View Selection in Knot Deformation

Juan Lin
Dept. of Comp Eng. & Comp Sci.
Speed School of Engineering
University of Lousiville
Louisville, USA
juan.lin@louisville.edu

Hui Zhang (corresponding author)
Dept. of Comp Eng. & Comp Sci.
Speed School of Engineering
University of Lousiville
Louisville, USA
hui.zhang@louisville.edu

***Abstract*—Extracting good views from a large sequence of visual frames is quite difficult but a very important task across many fields. Fully automatic view selection suffers from high data redundancy and heavy computational cost, thus fails to provide a fast and intuitive visualization. In this paper we address the automatic viewpoint selection problem in the context of 3D knot deformation. After describing viewpoint selection criteria, we detail a brute-force algorithm with a minimal distance alignment method in a way to not only ensure the global best viewpoint but also present a sequence of visually continuous frames. Due to the intensive computation, we implement an efficient extraction method through parallelization. Moreover, we propose a fast and adaptive method to retrieve best viewpoints in real-time. Despite its local searching nature, it is able to generate a set of visually continuous key frames with an interactive rate. All these combine provide insights into 3D knot deformation where the critical changes of the deformation are fully represented.**

***Index Terms*—visualization; best view selection; parallelization;**

I. Introduction

Automatic selection of best views is an important fields in computer vision related research. Since typical visualization of our 3D world structures starts with projection geometric information from 3D to 2D as its first step, a meaningful projection from an appropriate view point can help to present maximal information in the full dimensional space. The criteria for choosing such good views can be very different according the application needs. In this paper, we focus on the best view selection for 3D mathematical curves which can be deforming while keeping their topological structures; such dynamic structures are particularly useful in medical application [1] and mathematical simulations [2]. When examining the 2D projections of 3D curves, a well-chosen 3D presentation can provide not only salient features of the objects but also structural continuity during the entire deformation. However, finding an optimal 2D representation with a view-based approach for the changeable objects can be very challenging. Efficient projection finding and view feature extraction suffer from heavy computational cost and thus cannot easily meet the needs in interactive visualization applications. Moreover, when changing structure is visualized in real time , the traditional methods are unable to appropriately handle the dependencies between each process of arbitrary deformation and thus fail to present a series of continuous frames. In order to address all the challenges, we need to explore a new paradigm to communicate deforming structures in the most efficient and fast way.

This work was supported in part by National Science Foundation grant #1651581 and the 2016 ORAU's Ralph E. Powe Junior Faculty Enhancement grant.

We start with a fairly familiar viewpoint entropy concept and extend the criterion for the evaluation of the animation curves that not only features the maximum information, but also contains the less disturbance. Having established the mechanisms, we proceed to a brute-force method and its parallelization that guarantees a global best view. To further improve the visual experience, we proposed an adaptive algorithm based on local search where successive terms can be presented with an interactive rate. All these combine to fulfill the need where a meaningful and minimal set of representative views are created to visualize a deformation sequence that otherwise requires a huge number of visual frames.

The rest of our paper is organized as follows. Section 2 overviews related work on computational methods to identify the effective viewpoints and 3D model alignment. Section 3 describes the brute-force approach with minimum distance match as well as the parallel version. Section 4 focuses on an adaptive viewpoint selection algorithm to improve the performance. We then finally present our application study and the conclusion.

II. Related Work

Our fundamental techniques are based on a wide variety of prior art, mainly concern with viewpoint selection and 3D model alignment. With advances in 3D model acquisition technologies, the problem of selecting an optimal set of frames from high redundancy and high computation cost of capture to present useful information gets consistent attention. Assisted with the incorporation of 3D graphic, information science and mathematical theory, maximum information from a minimal set of views can be abstracted automatically. Kamada and Kawai [3] minimize the degenerated edges to select the viewpoint, and an parallel projection is applied to determine the viewing direction. A viewpoint entropy is widely used to define the best view of a scene [4]. The main idea is based on entropy theory where a good viewpoint should contain as much geometric information as possible. And an extra adaptive view selection method is designed to accelerate the computation

[5]. It starts with six points and predicts the entropy with middle points recursively, obtains a large speed-up compared with the brute-force method. Colin [6] develops an model to calculate the good viewpoint with an octree. Lee et al. [7] define mesh saliency with a center-surround operation on Gaussian-weighted mean curvatures, and it is able to capture the appealing regions automatically.

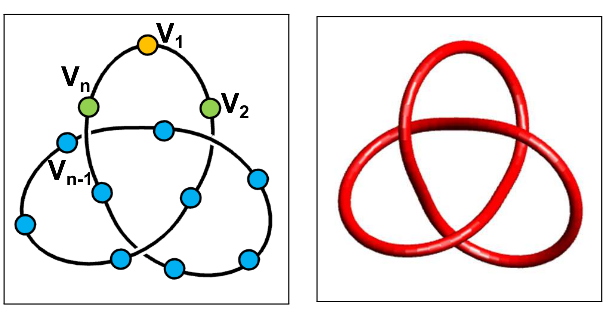
For the model alignment, Alexa et al. [8] present an engaging shape interpolation to find the optimal least-distorting transformation, the implementation can be applied not only on morph and contour blending but also on texture colors. Henry et al. propose a minimum projection area-based alignment method with three principal axes [9], an optimization method based on Particle Swarm Optimization is adopted to decide the minimum projection. Such method can be used on rigid/non-rigid models. The singular-value decomposition (SVD) is often used to calculate the rigid transformation [10] [11], while it fails to address the reflection problem. Cashbaugh and Kitts [12] adopt a linear regression to write the transformation matrix as a series of linear equation, and they sort out the reflection problem through a pseudoinverse matrix. With only paired data sets, a transformation matrix can be generated effectively and accurately.

III. Brute-Force Algorithm

In this section, we introduce our viewpoint criterion, a brute-force algorithm using such criterion to generate a list of viewpoints, a minimal distance match method to minimize the distortion of the intermediate shape, and the parallel framework to accelerate the procedure.

A. Data Representation

An initial diagram of a 3D curve is represented through a group of cylinders with given length L and radius R. The $i-th$ cylinder is connected with a pair of vertices (v_i, v_{i+1}), where $i \in [1, n-1]$. The 3D representation can be obtained by projecting each vertex 3D xyz-space to 2D xy-plane. The curve moves with certain restrictions to guarantee the geometry type will not be changed (see Figure1).



(a) (b)

Fig. 1: An example of curve representation in our work. (a) A smooth 2D topological structure represented using an array of line segments, nodes, and interpolation splines. (b) The 3D render of the curve with color, texture and shading.

B. View feature extraction

Our approach is based on a general viewpoint entropy measure obtained from Information Theory [13]. The main idea is to use a probability distribution to represent the relative area of the projected faces around a sphere viewpoint. A larger viewpoint entpropy contains more useful information, the best view will be the one with maximum viewpoint entropy.

During the deformation, the curve transforms in a unsupervised way. To trace an explicit transformation, it is important to identify the meaningful changes that occur in the process. Here a crossing number is defined, that is, the smallest number of crossing of any projective diagram of a 3D curve. The less crossing number means the less complicated structure. Then we redefine new viewpoint entropy formula, which features the maximum information as well as the minimum complexity :

$$I(K) = \sum_{i=1}^{N} (-\frac{L_i}{L_t} log \frac{L_i}{L_t} + V(i)) \tag{1}$$

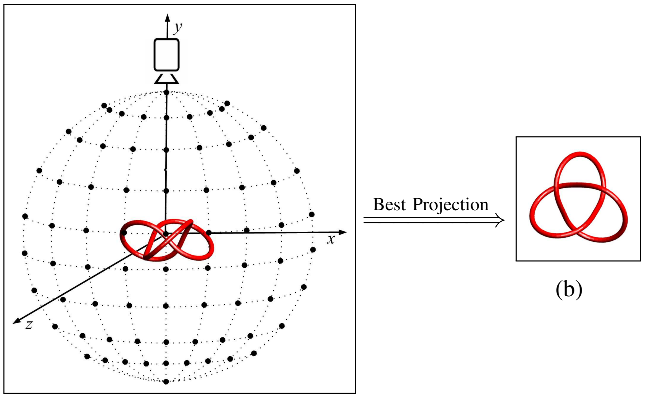
where L_i represents the projected length of curve segment i, L_t is the total length of the projected curve; $V(i)$ is the visibility test function for curve segment i, $V(i)$=-1 if the segment is crossed by another segment, and otherwise $V(i)$=+1. Here the larger length of projected curves will contribute to a larger entropy value, and the number of crossings in the projection contributes to the entropy value as a penalty. In this way, the best diagram is identified as one that contains the maximal length of curves and the minimal number of crossing in the diagram.

With the viewpoint entropy defined above, the deformation requires to be examined with a huge sampling space of projections so as to make sure the best viewpoint can not be omitted. In order to find the global best viewpoint, here a brute-force method is presented. The camera rotates from 1 to 360 degree incrementally around x, y, and z axis respectively(see Figure 2). In each position, the entropy is calculated and compared to the current maximum value. This process is repeated until all of position are examined.

C. The minimal distance alignment

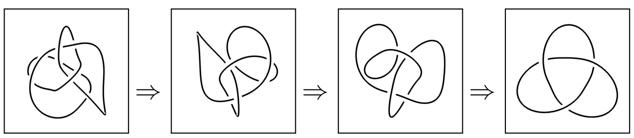
The viewpoint entropy method can help us identify the best view for a dynamic curve. However there is a large probability to exhibit visual discontinuities from one diagram to another. This is because the brute-force method searches for the best diagram with the maximum viewpoint entropy covers a serial of angles around the 3-dimensional space, the resultant diagrams with various rotation direction can share the same viewpoint entropy value. Then the orientation in the original deformation cannot be preserved. It may cause the disjunction between the intermediate shapes, leaves the complete deformation difficult to trace.

Figure 3 shows a discontinuous relaxation for a classical structure in Knot theory names trefoil knot since the brute-force method fails to identify the difference among the diagrams that owns the same entropy.



(a) (b)

Fig. 2: (a) The viewpoints generated by the brute-force method. The viewpoint incrementally changed 1 degree along x, y, and z axis respectively. (b) The best projection presented by current viewpoint.



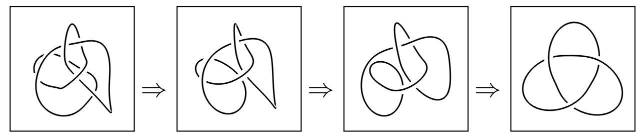
(a) (b) (c) (d)

Fig. 3: A discontinuous relaxation for a trefoil knot. These frames fails to preserve their original orientation. (a) Initial conformation with $crossNum = 9$. (b) A diagram rotates 180 degrees clockwise with $crossNum = 6$. (c) A diagram makes mirror flip with $crossNum = 4$. (d) Final conformation with $crossNum = 3$.

To maintain the original direction, we need to establish a correspondence between the original structure and the following list of candidate projection. One way is to adopt 3D model alignment method, while it is time-consuming and error-prone. Here a simple least-squares method (see Eq. 2) is proposed. Compared to other projections, the most similar structure suffers least change from the original structure, the minimum distance value is obtained accordingly.

$$d(\mathbf{K}_p, \mathbf{K}_q) = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{v}_{pi} - \mathbf{v}_{qi}\| \tag{2}$$

During the force-brute method, we examine the visual frame with the minimal number of crossing **and** the least square distance from the previous frame at the same time. This new searching criterion ensure the extract frames preserve the best visual continuity in the extracted sequence. Algorithm 1 describes the process. Initially, the object is placed in the middle of scene. It rotates along x-axis, y-axis and $z-axis$ incrementally. Each rotation generates a new projection value. If this new projection value is larger than the old one, we continue to compare the minimum distance, if the value is less than the current one, it means a more closer projection is found. Such process keeps comparing until a complete 360*360*360 angle combination is done. And the resultant diagram is the one that obtains from the best viewpoint position as well as minimum distance from the last frame. Figure 4 shows a successive relaxation for trefoil knot. The diagram in Figure 3 (b) and Figure 4 (b) shares the same entropy value, while Figure 3 (b) rotates 180 degrees clockwise from Figure 4 (b), arises a visual distortion in the deformation. With the correction of minimal distance, Figure 4 (b) is selected as resultant diagram. Figure 3 (c) and Figure 4 (c) show the similar situation. Overall, compared to Figure 3, the minimal distance strategy helps to identify the most similar diagrams during the deformation, gives an appropriate illustration to display the dependencies between the diagrams.



(a) (b) (c) (d)

Fig. 4: A successive relaxation for a trefoil knot.

Algorithm 1: Find the best viewpoint with the brute-force method

Input: Initial Layout
Output: A projection with the best viewpoint
Initialize the current projection $\mathbf{P_{max}}$ with the initial viewpoint, best entropy v_{max}, minimum distance d_{min} ;
for all viewpoints **do**
  Calculate current entropy v;
  if $v' >= v_{max}$ **then**
    Calculate current distance d ;
    if $d < d_{min}$ **then**
      Update $\mathbf{P}_{max}$;
      Update v_{max} with v ;
      Update d_{min} with d ;
    end
  end
end

D. Accelerate Brute-Force Algorithm

A brute-force method ensures the best viewpoint would not be missed and guarantee a global best solution. While this algorithm suffers heavy computational cost, for each moment, it requires traveling an angle combination of 360*360*360 to find the best result. Since the entropy calculation for each viewpoint has not data dependency and we only care about those with the same maximum value, it is easy to process the search in a parallel way. We then slightly change our algorithm into a parallel framework, so that an effective searching method is available. Algorithm 2 describes the process. We generate a list of best viewpoints in a parallel way at the beginning, deploy as many computing units as available. After that, the distance between the last frame and the diagram projected by current viewpoint are compared, the resultant viewpoint is the one with the highest entropy and lowest distance value. In Algorithm 2, the minimal distance comparison is excluded. Within a parallel pattern, the frequent data change would be devastating. And the number of projections sharing the same largest viewpoint value is far less than the amount number of viewpoints. So it is appropriate to remove this part out of the parallel execution framework. Figure 5 shows the illustration for parallel searching method. The multiple viewpoints are examined at the same time, which largely increases the searching efficiency.

Algorithm 2: Find the best viewpoint in Parallel

Input: Initial Layout
Output: Best projection presentation
Initialize a empty projection list LP, best viewpoint V_{max}, minimum distance d_{min} ;
Calculate all entropy v_i in each computing unit i ;
Barrier synchronization of all processes ;
Order the projection value and generate the LP with the maximum values ;
for each projection $\mathbf{P}$ in LP **do**
    if $d^{'} < d_{min}$ **then**
        Update $\mathbf{P_{max}}$ with $\mathbf{P}$;
        Update d_{min} with d ;
    end
end

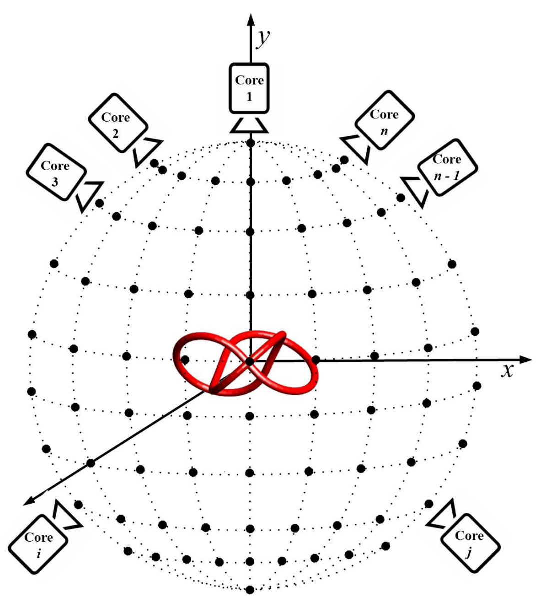


Fig. 5: The viewpoints generated by the parallel brute-force method. The viewpoint incrementally changed 1 degree along x, y, and z axis respectively and examined in a parallel way.

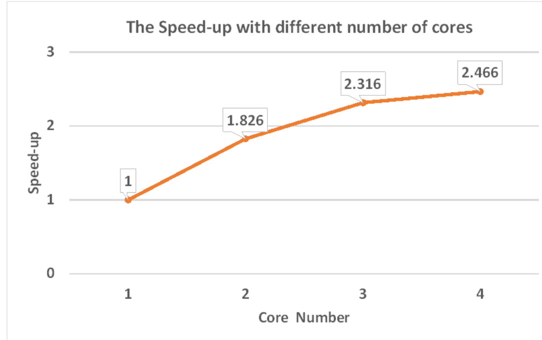


Fig. 6: Average Speed-up with different number of cores.

E. Experiments

Our implementation uses R as the major analysis tool. The RGL package is included to help the geometry's depiction, it uses OpenGL as the rendering backend and provides a real-time 3D engine. The algorithms run on a Lenovo PC desktop with Intel Core i7 CPU 1.8GHz. To accelerate the iteration in our simulation, we adopt the library of doParallel in R. The doParallel package acts as an interface between foreach and the parallel package of R 2.14.0 and later [14]. It provides a nice, efficient parallel programming platform for multiprocessor/multicore computers. As a blend of the snow and multicore packages, the doParallel package works well on Unix-like system, Windows and even their combination. The most advantage of this package is it is very easy to install and use.

We characterize the performance in term of Speed-up= T_s/T_p, where T_s is the execution time of the serial algorithm, and T_p is the execution time of parallel execution. To verify the performance of parallel algorithm, we test a set of curves with different number of nodes ranges from 46 to 500. And the average Speed-ups with 1 to 4 cores are shown in Figure 6 where the Speed-up is growing gradually long with the increasing number cores. And the parallel algorithm obtains reasonable speedup of 2.47x when executed on the multi-core system with 4 cores.

Figure 7 shows the comparsion with different number of nodes running wit 4 cores. With the increasing number of nodes, and the execution time for each structure is growing dramatically, while the parallel algorithm takes far less time and the Speed-up keeps stable around 2.3x.

IV. Adaptive View Selection Method

In this section, we will focus on a more efficient approach to present the best projection from long deformations. Although Our basic idea presented in Section III ensures a global best solution, it is compute-intensive, and thus very time-consuming and nearly impossible to be helpful for our real-time presentation. Here we devote to present a more flexible local search algorithm boosts the efficiency and accuracy simultaneously.

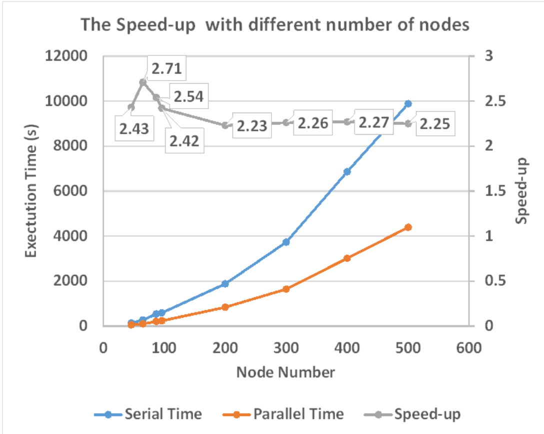


Fig. 7: The performance for curves with different number of nodes.

A. Overall Scenario

Our algorithm starts with a coarse-grained space to calculate the viewpoint entropy value at each sampling vertex, and a fine-grained local search follows to find the best viewpoint recursively until a final optimal viewpoint is reached. The main steps include:

1) Generate an initial set of viewpoints;
2) Evaluate each viewpoint and sort the viewpoint set in the decreasing order, generate an initial triangular mesh with the top three best position;
3) Evaluate the middle points of each edge, generate a new triangle;
4) Apply last step (3) iteratively until the terminal criteria satisfies.

B. Initial Viewpoints Generation

The performance bottleneck in our original method lies in the function to identify critical changes at each time point, which is very compute-intensive because it calculates the curve's crossing number from all possible viewpoints in 3-dimensional space. It is possible to make the view selection much faster. For example, for the two nearby viewpoint, they usually present the similar projection. It is not necessary to examine all of them. Instead, if we start from the most promising area, pass through the similar position, our method can be more efficient compared to the original brute-force one.

There are several viewpoint sampling methods include longitude and latitude sampling method, random sampling method, and pseudo uniform method, etc. In order to obtain a reasonable sampling distance, cover an effective searching area as well as make sure not to miss the essential viewpoints, We start from sampling a set of points on the surface of the sphere based on the subdivision of a regular icosahedron (See Figure 8(a)). The first level subdivision produce 12 sampling points around the sphere, which covers elementary viewpoints. By applying the loop subdivision rule, we are able to narrow down the searching space gradually.

C. Initial Viewpoints Evaluation

After the generation of the initial candidate viewpoints, it is essential to evaluate these viewpoints and find a "good" starting point. Since our proposed algorithm does not cover a complete searching space, if the initial searching point fails to provide a correct searching direction, there is not chance we can correct later. So we evaluate and compare all of viewpoints with Equ 1, sort these entropy in decreasing order. If there are same entropy value, we go on to compare the minimum distance, and generate an initial triangular mesh with the top three best position with largest entropy value and minimum distance with last diagram (See Figure 8 (b), Figure 8 (f) shows current projection with this viewpoint).

D. Adaptive Viewpoints Selection

Next we use the three mid-points on this viewing triangle's edges to generate three new viewpoints for the next round of viewpoint search (see Figure 8(c)). By calculating the viewpoint entropy values for the three views, we now pick the new best view (see the mid-point in yellow in Figure 8(d), and Figure 8 (g) shows current projection with this viewpoint), and this lead to us identifying the next six adjacent views (see the 6 in green in Figure 8(e)). Our view search procedure is recursively performed, leading to a finer searching area in each iteration, until the terminating condition satisfies (see Figure 8 (h)).

The description of this procedure is sketched in the following algorithm (See Algorithm 3). After the stage of initial configuration, the process is executed iteratively until none of the projection value is higher or sampling points are at a very close distance over a threshold. With the adaptive viewpoint method defined above, the best diagram can be identified in an more efficient way.

E. 3D alignment

The adaptive view selection obtains an approximate optimal projection for current diagrams. While the curves relax themselves in 3D space, when the critical change occurs, the projection may vary greatly from the last diagram, thus leaves structural discontinuity. In order to provide the desired visual continuity and keep the computational efficiency, we apply a rigid transformation to "re-align" between the diagrams in the final visualization sequence.

We applied a homogenous transform [12] method to align two resultant diagrams. In general, this method calculate an 4*4 augmented transformation matrix between two frames with a linear regression. Through rewriting the transformation matrix as a series of linear equations and minimizing the square of the residuals of them, the values in each row of the transformation matrix are obtained. And the rotation sequences are recovered in a 3-2-1 way. Thats it , the object rotates in a order of z-axis, y-axis and x-axis. The Euler rotation angles can be obtained with these values.

Given two diagrams $\mathbf{P}_A$ and $\mathbf{P}_B$, a homogenous transformation matrix is defined in Eq. 3. It contains the rotation R and translation T at the same time. The second row is a

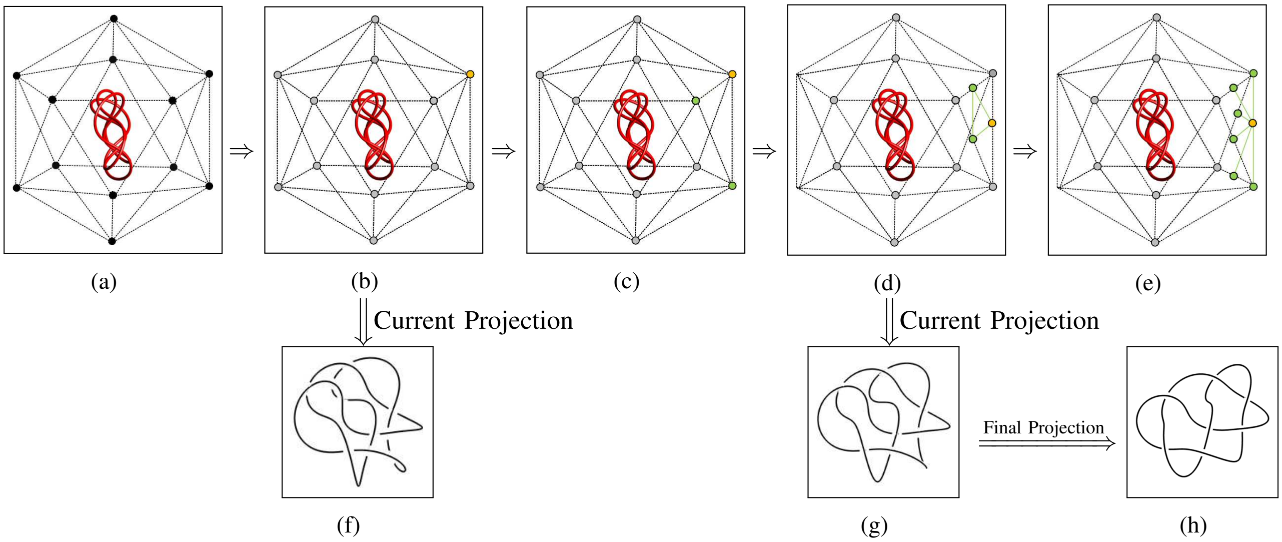


Fig. 8: The adaptive viewpoint selection procedure. (a) Initial 12 sampling viewpoints in the icosahedron. (b) The current best viewpoint. (c) The first viewing triangle around the current best viewpoint. (d) The first candidate viewpoints generate in the middle of edges. (e) The recursive generation of viewpoints. (f) The current projection presented by the viewpoint in (b). (g) The current projection presented by the viewpoint in (d). (h) The final best projection.

Algorithm 3: Best diagram identification with adaptive view selection

Input: Initial Layout
Output: Best Diagram with Maximal View Entropy Value
Initialize a set of viewpoints placed on icosahedron vertices around the object;
while un-visited viewpoints exist **do**
    find the best viewpoint with maximum entropy value (maximal projective length and minimal number of crossing);
    while the entropy value of current best view is better **do**
        Find all adjacent points, calculate all entropy values on the mid-point of all edges;
        Find the next viewpoint with the maximum entropy value and minimal number of crossing;
        Update new entropy value and un-visited viewpoints;
    end
end

orthonormal perspective and homogeneous scaling factor. In our case, it sets as 1 since we only focus on the deformation.

$$\begin{bmatrix} P_B \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} R & T \\ \mathbf{0} & \mathbf{1} \end{bmatrix} * \begin{bmatrix} P_A \\ \mathbf{1} \end{bmatrix} \tag{3}$$

where $R = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix}$, $T = \left[t_x, t_y, t_z\right]^t$.

With a linear regression to minimize the square of the residual, a volume matrix A is obtained (Eq.4) to generate each column of the transformation matrix (Eq. 5 - Eq.7).

$$A = \begin{bmatrix} \sum(x_{Ai}^2) & \sum(x_{Ai}y_{Ai}) & \sum(x_{Ai}z_{Ai}) & \sum(x_{Ai}) \\ \sum(x_{Ai}y_{Ai}) & \sum(y_{Ai}^2) & \sum(y_{Ai}z_{Ai}) & \sum(y_{Ai}) \\ \sum(x_{Ai}z_{Ai}) & \sum(y_{Ai}z_{Ai}) & \sum(z_{Ai}^2) & \sum(z_{Ai}) \\ \sum(x_{Ai}) & \sum(y_{Ai}) & \sum(z_{Ai}) & n \end{bmatrix} \tag{4}$$

where $i \in [1, n]$.

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \\ t_x \end{bmatrix} = [A]^{-1} \begin{bmatrix} \sum(x_{Bi}x_{Ai}) \\ \sum(x_{Bi}y_{Ai}) \\ \sum(x_{Bi}z_{Ai}) \\ \sum(x_{Bi}) \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \\ t_y \end{bmatrix} = [A]^{-1} \begin{bmatrix} \sum(y_{Bi}x_{Ai}) \\ \sum(y_{Bi}y_{Ai}) \\ \sum(y_{Bi}z_{Ai}) \\ \sum(y_{Bi}) \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \\ t_z \end{bmatrix} = [A]^{-1} \begin{bmatrix} \sum(z_{Bi}x_{Ai}) \\ \sum(z_{Bi}y_{Ai}) \\ \sum(z_{Bi}z_{Ai}) \\ \sum(z_{Bi}) \end{bmatrix} \tag{7}$$

Then a rotation matrix can be abstracted where the values are defined by the rotation sequences (Eq. 8).

$$R_A^B = \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{bmatrix} \tag{8}$$

The resultant rotation angles are calculated separately with this matrix in the form of the Euler angles.

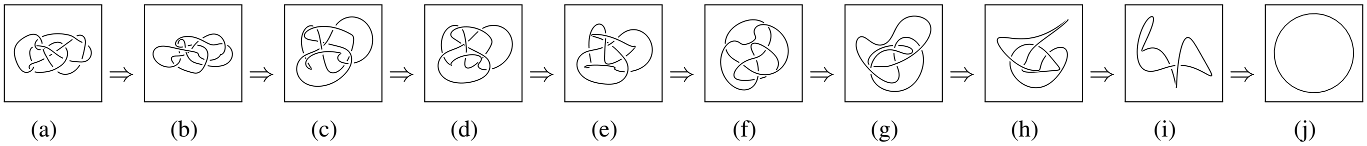


Fig. 9: The relaxation for complex unKnot with parallel brute-force method. (a) Initial conformation with $crossNum = 15$. (b)$crossNum = 13$. (c) $crossNum = 11$. (d) $crossNum = 10$. (e) $crossNum = 9$. (f) $crossNum = 8$. (g) $crossNum = 6$. (h) $crossNum = 4$. (i) $crossNum = 2$. (j) Final conformation with $crossNum = 0$.

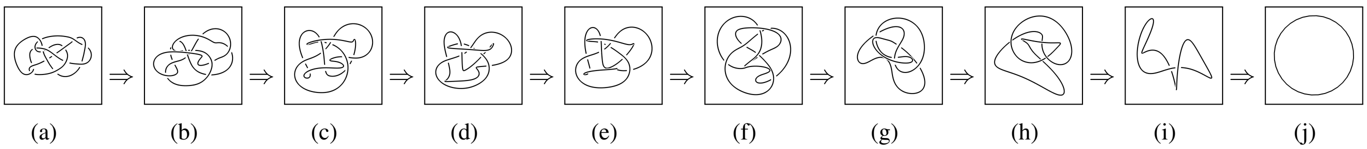


Fig. 10: The relaxation for complex unknot with adaptive best projection presentation.

TABLE I: Performance comparison between parallel brute-force and the adaptive view selection method

Node Number	Parallel Brute-Force Method Execute Time (s)	Adaptive Method Execute Time(s)	Speed-up
46	59.5	0.87	68.39
65	104.14	1.53	68.07
87	213.95	2.18	98.14
96	246.55	2.66	92.69
200	842.22	7.92	106.34
300	1650.38	15.85	104.12
400	3019.31	27.11	111.37
500	4392.2	40.75	107.78

Beside, each diagram is generated from a 3D object to a 2D projection, for the data frames, the value in all z-axis are zero, therefore A is not invertible in our case. We include Moore-Penrose method [15] to produce a pseudoinverse matrix. And an approximate inverse is generated through the Singular Value Decomposition (SVD). In this way, we acquire a valid transformation matrix to further produce a reasonable rotation angles. More detailed derivation and proof can be found in [12].

F. Performance and Cast Study

In order to verify the performance improvement, we test the same set of curves and make the comparison between brute-force and the adaptive view selection method. Table I shows the results. Despite the parallelism exploration, the searching space for the adaptive method is far less than the brute-force one. It greatly reduces the time complexity of the algorithm and helps to gain a large speedup. Beside, as the number of nodes grows, the Speed-up is able to increase accordingly.

To further illustrate a complete clue with our algorithms, we introduce a complex unknot in Knot Theory and present a complete relaxation with an the parallel brute-force algorithm . The resultant frames are presented in Figure 9 and Figure 10 respectively. The similar continuous diagrams shared by two figures indicates that, although the adaptive method only examines a small number of viewpoints, we can still obtain the same effect than with the brute-force method, moreover the time of computation reduces dramatically at the same time.

V. Conclusions

In this paper, aim to attack different level purpose, we have presented two different methods for the automatic selection of best views specific for 3D animation curves, a brute-force method makes sure a complete exploration. Embedding in a parallel computation framework, the performance boosts with the demanding computation resource. An adaptive viewpoint selection method is able to provide both scalability and high efficiency. Such work can be applied to fully simulate the important transformation with Knot Theory and other relative data-intensive research domains.

In the future, we plan to proceed to an extension to viewpoint selection mechanism for mathematical objects embedded in higher dimension space. This is particularly challenging since higher dimensional objects must be projected to three dimensions or less for understanding and interaction, thus how to identify the full features of the whole object and provide structural continuity for higher-dimensional projection remains a challenge to solve.

References

[1] J. Brown, J.-C. Latombe, and K. Montgomery, “Real-time knot-tying simulation,” The Visual Computer, vol. 20, no. 2-3, pp. 165–179, 2004.

[2] H. Zhang and A. Hanson, “Shadow-driven 4d haptic visualization,” IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 6, pp. 1688–1695, 2007.

[3] T. Kamada and S. Kawai, “A simple method for computing general position in displaying three-dimensional objects,” Computer Vision, Graphics, and Image Processing, vol. 41, no. 1, pp. 43–56, 1988.

[4] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Viewpoint selection using viewpoint entropy.,” in VMV, vol. 1, pp. 273–280, 2001.

[5] P.-P. Vázquez and M. Sbert, “Fast adaptive selection of best views,” in International Conference on Computational Science and Its Applications, pp. 295–305, Springer, 2003.

[6] C. Colin, “Automatic computation of a scene’s good views,” in Proc. MICAD, vol. 90, 1990.

[7] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” ACM transactions on graphics (TOG), vol. 24, no. 3, pp. 659–666, 2005.

[8] M. Alexa, D. Cohen-Or, and D. Levin, “As-rigid-as-possible shape interpolation,” in Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 157–164, ACM Press/Addison-Wesley Publishing Co., 2000.

[9] H. Johan, B. Li, Y. Wei, et al., "3d model alignment based on minimum projection area," The Visual Computer, vol. 27, no. 6-8, p. 565, 2011.
[10] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," IEEE Transactions on Pattern Analysis & Machine Intelligence, no. 4, pp. 376–380, 1991.
[11] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," IEEE Transactions on pattern analysis and machine intelligence, no. 5, pp. 698–700, 1987.
[12] J. Cashbaugh and C. Kitts, "Automatic calculation of a transformation matrix between two frames," IEEE Access, vol. 6, pp. 9614–9622, 2018.
[13] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Automatic view selection using viewpoint entropy and its application to image-based modelling," in Computer Graphics Forum, vol. 22, pp. 689–700, Wiley Online Library, 2003.
[14] S. Weston and R. Calaway, "Getting started with doparallel and foreach," 2019.
[15] R. Penrose, "A generalized inverse for matrices," in Mathematical proceedings of the Cambridge philosophical society, vol. 51, pp. 406–413, Cambridge University Press, 1955.