

Building new tools for Synthetic Image Animation by using Evolutionary Techniques

Jean LOUCHET^{†‡}, Michael BOCCARA[†], David CROCHEMORE[†], Xavier PROVOT[‡]

[†] ENSTA, Laboratoire d'Electronique et d'Informatique
32 boulevard Victor 75739 PARIS cedex 15, France
33-1-45 52 60 75

[‡] INRIA, projet SYNTIM
Rocquencourt, B.P. 105 78153 LE CHESNAY Cedex, France
33-1-39 63 54 38

e-mail: louchet@bora.inria.fr

Abstract

Particle-based models and articulated models are increasingly used in synthetic image animation applications. This paper aims at showing examples of how Evolutionary Algorithms can be used as tools to build realistic physical models for image animation.

First, a method to detect regions with rigid 2D motion in image sequences, without solving explicitly the Optical Flow equation, is presented. It is based on the resolution of an equation involving rotation descriptors and first-order image derivatives. An evolutionary technique is used to obtain a raw segmentation based on motion; the result of segmentation is then refined by an accumulation technique in order to determine more accurate rotation centres and deduce articulation points.

Second, an evolutionary algorithm designed to identify internal parameters of a mass-spring animation model from kinematic data ("Physics from Motion") is presented through its application to cloth animation modelling.

Keywords

Computer vision, motion analysis, image animation, evolutionary algorithms.

1 Introduction

In this paper, we shall examine through two examples, how evolutionary techniques can contribute to the resolution of problems at the border of the image analysis and image synthesis domains, where optimisation methods tend to play an increasing role and, in our opinion, evolutionary techniques bring interesting new possibilities, notably through their abilities to cope with large numbers of unknown variables or noisy data, to use heterogeneous cost functions, and to find families of solutions rather than a single optimum.

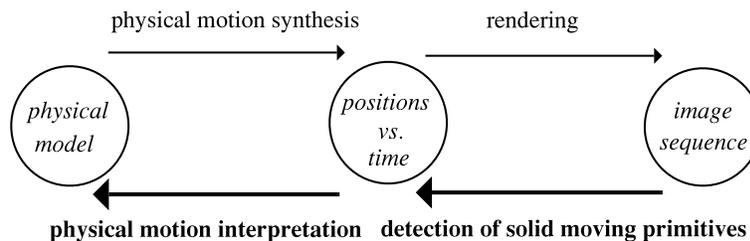
Reprint from: *Artificial Evolution '95*, Proceedings of the EA95 Workshop in Brest, France, Sep. 1995, Springer, to appear in 1996. © 1996 Springer-Verlag/Wien

Both examples shown are inspired by an image synthesis point of view: how to achieve realism of motion in synthetic image animation. Our basic assumption is that motion realism may only be achieved through the explicit use of physical laws to generate animated image sequences. A long-term goal is to help the human animator build physically realistic image sequences without the task of evaluating himself visually the physical consistency of motion. It is therefore important to develop physical model-based animation techniques as the basis of animation tools. To be realistic, such a model should be identified from *real world* image sequences.

In this scope, we devised a family of mechanical models based on points with masses, and interactions (bonds) modelled by energy potentials. We developed ([L94], [L94b]) a general method to identify the set of parameters of such a structure, from given particle kinematic data, based on an evolutionary algorithm technique: being given a set of particles and their positions in function of time, it consists in tuning up a set of mechanical parameters in order to give the object (i.e. its set of geometric primitives) a behaviour which minimises a quadratic norm of the difference of the generated trajectories from the observed ones.

The second part of this paper will show an application of this evolutionary technique to identify internal mechanical parameters of a cloth sample from its kinematics.

The first part of the paper is devoted to another important question still unanswered: how is it possible to build the kinematic data themselves from real image sequences? We describe an algorithm which combines evolutionary methods with more classical approaches to detect solid 2D motion components in image sequences (corresponding to the bottom right arrow).



The cloth application described in the second part of the paper, corresponds to the bottom left arrow.

2 Detecting Rigid Motion in Image Sequences: a phenomenological analysis.

2.1 Instantaneous 2-D rotation centres

Let $I(x, y, t)$ an image sequence. I is the intensity (grey level) of pixel (x, y) at time t . To each point (x, y, t) corresponds a speed vector with coordinates (V_x, V_y) .

The classical Optical Flow hypothesis assumes that the image of each physical point moving in the scene, keeps a constant radiance. This results in a relationship between local speed vectors and the local derivatives of function $I(x, y, t)$ [HW83]:

$$\frac{\partial I}{\partial x} V_x + \frac{\partial I}{\partial y} V_y + \frac{\partial I}{\partial t} = 0$$

However, to solve this equation and compute velocity fields, it is necessary to make

some regularity assumptions, as local derivatives at a given pixel do only give one equation for two unknowns V_x and V_y . Major Optical Flow resolution algorithms are described and compared for example in [BFB94]. They use elaborate techniques to exploit assumptions on topological properties of the physical objects represented (connectivity...), without which the above equation would be completely unconstrained.

Once the velocity field calculated, it would be theoretically possible to use it to determine rotations in the image, in order to extract solid rotating regions; but in practice the speed components are too noisy to be able to differentiate them safely.

With the same Optical Flow hypothesis on the conservation of pixels' radiance along time, it may be shown from geometrical considerations [BL95], that if a region is moving with a solid apparent 2D motion, then:

$$\boxed{-\omega \frac{\partial I}{\partial x} (y - \eta) + \omega \frac{\partial I}{\partial y} (x - \xi) + \frac{\partial I}{\partial t} = 0}$$

where:

ξ, η are the coordinates of the Instantaneous Rotation Centre (IRC);

ω is the rotation speed.

This formula ("IRC equation") does not involve second derivatives, but gives a relationship between easily computed values at the current pixel (pixel's coordinates and first-order derivatives) and the three rotation parameters (ω, ξ, η). It is rather similar to the Optical Flow equation, in that it links several second-order motion descriptors with the local first-order derivatives of function $I(x, y, t)$.

Let us now focus on how to use it to detect *solid motion* in the 2D image sequence. All the pixels in a solid region will share a common IRC (ξ, η) and a common rotation speed ω . Moreover, the equation above states that for each pixel (x, y) and each value of ω , all corresponding potential rotation centres are on one straight line. Therefore if the scene contains a sufficiently large solid region, and if we suppose that we know the rotation speed ω , then these straight lines will normally converge onto the rotation centre. The first resolution method (§ 2.2) uses an accumulation method to find a single solid motion; an evolutionary preprocessing algorithm is then introduced in § 2.3 to find multiple motion primitives and exploit the spatial coherence of moving regions.

2.2 An accumulation algorithm to find a single IRC

The following algorithm is based on the fact that resolving the IRC equation does not require solving explicitly the Optical Flow equation. For each value of ω and each image pixel (x, y) , the equation gives a straight line in the (ξ, η) domain; we know that the local IRC belongs to this line. The idea is that a "good" IRC will belong to many such lines. We implemented an algorithm based on a vote technique inspired by the Hough method [BFB94]. The main steps of the algorithm are:

- define an *accumulation space* in the domain (ξ, η) , i.e. a 2-dimensional buffer initialised at 0;
- for each value of ω in a given interval:
 - for each pixel (x, y) , calculate the local derivatives and increment the values

in the accumulation space domain (ξ, η) along the line defined by the IRC equation

$$-\omega \frac{\partial I}{\partial x}(y - \eta) + \omega \frac{\partial I}{\partial y}(x - \xi) + \frac{\partial I}{\partial t} = 0$$

- detect the position and value of the maximum peak, thus defining a function $F: \omega \rightarrow (\xi, \eta, \text{peak value})$
- for the value ω_i detected, $F(\omega_i)$ gives the coordinates (ξ_i, η_i) of the IRC.;
- the corresponding region is the set of pixels which verify the IRC equation.

Experimental results

We tested the algorithm on a natural image sequence from a 2D scene. The scene consists here in a C-shaped object moving on a textured background. Two consecutive 256×256 frames are shown. The rotation centre is a screw near the centre of the image, with coordinates (69,140).

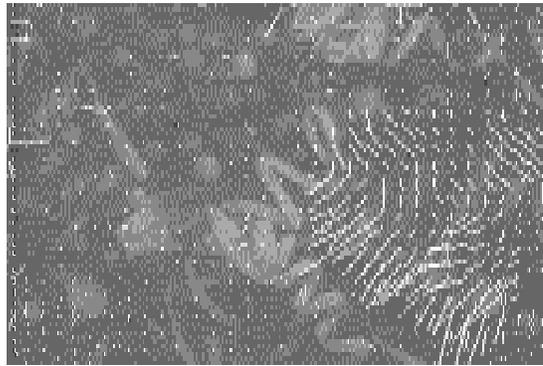


Fig. 1 - A frame from the sequence, with velocity field calculated using a classical method.

We only used image pixels with a sufficient gradient norm value to increment the accumulation space..

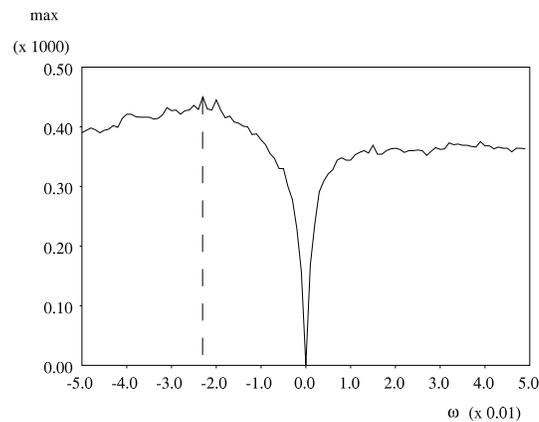


Fig. 2 - Accumulation space maxima vs. ω

Exploring values of ω between $-.05$ and $.05$ radians per frame, gives the best peak for $\omega = -.023$ (fig. 2) and the corresponding accumulation space (fig. 3), giving a good estimate of the coordinates of the IRC: $\xi = 72, \eta = 142$

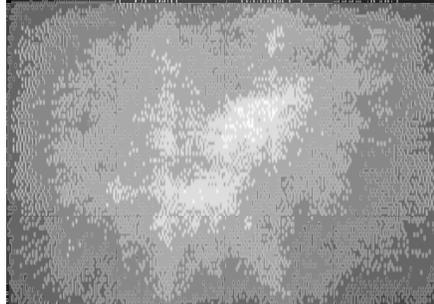


Fig. 3 - The Hough accumulation space for ω_{\max}

Conversely, the image pixels corresponding to the rotating object should verify the IRC equation with $\omega = -0.023, \xi = 72, \eta = 142$. Figure 4 shows brighter intensities for pixels verifying the IRC equation with parameters values as found (image pixels with insufficient space derivatives are eliminated first).

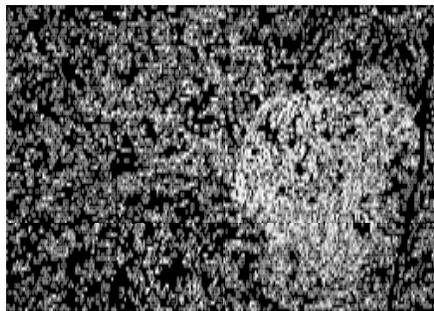


Fig. 4 - Blob detection: brighter pixels verify the equation of motion detected.

2.3 Finding multiple IRCs: Rigid Motion Segmentation

2.3.1 Outline of the algorithm

The algorithm described above, uses a single accumulation space to detect the rotation centre. Therefore, it cannot detect multiple motion as occurring in the general case. Moreover, it makes no assumption on the topology (connectivity) of solid moving regions. The values in the accumulation space are cluttered because of the small size of the objects actually moving compared to the full image size. To solve simultaneously these problems, we devised a new algorithm in three steps:

The first step consists in a primary, raw detection of multiple rotation components. We chose to use an evolutionary algorithm with sharing, to give multiple concurrent solutions.

The next step consists in refining the shapes of these components in order to get a better coverage of the apparent individual rigid objects. We used an optimisation method

based on active contours (snakes, [IP94]).

The third step consists in refining the rotation characteristics of these shapes by using an accumulation method as in section 2.1, but restricted to each of the refined shapes determined above.

2.3.2 An evolutionary algorithm to combine local and topological constraints

In order to detect multiple rotation components, we define a population consisting in rotation descriptors, and make it evolve through an evolutionary algorithm. The basic idea is to *introduce the topological constraint into the population individuals themselves*. An individual will therefore consist in the association of a rotation, with a circular neighbourhood of a pixel: the cost function of an individual will be low if the pixel values inside the neighbourhood are consistent with the rotation.

Coding

Each individual i is an ordered sequence $(x_i, y_i, r_i, \xi_i, \eta_i, \omega_i)$ of real numbers, where: x_i, y_i are the coordinates of the centre of a disc; r_i the radius of the disc; ξ_i, η_i the coordinates of a rotation centre; ω_i the rotation value.

The aim of the algorithm is to make the population converge to a final population in which the discs have a fair repartition on the image area, do not overlap, and their corresponding rotation descriptors ξ_i, η_i, ω_i be in accordance with the local motion inside the disc.

Cost function and sharing

The cost function is based on a normalized version of the IRC equation:

$$Cost(i) = \alpha \sum_{(x-x_i)^2 + (y-y_i)^2 < r_i^2} \left[\frac{-\frac{\partial I}{\partial x}(y-\eta_i) + \frac{\partial I}{\partial y}(x-\xi_i) + \frac{1}{\omega_i} \frac{\partial I}{\partial t}}{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \right] + \beta \cdot \frac{R}{R+r_i}$$

The "over-normalizing" term in the denominator, helps preventing the discs from spreading over image regions with low gradient values. An additional term gives a slight advantage to larger discs. R, α, β are parameters.

Sharing prevents the population from concentrating on near-identical individuals, and is implemented through the addition of an extra cost to individuals with overlapping discs: the cost function is first calculated for each individual, and the individuals are sorted accordingly. The shared cost function is initialised as equal to the cost function. Then, all pairs of individuals are examined in turn: for each overlapping pair, the shared cost function of the individual with higher (old) cost function is incremented proportionally to the overlapping surface, while the shared cost function of the other individual is kept unchanged.

Selection

At each generation, the individuals are sorted again, according to their shared cost function values. The selection process is controlled by the rank rather than by the shared cost function value. The 20% most performing individuals are then kept, the remaining 80% are deleted and replaced by new individuals created by the mutation process.

Mutations

At each generation, four different mutation processes are applied in parallel. No crossover process is applied.

The first one consists in creating totally new individuals through random functions. The other ones consist in creating three slightly altered copies of each of the 20% best individuals selected. The two first ones are obtained through applying a random noise to the disc's position and radius, and the rotation speed ω_i . The last one consists in projecting (with a damping coefficient) the rotation centre (ξ_i, η_i) onto the straight lines defined by equation [3]. The projection is repeated for each pixel in the disc.

Initialising

The population is initialised with random values for $x, y, r, \xi, \eta, \omega$ (between given bounds).

Results

With a typical population of 100 individuals, a reasonable stability is obtained after 50 to 100 generations. The final 20% best individuals are retained.

The image below shows the results of primary rotation segmentation after 50 generations on a population of 150 individuals. The circles represent neighbourhood discs. Small squares are the rotation centres found. The straight line from the rotation centre to the corresponding disc, is omitted when the rotation centre is out of the picture. The shorter straight lines from the discs centres represent the local estimated velocities.

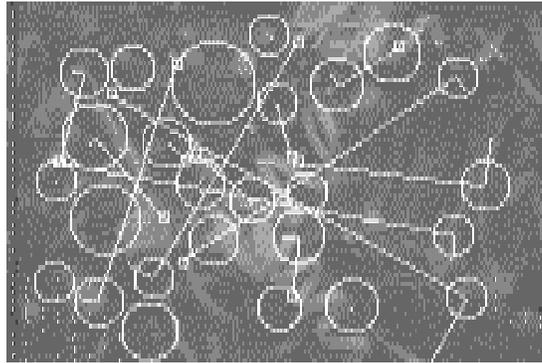


Fig. 5 - Result of evolutionary preprocessing

2.3.3 Refining shapes using snakes

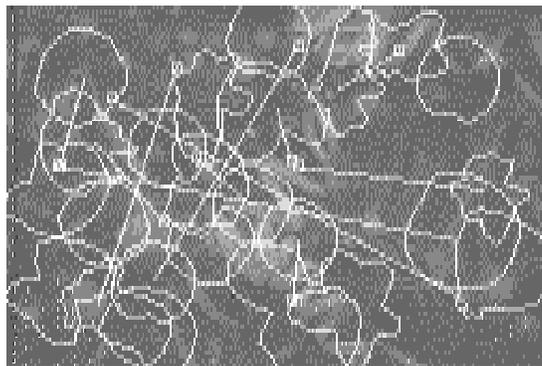


Fig. 6 - Refining the discs' shapes using snakes.

This step consists in refining the shapes found, and deform the circular discs to get more complex shapes by using a snake (active contour) technique, where regions are defined by their radial equation. The regions' cost incorporates a stiffness term to avoid erratic contours; a simple algorithm inflates the contour and retain it when it gives a better energy value. Stabilisation is obtained after typically 5 to 20 iterations (figure 6).

2.3.4 Determining rotations

This last step consists now in refining the rotation characteristics of each region. We apply the algorithm of section 2.2, exploring values of ω around the initial value ω_i found by the evolutionary preprocessing. The essential difference is that values in the domain (ξ_i, η_i) are incremented only for the pixels (x, y) belonging to the refined region (fig. 6) determined above, thus allowing both to reduce noise and potentially detect as many solid motion elements as the number of discs detected in 3.2.1. Experimental results show that the IRCs corresponding to a single solid region do concentrate fairly well into a single point, as shown in figure 7.

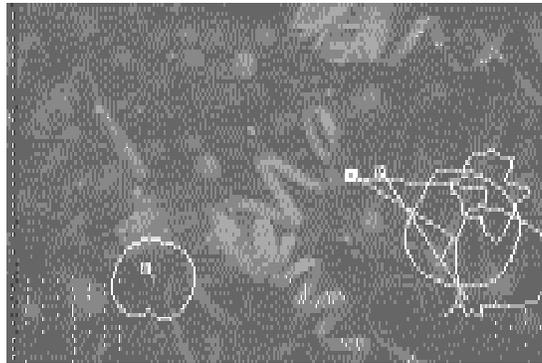


Fig. 7 - Refining the rotations detected.

2.4 Conclusion

The method above allows to detect regions with solid 2D motion in an image sequence. The specific role of the evolutionary algorithm used here is to allow simultaneous determination of unknown motion of unknown image regions, involving several independent constraints: a connexity constraint on the shape of regions, and a motion consistency constraint on grey levels derivatives. The evolutionary preprocessing algorithm gives also a good answer to the need of finding several simultaneous solutions to a problem, as it is often needed in other image processing applications. It is rather fast (a fraction of the time taken e.g. by snake optimisation) and gives good first-approximation solutions which can then be refined using more conventional approaches.

Here, we do not use any crossover mechanism: this would be no benefit, because a combination of e.g. the disc of one individual with the rotation of another individual, would be physically meaningless and have no reason to yield a new individual with a lower cost.

3 Identifying internal parameters of a cloth animation model

3.1 Physical model identification

An interesting property of the algorithm described above was the ability of an evolutionary approach to find easily multiple approximate solutions to an image processing problem, and to exploit the topological consistency of images by creating individuals strongly related to topological constraints.

Another useful property in image applications corresponds to the current trend towards model-based image interpretation. Fitting a model to given image data requires in particular to be able to optimise functions depending on large numbers of parameters. We described in [L94] an evolutionary algorithm to identify the internal parameters (spring lengths and stiffness...) of a physical animation model to fit given motion data. An original feature was there the use of multiple cost functions, again to exploit more efficiently the 3-D topology of the object to be identified, resulting in the algorithm's convergence in an average number of generations independent of the number of particles (and therefore the number of parameters) involved.

However, in many cases, the same bond may be replicated a great number of times with identical parameters, in a single object. This is the case in cloth models or in several molecular modelling problems. The model parameters are then independent of the location of the bond considered.

This section describes such an evolutionary identification approach, using a cloth animation physical model developed by X.Provot [P95], and giving a visual counterpart to the quality of parameters' identification.

3.2 A Mass-Spring Cloth Model

The elastic model is a periodic mesh of masses, each one linked to its neighbours by massless springs of non-zero natural length at rest. The links between neighbours are:

- springs between masses $[i, j]$ and $[i+1, j]$, or between masses $[i, j]$ and $[i, j+1]$: "structural springs";
- diagonal springs between masses $[i, j]$ and $[i+1, j+1]$, or $[i+1, j]$ and $[i, j+1]$: "shear springs" ;
- double-length springs between masses $[i, j]$ and $[i+2, j]$, or $[i, j]$ and $[i, j+2]$: "flexion springs".

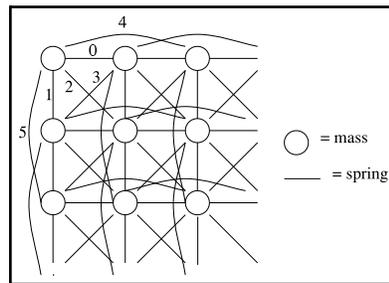


Fig. 8 - The periodic mesh of masses and springs used in our model.

We chose not to introduce any "ternary" bond. In practice, if not strictly equivalent, a

flexural stiffness effect can be achieved by giving planar internal stress, e.g. by increasing the lengths of type 2 and 3 springs while reducing the lengths of types 0 and 1.

Motion is obtained by calculating at each step: mutual distances between particles, resulting forces on particles, particles' accelerations, new speeds and new positions.

To take into account the non-linear elasticity of cloth, an 'inverse dynamics' procedure is applied if necessary to the two ends of the spring, so that its deformation rate cannot be greater than τ_c .

3.3 Identifying mechanical parameters of cloth

The fabric model above uses (in the case of homogeneous fabrics) equal masses, and 6 different spring types. Each spring being described by three parameters, the fabric is completely described by 18 parameters. In a first approach to the identification problem, in order to reduce the algorithmic cost of analysis, we have partly simplified this general model, making the assumption the fabric is isotropic and all the springs share a common stiffness value. The simplified model contains 5 parameters:

- the springs stiffness
- the elongation rate;
- lengths at rest of springs 0 and 1;
- lengths at rest of springs 2 and 3;
- lengths at rest of springs 4 and 5.

The cost function is based on the difference between the predicted and actual behaviours. One formulation could be:

$$f(\text{parameters}) = \sum_{\text{time steps}} \sum_{i,j} [(x_p - x_r)^2 + (y_p - y_r)^2 + (z_p - z_r)^2 + \Delta t^2 ((vx_p - vx_r)^2 + (vy_p - v$$

where

$x_r, y_r, z_r, vx_r, vy_r, vz_r$ are the actual (recorded) positions and velocities of particles;

$x_p, y_p, z_p, vx_p, vy_p, vz_p$ are the positions and velocities of particles as predicted by the model. Δt is an arbitrary coefficient.

In order to prevent a quick divergence between actual and predicted coordinates, which would result in the cost function being too sensitive to the parameters values, positions and velocities are predicted using actual positions and velocities at the preceding time step, rather than values predicted over several time steps.

The major problem with this cost function is its high computational cost (about 1 minute on a Sparc 4 for 50 time steps), remembering it will have to be calculated several thousands times in the identification process. Therefore we have defined a new "small" cost function, which only involves one frame of the sequence. It proved to be sufficient in practice to obtain good estimates of parameters (see below) if the frame chosen is far enough from the initial conditions.

To optimise this cost function, we devised an Evolutionary Algorithm. The *individuals* are tentative sets of model parameters. An individual is a n-uple of real numbers.

The population is randomly *initialised*, then evolves using three random basic processes controlled by the cost function values: a *selection* process, a *mutation* process and a *crossover* process.

Selection

At each generation, individuals are sorted according to their cost function values. The selection process is guided by *ranking* rather than by the cost function value in itself. We chose to keep the 50% most performing individuals unchanged, the remaining 50% are deleted and replaced by new individuals created by the mutation (20%) and crossover (30%) processes.

Mutations

At each generation, 20% new individuals are created through mutations of randomly chosen parameters among the 80% most performing individuals.

Crossover

At each generation, 30% new individuals are created through uniform crossover. For each new individual to be created, we choose a number of parents equal to the number of model parameters, among the 50% most performing individuals .

Initialising

The population is initialised with random values for all the parameters. However, natural spring length values are chosen around the average length values observed on the reference data.

3.4 Experimental results

We tested the algorithm in the case of a cloth hanging from two corners, with different parameter values. In order to check the suitability of the cost function, we calculated its theoretical values using the simplified 5-parameter model in a neighbourhood of the reference values. The following diagrams show variations of theoretical cost functions, in function of two parameters: the spring stiffness K and elongation rate τ , for two parameter sets. The cost function has been calculated on the domain $\tau \in [0.995, 1.01]$, $K \in [2.0, 4.0]$. Spring natural lengths are fixed to their reference values. The accuracy of the new cost function appears good in the absence of noise, and very sensitive to the elongation rate: the last image in the animation corresponds to a state where the elongation rate plays a dominant role.

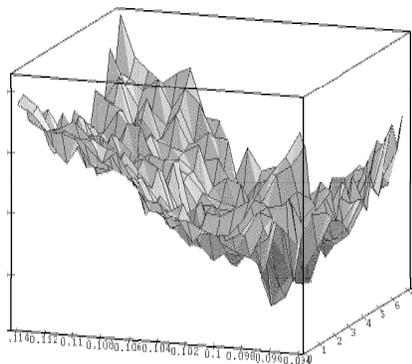


Fig. 9. Cost function for a reference trajectory computed with the fixed parameters $K = 4$ and $\tau_c = 0.1$.

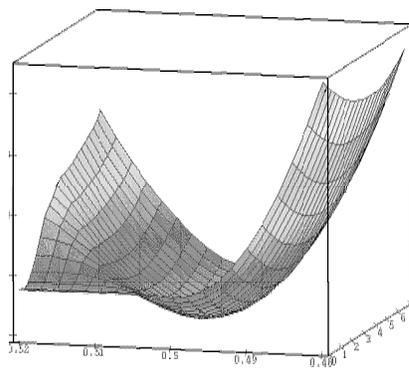


Fig. 10. Cost function for a reference trajectory computed with the fixed parameters $K = 4$ and $\tau_c = 0.5$.

Convergence is generally obtained after 50 to 100 generations and gives results in good accordance with the reference. In some cases, the cost function suffers from numerical stability problems, as it can be seen above on the left, but this only results in a less accurate identification. The listings below shows the parameters values for the 5 best individuals in a 100 individual population, after 49 generations, in typical good and bad situations, with a 17×17 mesh. The first line shows the reference values.

	/	101	/	123	/	145	/	K	/	tau	/	cost	/
reference		0.093750		0.132583		0.187500		4.000000		0.500000		0.000000	
candidate	4:	0.093984		0.133019		0.187106		3.938169		0.497108		0.024129	
candidate	26:	0.093887		0.133019		0.187111		3.906064		0.497108		0.022811	
candidate	57:	0.093960		0.133019		0.187107		3.926550		0.497108		0.022180	
candidate	39:	0.093954		0.133019		0.187108		3.927238		0.497108		0.021845	
candidate	88:	0.093984		0.132705		0.187083		4.084211		0.497108		0.014619	

In the second situation:

reference		0.093750		0.132583		0.187500		4.000000		0.500000		0.000000	
candidate	73:	0.086170		0.132667		0.199857		0.394229		0.630850		0.645847	
candidate	7:	0.086170		0.132667		0.198607		0.394229		0.630850		0.645059	
candidate	1:	0.086170		0.132670		0.199827		0.400431		0.630850		0.644962	
candidate	40:	0.086170		0.132668		0.199821		0.406031		0.630850		0.644202	
candidate	51:	0.086170		0.132671		0.200991		0.435233		0.630850		0.642728	

In the first case, the cost of 0.014 corresponds to an average error per frame of about one twentieth of the springs' natural length and gives very small visual differences with the original sequence (Figure 11, middle column).

In the second case, the cost function has been trapped into the valley visible on the left side of figure 10 and the number of generations was not sufficient to reach the global optimum. The cost of 0.64 corresponds to a quadratic average error per frame on particles' positions about one half the structural springs' natural length.

Figure 11 shows in the left column, the original image sequence generated using arbitrary parameters. The centre column shows a reconstruction of the sequence using only the parameters resulting from the first identification case above (with final cost function 0.014619), the column right shows another reconstruction using the set of parameters obtained in the second case (cost 0.642726). These examples show how premature results of the evolutionary algorithm affect the visual results in animation, and suggest that in this case, the criterion used to stop the evolutionary algorithm should be based on the cost function values rather than the number of generations.

4 Conclusion

Realistic image animation requires explicit modelling of the hidden processes which control the objects' observable behaviours in the scene. Such models, based on general physical knowledge, are often complex. They need to be run efficiently to create images, but also to be *built accurately from real life data*. The examples above, show two applications of Evolutionary Algorithms to the resolution of problems on the border between Computer Graphics and Image Processing, in both kinematic animation and physical animation modelling. They suggest that Evolutionary techniques are likely to play a significant role in the field of model-driven image analysis, where statistical methods alone, like Hough's accumulation algorithms, are not able to cope

with the large numbers of parameters often involved in physical or even phenomenological models.

Moreover, they illustrate the strong link between the *semantics* of the function being optimised and the corresponding *evolutionary scheme*: in particular, crossover was omitted in our first example because of its physical meaninglessness; in the second example, the use of multiple cost functions, shown in [L94] to be essential in the parameter identification of complex mass-spring objects, is left aside in the special case of fabrics modelling where the same set of parameters is used throughout the object.

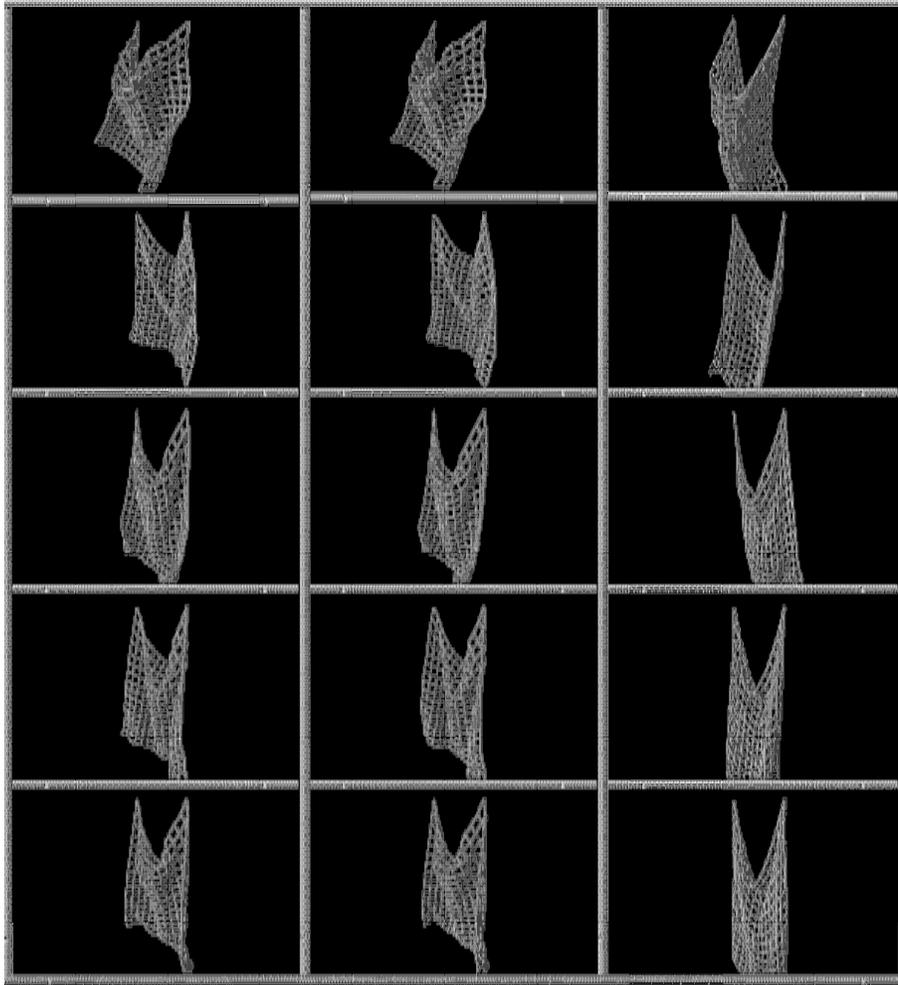


Fig. 11. The original animation (left column), and two reconstructions of the animation using different identification results (centre column: cost = 0.014; right column: cost = 0.642).

References

- [BHW94] D.Breen, D. House, M. Wozny, "Predicting the drape of woven cloth using interacting particles", Proc. Siggraph 94, Comp. Graph. Proc., 1994, pp. 365-372.
- [BL95] M.Boccaro, J.Louchet "Recherche de points d'articulation dans les séquences d'images", internal report, Ecole Nationale Supérieure de Techniques Avancées, 1995.
- [G89] D.A.Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley 1989.
- [GVP90] M.-P. Gascuel, A. Verroust, C.Puech "Animation with collisions of deformable articulated bodies", Eurographics Workshop on Animation & Simulation, Sep. 1990.
- [HW83] B.K.P.Horn, E.J.Weldon Jr., "Determining Optical Flow", Artificial Intelligence 17: 185-204, 1981.
- [IP94] Jim Ivins, John Porrill, *Statistical Snakes: Active Region Models*, British Machine Vision Conference, York, Sep. 1994.
- [K92] John Koza, *Genetic Programming*, MIT Press, 1992.
- [L94] J. Louchet, "An Evolutionary Algorithm for Physical Motion Analysis", British Machine Vision Conference, York, Sep. 1994.
- [L94a] J.Louchet, "Identification évolutive de modèles physiques d'animation", Journées Evolution Artificielle 94, Toulouse Sep. 1994.
- [LJFCR91] A. Luciani, S. Jimenez. J.L. Florens, C. Cadoz, O. Raoult, "Computational Physics: a Modeller Simulator for Animated Physical Objects", Proc. Eurographics Conference, Wien, Sep. 1991, Elsevier.
- [P95] X.Provot, "Deformation Constraints in a Mass-Spring Model to describe Rigid Cloth behavior", Graphics Interface 1995, Québec, April 1995.