Optical Flow Field Estimation With Markov Random Fields Using Evolutionary Algorithms

S. Voisin, A. Dipanda and C. Bourgeois-République
Université de Bourgogne, Laboratoire LE2I (CNRS UMR 5158),
BP 47870 – 21078 DIJON CEDEX, FRANCE
adipanda@u-bourgogne.fr

Abstract

This paper presents a new method for estimating the optical flow field using the MRF modeling. In the MRF framework, the estimation problem amounts to the minimization of an energy function. We propose an Evolutionary Algorithm (EA) method to solve this minimization problem. It is based on a divide-and-conquer strategy which adequately uses the markovian property. Experimental results show the effectiveness of the method.

1. Introduction

Optical flow is the distribution of apparent velocities of movement of the brightness patterns in an image. Several models of optical flow estimation are proposed in the literature [1]. Methods which represent the optical flow with at least one independent motion vector per pixel, the *dense motion field*, generally use the spatio-temporal variations of the brightness in the image sequence. Most of them use a constraint based on the gray level variations with the assumption that, under ideal conditions, the intensity of a pixel is constant between two successive images.
In this paper, the optical flow is estimated by a bayesian approach. The Markov Random Fields (MRFs) are used to model the motion field. MRFs are quite suitable for modeling the global and local properties in an image. Optical flow estimation modeled through MRFs leads to the minimization of a global energy function which specifies nonlinear interactions between different image features (gray levels in two successive images, motion vectors). Generally this is a difficult problem due to the fact that it involves a large number of unknowns and the function has many local minima. This minimization can be achieved either by stochastic relaxation algorithms such as the simulated annealing [2] or by deterministic relaxation algorithms for example the Iterative Conditional Modes (ICM) method [3] Deterministic algorithms can be missed towards a local optimum while stochastic algorithms converge to the optimal solutions but require much computational time.
In this study, we propose to use Evolutionary Algorithms (EAs) to solve this minimization problem. EAs are a type of stochastic search method whose functioning is inspired by natural selection and the principles of evolution [4]. EAs have received a great deal of attention in the recent past and are widely used in diverse areas of image processing [5-7]. However, to my knowledge, they have never been used to estimate the dense motion field. Our goal is to show that by suitably exploiting the markovian property, EAs can successfully be used to estimate motion.

2. Background on MRFs

2.1. Motion estimation modeling with MRFs

Let $S=\{s_1,s_2,....,s_N\}$ be a finite set of sites of an image defined on a *2-D* lattice and η a neighborhood system on it (see Figure1a). Let $D=\{\vec{d}=(d_u,d_v)^T\}$ be a finite set of discrete-value vectors, i.e. d_u and d_v take discrete values in the range $[-d_{max},d_{max}]$. Let $y=(y_t,y_{t+dt})$ represent the two successive images in an image sequence at times t and $t+dt$. The goal of the motion estimation process is to calculate a displacement field label $x=\{\vec{d}_s, s\in S\}$ on the basis of observation y. In this last expression, $\vec{d}_s=(d_1,d_2)^T$ is a 2-D vector attributed to the site $s=(s_1,s_2)$ in the image y_t when this one has moved to the site $r=(s_1+d_1, s_2+d_2)$ in the image y_{t+dt}.
Assuming Markov properties of observations and motion labels implies that the interactions between

the different variables on each site of S remain local with respect to the neighborhood system η (see [3] for a detailed presentation of MRFs). The Hammersley-Clifford theorem allows to write $P(x/y)$ as a Gibbs distribution in the following form:

$$P(x/y)=\frac{1}{Z}\exp(-U(x/y)) \quad (1)$$

where Z is a normalizing constant called the *partition function* and $U(x/y)$ is an energy function defined by:

$$U(x/y)=\sum_{c\in C}V_c(x,y) \quad (2)$$

In this formula, V_c is the potential function associated with the clique c which consists of either a single site or a set of neighboring sites, and C is the set of all the cliques derived from the neighborhood system η (see Figure 1).

With the *Maximum A Posteriori* (MAP) estimation criterion, the most likely displacement field label x^* must maximize the conditional probability $P(x/y)$ which, from Baye's rule, is proportional to $P(y/x)P(x)$. Thus the following optimization problem has to be solved:

$$\max_x(P(y/x)P(x)) \quad (3)$$

Within this framework, finding the MAP estimate can be expressed by the following equation:

$$\max_x(\exp\{-(U(y/x)+U(x))\}) \quad (4)$$

Finally, the displacement field label estimation problem amounts to the minimization of a global energy function composed of two terms:

$$U(x/y)=U(y/x)+U(x) \quad (5)$$

The first term is referred to as the *brightness constraint* and is related to the constancy of the gray levels. It is given by:

$$U(y/x)=\sum_{s\in S}\left(DFD(s)\right)^2 \quad (6)$$

DFD is the Displaced Frame Difference defined by:

$$DFD(s)=y_t(s)-y_{(t+dt)}(s+\vec{d}_s) \quad (7)$$

The second term of the energy function is referred to as *the regularization term* and is written:

$$U(x)=\sum_{s\in S}\Big(\sum_{\{s,r\}\in C_s}\phi(\|\vec{d}_s-\vec{d}_r\|)\Big) \quad (8)$$

where C_s is the set of cliques containing the site s, ϕ is a potential function, $\vec{d}_s$ (respectively $\vec{d}_r$) is the displacement vector on sites s (respectively r), and $\|.\|$ is a norm.

In the standard regularization expressed by Tikhonov [8] the potential function ϕ is given by:

$$U(x)=\beta^2\sum_{\{s,r\}\in C_s}\|\vec{d}_s-\vec{d}_r\|^2 \quad (9)$$

where β is a weighting parameter that controls the respective contributions of the two terms $U(y/x)$ and $U(x)$ within the global energy.

Note that the global energy can be written as a sum of local energies calculated successively on each site. Thus we have:

$$U(x/y)=\sum_{s\in S}(U_s(y/x)+U_s(x)) \quad (10)$$

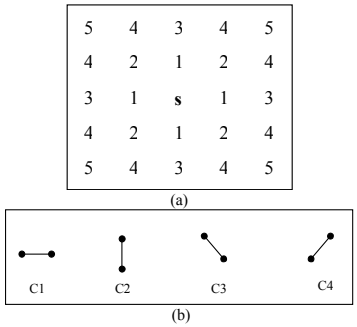

Figure 1. (a) Neighborhood system defined on the pixel s. (b) Different cliques of two elements derived from the two order neighborhood.

2.2. MAP estimation

The MAP estimation can be performed using either stochastic or deterministic methods. The simulated annealing is the most commonly used stochastic method. A temperature parameter T is introduced into the a posteriori probability as follows:

$$P(x/y)=\frac{1}{Z}\exp(-U(x/y)/T) \quad (11)$$

The process consists in generating configuration samples by using a stochastic relaxation (Gibbs sampler for example). The temperature decreases at each iteration according to a given schedule such as:

$$T_k=\varphi(T_0,k) \quad (12)$$

where k is the iteration number.

It has been proven that the process converges to the global optimum for any starting configuration if the initial temperature T_0 is sufficiently high and if the final temperature reaches 0 at an almost logarithmic rate. In practice, in order to reduce the number of iterations required for the convergence, one usually takes:

$$\varphi(T_0,k)=a^kT_0 \quad (13)$$

where the coefficient α is slightly less than 1.

A deterministic alternative to the simulated annealing method is the ICM in which the global energy is minimized by sequentially updating the different sites of the motion field. At a given location, the value assigned to a site is the one that minimizes the local energy function which depends on the visited site and also on its neighbors. The

result is a local optimum in most cases, and it depends on the initial configuration. The simplest way to initialize the process is by setting the null vector (i.e. vector $\vec{0}$) to all the displacement field.

3. Motion estimation using EAs

EAs are adaptive procedures that find solutions to problems by using an evolutionary process based on natural selection. A GA uses a fixed- or a variable-size, finite *population* of potential solutions to a problem. Each individual solution is encoded as a *chromosom*e made up of a string of *genes* which take values in either a binary or a non-binary alphabet. A GA comprises three main stages: evaluation, selection and mating. They are applied cyclically and iteratively until a saturation or other stopping criterion is satisfied. At the evaluation stage, each chromosome is assigned a *fitness* value which represents its ability to solve the problem. The fitness function directly relates to the objective function to be minimized or maximized.

3.1. Basic Evolutionary Algorithm

3.1.1. Coding and fitness: Recall that our goal is to estimate the displacement vector for each site in the image. Let NxM be the size of the images of the sequence. A chromosome corresponds to a possible solution of the optical flow and is encoded as a NxM grid of genes which are 2D vectors. Each coordinate of a gene takes an integer value in the interval $[-a, a] \subset Z$ (a is the maximum motion magnitude in the two directions of a pixel between two consecutive times.
The fitness of a chromosome corresponds to the energy calculated from the Equation 10. So, the better the chromosome, the smaller its fitness value.

3.1.2. Selection: We have implemented an elitist strategy. First, all the chromosomes of the current generation are sorted based on their fitness value. Second, some of the best chromosomes are incorporated into the next generation. The number of selected best chromosomes is calculated with respect to a rate p_s. Third, the best individual from two individuals randomly chosen from the whole current population is selected. This tournament is repeated to choose a second individual. These individuals participate in the crossover to create children which are incorporated into the next generation. The process is iterated until the next generation is complete.

(-2,1)	(1,1)	(1,1)	(0,-1)	(1,1)	(1,1)	(1,-1)	(1,-1)
(0,-1)	(0,-1)	(0,-1)	(1,1)	(0,-1)	(0,-1)	(1,-1)	(1,-1)
(0,-1)	(0,-1)	(1,1)	(1,-1)	(0,0)	(2,-2)	(2,-2)	(1,1)
(2,-2)	(1,-1)	(1,-1)	(1,-1)	(1,1)	(2,-2)	(2,-2)	(0,-1)
(2,-2)	(-2,1)	(0,0)	(2,0)	(2,1)	(0,-1)	(0,-1)	(-1,1)
(-2,1)	(2,1)	(2,1)	(0,0)	(2,1)	(0,0)	(0,0)	(0,0)
(-2,1)	(0,-1)	(1,-1)	(-1,-1)	(2,1)	(0,0)	(0,0)	(0,0)
(-2,1)	(0,-2)	(0,-2)	(0,-2)	(-1,-1)	(-2,1)	(-2,0)	(1,1)

Figure 2. Chromosome example (8x8 image size - a=2).

3.1.3. Crossover: The crossover is performed with a probability p_c. We have implemented two different crossover operators. The first one is the uniform 2D *two-point* crossover operator which swaps a randomly chosen part of two parents to produce two children (see Figure 3).
The second crossover operator is the *arithmetic crossover*. Two children are produced from two parents as follows:

$$\text{Child1}(i,j)=\alpha.\text{Parent1}(i,j)+(1-\alpha).\text{Parent2}(i,j) \quad (14a)$$

$$\text{Child2}(i,j)=(1-\alpha)\text{Parent1}(i,j)+\alpha.\text{Parent2}(i,j) \quad (14b)$$

where α is a random number, with uniform distribution in [0,1], $1 \le i \le N$ and $1 \le j \le M$.
To choose one of the two crossover operators, a low probability p_l is assigned to the arithmetic crossover.

3.1.4. Mutation: The mutation is performed on a chromosome with a probability p_m. We have implemented two mutation operators. The first one is the *two-point* mutation which swaps the loci of two randomly chosen genes. The second mutation operator is a self-adaptation mutation which performs a local minimization near a chromosome. Some genes of the chromosome to be mutated are randomly chosen for modification. For each selected gene, a new value is chosen randomly and uniformly in the range of definition, to be tested. If the local energy calculated on the image site corresponding to the gene decreases, the new gene value is maintained, otherwise the change is refused.

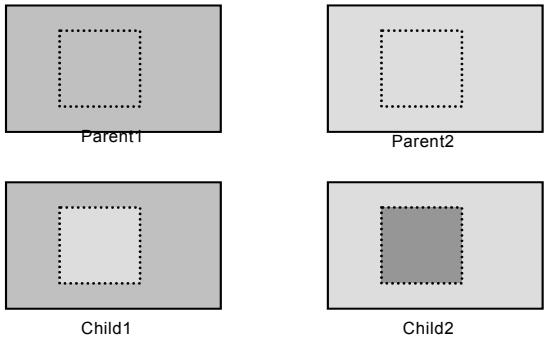

Figure 3. 2D two-point crossover.

To choose one of the two mutation operators, a low probability p_{mm} is assigned to the self-adaptation mutation while $(1-p_{mm})$ is assigned to the two-point mutation.

Applying the basic evolutionary algorithm consists in randomly generating an initial population of chromosomes, then the different steps described above are iteratively performed until a stopping criterion. The major limitation of the basic evolutionary algorithm is its relatively slow convergence. Indeed, the algorithm essentially searches through various combinations of the chromosome gene values and the number of possible combinations equals to $2(a+1)^{NxM}$. A rigorous analysis of the algorithm computational complexity is difficult to perform since the general convergence of an evolutionary algorithm is hard to quantify. However, as shown in Figure 4, the computation cost increases exponentially with the chromosome size (i.e. the image size). So, we propose a *divide-and-conquer* strategy in order to limit this computational time drawback.

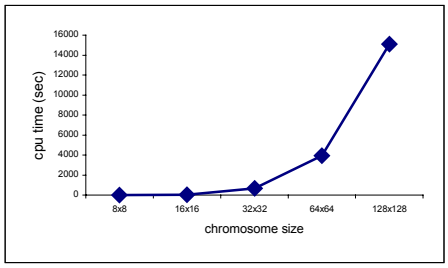

Figure 4. GA computation time versus chromosome size.

3.2 The Divide-and-Conquer strategy

The main idea of the proposed optical flow estimation strategy with GA consists in dividing the image into smaller image parts each of which is independently handled and the results are adequately combined in order to obtain the final result. This strategy is based on the markovian property of the observations and labes. Indeed, the computation of the local energy on a site depends only on its neighbors (i.e. elements of cliques that contain the site). Thus the estimation can be performed independently (i.e. in parallel) at two different sites which are not neighbors. Let *Np* be the number of image parts and let *Tp* denote the execution time needed for an image part. The global execution time for processing the whole image is equal to *TpxN,* which is linear according to *Tp*. However, because of the border effects, dividing the image will generate a mosaic effect on the final global result. Therefore at least two steps are necessary in order to solve this drawback. In the second step, the image division is shifted from the first step image parts so that by the combination of the results obtained from the two steps the gaps created in each step are filled (see Figure 5).

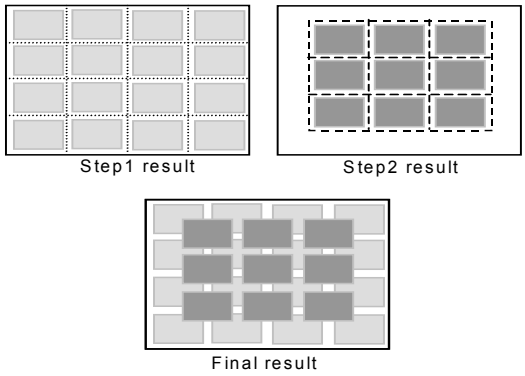

Figure 5. Scheme of the divide-and-conquer strategy.

4. Experimental results

In this section we present experimental results obtained with the proposed GA method (called DCGA) on one synthetic and two real world image sequences. In all cases, these results are compared with the ones obtained on the one hand with the ICM and on the other hand with the simulated annealing. The synthetic sequence (called *Disc-Square)* concerns two objects on a white background: a gaussian brightness disc which undergoes a rotation in the trigonometric direction, and a square which translates one pixel up and to the right (see Figure6a). The real world sequences are the well known sequences *Train* and *Interview* (see Figure6b and Figure 6c).

For the sake of visibility, the estimated dense motion fields are represented by gray level images. Figure 7 shows the different gray levels attributed to each of

the nine possible displacement vectors on a given pixel.

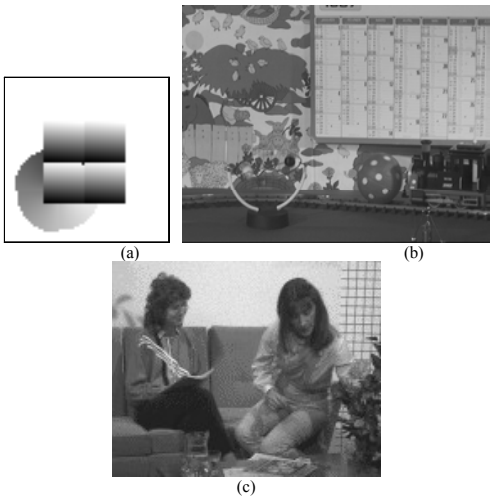

Figure 6. Single images from the respective experimental image sequences. (a) *Disc-Square* sequence; (b) *Train* sequence; (c) *Interview* sequence.

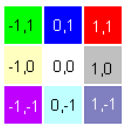

Figure 7. Gray level representations of displacement vectors.

Table 1 and Table 2 respectively give the parameter values of the DCGA and the simulated annealing.

Table 1. DCGA parameter values.

Elite rate p_s	0.04
Crossover rate p_c	0.8
Arithmetic crossover rate p_l	0.2
Mutation rate p_m	0.1
Adaptive mutation rate p_{mm}	0.1
Maximal generation number	200
Chromosome size	8x8
Population size	50

Table 2. Simulated annealing parameter values.

Decreasing coefficient α	0.9
Initial temperature (T_0)	250
Maximal iteration numbers on synthetic (and real) image sequence	200 (1000)

With the *Disc-Square* sequence, the ICM and the simulated annealing are randomly initialized, while the processes are initialized to the null displacement field with both the *Train* and the *Interview* sequences. Furthermore, the number of iterations of the ICM is not limited, but the process is stopped if the number of modified sites during the last iteration is less than 10.

Figures 8, 9 and 10 show the optical flow field results obtained from the three sequences with respectively the DCGA, the ICM and the simulated annealing. As seen in Figure 8a the *Disc-Square* DCGA result is almost perfect. The square translation and the disc rotation are well detected. Moreover as expected, no motion is estimated in the image background. However, particularly in the square, one can observe the mosaic effect due to the splitting step of the DCGA. Concerning the ICM and the simulated annealing results (respectively Figures 8b and 8c) it can be noted that in both cases the square translation is well estimated contrary to the disc rotation motion in which there are many incorrect displacement vectors. In addition, the main drawback in these two results is the random motion detected on the image background.

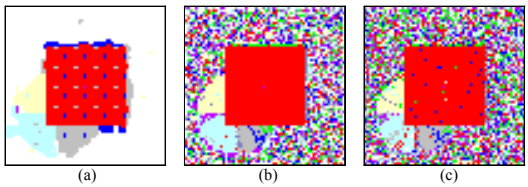

Figure 8. *Disc-Square* optical flow results. (a) DCGA result; (b) ICM result; (c) Simulated annealing result..

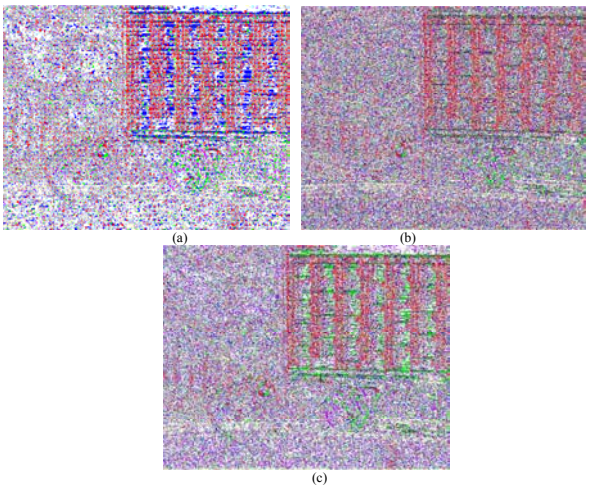

Figure 9. Results on the *Train* sequence. (a) DCGA result; (b) ICM result; (c) Simulated annealing result.

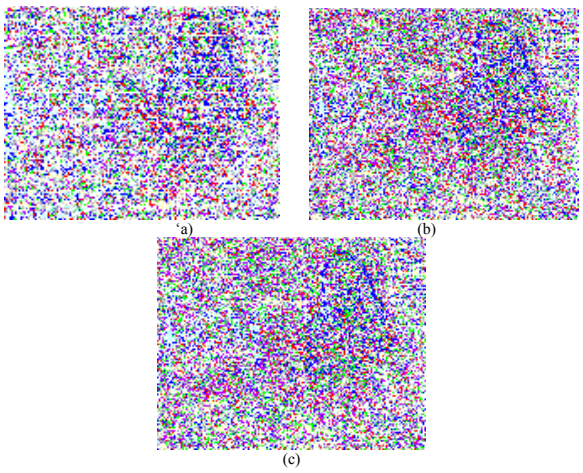

Figure 10. Results on the *Interview* sequence. (a) DCGA result; (b) ICM result; (c) Simulated annealing result.

The results on the real world sequences are more difficult to analyze (see Figures 9 and 10). Whatever the method, one can observe that the different feature motions in the images are globally well detected.

To quantify the goodness of these results, we have calculated the differences between the images at time ($t+dt$) and the images obtained from the images at time t and from the estimated motion fields. If the estimated displacement on a given site is incorrect, this difference value may not equal zero. Table 3 gives the numbers of "incorrect" pixels in the respective results. It can be observed that the number of pixels in the DCGA difference image is almost twice smaller than in the case of ICM and the simulated annealing.

Table 3. Numbers of incorrect pixels.

Methods	DCGA	ICM	Sim. Ann.
Numbers of incorrect pixels	4079	7395	7091

And finally, Table 4 shows the computation times involved by the three methods on the different image sequences. It must be noticed that as the DCGA and the Simulated Annealing are stochastic methods, thus these values are the mean values calculated from more that twenty runs. As expected, it can be observed that the ICM is much faster than the two other methods. However, the respective computation times of DCGA and simulated annealing are similar.

Table 4. Computation times of the different methods.

Methods	DCGA	ICM	Sim. Ann.
Disc-Square (64x64)	47	0.3	42.3
Train (512x400)	1234	21.4	1140
Interview (128x160)	130	1.75	120

5. Conclusion

An Evolutionary Algorithm method is proposed for estimating the optical flow field using a MRF modeling. It is based on a divide-and-conquer strategy which adequately uses the markovian property. Experimental results on synthetic and real sequence images show that the method provides quite satisfactory results while the execution time is significantly reduced compared to the basic EAs.

6. References

[1] J.L. Barron, D.J. Fleet and S.S. Beauchemin "Systems and Experiment – Performance of Optical Flow Techniques". *Intern. Jour. of Comp. Vis.*, vol. 12 (1), 1994, pp. 43-77.

[2] S. Kirkpatrick, C.D. Gellat and M.P. Vecchi "Optimisation by Simulated Amealing". Research Report rc, IBM, 1982.

[3] J. Besag. '' On the Statistical Analysis of dirty pictures''. *Jour. Royal Statist. Sos*., Vol. 48 (3), 1986, pp. 259-302.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.

[5] P. Andrey: "*Selectionist relaxation: genetic algorithms applied to image segmentation*". Image and Vision Computing, Vol. 3 (4): 1999, pp. 175-187.

[6] A. Dipanda, S. Woo, F. Marzani and J.M. Bilbault. "3-D shape reconstruction in an active stereo vision system using genetic algorithms". *Pattern Recognition*, Vol. 36, 2003, pp. 2143-2159.

[7] P-H. Chang, J-J. Leou and H-C. Hsieh, "A genetic algorithm approach to image sequence interpolation", *Signal Processing: Image Comm.*, vol. 16, 2001, pp. 507-520.

[8] A.N. Tikhonov, "Regularization of Incorrectly Posed Problem", *Sov. Math. Dokl*., vol. 4, 1963, pp. 1624-1627.