

Evolving Physical Models to Understand Motion in Image Sequences

Bogdan Stanciulescu, Jean Louchet
ENSTA – Ecole Nationale Supérieure de Techniques Avancées
32, boulevard Victor, 75739 Paris Cedex 15, France
Phone: +33-1-45524094, Fax: +33-1-45528327
email: {stanciul,louchet}@ensta.fr

ABSTRACT: A new method to identify internal parameters of a mass-spring deformable object model from kinematic data is presented. It uses an evolutionary model identification algorithm, with several original features concerning convergence and adaptivity. Experimental results are presented.

KEYWORDS: evolutionary strategies, particle systems, image animation, neural networks, physical modelling, pattern recognition

1. MOTION UNDERSTANDING: ‘HOW’ OR ‘WHY’?

Identification of image animation model parameters from real life data may be considered as one of the tasks of pattern recognition. Kinematic model-based image sequence interpretation may not be sufficient for some applications where mechanically oriented descriptions may be needed (e.g. motion prediction or simulation). When analysing images using geometric, photometric and motion modelling tools borrowed from computer graphics, the links between model parameters and their pixel-level effects may not be obvious and it then becomes necessary to explore the parameter space in order to find out the best match between real and synthetic images.

Among many animation-oriented modelling tools, we chose particle-based models as they are adapted to realistic simulation of a broad class of physical processes. Their modularity is a strong argument in the scope of identification.

2. A PARTICLE-BASED MODELLING SYSTEM

In particle-based animation models Reeves (1983), particles’ movements are generated through Newtonian point mechanics: for each particle i , with mass m_i , $f_i = m_i \ddot{x}_i$ where f_i is the total force applied and \ddot{x}_i the particle acceleration. The force model inputs are the particles’ positions, velocities and environment data.

Most particle-based models developed in the last decade e.g. Van Overveld (1995) did not consider the parameter identification problem. The Cordis-Anima particle system modelling language developed by Luciani et al. Luciani (1991), allows to model a wide range of objects, from complex mechanisms to materials such as sand, powder, smoke, turbulent fluids Luciani (1995), human flows Bouvier (1995). Based on these, we implemented a modelling system consisting of *points* with masses, and *interactions* (bonds). An object is defined as *a set of particles having their masses, positions and speeds, and of bonds between these particles*. Three bond types are considered: *unary* bonds, involving one particle (e.g.

viscosity, gravitation, etc.); *binary* (“stretch”) bonds between two particles; *ternary* (“flexion”) bonds, between three particles. Each bond (whatever its type) may consist of two components:

- an elastic (“spring”) component which generates a force only depending on particles positions. It may include non-linearities and upper and lower distance thresholds (“conditional bonds”).
- a viscous (“damper”) component which generates forces depending on the particles’ mutual speeds. Dampers may also include linear non-linearities and distance thresholds.

An object is constructed in three steps:

- define particles with their masses and initial positions/speeds;
- create a catalogue of bonds with their mechanical properties (stiffness, rest length, damping coefficients etc.);
- complete object building by installing bonds taken from the catalogue.

Once the object is defined, the animation program calculates, for each time step, the forces applied to each particle and deduces the particles’ accelerations, speeds and positions at the next time step, enabling calculation of image sequences.

3 EVOLUTIONARY OPTIMISATION TO IDENTIFY THE MODEL

Our target is to identify a mechanical model of an object from kinematic data. We suppose these kinematic data have been obtained through low-level image data processing.

3.1 COST FUNCTION AND EVOLUTIONARY STRATEGY

Model identification can be considered as the optimisation of a cost function defined as a generalised distance in the space of trajectories, between the trajectory predicted by the model being evaluated, and the real given trajectory. The more realistic the model is, the lower its cost will be.

We observed that in general, conventional numerical optimisation techniques are unsuccessful on these cost functions, often getting trapped into local minima. Therefore we used an Evolutionary Strategy approach, inspired by biological evolution Rechenberg (1994). It consists in creating a random initial population of models, coding them as parameter vectors, and letting the population evolve through the effects of *selection*, *crossover* and *mutation* operators.

3.2 LOCAL COST FUNCTIONS

We experienced that a conventional evolutionary scheme loses rapidly efficiency and precision with larger number of particles and bonds to be identified. Therefore in order to better exploit the problem’s specificities we designed a *short-term* cost function which calculates the square sum of differences between the real reference trajectory points, and those predicted by the model using the preceding time step. We could then exploit the fact that the *position of a particle at a given time only depends on the recent history of its neighbourhood*. The position of a particle at time $t+1$ only depends on the positions and speeds of its neighbours at time t . This allows to split the short-term cost function into each particle’s contributions, called the “*local cost functions*”:

$$local_cost(i) = \sum_{\text{time steps}} \left((x_{\text{short term predicted}} - x_{\text{real}})^2 + (y_{\text{s.t.pred}} - y_{\text{real}})^2 + (z_{\text{s.t.pred}} - z_{\text{real}})^2 \right)$$

$$global_cost = \sum_{\text{particle } i} local_cost(i)$$

The local cost functions allow better parameter separability, resulting in *the number of generations required for convergence not depending any more on the number of unknowns or the object’s complexity*.

3.3 GENETIC OPERATORS

In this Section, we give some details of the identification algorithm, corresponding to the identification of *linear, non-conditional binary spring* bonds. The operators dedicated to the other parameters are very similar.

Coding and mutabilities

Each individual is coded as a sequence of bond parameter values. In order to provide the algorithm with a *self-adaptive* behaviour Bäck (1995), we introduced “mutability” parameters, which control the standard deviation of mechanical parameter mutations. The mutabilities have no effect on the behaviour or cost of the individuals, but on the way they evolve. Thus, the balance of robustness and precision remains under the control of input data and cost function, resulting in a much better accuracy of parameters’ final estimation without real penalty on execution speed.

Initialising

The initial population (typically 100 individuals) is created by generating random values for each parameter. The parameters of a simple binary spring are: rest length, stiffness and the corresponding mutabilities. Length parameters are initialised using average observed distances, other parameters using Log-Gaussian repartitions.

Mutations

Four mutation types are used concurrently, creating entirely or partly new individuals, or creating modified replicas of existing individuals using random values with mutability-controlled standard deviations. The individuals partially replicated are chosen with a bias in favour of those having good local cost function values at the sites involving the bond to be mutated. Mutabilities are transmitted and mutated under the global cost’s control, which allows them to participate to global evolution and help the evolutionary strategy to improve precision when adequate.

Multi-parent crossover

For each bond type, all the particles which are an extremity of an occurrence of this bond type are examined; for each of these particles, a parent is chosen randomly, with a bias in favour of locally good ones. The new values of parameters are calculated as the mean values of all the parameters taken from all the parents chosen for this bond type.

Selection

The selection process used here is an ‘elitist’ Goldberg (1989), random choice on the population, biased in favour of the globally best individuals. In addition, we use a variant of the Powell method to locally optimise the cost function every 100th generation beginning at the 400th, as an earlier application leads to premature convergence into local minima.

4 EXPERIMENTAL RESULTS

4.1 GENERAL IDENTIFICATION AND NOISE RESISTANCE RESULTS ON SYNTHETIC DATA

In order to test the identification algorithm's performance on synthetic data, we used an experimental protocol which consists of building an arbitrary object and initial conditions and calculating a trajectory: the test consists in 'forgetting' the object parameters and use the evolutionary strategy described above in order to retrieve these parameters from the given synthetic trajectory. The curves below show convergence results in function of the number of generations on inelastic object consisting of 15 masses, 10 different linear bond types (20 parameters) and 31 installed bonds, with different noise levels on the trajectory. The identification algorithm uses 100 consecutive images and a population of 100 individuals. In order to test the algorithm's noise resistance, we added a Gaussian random noise to the kinematic data with different standard deviations.

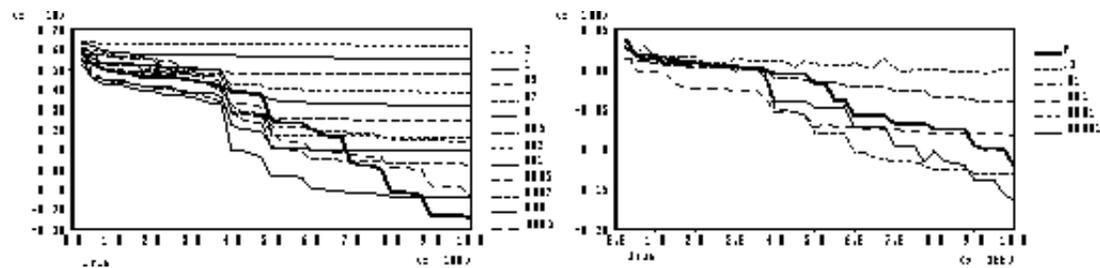


Figure 1: Log(cost function)

Log(precision of parameters' estimation)

The asymptotic levels can be interpreted as a distance between the given noisy kinematic data and the closest physically constructible kinematics. Other tests show that convergence is very good after 500 to 1000 generations, independently of the number of parameters to be identified. In the typical convergence results shown above, the average parameter estimation error after 1000 generations is lower than 0.01% for the three lowest noise levels shown.

4.2 CLOTH MODELLING

Provot (1995) developed a cloth model consisting of a periodic mesh with 6 spring types: horizontal and vertical "structural springs" between neighbours, diagonal "shear springs" and double-length "flexion springs". An inverse dynamics constraint allows a maximum elongation rate τ_c . The model includes 18 parameters (6 bond references described by 3 parameters each). Thanks to the replication of large numbers of identical springs in varied elongation states, we could only use a single image, corresponding to the equilibrium state at the end of the synthetic sequence. Convergence is obtained after about 50 to 100 generations and gives results in good accordance with the reference. The table below shows typical values of 5 parameters of the best individual in a 100 individual population, at the 50th generation, with a 1717 mesh. The first line shows the reference values (which were not known by the algorithm):

| | length01 | length23 | length45 | stiffness01 | τ elongation rate | cost |
|----------------------|----------|----------|----------|-------------|------------------------|----------|
| <i>reference</i> | 0.093750 | 0.132583 | 0.187500 | 4.000000 | 0.500000 | 0.000000 |
| estimated value | 0.093984 | 0.132705 | 0.187083 | 4.084211 | 0.497108 | 0.014619 |
| estimation error (%) | 0.24 | 0.09 | 0.27 | 2.1 | 0.58 | - |

Table I: Typical values obtained for 5 parameters

The cost of 0.0146 corresponds to an average error on particles' positions of about 5% of the mesh period and gives hardly perceptible visual differences with the original sequence.

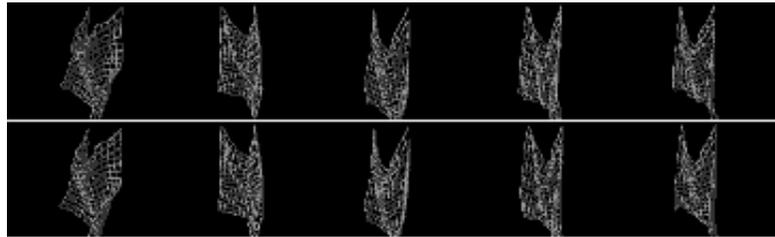


Figure 2: Five frames from the original animation and their reconstructions from identified parameters (cost = 0.0146)

4.3 FLUID MECHANICS

Particle-based models have long been used to simulate fluid objects like clouds or smoke. After Luciani (1995) who obtained visually convincing simulations of turbulent fluid flows using conditional viscous and elastic bonds, we examined how our identification algorithm could reconstruct the parameters of such a turbulent fluid model.



Figure 3: Four images of a jet penetrating into a fluid (frames nos. 100, 200, 300 and 400 from a sequence)

The main difference with the general model is the high number of bonds installed (and therefore the computational load of the cost function) compared to standard flexible objects (about 80000 bonds in the 2-D example above) even if not all are activated at the same time. With 100 individuals and after 1000 generations, the algorithm gives very good estimates of the initial parameters, with most estimation errors lower than 1%.

| | gas visc | gas visc n,l | visc. dist | jet visc | jet visc n,l | jet visc. dist | elast | elast. dist |
|----------------------|----------|--------------|------------|----------|--------------|----------------|-------|-------------|
| <i>reference</i> | 2.5 | 1.25 | 1.5 | 4 | 2 | 2 | 2 | 1.1 |
| estimated value | 2.499 | 1.248 | 1.500 | 3.962 | 2.052 | 2.000 | 2.000 | 1.100 |
| estimation error (%) | 0.04 | 0.16 | 0.0 | 0.96 | 2.6 | 0.0 | 0.0 | 0.0 |

Table II.: Experimental results of identification algorithm

4.4 HORSE WALK AND ACTIVE MUSCLES

We processed kinematic data obtained from measurements on a race horse at a trot. The horse has been fitted with 35 3-D markers sampled at 50Hz. Each leg includes 5 markers, the remaining 15 correspond to the body and the head. Masses were estimated using standard veterinary data. Using linear springs gives a good convergence, but the parameter values obtained give an unstable model. Introduction of dampers only gives a slightly (2%) lower final cost function and better model stability. The visually still “rubber-like” motion obtained strongly suggests introducing the equivalent of muscle control into the available bond types and into the identification algorithm. This is why we have introduced active controllers into the particle system. An ‘active muscle’ consists of a visco-elastic bond, associated with a feed-forward neural

controller whose inputs are the elongations and elongation speeds of other bonds. The internal parameters of the neurones are identified using a straightforward extension of the evolution strategy described above. Experiments on synthetic data which had been generated without neural controllers show little differences, but experiments on the horse sequence show that with neural controllers, consistently lower cost values (and therefore more accurate modelling) are obtained.

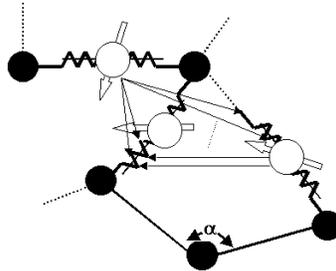


Figure 4: A mass-bond model with neural controllers

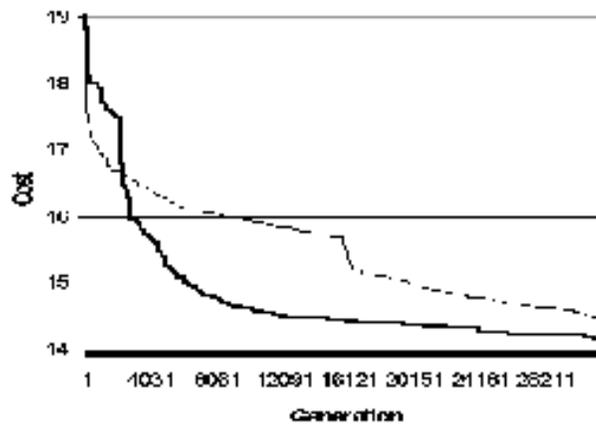


Figure 5: Log (cost function) without (thin) and with active muscles (bold), in function of the number of generations.

5 CONCLUSION

The motion model identification technique presented in this paper has proven successful to reconstruct model internal parameters from kinematic sequences, including non-linear, conditional, elastic or viscous bonds in periodic or non-periodic objects, and thus provide the particle-based motion modelling tool with the ‘measuring instrument’ which should always be associated to a physical model. We also extended the particle-based approaches to much complicated behaviours, like the living-beings motion, by enriching our physical model with low-level control capabilities, here represented by the neural command system. For the future we intend to develop low-level image processing methods to extract particles extracting particles’ trajectories from image sequences.

6 REFERENCES

Bäck, T.,1995, “*Evolution Strategies: an alternative evolutionary algorithm*”, Artificial Evolution 1995, Brest.

- Bouvier, E., Cohen, E., 1995, "*Simulation of Human Flow with Particle System*", *Simulators International* XII, 27 (3), April 1995, pp. 349-354.
- Goldberg, David E., 1989, "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison Wesley, 1989.
- Holland, J., 1975, "*Adaptation in Natural and Artificial Systems*", MIT Press (reprint, 1992).
- Luciani, A., Habibi, A., Vapillon, A., Duroc, Y., 1995, "*A Physical Model of Turbulent Fluids*", Eurographics Workshop on Animation and Simulation, Maastricht, Springer, September 1995.
- Luciani, A., Jimenez, S., Florens, J. L., Cadoz, C., Raoult, O., 1991, "*Computational Physics: a modeller simulator for animated physical objects*", Proc. Eurographics Conference, Wien, Austria, Elsevier, September 1991.
- Provot, X., 1995, "*Deformation Constraints in a Mass-Spring Model to describe Rigid Cloth behaviour*", Graphics Interface 1995, Québec, Canada.
- Rechenberg, I., 1994, "*Evolution Strategy*", in J.M. Zurada, R.J. Marks II, C.J. Robinson, Computational Intelligence imitating life, IEEE Press (1994) pp.147-159.
- Reeves, W.T., 1983, "*Particle Systems - A Technique for Modelling a Class of Fuzzy Objects*", Computer Graphics (Siggraph) vol. 17 n° 3, pp. 359-376.
- Van Overveld, K., Barenbrug, B., 1995, "*All you need is force: a constraint-based approach for rigid body dynamics in computer animation*", Eurographics Workshop on Animation and Simulation, September 1995, Springer, pp.80-94.