Journal of Math-for-Industry, Vol. 4 (2012A-7), pp. 49–53

Math-for-industry
Education & Research Hub

Modeling through self-assembly

Jos Stam

Received on February 1, 2012

Abstract. In this paper we explore a new paradigm for modeling geometric structures through self-assembly. This approach is inspired by the new emerging field of nano-technologies. At the very small nano-scales the laws of physics are different from the ones at the scales we are used to in daily life. Gravity is negligible and Brownian motion induced by heat is a crucial factor. In fact the latter provides the vital force that drives the process of creating new shapes at nano-scales: heat induced noise makes it possible for programmed DNA chains with free bonds to form and create shapes by bonding with other strands. The key challenge is how to program the DNA strands to create specific shapes. In this paper we introduce some concepts of this exciting new area of research and describe a couple of concrete self-assembly modeling examples. The goal of this paper really is two-fold: (1) to show an illustration of self-assembly at work in an appealing way using computer graphics and (2) to bring this exciting field to the attention of researchers in other fields.

Keywords. molecular biology, self-assembly, computer graphics, geometric modeling, physics-based simulation

1. Introduction

Traditionally computer graphics geometric modeling is concerned with the creation of familiar shapes like buildings, cars, tables and characters. That is to say shapes at our human scale that we are used to. Things we can see, touch, smell, hear and taste in our daily life. Obvious applications of geometric modeling are in the area of computer aided design (CAD) that help create real objects and also in the area of three-dimensional content creation for the entertainment industry to produce movies and games. All of these applications aim at creating models that result in one way or another to shapes that are directly available to our senses.

On the other hand there is another world at vastly smaller scales that is inhabited by DNA chains and proteins. This is the world of micro-biology. In this realm objects are measured in nano-meters which is about one thousands of the width of a human hair. At this scale geometric modeling takes on another character. Instead of building shapes through a top down design process, shapes emerge through a bottom up dynamical process, they are built through local interactions. Elements such as DNA chains assemble by bonding with other elements. These bonds are made possible by the random motions that these chains undergo due to thermal noise. Through this process free bonds are brought in proximity resulting in connections that self-assemble into larger structures. The dynamics of the chains is therefore a key ingredient to the modeling process. At the nano-scale modeling and simulation are really intimately connected. This is what makes it such an exciting modeling paradigm. This paper explores how this paradigm can be achieved using computer graphics tools. We model DNA chains as one-dimensional curves. The curve dynamics are ruled by simple constraints such as stretch, bend and self-collisions. Each end point of the curve is labeled by a bond constraint. Only end points of the curves with the same label can bond together when brought in proximity. The bonds form because the vertices of the curves are constantly being subjected to random noise. The amount of noise is controlled through a cooling process. Early on in a heating phase the curves are shaken vigorously to create bonds. The final shape emerges through a cooling process which reduces the amount of noise applied. We mention in passing that this process is very much akin to popular simulated annealing algorithms that attempt to find a global minimum of some arbitrary function [4]. However, we will not further explore this analogy here.

This work builds on top of our Nucleus solver [7]. We add specific constraints to this solver to enable self-assembly.

In this paper we introduce some basic concepts from micro-biology to motivate our modeling system. The goal of this paper is not to directly map our simulations to real strands of DNA used in actual experiments. Our aim is to present a new way of modeling geometrical structures appearing at the nano-scale. In the next section we briefly review some concepts from microbiology that inspired this work. We then present our model followed by results that illustrate the concepts. We conclude this paper with out-

This is an invited paper presented at the Forum "Math-for-Industry" 2011.

lining exciting directions for future work.

2. Micro-Biology: DNA and Self-Assembly

Micro-biology is a fascinating subject. The goal here of course is not to give an expert introduction to the subject, just those aspects that inspired this work. Fortunately there are many books written by researchers in the field that do a great job at explaining this subject to laymen like ourselves. We arbitrarily single out Goodsell's "The Machinery of Life" as a good introduction to a computer graphics audience [2]. In particular this book illustrates most concepts with beautiful visualizations and clear prose. In this section we will cover the aspects of micro-biology that inspired us most: namely the structure and dynamics of DNA chains.

Figure 1: A DNA chain is a helix formed by two strands connected by complementary nucleotides A, T, G and C.

Figure 1 shows a portion of a DNA chain whose structure was first uncovered by Watson and Crick [8]. What is DNA? It is a macromolecule formed of two strands connected by complementary nucleotides: Adenine (A), Thymine (T), Guanine (G) and Cytosine (C). As shown in Figure 1 the only allowable bonding pairs are (AT), (TA), (GC) and (CG). DNA of course encodes our genetic code. But this is irrelevant for the purposes of this paper. We are more interested in its ability to form nano-structures. Seeman in his seminal works is probably one of the first researchers to explore this line of research in the early eighties, e.g., [6].

DNA chains are of finite size and one way to allow them to self-assemble is to shift the strands so that unpaired nucleotides are exposed and allowed to bond with their complementary partner from another strand. This is illustrated in Figure 2. There two nucleotides are exposed and in this case the two strands can bond and form a larger one through (AT) and (CG) pairings. In general adding N free nucleotides allows one to encode 4^N different pairings which can be uniquely labeled. These potential bonds at the extremities of the chains are often referred to as sticky ends.

However, these pairings result only in one-dimensional shapes. To get more interesting shapes there is a way to pair K single strands to form junctions as shown in Figure 3. For convenience we will call them stars of valence

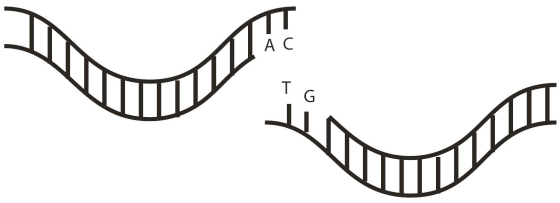

Figure 2: DNA with sticky ends can bond through complementary pairs of nucleotides.

K. See [6] how these strands can be synthesized in practice especially using a hierarchical process [3]. By adding sticky ends to each of these stars we can assembly arbitrary geometric shapes.

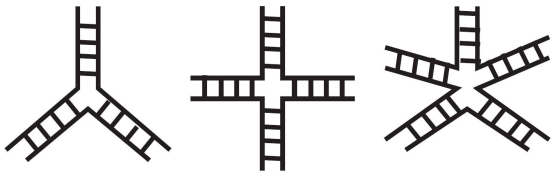

Figure 3: Connector DNA building blocks made from K (= 3, 4 and 5 here) strands can be used to create bonds between various DNA helices.

The vital ingredient after each DNA chain has been designed is the addition of thermal heat so that sticky ends with complementary labels will bond. Temperature is related to the kinetic velocity of the particles in the medium. At the nano-scale these particles act as a noise perturbing the motion of the DNA chains. This allows complementary sticky ends to get in close proximity and form bonds. The basic scenario is thus to heat up the solution containing the DNA chains and allow them to form bonds and then through cooling to let them reach an equilibrium state. This is not always easy to achieve and the cooling rate and initial concentration of DNA chains is key to the success of the self-assembling process in practice [3].

We understand that this description is overly terse and leaves out many interesting issues. However, it is sufficient to motivate the models and simulations that we will describe next. We claim no expertise in the area of micro-biology but we think that the fact that at least in principle this approach is feasible in practice is very exciting.

3. Geometric Modeling through Self-Assembly

We now translate these ideas into the realm of computer graphics. We model each DNA chain as a piece-wise linear non-manifold curves as shown in Figure 4. Each sticky end is encoded with a unique integer label: only the ends of each curve with the same label can bond when in close proximity. In theory one could always map this approach back to DNA chains with sticky ends. Given M labels one

would then need to synthesize DNA chains with sticky ends of size $\log_4(M)$.

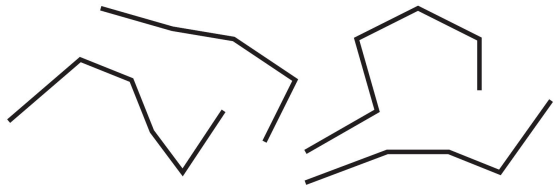

Figure 4: We model each DNA chain as a collection of piece-wise linear dynamic curves.

Our first step is to take an arbitrary mesh and split each edge into two to create V stars: one for each vertex, where V is the number of vertices as shown in Figure 5.

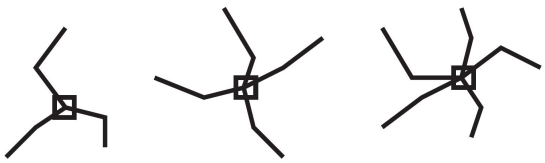

Figure 5: For each DNA star we create a vertex with K curves emanating from it.

At this point to synthesize the mesh from these DNA stars we use a dynamics solver for non-manifold curves. Recently there have been many such solvers one can choose from [7, 1]. The physical properties of the DNA stars exhibit the usual properties of dynamical curves: stretching and bending. Bending is especially important if the DNA star chains only have a few nucleotides. In fact the bending constraint helps the shapes to emerge naturally and not to get overly tangled. Another key factor in the simulation is the treatment of self-collisions between segments. To summarize our model has the following parameters:

- Stretch strength
- Bending strength
- Start temperature
- Final temperature
- Cooling rate
- Bond forming radius
- Bond strength
- Bond damping

In practice the temperature is just the amount of noise which can be related through a thermodynamic relation to a real temperature if needed. All these parameters influence the success of a self-assembling structure as we will show next in the examples section of this paper.

Our system is based on a particle based solver with constraints [7]. We used the standard constraints in that solver to handle stretching, bending and self-collisions. Through the API we added a Brownian noise force and a bonding strategy. For the latter we used a hash-table to determine when sticky ends (free labeled vertices) are in close proximity in order to decide when to create spring-like bonds between them. Once these bonds are created they are part of the solution process. We want to emphasize that our approach is solver agnostic. Any curve-based solver with a flexible API could have been chosen.

4. Results and Demo

We have written a demo program that takes as an input a mesh. Figure 6 shows snapshots of a simulation that self-assembles a single tetrahedron. The background fades from red to blue and indicates the current temperature. Notice how the edges bond when in close proximity. Figure 7 shows snapshots of various self-assembling meshes. In general the success of the assembly depends on the cooling rate and the bond strength. Animations of these figures can be found at `http://www.autodeskresearch.com/publications/jmi2012`.

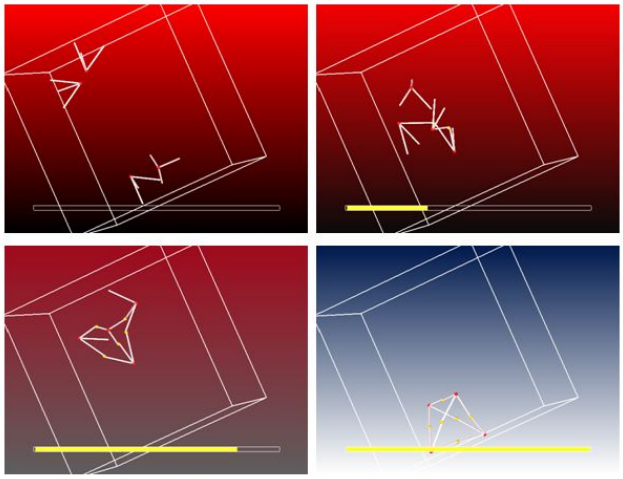

Figure 6: Four frames of a simulation of strands self-assembling into a tetrahedron.

In Figure 8 we show snapshots from a simulation that uses a dual approach. Instead of creating star-like DNA strands we assign one DNA strand per edge of the mesh. The tips of these strands can only bond with other strands that shared the same vertex in the original mesh. To keep the shape from collapsing we added an artificial repulsion force between bonds. It is not clear if this strategy could be achieved in practice but it nicely illustrates the paradigm of self-assembly.

Figure 9 shows a snapshot from a demo that was shown at TED Global 2011 in Edinburgh, Scotland. There strands self-assemble to form the TED logo.

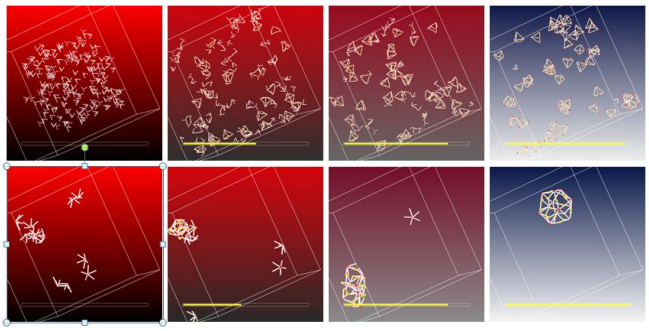

Figure 7: Sequences of 49 self-assembling tetrahedra (top) and an icosahedron (bottom).

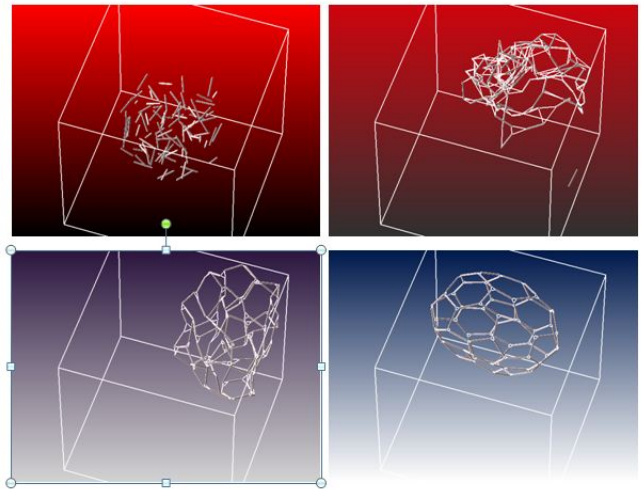

Figure 8: Four frames of a simulation of strands self-assembling into a bucky ball.

5. Conclusions and Future Research

In this paper we have introduced a modeling paradigm based on the concept of self-assembly. It was inspired by the field of micro-biology where shapes are synthesized by DNA strands that can stick together in restricted ways. This approach is useful to build static structures. A lot more research needs to be done to produce more complex shapes. In fact for some complicated shapes our simulations get tangled up because of self-collisions or never self-assemble because the cooling rate is too fast, etc. The choice of parameters is a key factor in synthesizing the right shapes.

Another challenge is to come up with self-assembling machines (nano-robots) which can perform specific tasks. These are equivalent to proteins in biology. This is currently a hot topic of research in micro-biology. For example, there is a lot of research in synthesizing nano-robots that will target and kill specific cancer cells. We also hope that this paper will inspire researchers in other fields to study this fascinating topic. For example, we believe that micro-biology research could potentially benefit from computer graphics tools and expertise. There are also interesting mathematical problems that arise from micro-biology. See for example the thesis by Reishus [5]. Another good example is how knot theory is relevant in this area as described in the overview article [9].

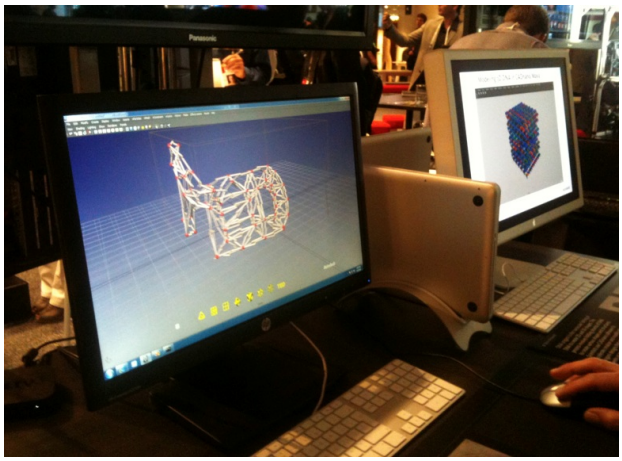

Figure 9: Our self-assembly demo was shown at TED Global 2011.

Acknowledgements

Thanks to Carlos Olguin for introducing me to the fascinating world of micro-biology and many fruitful discussions. Also thanks to Wei Zhao from our Shanghai office for implementing the TED demo in our MAYA software.

References

[1] Bergou, M., Wardetzky, M., Robinson, S., Audoly, B., and Grinspun, E.: Discrete Elastic Rods, Transactions on Graphics (SIGGRAPH 2008), **27**, 3, 63:1–63:12, 2008.

[2] Goodsell, D.: The Machinery of Life, Springer-Verlag, 1993.

[3] He, Y., Ye, T., Su, M., Zhang, C., Ribbe, A. E., Jang, W., and Mao C.: Hierarchical Self-Assembly of DNA into Symmetric Supramolecular Polyhedra, Nature, **452**, 13 March 2008, 198–202.

[4] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P.: Optimization by Simulated Annealing, Science, **220**, 4598, 671–680.

[5] Reishus, D.: On the Mathematics of Self-Assembly, Ph.D. Dissertation, Department of Computer Science, University of Southern California, December 2009.

[6] Seeman, N. C.: Nucleic Acid Junctions and Lattices, Journal of Theoretical Biology, **99**. 1982, 237–247.

[7] Stam, J.: Nucleus: Towards a Unified Solver for Computer Graphics, in *2009 Con-ference Proceedings:*

IEEE International Conference on Computer Aided-Design and Computer Graphics, Yellow Mountain, China, August 2009, 1–11.

[8] Watson, J. D., and Crick, F. H.: Molecular Structure of Nucleic Acids; A Structure For Deoxyribose Nucleic Acid, Nature, **171**, 4356, 25 April 1953, 737–738.

[9] De Witt, S.: Lifting the Curtain: Using Topology to Probe the Hidden Action of Enzymes, Notices of the AMS, Volume 45, Number 5, May 1995, 528–537.

Jos Stam
Autodesk Research
210 King Street East, Toronto, Canada M5A 1J7
E-mail: Jos.Stam(at)autodesk.com