, small width window (4, 5 or 8 pixels) presents good results.

We perform 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 method, we achieved 83.9% with a 5×30 window, 36 or 52 bins in the histogram and $\alpha = 2.0$. The Table 4.7 summarizes the results for the three methods.

Method	Configuration	Recognition
Our method	Window 5×30, 60 bins, $\alpha^2 = 1.0$	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%

Tabela 4.7: Best configuration for each method

We also analyse the influence of the number of bins of the histogram for each method. The Figure 4.15 shows the results. We observe that the recognition rate stabilizes and our method outperforms the classical approaches for histograms greater than 45 bins.

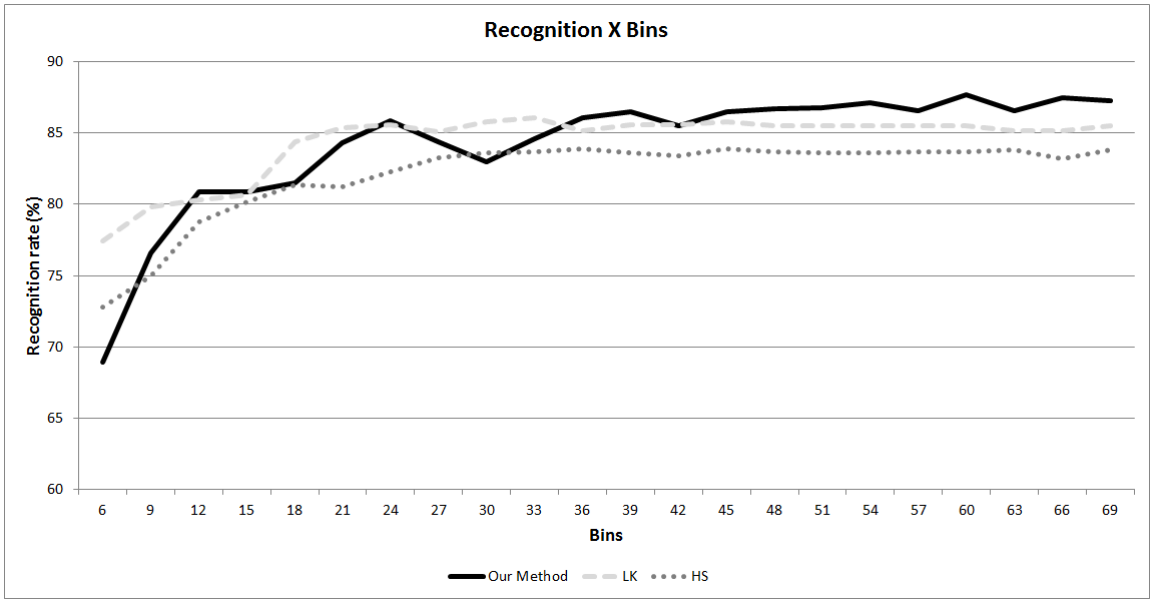


Figura 4.15: Recognition rate for different number of bins.

The confusion matrix for the best configuration of our method is showed in the Table 4.8. One may observe that the major mislabelling 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 mislabelling is expected.

The Table 4.10 and 4.9 show the confusion matrix for Horn and Schunck and Lucas and Kanade methods. We observe that our method identify better the movement jogging

	Boxing	Handclapping	Handwaving	Jogging	Running	Walking
Boxing	95.8%	4.2%	0.0%	0.0%	0.0%	0.0%
Handclapping	6.9%	92.4%	0.7%	0.0%	0.0%	0.0%
Handwaving	6.9%	2.1%	91.0%	0.0%	0.0%	0.0%
Jogging	0.0%	0.0%	0.0%	87.5%	4.2%	8.3%
Running	0.0%	0.0%	0.0%	27.8%	69.4%	2.8%
Walking	0.0%	0.0%	0.0%	6.9%	2.1%	91.0%

Tabela 4.8: Confusion matrix for our method using our best configuration (Window 5×30, 60 bins, $\alpha^2 = 1.0$).

than the other approaches. Comparing 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.

	Boxing	Handclapping	Handwaving	Jogging	Running	Walking
Boxing	91.6%	8.4%	0.0%	0.0%	0.0%	0.0%
Handclapping	8.3%	89.6%	2.1%	0.0%	0.0%	0.0%
Handwaving	3.5%	3.5%	90.3%	0.0%	2.8%	0.0%
Jogging	0.0%	0.0%	0.0%	82.6%	8.3%	9.0%
Running	0.0%	0.0%	0.0%	19.4%	70.8%	9.7%
Walking	0.0%	0.0%	0.0%	3.5%	4.9%	91.7%

Tabela 4.9: Confusion matrix for Lucas and Kanade method using the best configuration (window 4×30, 33 bins).

	Boxing	Handclapping	Handwaving	Jogging	Running	Walking
Boxing	95.1%	4.2%	0.7%	0.0%	0.0%	0.0%
Handclapping	1.4%	81.2%	17.4%	0.0%	0.0%	0.0%
Handwaving	6.2%	7.6%	85.4%	0.0%	0.7%	0.0%
Jogging	0.0%	0.0%	0.0%	80.6%	12.5%	6.9%
Running	0.0%	0.0%	0.0%	18.1%	79.2%	2.8%
Walking	0.0%	0.0%	0.0%	14.6%	3.5%	81.9%

Tabela 4.10: Confusion matrix for Horn and Schunck method using the best configuration (Window 5×30, 36 bins, $\alpha = 2.0$).

We further analyse the performance of our method in the problem of human action recognition with the challenging database Hollywood2. The computations were performed using the best configuration found for each method in the KTH database. Table 4.11 shows the achieved results, where one may note that our method outperform the classical methods. We did not perform more tests in Hollywood2 database due to the high computational cost of this database.

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

Tabela 4.11: Results in the Hollywood2 database.

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 database in an Intel® Xeon® 2.20GHz processor with 32GB of memory. For the classical methods, we computed for the same database with 21 frames per second for the Horn and Schunck and 25 for the Lucas and Kanade methods. Using 35 videos from Hollywood database 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 databases, respectively.

5 CONCLUSION

In this work we presented a differential optical flow method based on the energy of wave equation. The 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. The decoupling makes our approach local and suitable for parallelization.

Our method is not able to calculate the flow in all the image points. 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 in the points that satisfies them. Therefore, the resulting flow field is sparse. In spite of the sparsity, we have high quality flow in the detected points.

In order to evaluate the performance of our method, we performed comparative tests with the classical Horn e Schunck (1981) and Lucas e Kanade (1981) methods. This choice is justified by the fact that these classical approaches are basis of recent differential methods (RASHWAN et al., 2013; BRUHN et al., 2005). For the performance measures proposed by Baker et al. (2011), we verified that our method performs similar results to the classical ones.

Our flow was applied in the problem of human action recognition using histograms of optical flow. The best result of our method in the KTH database 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 (PEREZ et al., 2012; MOTA et al., 2013). These recognition ratios 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 $0.6 < \alpha^2 < 2.6$.

For future works, we intend to include global elements in the flow computation, which

could reduce the sparsity of the flow. The first attempt was to constrain the flow rotational in the energy functional. However, this approach did not converge since the convergence condition depends on all the points of the image. We also want to further study the application of our optical flow method in the human action recognition problem. Another possibility is to use our convergence conditions as a feature detector.

REFERÊNCIAS

ADELSON, E. H.; BERGEN, J. R. Spatiotemporal energy models for the perception of motion. **J. OPT. SOC. AM. A**, v. 2, n. 2, p. 284–299, 1985.

BAKER, S.; SCHARSTEIN, D.; LEWIS, J. P.; ROTH, S.; BLACK, M. J.; SZELISKI, R. A database and evaluation methodology for optical flow. In: **In Proceedings of the IEEE International Conference on Computer Vision**, 2007.

BAKER, S.; SCHARSTEIN, D.; LEWIS, J. P.; ROTH, S.; BLACK, M. J.; SZELISKI, R. A database and evaluation methodology for optical flow. **Int. J. Comput. Vision**, Kluwer Academic Publishers, Hingham, MA, USA, v. 92, n. 1, p. 1–31, mar. 2011. ISSN 0920-5691. Disponível em: <http://dx.doi.org/10.1007/s11263-010-0390-2>.

BARNARD, S. T.; THOMPSON, W. B. Disparity analysis of images. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, Los Alamitos, CA, USA, v. 2, n. 4, p. 333–340, 1980. ISSN 0162-8828.

BOUGUET, J.-Y. **Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm**, 2000. Disponível em: <http://robots.stanford.edu/cs223b04/algo_tracking.pdf>.

BROX, T.; BRUHN, A.; PAPENBERG, N.; WEICKERT, J. **High accuracy optical flow estimation based on a theory for warping**, May 2004. 25-36 p. (Lecture Notes in Computer Science, v. 3024). Disponível em: <http://lmb.informatik.uni-freiburg.de//Publications/2004/Bro04a>.

BROX, T.; MALIK, J. **Large displacement optical flow: descriptor matching in variational motion estimation**. 2011. 500-513 p. Disponível em: <http://lmb.informatik.uni-freiburg.de//Publications/2011/Bro11a>.

BRUHN, A.; WEICKERT, J.; SCHNöRR, C. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. **International Journal of Computer Vision**, v. 61, p. 211–231, 2005.

DUYNE, S. A. V.; III, J. O. S.; VAN, S. A.; JULIUS, D.; III, O. S. **Physical Modeling with the 2-D Digital Waveguide Mesh**. 1993.

EFROS, A. A.; BERG, A. C.; BERG, E. C.; MORI, G.; MALIK, J. Recognizing action at a distance. In: **In ICCV**, 2003. p. 726–733.

FARNEBÄCK, G. Very high accuracy velocity estimation using orientation tensors, parametric motion, and simultaneous segmentation of the motion field. In: **Proceedings of the Eighth IEEE International Conference on Computer Vision**, 2001. I, p. 171–177.

FLEET, D. J.; JEPSON, A. D. Computation of component image velocity from local phase information. **International Journal of Computer Vision**, v. 5, n. 1, p. 77–104, 1990. Disponível em: <http://dblp.uni-trier.de/db/journals/ijcv/ijcv5.html#FleetJ90>.

FONTANA, F.; ROCCHESSO, D. A New Formulation of the 2D-Waveguide Mesh for Percussion Instruments. In: **Proceedings of XI Colloquium on Musical Informatics 1995**, 1995. p. 27–30.

GIANCOLI, D. **Physics for scientists and engineers with modern physics**, 1989. (Physics for Scientists and Engineers with Modern Physics, v. 1). ISBN 9780136666523. Disponível em: <http://books.google.com.br/books?id=2USuwjNEbw4C>.

GIROSI, F.; VERRI, A.; TORRE, V. Constraints for the computation of optical flow. In: IEEE. **Visual Motion, 1989., Proceedings. Workshop on**, 1989. p. 116–124.

HAUSSECKER, H. W.; FLEET, D. J. Computing optical flow with physical models of brightness variation. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 23, n. 6, p. 661–673, jun. 2001. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/34.927465>.

HORN, B. K. P.; SCHUNCK, B. G. Determining optical flow. **ARTIFICAL INTELLIGENCE**, v. 17, p. 185–203, 1981.

HWANG, S. H.; LEE, S. U. A hierarchical optical flow estimation algorithm based on the interlevel motion smoothness constraint. **Pattern Recognition**, v. 26, n. 6, p. 939–952, 1993. Disponível em: <http://dblp.uni-trier.de/db/journals/pr/pr26.html#HwangL93>.

LANCZOS, C. **The Variational Principles of Mechanics**, 2012. (Dover books on physics and chemistry). ISBN 9780486134703. Disponível em: <http://books.google.com.br/books?id=ZWoYYr8wk2IC>.

LAPTEV, I.; MARSZAŁEK, M.; SCHMID, C.; ROZENFELD, B. Learning realistic human actions from movies. In: **Conference on Computer Vision & Pattern Recognition**, 2008. Disponível em: <http://lear.inrialpes.fr/pubs/2008/LMSR08>.

LATTER, J. Tsunamis of volcanic origin: Summary of causes, with particular reference to Krakatoa, 1883. **Bulletin of Volcanology**, v. 44, n. 3, p. 467–490, 1981. Disponível em: <http://dx.doi.org/10.1007/bf02600578>.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In: **Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2**, 1981. (IJCAI'81), p. 674–679. Disponível em: <http://dl.acm.org/citation.cfm?id=1623264.1623280>.

MARSZAŁEK, M.; LAPTEV, I.; SCHMID, C. Actions in context. In: **Conference on Computer Vision & Pattern Recognition**, 2009. Disponível em: <http://lear.inrialpes.fr/pubs/2009/MLS09>.

MOTA, V. F.; PEREZ, E. A.; MACIEL, L. M.; VIEIRA, M. B.; GOSSELIN, P. H. A tensor motion descriptor based on histograms of gradients and optical flow. **Pattern Recognition Letters**, ago. 2013. ISSN 01678655. Disponível em: <http://dx.doi.org/10.1016/j.patrec.2013.08.008>.

MOTA, V. F.; PEREZ, E. de A.; VIEIRA, M. B.; MACIEL, L. M.; PRECIOSO, F.; GOSSELIN, P. H. A tensor based on optical flow for global description of motion in videos. In: **SIBGRAPI**, 2012. p. 298–301. ISBN 978-1-4673-2802-9. Disponível em: <http://dblp.uni-trier.de/db/conf/sibgrapi/sibgrapi2012.html#MotaPVMPG12>.

MYINT-U, T.; DEBNATH, L. **Linear Partial Differential Equations for Scientists and Engineers**, 2007. ISBN 9780817645601. Disponível em: <http://books.google.com.br/books?id=Zbz5_UvERIIC>.

NGUYEN, D. D.; JEON, J. W. Tuning optical flow estimation with image-driven functions. In: **ICRA**, 2011. p. 4840–4845. Disponível em: <http://dblp.uni-trier.de/db/conf/icra/icra2011.html#NguyenJ11>.

PEREZ, E. de A.; MOTA, V. F.; MACIEL, L. M.; SAD, D. O.; VIEIRA, M. B. Combining gradient histograms using orientation tensors for human action recognition. In: **ICPR**, 2012. p. 3460–3463. ISBN 978-1-4673-2216-4. Disponível em: <http://dblp.uni-trier.de/db/conf/icpr/icpr2012.html#PerezMMSV12>.

RASHWAN, H. A.; GARCÍA, M. A.; PUIG, D. Variational optical flow estimation based on stick tensor voting. **IEEE Transactions on Image Processing**, v. 22, n. 7, p. 2589–2599, 2013.

SAKAINO, H. Fluid motion estimation method based on physical properties of waves. In: **CVPR**, 2008.

SANTEN, J. P. H. van; SPERLING, G. Elaborated reichardt detectors. **J. Opt. Soc. Am. A**, OSA, v. 2, n. 2, p. 300–320, Feb 1985. Disponível em: <http://josaa.osa.org/abstract.cfm?URI=josaa-2-2-300>.

SCHULDT, C.; LAPTEV, I.; CAPUTO, B. Recognizing human actions: A local svm approach. In: **Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03**, 2004. (ICPR '04), p. 32–36. ISBN 0-7695-2128-2. Disponível em: <http://dx.doi.org/10.1109/ICPR.2004.747>.

WANG, H.; KLASER, A.; SCHMID, C.; LIU, C.-L. Action recognition by dense trajectories. In: **Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition**, 2011. (CVPR '11), p. 3169–3176. ISBN 978-1-4577-0394-2. Disponível em: <http://dx.doi.org/10.1109/CVPR.2011.5995407>.