Finding the Exact Optical Flow: a Maximum Flow Formulation

Martin Bonneville Jean Meunier
Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128, Succ. Centre-Ville
Montréal(Qc), Canada, H3C 3J7
$<$bonnevil, meunier$>$@iro.umontreal.ca

Abstract

This paper introduces a new efficient method for finding the optical flow for a special type of motion and smoothness constrain. The optimisation is mapped into a maximum-flow problem in a graph, which is solved efficiently and leads to an optimal solution. Previously, similar methods were used to solve stereo correspondence [11, 9] and maximum *a posteriori* estimation problem in Markov random fields [2, 3, 6, 10].

After a general introduction to optical flow, we present a modified version of Horn and Schunck's method [7], then we present how we map the optical flow optimisation problem into a flow graph. Experimental results on simulated data are then provided and compared to the modified Horn and Schunck's method. They clearly show the potential of our formulation with normalised errors of 9.90% with our method vs. 19.87% with the Horn and Schunck's method. Both algorithms were also tested with realistic data in cineangiography of artery to assess blood flow. Again, our method shows a better velocity profile assessment (1.97% vs. 8.32% of error). These results were expected because the maximum-flow formulation is not iterative and leads to a global optimum.

1 Introduction

There is an abundant literature about the computation of the optical flow (see Barron *et al.* [1] for a summary of the major methods). In this paper, we focus our attention on the optimisation involved in a special gradient-based method and show how to map the optimisation into a flow problem in a graph. Similar works have been done in the field of stereo correspondence [3, 9, 11], image segmentation [10] and maximum *a posteriori* estimation in Markov random fields [2, 3, 6]. Before getting to our method, we introduce the basics of gradient-based optical flow calculation [1, 7], then we present a modified version of Horn and Schunck's algorithm [7] and finally, we proceed with our new method based on a maximum-flow formulation.

2 Optical flow

A 2-D sequence of two or more images is mathematically described as a function $I(i,t)$, where I is the image intensity at time t and at position $i = (x,y) \in \mathcal{S}$ and $\mathcal{S} = \{0,\ldots,m-1\}$ is the set of all pixels. By the chain rule of derivation, we obtain a formulation for the total rate of change of brightness

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} \tag{1}$$

where $\partial I/\partial x$, $\partial I/\partial y$ and $\partial I/\partial t$ can be computed directly from a pair of images $I(i,t)$ and $I(i,t+\Delta t)$ and will be abbreviated by I_x, I_y and I_t. It remains to determine the brightness change dI/dt and the x and y components of velocity, denoted by $u = dx/dt$ and $v = dy/dt$ respectively. To solve equation (1), for the velocities (u,v) and the brightness change dI/dt, additional constraints must be applied to restrict the allowable motions. Such constraints are constructed according to prior knowledge we have about the real motion and brightness change. Horn and Schunck [7] suppose the intensity is conserved, so the brightness change of a particular point in the image is equal to $dI/dt = 0$. Equation (1) is rewritten as

$$\frac{dI}{dt} = I_x u + I_y v + I_t = 0 \tag{2}$$

Because this is a single equation with two unknowns, Horn and Schunck [7, 1] added a smoothness constraint to solve the optical flow. We now proceed with our modified version of their algorithm.

2.1 Modified Horn and Schunck's method

In our applications, we set $v = 0$ because there is no motion along the y-axes. Equation (2) is reformulated as

$$I_{x_i} u_i + I_{t_i} = 0 \tag{3}$$

Let $f_i = ((u_i, 0))^T$ designate the optical flow vector at pixel i. Thus, $f = \{f_i \mid i \in \mathcal{S}\}$ is the optical flow field to compute. Equation (3) shows that we can find a unique solution by setting $u_i = -I_{t_i}/I_{x_i}$ (if $I_{x_i} \neq 0$). But, when noise is taken into consideration, additional constraints must be imposed on the flow in order to find a solution that reflects some prior knowledge about the real flow field we are seeking. Thus, we should instead minimise

$$V(I_i \mid f_i) = (I_{x_i} u_i + I_{t_i})^2 \tag{4}$$

where $V(I_i \mid f_i)$ is the likelihood potential of I_i knowing the motion f_i. An important constraint reflecting prior knowledge about the field is smoothness. The flow at nearby pixels in the image should be similar because the motion of intensity points of an object presents some coherence and therefore the flow in a neighbourhood should change smoothly. One way to ensure such coherence is to constrain the local change in velocity by minimising the squared magnitude of the x-derivative of the velocity over the whole set of pixels S:

$$V(f_i) = (\frac{\partial u_i}{\partial x})^2 \tag{5}$$

where $V(f_i)$ is called the prior potential. The posterior potential is obtained by combining equation (4) and equation (5) into a weighted sum,

$$V(f_i \mid I_i) = V(I_i \mid f_i) + \beta V(f_i) \tag{6}$$

The problem of finding the optical flow can be reformulated as the minimisation of the posterior energy, which is the integration, over all pixels $i \in \mathcal{S}$, of the posterior potential described by equation (6)

$$E(f \mid I) = \int_{\mathcal{S}} V(f_i \mid I_i)\, d\mathcal{S} \tag{7}$$

A solution to this minimisation problem is obtained by variational calculus and an iterative form of the solution is [7]

$$u_i^{k+1} = \overline{u}_i^k - \frac{(\overline{u}_i^k I_{x_i} + I_{t_i}) I_{x_i}}{\beta + I_{x_i}^2} \tag{8}$$

where $\overline{u}_i^k$ is the local average (excluding the current pixel velocity u_i) of the velocity in a neighbourhood centred at pixel i at the kth iteration. Because of the iterative aspect of the algorithm, we have set $u_i^0 = -I_{t_i}/I_{x_i}$. This is correct, because it corresponds to the minimum of the likelihood potential of equation (4) if we suppose no smoothness (i.e. we start with the most noisy flow field and we later smooth it).

There are many drawbacks in the solution provided by the iterative form in equation (8). First, there is no guarantee of reaching the global minimum of the posterior energy of equation (7). As a result, there will be a lack of global smoothness in the flow field (in our application to the calculation of the blood flow in artery, it is crucial to have global smoothness). Second, we need to set the number of iteration or some stopping criteria and we don't have any guarantee of convergence.

In the next section, we present a new method to compute the global minimum of the posterior energy with a slightly different prior potential.

3 Maximum-flow method

In our new method, we use a graph formulation of the problem of finding a suitable motion field. In this way, we have to bound and discretize the values u can take. This is correct, because differential optical flow methods are only valid to recover small motion. We define $\mathcal{U} = \{u : u = u_{min} + h\Delta u, h = 0, \ldots, M-1 = (u_{max} - u_{min})/\Delta u\}$ as the set of M finite values u can take.

Because of some limitations of our algorithm, we cannot use the same smoothness constraint as the one described in equation (5). Instead, we constrain the local change in velocity by minimising a weighted sum of the absolute value of the x and y components of the velocity gradient:

$$V(f_i) = C_1 |\frac{\partial u_i}{\partial x}| + C_2 |\frac{\partial u_i}{\partial y}|$$

where C_1 and C_2 are positive constants. A discrete formulation of this prior potential is then

$$V_{i,i'}(f_i, f_{i'}) = \beta(i,i') |u_i - u_{i'}| \tag{9}$$

where $\beta(i,i')$ is a positive function defined on the neighbourhood system $\mathcal{N} = \{\mathcal{N}_i : i \in \mathcal{S}\}$, where $\mathcal{N}_i$ is the set of pixels in the neighbourhood of pixel i, such that $i \notin \mathcal{N}_i$.

The problem of finding a suitable u can be restated as the minimisation of the following posterior energy

$$\begin{aligned} E(f \mid I) \quad = \quad & \sum_{i \in \mathcal{S}} \sum_{i' \in \mathcal{N}_i} \beta(i,i') |u_i - u_{i'}| \quad + \\ & \sum_{i \in \mathcal{S}} V(I_i \mid f_i) \end{aligned} \tag{10}$$

3.1 Maximum-flow formulation of the energy global optimisation

In this section, we will show how a particular flow problem in a graph can be used to optimise globally the posterior energy function of equation (10). Recall that the objective in a maximum-flow problem is to maximise the flow between two points, the source s and the sink t via various intermediate junctions. A maximum-flow problem can be modelled by a graph (see figure 1). The source, the sink and junctions are represented by vertices, while the edges represent

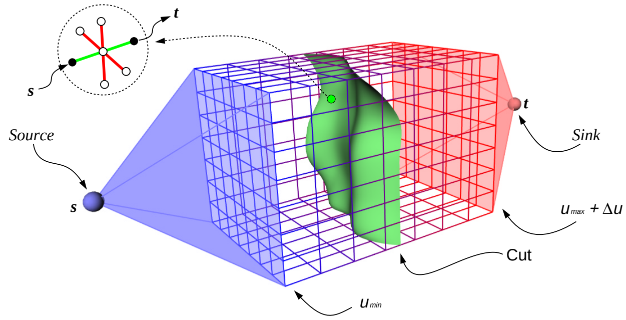

Figure 1: flow graph used to minimized the posterior energy funciton of equation (10). The source s is connecte to all assignments with velocity $u = u_{min}$ and the sink t is connected to all assignments with velocity $u = u_{max} + \Delta u$. The neighborhood system $\mathcal{N}$ used in this figure the 4-neighbors neighbohood system.

the "pipes" through which "material" flows. The capacity of an edge represents the maximum amount of flow that can go through it.

A cut through a flow graph is any set of edges that contains at least one edge from every path from the source to the sink (see figure 1). The cut value is simply the sum of the flow capacities of the cut edges. An important theorem [4] relates the cut value and the maximum flow, it states that the edges saturated by the maximum flow through the graph will form the minimum cut. Therefore, from now on we will focus on the transposition of the minimisation of equation (10) into a minimum cut problem. This problem will be solved later on with a maximum flow algorithm (see section 3.4).

In the following sections we will describe with more details the minimum cut problem adapted to our specific needs.

3.2 The graph

Consider the graph $G = (V, E)$ depicted in figure 1. The vertex set V is defined as

$$V = V' \cup \{s, t\}$$

where s is the source, t is the sink and V' is the set of all possible assignments,

$$V' = \{(i, u) : i \in \mathcal{S}, u \in \mathcal{U} \cup \{u_{max} + \Delta u\}\}$$

The set of edges E is split into three different types. First, the source is connected to all assignments with value u_{min} and the sink to all assignments with value $u_{max} + \Delta u$. The second set $E_{penalty}$ contains the penalty edges, they are defined by the neighbourhood system $\mathcal{N}$ that establishes the connections between corresponding assignments (vertices) with the same value u. The third set E_{motion} contains the motion edges. They correspond to single forward connection between a vertex with value u, to another one with value $u + \Delta u$ at the same pixel. The set of edges is thus defined as

$$\begin{aligned} E &= \{(s, (i, u_{min})) : i \in \mathcal{S}\} \\ &\cup \{((i, u_{max} + \Delta u), t) : i \in \mathcal{S}\} \\ &\cup E_{penalty} \cup E_{label} \end{aligned}$$

with

$$\begin{aligned} E_{penalty} &= \{((i, u), (i', u)) : i, i' \in \mathcal{S}, i' \in \mathcal{N}_i, u \in \mathcal{U}\} \\ E_{motion} &= \{((i, u), (i, u + \Delta u)) : i \in \mathcal{S}, u \in \mathcal{U}\} \end{aligned}$$

3.3 Edge capacity

One way to map the minimisation of the energy function of equation (10) into a flow graph problem is to set the edge capacities properly. First, we set the capacity of edges directly connected to the source and the sink to infinity, $cap(s, (i, u_{min})) = cap((i, u_{max} + \Delta u), t) = \infty$. Now, it remains to define the capacities of motion edges and penalty edges.

3.3.1 Motion edges

The role of motion edges is to model the likelihood potential of equations (10) and (4). We will set their capacities to $cap((i, u), (i, u + \Delta u)) = V(I_i \mid f_i = u)$. This way, cutting a motion edge $((i, u), (i, u + \Delta u))$ corresponds to assign value u to pixel i.

3.3.2 Penalty edges

The penalty edges incorporate the smoothness constraint we have described by the prior potential $V_{i,i'}(f_i, f_{i'})$ defined in equation (9). In this way, if two neighbouring pixels are assigned a different motion value ($f_i = u$ and $f_{i'} = u'$, $u \neq u'$), the prior potential will be $\beta(i, i')|f_i - f_{i'}| = \beta(i, i')|u - u'|$. Therefore, we can define the capacity of a penalty edge as $cap((i, u), (i', u)) = \beta(i, i')$, where $i' \in \mathcal{N}_i$.

3.4 Complexity of the maximum-flow algorithm

As mentioned before, a minimum cut separating the source s from the sink t can be efficiently obtained by computing the maximum flow of the graph. The minimum cut will be the set of edges saturated by the maximum flow and will contain the configuration that minimises the posterior energy in equation (10).

To compute the maximum flow in the graph, we use the *preflow-push relabel* algorithm (see [4, 5]). A typical graph to compute the MAP-MRF contains v vertices equal to the number of pixels multiplied by the number of discretized velocities. In our application, $v = mM$ and the number of edges e depends on the neighbourhood system $\mathcal{N}$, which is fixed. Thus, the number of edges is proportional to the number of vertices, $e = O(v) = O(mM)$. The worst case performance of the algorithm is then

$$O(ve\,log(v^2/e)) = O(m^2M^2\,log(mM))$$

Moreover, the average performance, estimated as $O(m^{1.3}M^{1.3})$ (see [5]), is even better because of the simple graph topology (a mesh) and locality of node connections.

4 Experiments and results

In this section, we compare the performance of our maximum-flow algorithm to compute the optical flow to the modified Horn and Schunck's method presented in section 2.1. Because we only need to find the x component of velocity (u), a suitable measure of performance is the normalised square error between the computed value of u and its real value.

$$error = \frac{\sum_{i\in\mathcal{S}}(u_i - u_i^{real})^2}{\sum_{i\in\mathcal{S}}(u_i^{real})^2} \times 100 \qquad (11)$$

For example, with this error measure, a motion field equals to $2u^{real}$ or 0 gives a 100% error.

First, we compare performances over simulated images (see section 4.1) and then we show an application to the computation of the blood flow in artery.

4.1 Validation over simulated deformation

To assess the usefulness of our method, we have generated a shearing deformation (see figure 2b) and applied it to a 64×64 image of Lenna (see figure 2a). In this experiment, we have bounded the motion to be $u \in [-10, 10]$ with a precision of $\Delta u = 1/100$ and the neighbourhood system $\mathcal{N}$ is fixed to the 4-neighbors system. Because the motion is less dependant along the y-axis, we set $\beta(i,i') = 0$, that is, when $i' \in \mathcal{N}_i$ and i' is north or south of pixel i. For the other pixels in $\mathcal{N}_i$, $0 < \beta(i,i') < \infty$. In figure 2c, we show the trivial solution $u = -I_t/I_x$, if $I_x \neq 0$. In figure 2d, we show the best result obtained by the modified Horn and Schunck's algorithm ($\beta = 50$ and 1000 iteration steps). Finally in figure 2e, we present the result obtained from the maximum-flow calculation of the optical flow, with $\beta(i,i') = 1$. The global error in the calculation of u is 69.61% for the trivial solution, 19.87% for the modified Horn and Schunck's approach and our optimal maximum-flow method performed well with only a 9.90% error.

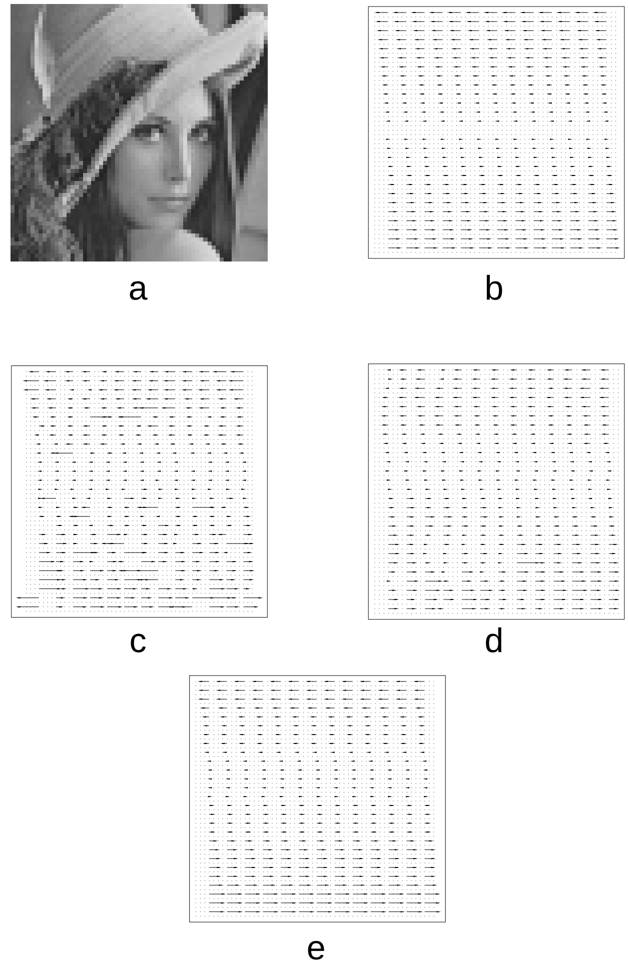

Figure 2: Shearing motion results on Lenna, a) original 64×64 image of Lenna, b) ground truth shearing motion field, c) trivial solution: $u = -I_t/I_x$, d) modified Horn and Schunck's solution and e) results of the maximum-flow approach.

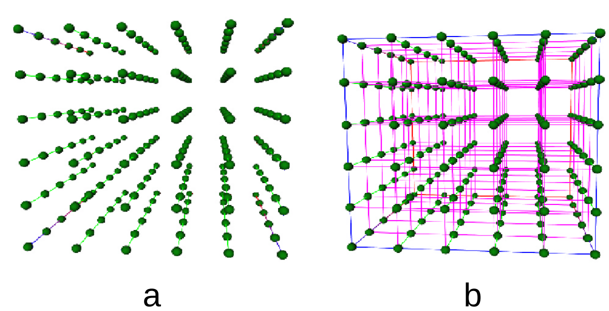

Figure 3: Configuration of the graph G for a) the straigth forward solution $u = -I_t/I_x$ (i.e.: $\beta(i,i') = 0$) and b) the minimization of the posterior energy funciton ($\mathcal{N}$ is the 4-neighbors system).

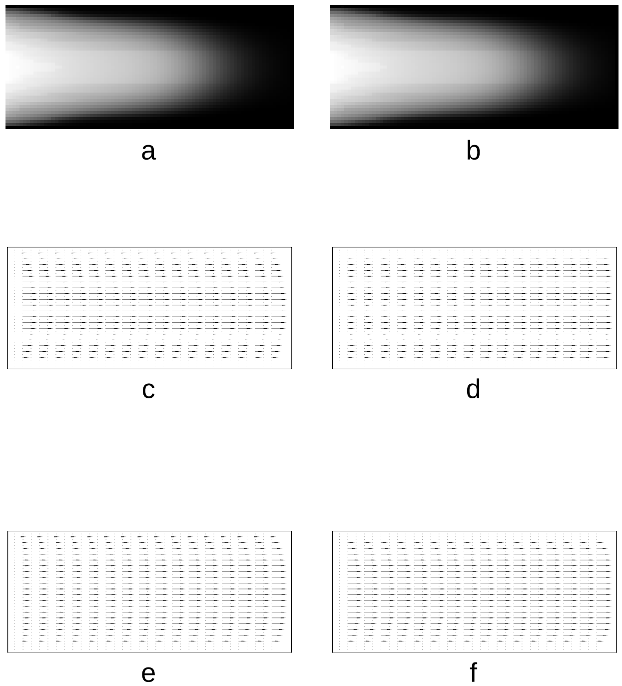

Figure 4: Results on blood flow in artery, a) image of the contrast material at time $t = 0$, b) image of the advanced contrast material at time $1/100$ sec., c) ground truth blood flow in the artery between the two images, d) results of the trivial solution ($u = -I_t/I_x$), e) results of the modified Horn and Schunck's method and f) results of the maximum-flow calculation of the optical flow.

4.2 Interpretation of the smoothness parameter

Setting $\beta = 0$ in both prior potential of equation (5) and (9) corresponds to assuming no smoothness and this is equivalent to set $u = -I_t/I_x$ for all $i \in \mathcal{S}$ if $I_x \neq 0$. In our maximum flow (see figure 3b), it is equivalent to remove the set of penalty edges $E_{penalty}$ from the whole set of edges of graph G and we obtain the graph of figure 3a.

Conversely, setting $\beta = \infty$ in our maximum flow methods leads to the average motion field (flat solution).

4.3 Application to blood flow computation in cineangiography of arteries

To test the algorithm in a more realistic situation, we used an image formation model presented elsewhere [8] for angiographic (X-ray) imaging of artery using the physical equations governing blood dynamics and contrast material dispersion (the contrast material which is opaque to X-ray is necessary to "see" the blood motion, otherwise blood would be invisible). In figure 4a and b, two such images show the blood (contrast material) advance in a vessel at two different instants in time. The vessel diameter is 4 mm, the image size is $10\text{mm} \times 4\text{mm}$ corresponding to 82×41 pixels (pixels are not squares) and the time sampling interval is 1/100 sec. Using these images, it is easy to compare the optical-flow-computed blood flow with the true flow used for the simulations. The theoretical flow is actually parabolic and the maximum (image) velocity in the vessel is 2 cm/sec. First, we show the result of the trivial solution (figure 4d), this is followed by figure 4e, which present the motion field computed with the modified Horn and Schunck's algorithm and then we present the maximum-flow calculation of the optical flow (figure 4f). The errors are respectively 11.25%, 8.32% and 1.97% showing again the superiority of the maximum-flow approach. The maximum velocity computed with our algorithm is 1.69 pixel (2.06 cm/sec) which is about a 3.05% error from the real maximum speed of 1.64 pixel (2.00 cm/s).

5 Discussion and conclusion

In this paper, we have presented a new method for computing the optical flow from a sequence of images. This method provides a scheme to compute exactly the global minimum of the posterior energy of equation (10). Another major advantage is that the solution is direct (not iterative) and thus we don't have to set the number of iterations or define a stopping criterion. However, in its current formulation, the algorithm is limited to one-component optical flow.

Notice that the 2D standard Horn and Schunck's algorithm would perform worse then the modified version of the algorithm, because it is less constrained ($v \neq 0$ in our case $v = 0$ which is *a priori* knowledge).

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. System and experiment, performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[2] M. Bonneville, S. Roy, J. Meunier, and A. C. Evans. Using maximum flow for exact maximum a posteriori estimation in image restoration and segmentation. In *Submitted to ICIAP'99*.

[3] Y. Boykov, O. Veksle, and R. Zabih. Markov random fields with efficient approximations. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. McGraw-Hill, New York, 1990.

[5] A. V. Golberg and S. B. Rao. Length function for flow computation. Technical Report 97-055, NEC Research Institute, Princeton NJ, 1997.

[6] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51(2):271–279, 1989.

[7] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[8] B. Imbert, J. Meunier, R. Mongrain, and J. Hudon. Stenosis parameter assessment from contrast medium tracking in cineangiography with an optical flow method. In *SPIE Medical Imaging 1997*, pages 631–640, Newport Beach, CA, 1997.

[9] H. Ishikawa and D. Geiger. Occlusions, discontinuities, and epipolar lines in stereo. In *Fifth European Conference on Computer Vision*, Freiburg, Germany, 1998.

[10] H. Ishikawa and D. Geiger. Segmentation by grouping junctions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, 1998.

[11] S. Roy and I. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *6th International Conference on Computer Vision*, Bombay, India, 1998.