PROCESSING ELASTIC PATTERNS THE MASSIVELY PARALLEL WAY

Francis Van Aeken † — Yves Demazeau ‡

Artificial Intelligence Laboratory, Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussel

ABSTRACT

In 1989, Yves Demazeau et al. developed at LIFIA the concept of the Elastic Pattern as a new tool to be used in early computer vision [Demazeau 89]. Instances of the model have been implemented on both sequential and parallel architectures. Lately, we have been redesigning the model from a Multi-Agent point of view. One of the major motivations for doing so was the fact that Elastic Patterns can be much more easily implemented in a parallel distributed fashion, when they are described as Multi-Agent systems. In this paper, we describe the original and the parallel distributed concept, and suggest in what way instances of these models can be processed by a distributed array processor.

THE ORIGINAL MODEL

The original Elastic Pattern finds its roots in, on the one hand, the Elastic Net Method [Durbin 87], and, on the other hand, Active Contour Models [Kass 87]. Instead of considering the pattern as a mathematically describable curve, which energy one tries to minimize, however, it is conceived as a discrete set of elements that interact with each other and with the environment by means of forces. The resulting system can be considered a complex dynamical one [Steels 88]. The original Elastic Pattern was developed as a specific system for contour detection in early computer vision. From the beginning, however, several applications in different fields were envisioned.

Although we will use Multi-Agent terminology when describing the Elastic Pattern models, one has to realize that we are dealing with simple entities rather than with complex agents. Still, we believe that the research reported in this paper contributes to the Multi-Agent field, and that the use of Multi-Agent terminology is appropriate.

As basic elements of a general Elastic Pattern, we can distinguish the agents, their environment, the interaction between agents themselves, and the interaction between agents and their environment.

The Agents

An Elastic Pattern consists of a fixed number of similar agents. In the original concept, these agents have an index which is also fixed. All agents are constructed in the same way, but their behavior can vary widely, depending on the state of their local environment.

† Francis Van Aeken can be reached by email at francis@arti.vub.ac.be.

‡ Yves Demazeau can be contacted by email at demazeau@arti.vub.ac.be. His permanent address is LIFIA/IMAG/CNRS, 46 Av. F. Viallet, F-38031 Grenoble Cedex (demazeau@lifia.imag.fr). He has been involved in the research reported as a visiting professor at the Vrije Universiteit Brussel, holding a SWIFT AI chair, and with support from CNRS.

Every agent is given at least three attributes: a mass, a position and a velocity. All quantities are expressed as real scalars or real vectors.

The Environment

The environment is a finite discretized subset of n-dimensional space, with a number of values and symbols associated with each element. This environment can be a static or a dynamic one. A static environment could be the map of intensity values of a still picture. We would be dealing with a dynamic environment if we, for example, were processing a sequence of video frames.

In the simplest, two dimensional case, we have a finite grid with only one value associated with each element.

The Interaction between Agents

Each agent has a number of relationships with other agents. These relationships are fixed on a small time-scale, forcing the agents to form a pattern, and, thus, to represent knowledge, but might vary over a longer period of time. The *state* of the relationship, though, is free to vary over short time spans. This accounts, in part, for the elasticity of the pattern.

In the simplest case, the relationships consist of forces exerted by each agent on its neighbors. These forces are comparable to the internal forces found in Elastic Nets and Active Contour Models. Spring-like models, for example, have turned out to be most useful.

The Interaction with the Environment

Each agent interacts with the elements of the surrounding environment. The size of this surrounding environment is unlimited and fixed in the original Elastic Pattern concept.

Again, in the simplest case, the relationships consist of forces exerted by the elements of the environment on the agents. These forces are the equivalent of the external forces used in Elastic Nets and Active Contour Models. Combinations of spring-like and gravitational models seem to be suitable.

Apart from these external forces, however, there is another important interaction that should be considered: friction. Friction, immobilizing agents on elements with a high friction-coefficient, enables the pattern to gradually lock on interesting features.

Of course, just as in real-life static friction, this counteractive force will disappear when internal or external forces get too high.

PROCESSING THE PATTERN I

The first implementations of Elastic Patterns were realized on a sequential architecture, a Sun-3. To speed up the system, and to make use of the parallelism inherent to the model, we reimplemented it on an DAP, a Distributed Array Processor, manufactured by Active Memory Technology [AMT 88]. Before we explain how the Elastic Pattern is actually processed, we briefly discuss the AMT DAP.

The AMT DAP

The core of the AMT DAP 500 is a 32 by 32 grid of 1-bit processing elements, which are connected to their neighbors and to high speed horizontal and vertical communication channels. Each clock cycle, all 1024 processors perform in parallel the same boolean operation on their private data, making the DAP a Single Instruction, Multiple Datastream (SIMD) machine.

Data-structures well suited to the DAP are 32- and 1024-dimensional vectors (*short* and *long DAP vectors*), and 32 by 32-dimensional matrices (*DAP matrices*).

Program execution is controlled by a 32-bit sequential processor.

The Processing

The agents are represented as an array of positions, velocities and, optionally, masses. Because of the use of long DAP vectors, a number of computations can be performed in parallel. Note, however, that we need full occupancy of the vectors, i.e. a population of 1024 agents, in order to exploit the parallelism in full.

The environment is represented as a set of positions and corresponding weights. These weights correspond to contrast-values at relevant positions in the environment. To prevent excessive computation, not the entire environment, but a subset, containing the elements with the highest weights, is represented. The cardinality of this subset is restricted to multiples of 1024, so that it can be represented as a number of long DAP vectors, and parallel computation of the external forces is optimized.

In figures 1 to 3 an experiment performed using the original model is illustrated.

THE PARALLEL DISTRIBUTED MODEL

The mapping of the original Elastic Pattern model on the DAP architecture is not entirely satisfying. Because of the agents' unlimited field of perception, really distributed computation is impossible. As a matter of fact, we are only simulating a naive physics system, and might as well use any number cruncher, whether it has a parallel architecture or not.

A key step in making the model parallel distributed is the introduction of two elements from general Multi-Agent systems: discretization of all aspects of the agent and scope.

We discuss the new concepts introduced and the consequences for the Elastic Pattern's basic elements.

Discretization

In the original Elastic Pattern, the positions and velocities of agents are real, continuous quantities. Moreover, agents do have a mass, but do not occupy any space.

The environment, however, is a discrete one. So, it seems only natural to give the agents, too, a discrete nature. A major motivation for introducing discretization is the fact that we like to use an analogical representation [Steels 89], that must be, on digital computers, discrete.

Taking into account the new nature of the agents, it is easy to consider them occupying a certain space related to the resolution of the discretization. This way of looking at things does not allow for space-sharing between agents. Connected with this are some advantages and disadvantages. On the one hand, agents are easier to represent and to process analogically since they cannot overlap. On the other hand, conflicts will occur when two agents want to occupy the same cell: some form of conflict-resolution must be introduced.

Scope

In the original Elastic Pattern, agents are able to perceive other agents and environmental elements, no matter how far they are located. Perception is also immediate and independent of the distance.

At first sight, this approach has only advantages: certainly an agent cannot have *too much* information. This might not be true: too much information can confuse an agent. Moreover, it seems natural that only the close neighborhood of an agent is of immediate relevance to that agent. Indeed, it would be strange if a far away located entity would have an immediate and profound influence on the behavior of an agent. Another motivation to limit the field of view of the agents is the fact that unlimited perception capabilities are obviously incompatible with distributed processing. Indeed, the related communication overhead would make any implementation impractical.

Because of all this, we have introduced the notion of *scope*. The scope of an agent is the subset of the environment that is visible to that agent. Basically, there are two kinds of scope: an internal one, related to the perception of other agents, and an external one, related to the perception of environmental elements. The internal and external scope can vary from agent to agent and from cycle to cycle, but is always bounded by some predefined maximum. Careful tuning of the scope, what can be incorporated in the agent's behavior, can be an important contribution to the system's intelligence.

Consequences for the Basic Elements

The agents are now defined as entirely discrete entities, with two attributes added: an internal scope, that defines the field of view in regard to other agents, and an external scope, that defines the subset in which environmental entities are visible. Both scopes are controlled by the agent itself. If we want the agents to be explicitly linked, as is the case in the original model, other attributes have to be added: an identity, a symbol that is needed to identify agents, and a number of links, which describe the identities of the agents that the agent is linked with.

The environment, being already discrete in the original system does not have to be changed.

Interaction between agents themselves is still guided by internal forces, but, different than in the original model, an agent is not influenced by all other agents. Only the agents that are within its internal scope *and*, in the case of an explicitely linked pattern, that it is linked with can exert forces on it.

As in the original model, interaction between agents and their environment consists of external and frictional forces applied by the environment on the agents. In the new system, however, an agent is only sensitive to environmental entities within its external scope.

PROCESSING THE PATTERN II

Both the configuration of agents and the environment are represented as 2-dimensional maps composed of DAP matrices. This analogical representation, combined with a bounded scope, allows for a very efficient implementation of the Elastic Pattern on an array processor. Indeed, by mapping the processor grid on the relevant subset of the analogical representation, all interactions with the environmental entities and neighboring agents within the two scopes can be handled in parallel. The mapping process consists of shifting and masking operations, both very fast on the DAP.

Note, however, that the agents are processed one by one, and that a sequencer is necessary to handle the different requests. Obviously, performance can be considerably increased by eliminating this sequential processing, and distributing the computation associated with each agent over a Multiple Instruction, Multiple Datastream (MIMD) machine. In an ideal case, we would have one MIMD computer co-operating with one or more SIMD machines.

An experiment performed using the parallel distributed model is illustrated in figures 4 to 6.

CONCLUSIONS

The introduction of a new concept, the Elastic Pattern, has enabled us to make new use of array processors in early computer vision. This would not have been possible without inspiration from Multi-Agent Systems. Indeed, the original Elastic Pattern model, in which there is no notion of discretization and scope, turned out to be ill suited to a parallel distributed architecture.

On the other hand, the concepts of discretization and scope were partially inspired by the parallel architecture itself, so that the relationship between the conceptual and hardware level is actually a bilateral one.

FIGURES

Figure 1: initial configuration.

Figure 2: intermediate configuration.

Figure 3: final configuration.

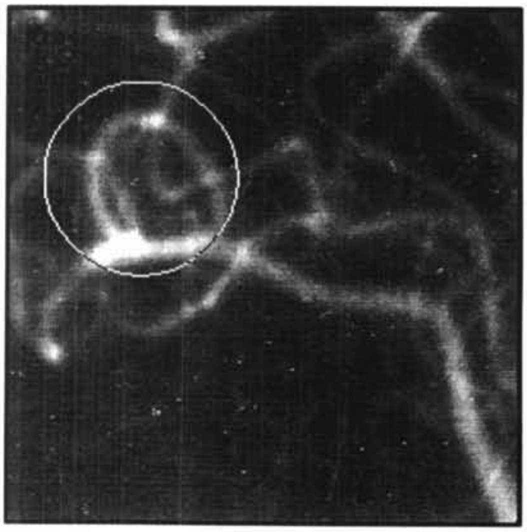

Figure 4: initial configuration.

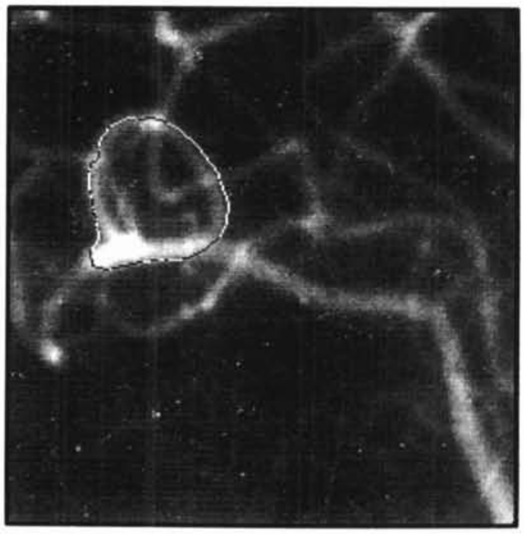

Figure 6: final configuration.

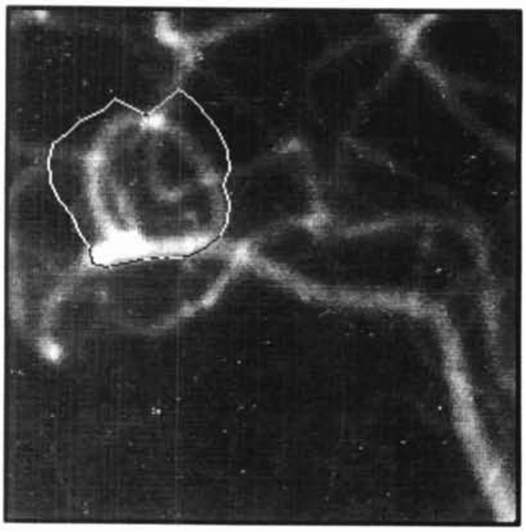

Figure 5: intermediate configuration.

REFERENCES

[AMT 88] Active Memory Technology, *DAP Series Technical Overview*, 1988.

[Demazeau 89] Demazeau Y., Bourdon O., Lebrasseur M., *Contours et Illusion de Contours: Un Gabarit Elastique pour l'Extraction de Formes*, 1er Workshop Régional de Sciences Cognitives, CNRS-LASCO3, Chichilianne, March 1989.

[Durbin 87] Durbin R., Willshaw D., *An Analogue Approach to the Travelling Salesman Problem Using an Elastic Net Method*, Nature, vol. 326, April 1987, pp. 689-691.

[Kass 87] Kass M., Witkin A., Terzopoulos D., *Snakes: Active Contour Models*, Proceedings of the 1st International Conference on Computer Vision, London, IEEE, 1987, pp. 259-268.

[Steels 88] Steels L., *Artificial Intelligence and Complex Dynamics*, AI-MEMO no. 88-2, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, January 1988.

[Steels 89] Steels L., *Exploiting Analogical Representations*, AI-MEMO no. 89-19, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, December 1989.

The original images used in the experiments illustrated were supplied by the *GRECO-Traitement du Signal et Imagerie* program promoted by the French Ministère de la Recherche et de la Technologie.