Computational Intelligence and Applications

Editors

Ajith Abraham
Monash University, Australia,

Baikunth Nath
University of Melbourne, Australia,

M. Sambandham
Morehouse College, Atlanta, Georgia, U.S.A., and

P. Saratchandran
Nanyang Technological University, Singapore

Dynamic Publishers, Inc., U.S.A

ISDA 2002

per si

Second International Workshop on Intelligent Systems Design and Applications

August (07-08), Atlanta, Georgia, USA

ISDA'02 Organizing Committee

General Chair

Ajith Abraham
Faculty of Information Technology,
School of Business Systems,
Monash University (Clayton Campus), VIC 3168, Australia
Phone: +61 3 9905 9766, eFax: +1 (509) 691-2851
Email: ajith.abraham@ieee.org
URL: http://ajith.softcomputing.net

Program Chairs

P. Saratchandran
Division of Control & Instrumentation,
School of Electrical and Electronic Engineering,
Nanyang Technological University, Singapore 639798
Telephone: +65 790 5029
Email: epsarat@ntu.edu.sg

Baikunth Nath
School of Computer Science and Software Engineering
The University of Melbourne, Victoria 3010, Australia
Telephone: +61 3 8344 9316
Email: baikunth@unimelb.edu.au

Local Organizing Chair

M. Sambandham
Department of Mathematics
Morehouse College, Atlanta, GA 30314, USA
E-mail: msamband@morehouse.com
Phone: (404) 215-2614, Fax: (404) 589-1661

Information Analysis, Knowledge Discovery and Data Mining

Other Intelligent Techniques and Applications

Computational Intelligence and Applications (2002), 241-244

An Intelligent Method for Processing String in 3-D Based on its Minimum Energy

Rahmat Budiarto[1], Abdullah Zawawi Hj. Taib[1], Zaharin Yusoff[1], Masashi Yamada[2]

[1]School of Computer Sciences, USM
11800 Minden, P. Pinang, Malaysia

[2]School of Computer and Cognitive Sciences,
Chukyo University, Toyota, 470-0383 JAPAN

1. INTRODUCTION

The problems associated with geometrical model of string (rope) in computer graphics are preserving topology and detail shape, and a large data volume in its processing. As an example, we consider string figures construction in Cat's Cradle game [3]. String figures in cat's cradle are designs formed from nothing more than a loop of string. Using their fingers, people weave string figures, and often a fabulous pattern is resulted (See Fig.1). The states change to new states according to inputs got from finger actions. It is necessary for real time I/O systems to employ high-speed processing to a large amount of data.

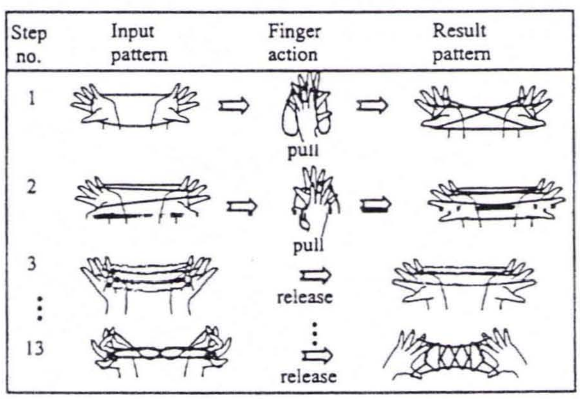

Fig. 1. A Cat's Cradle Construction

At present, there are many geometry models for representing string in three-dimension (3-D). We use Yamada's model [1], called as *Set of Crossings* and defined as follows.

$$D = \{C(x,y,z),\ p^1(x_1,y_1,z_1),\ p^2(x_2,y_2,z_2),\ p^3(x_3,y_3,z_3),\ p^4(x_4,y_4,z_4)\}, \quad (1)$$

where $C(x,y,z)$ is the coordinate of a crossing point C and a connecting point p^i ($i = 1,\ldots, 4$) is neighbor-crossing point. The model that is shown in Fig.3 gives us efficient data storage and easy to process structure. Our previous work proposed a processing method using geometrical and topological transformations of knots theory. However, when we verified the method to string figures of cat's cradle game, it failed to provide appropriate shapes for unsymmetrical patterns [2]. The method does not take into account physical characteristics of string, such as its energy and friction between parts of string.

As we can see from Fig.1, having done finger action(s) on the string loop, we can obtain invariant shapes of a string figure that have different energies.

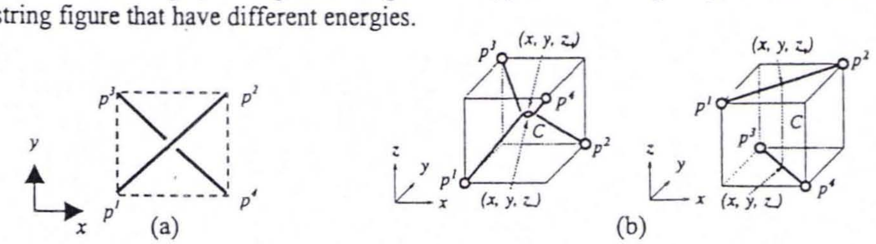

Fig.2. Crossing point and its types

A crossing point C in a string diagram is a point where two string parts cross each other. A crossing point C, which is looked down from z-axis direction is shown in Fig. 2(a).

The crossing points are classified into two types: type A and type B as shown in Fig. 2(b). Type A crossing points are composed of two string parts that the upper string touches and is tangled with the lower string. The type B crossing points are composed of two string parts that the two string parts do not touch each other. The crossing point has four connecting points (p^1, p^2, p^3, and p^4). A string diagram is constructed by a set of crossing points, connecting points and their connections.

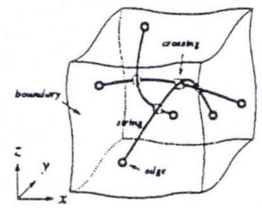

Fig.3. String diagram as a set of crossings

2. STRING (ROPE) PROCESSING

We represent a string figure by the string diagram of Fig.3. To find out a stable string diagram of a string figure we have to place all crossings of the string diagram on their proper positions in such a way that the energy of the string diagram is minimum.

In this paper we propose an intelligent searching by making use of *positional information* of crossing points in the diagram. The searching process is an adaptive search toward the *minimum energy* of the string diagram. Having obtained the positional information of all crossing points in the diagram with the minimum energy, we have to move all crossing points to their new positions. The movement should be done gradually to keep the minimum energy unchanged.

- **Topological Movements**

Before we determine the minimum energy of a string diagram, the string diagram will be simplified. We apply Reidemeister moves (Fig.4) [4] to a projection of string diagram on the xy plane. The Reidemeister move I and II reduce the number of crossing points in the string diagram, whereas, the Reidemeister III loses a bend of strings and reduces the length of string in the string diagram. We apply the Reidemeister moves to all type of crossing points.

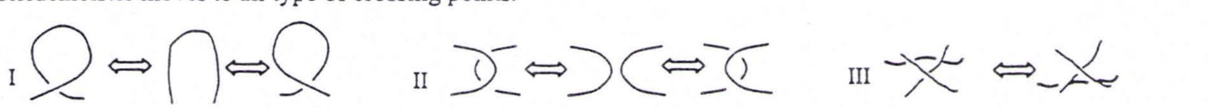

Fig. 4. Reidemeister Moves

- **Minimum Energy of The String**

To compute the energy, we use the Minimum Distance (MD) energy of regular n-polygonal knot, defined by J. Simon [5]. A polygonal knot K consists of several edges E_1, ..., E_n in the Euclidean space, which form a closed knotted loop. The ends of the edges are called the vertices of the knot. The energy contributed between E_i and E_j is

$$(L_i)(L_j)/(D_{ij}*D_{ij}) \qquad (2)$$

where, L_i is the length of E_i and D_{ij} is the minimum distance between E_i and E_j. The energy of K is obtained by summing such contributions over all E_i and E_j, which are not adjacent.

- **The Algorithm**

1. Determine the number of vertices n.
2. Generate the polygonal knot of the string diagram with equal segment length of $1/n$, and determine vertices coordinates.
3. Starting data: The n vertices of polygon K_0 in $\Re^3$. K_0 is assumed to have total length = 1.
4. For each pair of non-consecutive edges E_i, E_j of K_0, compute the minimum distance ε_{ij} between E_i and E_j. The minimum distance is computed in possibly several steps.
5. Compute the sum of $L(E_i)L(E_j)/MD(E_i, E_j)^2$ to obtain $U(K_0)$. If this is the desired $U(K_0)$ then stop.
6. Compare the various segment distances to compute the minimum of all the segment distances.
7. Perturb the $3n$ vertex coordinates of K, using a uniformly distributed random number generator to choose perturbations; call the new polygon K'
8. Select the polygon K_1, successor to K_0, as follows: Compute $U(K')$; if $U(K') < U(K_0)$ then divide the vertex coordinates of K' by the total length of K' to obtain K'', and let $K_1=K''$; otherwise keep K_0 and try another perturbation.

Starting with any n-segment polygon K_0 representing a knot type {K}, the algorithm generates a sequence of polygons K_0, K_1, ... such that the energies $U(K_i)$ decrease monotonically towards a limit u_∞ [5].

3. IMPLEMENTATION

The system consists of four components: Input interface, State Transformation, Energy Computation and Visualizer. Fig.5 shows the implementation model. The main functions of the input interface are to receive initial string diagram and finger actions from user, and then generate the geometry model (a set of crossing of the inputted string diagram). The Transformation part will transform the model into polygonal knot diagram in the form of coordinates of vertices and the number of vertices. Then, the polygonal knot diagram is sent to the Computation part for calculating the minimum energy and produces a stable diagram. Finally, the Visualizer part will handle the output for displaying the diagram.

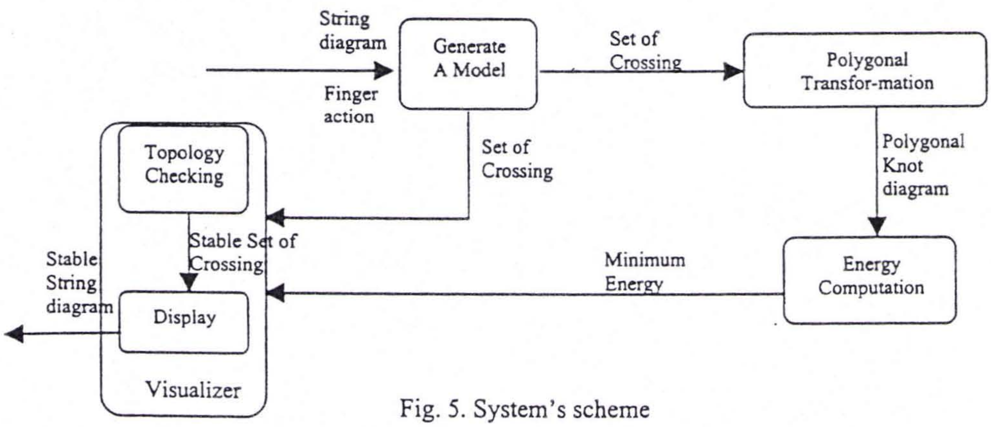

Fig. 5. System's scheme

4. EXPERIMENTS

The system is implemented on the SGI@230 machine. We run the system for generating *Broom* pattern and *Bridge* pattern of Cat's Cradle by using the following parameters: the number of vertices is 1000; number of iterations are 50, 100, 200, 500, 1000. Broom pattern and Diamond pattern represent unsymmetrical and symmetric pattern, respectively.

Fig. 6 shows the final step of Broom pattern construction, the input string diagram is obtained from the previous step of construction (Fig.6 bottom-middle). To obtain the final pattern we release part of strings from the Index and the Ring finger of the right hand, and pull a part of string to the Middle finger of the right hand on the string diagram. The pattern is asymmetric since only the right hand is pulled apart from the left hand. To obtain the final pattern of the Bridge pattern we release string part hung on the Middle fingers of the left and right hand (See Fig.7 left). The pattern is symmetric since both right hand and left hand are pulled apart each other. The minimum energy of the string diagram for number iteration 100, 200, 500 and 1000 are the same, since the pattern is symmetric.

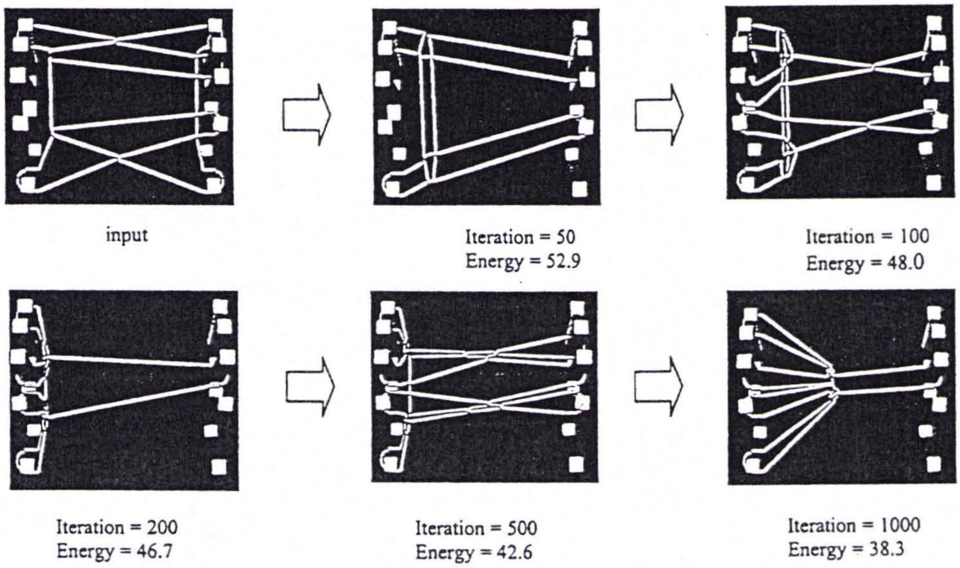

Fig. 6. Broom Pattern

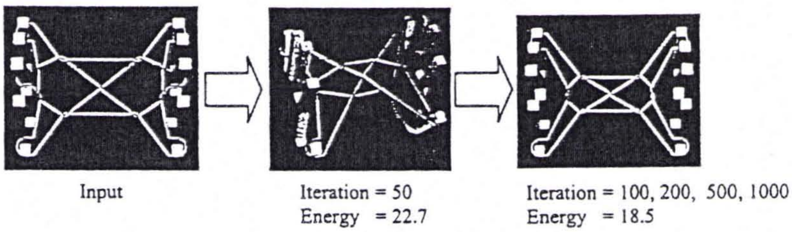

Fig. 7. Bridge Pattern

5. CONCLUSION

We implemented an intelligent string processing method through computing its minimum energy. The method provides a proper shape for asymmetric patterns, which cannot be provided by the system that uses geometrical transformation or the Genetic Algorithm.

ACKNOWLEDGEMENT

The author acknowledges the research grant provided by Universiti Sains Malaysia, Penang that has resulted in this article.

REFERENCES

1. M. Yamada, et al. (1994). A Cat's Cradle String Diagram Display Method Based On A Genetic Algorithm, Forma Magazine, Vol. 9, No. 1, pp.11-28.
2. R. Budiarto, et al. (1997). A 3-D String Figures Display System For Animating Cat's Cradle Game, Proceedings of The Fifth Pacific Graphics Conference, pp. 30-38.
3. `http://www.isfa.org/`
4. V. Katritch, et al. (1996). Geometry and Physics of Knots , Nature 384 , 142.
5. J. Simon, (1993). Energy Functions for Polygonal Knots, Knot Theory and its Ramify, No.3, pp.299-320.