Evolutionary identification of cloth animation models

Jean LOUCHET†‡, Xavier PROVOT‡, David CROCHEMORE†

† 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 provot@bora.inria.fr

Abstract

This paper presents an application of evolutionary genetic techniques to the identification of internal parameters of a mass-spring physically-based animation model.

A physical model of fabrics is first presented. It uses a mass-spring mesh and an inverse dynamics procedure in order to model the non-linear elasticity of fabrics.

A method to identify the internal parameters of the model from geometric data is then presented. It is based on a cost function which measures the difference in behaviour between the reference and the model, and an evolutionary minimisation algorithm.

Keywords

Physically-based modelling, animation, simulation, evolutionary algorithms.

Reprint from: *Computer Animation and Simulation '95*, Proceedings of the Eurographics Workshop in Maastricht, The Netherlands, Sep. 95, Springer, pp. 44-54.

1 Introduction

Cooperation between the processes of vision and image synthesis has been widely studied from the viewpoints of 3-D geometry and photometry, proving the mutual benefit of such an approach to both domains and leading to viewing vision as the process of *building synthesizable models*.

Here, while still adopting this philosophy of implementing image analysis techniques to fulfil the modelling needs of computer graphics, we shall focus on the problems of modelling motion and deformation of fabrics in image synthesis.

In order to be able to reproduce the behaviour of a cloth object, e.g. a flag in the wind, it is necessary to build the animation model itself, before using it to create images. To be visually realistic, these models should be identified from real world images. Thus, we may see modelling in computer graphics, as the process of *creating identifiable models*.

In the first part of this paper, we shall describe a system we devised to build visually realistic animations of fabrics, using a mass-spring physically based model.

In the second part, we present a method to identify such a model from given geometric data. Present results show the algorithm's ability to recover parameters from synthetic cloth animations.

2 A Mass-Spring cloth model

We are using the model described in [P95] which has proved to be more realistic than classical elastically deformable models used by Terzopoulos in [TPBF87] and by Thalmann in [CYMT92]. The linear elasticity of the elastic models used by the latter, leads indeed to unrealistic deformations of cloth objects which are successfully controlled in [P95] with a minimal increase in the computation cost.

Breen's particle systems [BHW94] have not been used to model the dynamic behaviour of cloth objects, and the computation time needed for such a system is much greater than that of the model we use. A reasonable computation time is required in order to use evolutionary algorithms.

2.1 The mesh

Our elastic model is a mesh of $m \times n$ virtual masses, each mass being linked to its neighbours by massless springs of non-zero "natural length" (i.e. 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], referred to as "structural springs" (types 0 and 1);
- diagonal springs between masses [i, j] and [i+1, j+1], or masses [i+1, j] and [i, j+1], referred to as "shear springs" (types 2 and 3);
- double-length springs between masses [i, j] and [i+2, j], or masses [i, j] and [i, j+2], referred to as "flexion springs" (types 4 and 5).

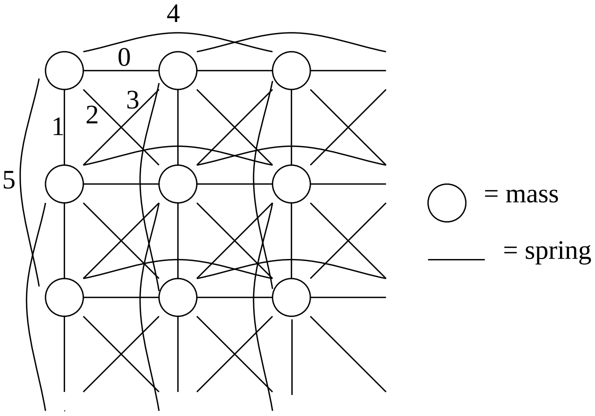

Fig. 1. The periodic mesh of masses and springs used in our model.

Shear springs are introduced in order to give the fabric a shear rigidity and prevent it from excessive and unrealistic distortion in its own plane.

"Flexion" double-length springs are under stress when flexion occurs in the cloth and therefore have a three-dimensional role. They may even be used to introduce preconstraints into the fabric and provide it with an extra degree of angular stiffness. For example, planar rigidity will be obtained by introducing type 5 springs with a natural length greater than twice the natural length of type 1 springs.

2.2 A Dynamical Model

The system considered is a mesh of $m \times n$ masses, each mass being located at time t on the point $P_{i,j}(t)$, where $i = 1... m$ and $j = 1... n$.

The evolution of the system is governed by the fundamental law of dynamics:

$$\vec{F}_{i,j} = \mu \vec{\gamma}_{i,j}$$

where:

μ is the mass of each point,

$\vec{\gamma}_{i,j}$ its acceleration,

$\vec{F}_{i,j}$ the force applied to point $P_{i,j}$,

$\vec{F}_{i,j}$ can be divided into internal and external forces.

The internal force results from the tensions of the springs linking $P_{i,j}$ to its neighbours:

$$\vec{F}_{int}(P_{i,j}) = -\sum_{(k,\, l \in R)} K_{i,j,k,l} \left[\vec{L}_{i,j,k,l} - l^{0}_{i,j,k,l} \frac{\vec{L}_{i,j,k,l}}{\left\| \vec{L}_{i,j,k,l} \right\|} \right]$$

where:

- R is the set of all ordered pairs (k, l) such that $P_{k,l}$ is linked to $P_{i,j}$
- $\vec{L}_{i,j,k,l} = \overrightarrow{P_{i,j}P_{k,l}}$
- $l^0_{i,j,k,l}$ is the natural length of the spring between $P_{i,j}$ and $P_{k,l}$
- $K_{i,j,k,l}$ is the stiffness of the spring between $P_{i,j}$ and $P_{k,l}$

The external forces depend on the types of loads we wish the model to be exposed to. Typical loads will be gravity, a viscous damping and a viscous interaction with an air stream (or wind).

Let $\vec{g}$ be the gravity; the weight of $P_{i,j}$ is given by:

$$\vec{F}_{grav;i,j} = \mu \vec{\gamma}_{i,j}$$

Viscous damping is modelled by:

$$\vec{F}_{visc;i,j} = -C_{visc}\vec{v}_{i,j}$$

where C_{visc} is a damping coefficient, and $\vec{v}_{i,j}$ the velocity of $P_{i,j}$.

The role of damping is to model a first approximation of the mechanical energy dissipation.

Finally, a viscous fluid moving at a uniform velocity $\vec{u}_{fluid}$ applies onto the surface of a body moving at a velocity $\vec{v}$, a local force

$$\vec{F}_{visc} = C_{visc}\left[\vec{n}.\left(\vec{u}_{fluid} - \vec{v}\right)\right]\vec{n}$$

where $\vec{n}$ is the normal vector to the surface. In the discrete case,

$$\vec{F}_{visc}\left(P_{i,j}\right) = C_{visc}\left[\vec{n}_{i,j}.\left(\vec{u}_{fluid} - \vec{v}_{i,j}\right)\right]\vec{n}_{i,j}$$

where $\vec{n}_{i,j}$ is the unit normal on the surface at the point $P_{i,j}$.

2.3 Time Integration

The elements described above allow us to compute the resulting force $\vec{F}_{i,j}(t)$ applied to the point $P_{i,j}$ at time t. The fundamental equation of dynamics can therefore be explicitly integrated through time by a simple Euler method:

$$\begin{cases} \vec{\gamma}_{i,j}(t+\Delta t) = \frac{1}{\mu}\vec{F}_{i,j}(t) \\ \vec{v}_{i,j}(t+\Delta t) = \vec{v}_{i,j}(t) + \Delta t\,\vec{\gamma}_{i,j}(t+\Delta t) \\ \vec{P}_{i,j}(t+\Delta t) = \vec{P}_{i,j}(t) + \Delta t\,\vec{v}_{i,j}(t+\Delta t) \end{cases}$$

where Δt is a chosen time-step.

2.4 Inverse dynamics constraints and deformation rates

In a classical mass-spring model, large deformations often occur locally in the mesh. This behaviour, caused by the fact that all the springs are linear springs, does not fit with the behaviour of real clothes. In order to prevent these unrealistic deformations, we devised an ad-hoc inverse dynamics procedure to the most deformed (or "super-elongated") springs in order to reduce their elongation.

At each given time step, the numerical integration is achieved using equation (2).

Then the deformation rates of all springs are computed.

If the deformation rate of a spring is greater than a critical given deformation rate τ_c (e.g. 0.1), then an inverse dynamics procedure is applied to the two ends of the spring so that its deformation rate exactly equals τ_c . This means that we want the length of the springs not to exceed their natural length by more than 10%. For more details, this procedure is described in [P95].

Thus, in a single computation, all the springs with a deformation rate exceeding τ_c after the numerical integration are adjusted to a more "reasonable" and realistic deformation rate.

3 Identifying a physical model of cloth from its motion

3.1 Introduction

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 cloth is described by a total of 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 that the fabric is isotropic and all the springs share a common stiffness value.

The simplified model contains 5 parameters (instead of 18):

- the springs' stiffness,
- the elongation rate,
- natural lengths of springs 0 and 1,
- natural lengths of springs 2 and 3,
- natural lengths of springs 4 and 5.

3.2 The cost function

The basic idea is to define and optimise a cost function which measures, for any set of model parameters, the difference between the behaviour predicted using these parameters, and the actual behaviour of the cloth.

Such a general cost function could be:

$$f(parameters)=\sum_{time}\sum_{i,j}\left[(x_p-x_r)^2+(y_p-y_r)^2+(z_p-z_r)^2+\Delta t^2\left((vx_p-vx_r)^2+(vy_p-vy_r)^2+(vz_p-vz_r)^2\right)\right]$$

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 practice, Δt will be of the same order of magnitude as the time step chosen in the animation.

In order to prevent a quick divergence between actual and predicted coordinates, which would result in the cost function being extremely sensitive to the parameters values and therefore difficult to optimise, positions and velocities will be predicted by the model 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 computational cost (about 1 minute on a Sparc10), remembering this cost function will have to be calculated several thousand times in the identification process. Therefore we have defined a new "small" cost function, which only involves the last time step in the sequence, rather than all the time steps. Experience shows it is sufficient in practice to give good estimates of parameters (see section 3.4).

3.3 An evolutionary strategy

Evolutionary Strategies play an increasing role in the optimisation of functions of several variables, especially in cases where a mathematical characterization of the function to be optimised is not easily obtainable ([G89], [K92]). Their main characteristics are generally a good noise resistance and their ability to escape multiple local minima. However, on convex domains, more classical methods are preferable on both points of view of precision and computational load. Experience shows [L94] that in similar cases of identification of complex physical animation models, it is advisable to use first an evolutionary strategy and then, if necessary, refine the results using a convex optimisation algorithm.

Here, the optimisation process will involve a *population*, the *individuals* of which are tentative sets of model parameters. An individual is a n-uple of real numbers.

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

The aim of the algorithm is to make the population converge to a final population in which the most performing individuals will be as close as possible to the optimal solution of the problem.

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 those from the 80% most performing individuals at the preceding generation.

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 in the preceding generation.

Initialising

The population is initialised with random values for all the parameters. However, in order to increase the algorithm's efficiency, 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 simulated cloth hanging from two corners, with different parameter values. Our reference trajectory is computed using a simulation with fixed parameters. We are testing if our evolutionary algorithm can converge toward the fixed parameters of the reference.

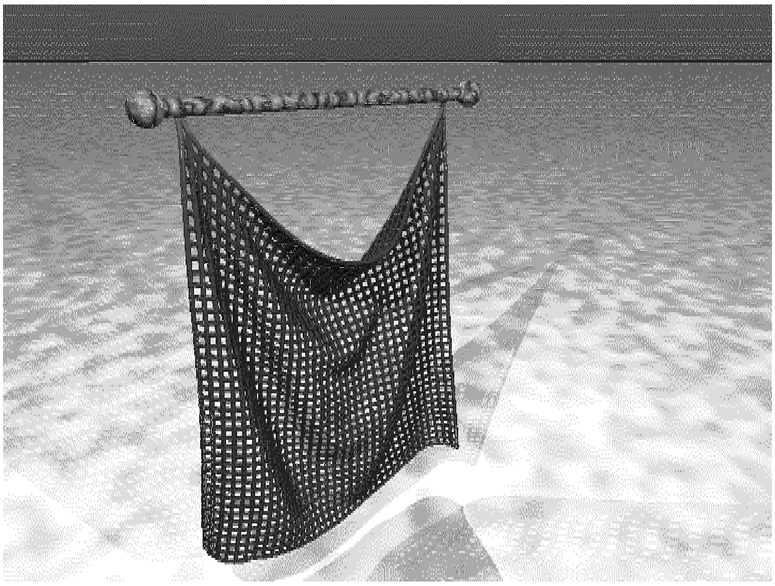

Fig. 2. A hanging cloth obtained with our model

In order to check the suitability of the cost functions, we calculated their theoretical values using the simplified 5-parameter model in a neighbourhood of the reference values.

The following diagrams (Figures 3 and 4) show variations of theoretical cost functions, in function of spring stiffness K and maximum elongation rate τ_c, for two parameter sets; spring natural lengths are fixed to their reference values.

The accuracy of the new (small) cost function appears to be good in the absence of noise, in spite of the fact it is based on only one frame from the synthetic animation. In fact, this is compensated by the large number of occurrences of identical springs in different elongation states. This should be true with any position of the cloth, except its initial state where all spring lengths equal their natural length and therefore give no information about spring parameter values.

The small cost function is generally very sensitive to the maximum elongation rate. This sensitivity may be explained by the fact the last image in the animation corresponds to a state where most springs are far from their rest position and where many of them are close to their maximum elongation rate.

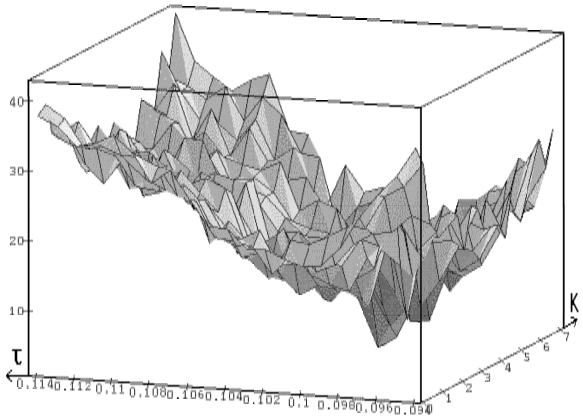

Fig. 3. Cost function for a reference trajectory computed with the fixed parameters $K = 4$ and $\tau_c = 0.1$.

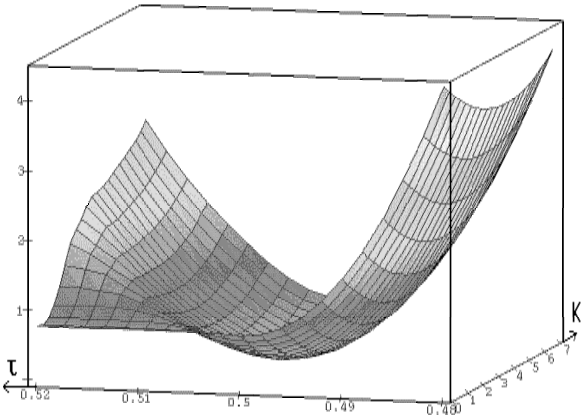

Fig. 4. Cost function for a reference trajectory computed with the fixed parameters $K = 4$ and $\tau_c = 0.5$.

In most cases, convergence is obtained after about 50 to 100 generations and gives results in good accordance with the reference. In some cases, the theoretical cost function suffers from apparent numerical stability problems, as it can be seen above in Figure 3, but this only results in a less accurate identification.

The listing below shows the values of parameters of the 10 best individuals in a 100 individual population, after 49 generations, in typical good and bad situations generated by the evolutionary algorithm, with a 17×17 mesh. The first line shows the reference values which were not known by the algorithm.

```
            /   l01   /   l23  /   l45   /   K    /  tau   /  cost  /
reference    0.093750 0.132583 0.187500 4.000000 0.500000 0.000000

candidate 95:0.093985 0.133019 0.187111 3.938169 0.497108 0.024182
candidate 12:0.093985 0.133019 0.187111 3.938169 0.497108 0.024160
candidate 65:0.093984 0.133019 0.187083 3.938169 0.497108 0.024137
candidate 31:0.093984 0.133019 0.187111 3.938169 0.497108 0.024133
candidate 82:0.093984 0.133019 0.187109 3.938169 0.497108 0.024130
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 other (bad) case:

```
            /   l01   /   l23  /   l45   /   K    /  tau   /  cost  /
reference    0.093750 0.132583 0.187500 4.000000 0.500000 0.000000

candidate 70:0.086170 0.132648 0.200459 0.363856 0.630850 0.650783
candidate 47:0.086170 0.132094 0.201230 0.370355 0.630850 0.650390
candidate 43:0.086170 0.132654 0.199978 0.369424 0.630850 0.649559
candidate 95:0.086170 0.132671 0.200991 0.386130 0.630850 0.648277
candidate 74:0.086170 0.132671 0.197468 0.386130 0.630850 0.646357
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 5, middle column).

In the second case, the cost function has been trapped into a local minimum and the number of generations was not sufficient to get a correct identification; the cost of 0.64 corresponds to a quadratic average error per frame on particles' positions slightly less than half the structural springs' natural length. In such a case, the model identified gives an important difference with the original animation as it can be seen in Figure 5, right column. This means that the criterion used to stop the evolutionary algorithm should be based on the cost function values rather than the number of generations.

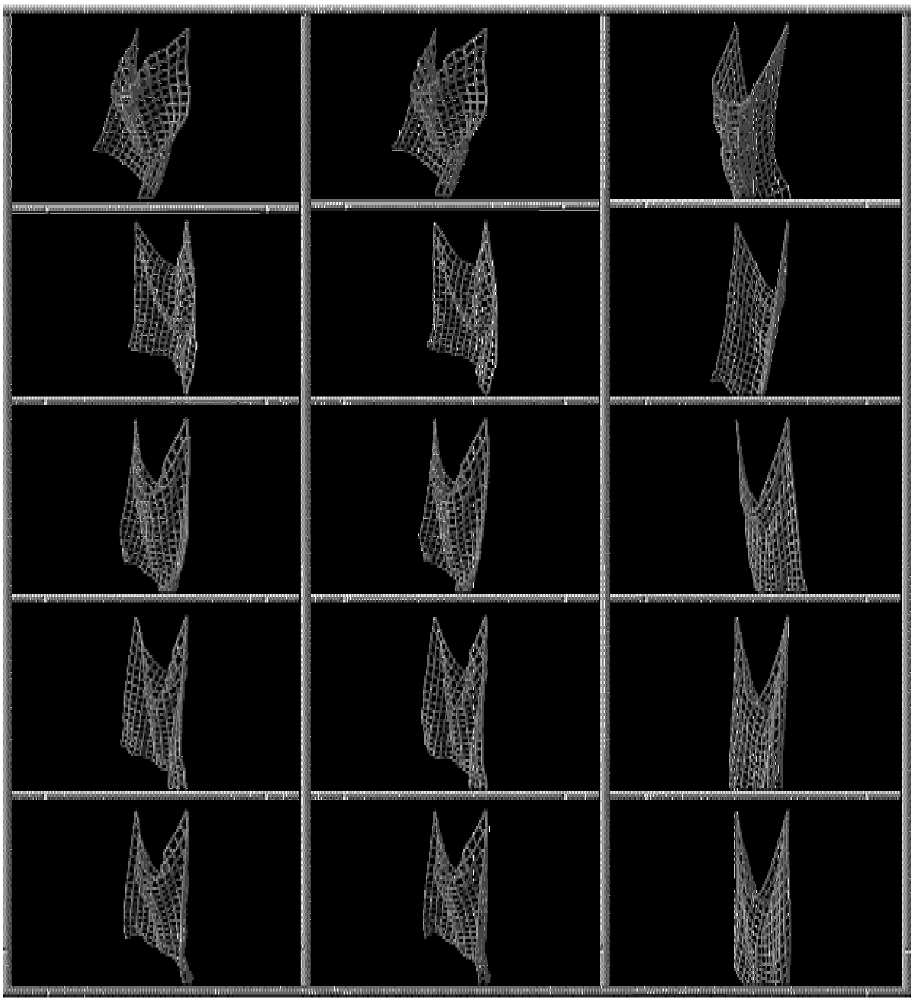

Fig. 5. The original animation (left column), and two reconstructions of the animation using different identification results (centre column: cost = 0.014; right column: cost = 0.642).

4 Conclusion

These first encouraging results demonstrate the use of computer-generated 3D geometric data to identify physical parameters. To apply this method to real images, it will be necessary, first to implement low-level vision algorithms to locate reference points on the cloth, then to adapt the cost function in order to identify parameters from 2-dimensional data (projections of 3D positions) extracted from real cloth images.

The given example shows the transferability of the model identification general method described in [L94], to a simplified cloth model using a reduced set of parameters. The main differences lie in the introduction of a non-linearity factor to prevent hyperelastic behaviours, and the existence of multiple occurrences of identical springs. The main consequence of which is the introduction of "short" cost functions based on single frames, made necessary by the otherwise excessive calculation cost. The number of parameters used in the model is not a critical issue in the identification process; in the general case, [L94] showed that the number of generations needed to converge is theoretically independent of the number of parameters, and that the calculation cost is about proportional to the number of parameters involved. Present research aims at verifying this in the application to cloth identification, using more complex models.

References

[AG85] W.W. Armstrong, M.W. Green, "The Dynamics of Articulated Rigid Bodies for Purposes of Animation", Proc. of Graphics Interface, 1985.

[BHW94] D. Breen, D. House, M. Wozny, "Predicting the drape of woven cloth using interacting particles", Proc. Siggraph 94, Computer Graphics, 1994, pp. 365-372.

[CYMT92] M. Carignan, Yang Ying, N. Magnenat-Thalmann, D. Thalmann, "Dressing Animated Synthetic Actors with Complex Deformable Clothes", Proc. Siggraph 92, Computer Graphics, 1992, Vol. 26, No. 2, pp. 99--104.

[G89] D. A. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley 1989.

[K92] J. Koza, "Genetic Programming", MIT Press, 1992.

[L94] J. Louchet, "An Evolutionary Algorithm for Physical Motion Analysis", British Machine Vision Conference, York, Sep. 1994.

[NM93] J. Thomas Ngo, Joe Marks, "Spacetime constraints revisited", proc. Siggraph 93, Computer Graphics, pp. 343-350.

[P95] X. Provot, "Deformation Constraints in a Mass-Spring Model to describe Rigid Cloth behavior", Graphics Interface 1995, Québec, April 1995.

[TPBF87] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, "Elastically Deformable Models", Proc. Siggraph 87, Computer Graphics, 1987, Vol. 21, No. 4, pp. 205-214.