https://doi.org/10.2352/ISSN.2470-1173.2019.1.VDA-683

Visualizing Mathematical Knot Equivalence

Juan Lin, Hui Zhang (corresponding author)
Department of Computer Engineering and Computer Science, University of Louisville, Louisville, KY, USA, 40292
juan.lin@louisville.edu
hui.zhang@louisville.edu

Abstract

We present a computer interface to visualize and interact with mathematical knots, i.e., the embeddings of closed circles in 3-dimensional Euclidean space. Mathematical knots are slightly different than everyday knots in that they are infinitely stretchy and flexible when being deformed into their topological equivalence. In this work, we design a visualization interface to depict mathematical knots as closed node-link diagrams with energies charged at each node, so that highly-tangled knots can evolve by themselves from high-energy states to minimal (or lower) energy states. With a family of interactive methods and supplementary user interface elements, our tool allows one to sketch, edit, and experiment with mathematical knots, and observe their topological evolution towards optimal embeddings. In addition, our interface can extract from the entire knot evolution those key moments where successive terms in the sequence differ by critical change; this provides a clear and intuitive way to understand and trace mathematical evolution with a minimal number of visual frames. Finally our interface is adapted and extended to support the depiction of mathematical links and braids, whose mathematical concepts and interactions are just similar to our intuition about knots. All these combine to show a mathematically rich interface to help us explore and understand a family of fundamental geometric and topological problems.

Introduction

In topology, knot theory [1] is a field that concerns the equivalence between classes of mathematical knots. While knot theory is a concept in algebraic topology, it has found applications to problems in physics [2] [3], chemistry [4], bioinformatics [5] and 3D printing [6] etc. A major task of knot theory is to find ways of telling whether two knot diagrams are diagrams of equivalent knots. In particular we will want to know if a given knot diagram represents a knot equivalent to an unknot, that is, a knot representable by a knot diagram without crossings. The question being asked here is not a geometric problem in the usual way we think about geometry, because the mathematical knots are slightly different than the everyday knots. The specific shapes of the knots don't matter to our question — If we nudge the knot images just a bit, it does not make any difference to the question, even though the geometric shape of the knot is changed. Knot equivalence is a topological question where we don't care about knots' length or angles. The objects of interest in this paper are mathematical knots that are considered infinitely stretchy and flexible, even if they also appear to be physical so they won't cut into themselves during a mathematically valid evolution.

A standard mathematical approach to understanding knot equivalence is in general quite challenging: one starts with a 3D knot structure sketched as a 2-dimensional (projective) image with cutaways at each crossing to indicate its 3D structure, and then draws a sequence of mathematically valid deformations applied to the knot image to generate its isotopy towards the final goal. One way of producing such deformations to show that a knot diagram can be transformed into another is to use a sequence of atomic moves, known as Reidemeister moves [7]. While the three Reidemeister moves are simple and intuitive, producing the entire procedure using the combinations of the Reidemeister moves in the right order is fairly difficult. It may take up to $236n^{11}$ moves for a knot diagram with n crossings to be distorted towards its isotopy [7]. In this work, we are motivated to model mathematical knots using a hybrid force model — on one hand they are infinitely stretchy string pieces with energies charged at the nodes, so that highly-tangled knots can evolve by themselves from high-energy states to minimal (or lower) energy states; on the other hand, they can be physical and real so one can nudge part of the knots to change their shapes, to tie or untie the knots as if they were the real ropes we are familiar with in our everyday life.

Related Work

Mathematical knot has been fundamental interests to visualization community. The advent of high-performance interactive computer graphics and visualization has opened a new ear for creating a tangible experience with these mathematical objects and their topological phenomena. For example, Andrew et al. present interactive methods to visualize geometric objects [8]. Wijk and Cohens' SeifertView focus on generating knots and links through Seifert surfaces [9]. In KnotApp [10], mathematical knots are written as fourier series and a method is presented to construct the contact surface of a knot shape using 3D printing. Other efforts include, e.g., methods to examine particular geometric features in high-dimensional space via their reduced-dimensional analogs (see e.g., [11] [8]), that allows interactive experiences with high-dimensional entities in three-dimensonal space accessible to us. Visualizing knot equivalence involves not only efforts to display static knot structures, but also to illustrate changing structures that require deformations or interactions, such as the moves to take one knot into an equivalent one. Several successful efforts already exist to visualize the knot dynamics. For example, Scharein's Knotplot is a widely used tool to draw and interact with mathematical knots [12]. Wu's work supplement Ming's knot program [13] with a local minimal MD energy calculation to associate knots with energies. Brown's work has been focused on 3D ropes' physical simulation, with a visualization tool developed to allow real-time manipulation of virtual ropes [14]. KnotSketch [15] adopts an enhanced version of Gauss code to facilitate the manipulation of virtual knots. Zhang et al. develop a family of

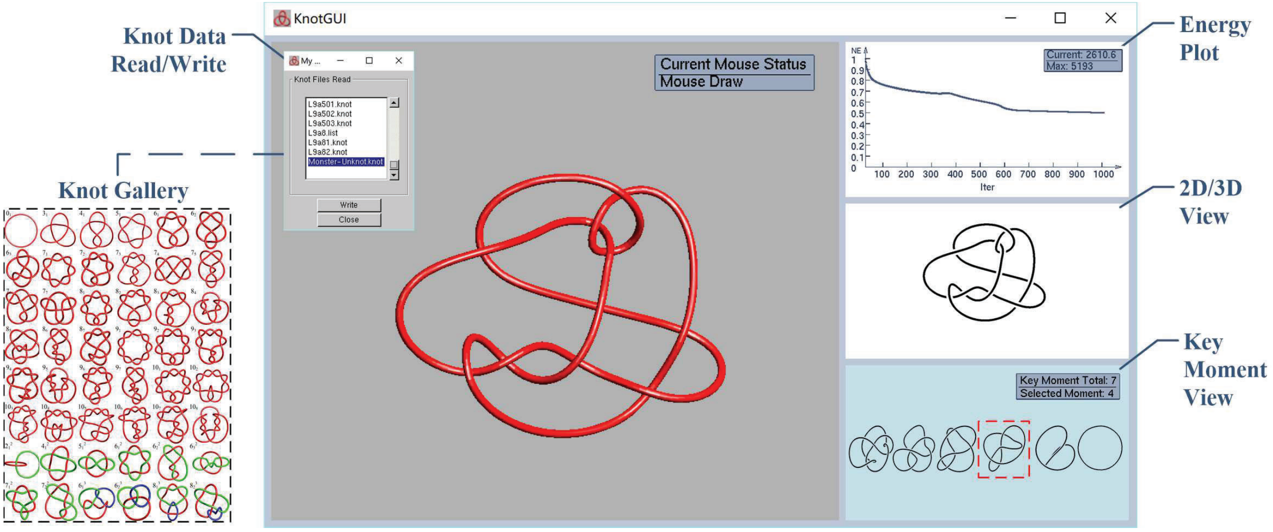


Figure 1: Visualizing Mathematical Knot Equivalence: our tool allows users to produce and trace equivalent knot diagrams through force-driven mathematical relaxation and physically-based interactions.

methods to visualize knots in 3D and 4D space [16] [17], and a visualization interface to interact with mathematical knots with the Reidemeister moves [18]. While most prior visualization efforts have been studying knots with the approved mathematical moves (i.e., the Reidemesiter moves), they typically provide expert's interfaces to display and interact with knots (and sometimes their evolved shapes) on a 2D screen. The work presented in this paper aims at an hybrid approach to bringing combined power of force-directed topological refinement methods and physically-based interactions to facilitate automatic optimization and understandable interaction with mathematical knots.

The rest of our paper is organized as follows. We will start with an overview of the interactive interface that allows one to sketch, edit, and polish knot diagrams. We then proceed to describing our force models and topological relaxation methods to refine knots' embeddings. Having established the main visualization paradigms, we proceed to a method to extract the critical frames as key moments from the long sequence of knot evolution. We next extend the our interface to the depiction of the mathematical links and braids and their equivalence. Finally we will present a preliminary user experience study, with a conclusion drawn in the end.

Overall Scenario

Figure 1 shows the typical screen image of our visualization interface. From the user's perspective, there are four major display components in the interface: a paper-like panel in the central area to sketch knots and explore their deformations, two display windows where the mathematical knot' energy over time is plotted and the knot's 3D rendering is provided, and a playback window to display the key moments captured from the entire knot deformation process.

The central panel is the main tool that allows users to visually explore knot/link structures and suggest deformations with automated relaxation and manual editing. More importantly, when users visually explore or make changes to the structures, all the windows are coordinated to allow users to toggle between synchronized 2D diagram, 3D rendering, and the associated energy plotting, which we will discuss more later. The playback window is a key frame player that allows users to access and trace the key moments during the whole mathematical deformation process. User operations in our interface are designed and supported in the hierarchy as follows:

- **Nodes and segments** — these are the basic geometric structures of the mathematical entities in this work, and our interface allows one to manually edit their properties, e.g., their coordinate values.
- **Knots, links, and braids** — these are the mathematical objects of interest in our work they can be visually explored with user-performed translation/rotation/scale, or topologically relaxed in an unsupervised way, and users can also intervene the relaxation by nudging parts of the structure with physically-based interactions, by treating part of the knots as 3D physical ropes.
- **Mathematical entities and phenomena** — all static and changing structures presented in our interface are visualized and traced, and can be exported as images and data files containng the geometric information.

Modeling Mathematical Knots with Forces

In this section, we focus on a family of methods used to implement the algorithms, interaction procedures, and user interfaces in our knot interface. Our fundamental techniques are based on a wide variety of prior art, including the 2D Reidemeister move interface for knot manipulation [18], sketching based knot interfaces described in [12, 16], and general graph layout algorithms presented in [19, 20].

Knot Representation and Creation

Mathematical curves are represented as node-link diagrams in our work. An initial diagram of a 3D cruve can be obtained

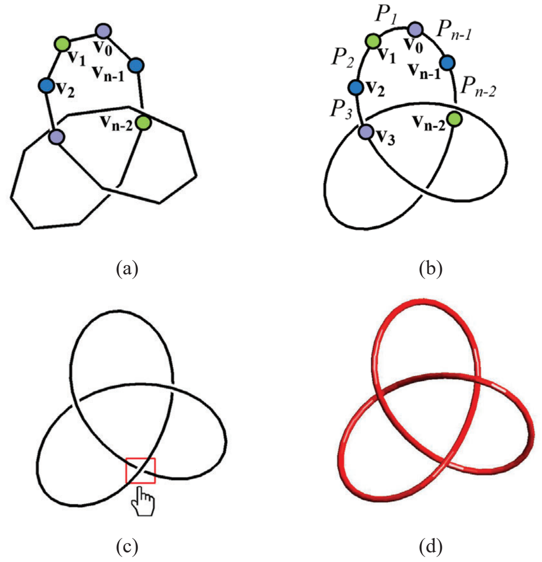


Figure 2: Knot representation and creation in our work. (a) Initial topological structure represented using an array of line segments and nodes. (b) An improved and smoother structure with interpolation splines. (c) Redefine the topological structure by toggling crossing orders. (d) The 3D rendering of the mathematical knot with color, texture, and shading.

by projecting each vertex from $\mathbb{R}^3$ ($xyz-$space) to $\mathbb{R}^2$ ($xy-$plane), but various parts of the diagram appear to touch each other due to the projective collapse of the z dimension. Let $\mathbf{K} = (\mathbf{V}, \mathbf{E})$ represent this initial diagram of a given smooth curve in $\mathbb{R}^3$, where $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2 ..., \mathbf{v}_n\}$ is the finite set of *vertices* of the polygon and $\mathbf{E}$ is the set of *edges* $\{\mathbf{e}_1, \mathbf{e}_2 ..., \mathbf{e}_n\}$.

Our method allows a user to mouse-click a series of "control points" for an initial planar diagram, and construct the 3D curve's topology by assigning each vertex a ternary "eye-coordinate" or, depth z: $z = 1$ for vertices on strands crossing *over* another section, $z = -1$ for vertices on strands crossing *under* another, and $z = 0$ for vertices on un-interrupted strands. The set of control points define the basic topology of a curve embedded in 3-dimensional space. The visual appearance can be significantly improved by computing a Cubic Spline [21] between each of the nodes along the knotted curve.

Consider Figure 2 for an illustration of our knot representation and creation. Figure 2(a) gives a piece of curve represented by an array of line segments connecting each pair of the control points defined by a user. While the 3D embedding in Figure 2 is sufficient to define the curve's topological structure, we can use an interpolation spline to provide an improved visual image in Figure 2(b). The sketched knot diagram in Figure 2(a) and (b) is an unknotted curve. Our interface allows users to toggle the crossings with mouse clicks on the crossing area to redefine the curve's topology. For example, the double click illustrated in Figure 2(c) brings the original unknotted curve into a true (and the simplest) mathematical trefoil knot.

In Figure 2(a)-(c), the mathematical curve is presented in a pen-and-ink style to create the classical "knot-crossing diagrams" by attaching a thickened curve segment in background color behind each of the curve segments in foreground color [16]. In this way, the two-dimensional projected curve is broken each time when it is occluded by another piece of the whole curve that is nearer to the 3D projection point. In Figure 2(d) we show an alternative 3D rendering view by attaching smooth and colored cylinders to replace the 2D line segments [22].

The Force Models

Our basic force model implements an infinitely stretchy string pieces with energies charged at each control point. Two knots are equivalent if one can be transformed into another by *stretching* or *moving* it around without tearing it or having it intersect itself. We are thus motivated to implement a hybrid force model that allows the mathematical curves to be stretchy and can relax by themselves in most cases. Meanwhile, the string pieces can also be moved around without tearing or self-intersection just as a 3D rope that we are familiar with in our everyday life.

Next we will discuss the force models and collision avoidance mechnisms we implement to stretch and move mathematical curves around, and observe their automatic topological evolution in our knot diagram tool.

Mathematical Relaxation. To simulate the dynamics of mathematical knots in $\mathbb{R}^3$, we now replace the knots' control points with electrostatically charged masses and replace each segment with a stretchy link to form a mechanical system [12]. The masses are placed in the initial layout. Then links between masses are infinitely strechy in that each mass is attracted by its two immediate neighbors and repelled by all other non-adjacent ones. Thus the positions of the masses and the length of the links are incrementally updated with the two forces, until a minimum energy state is achieved. The two types of forces being implemented for mathematical relaxation are defined as follows:

- Attractive mechanical force $\mathbf{F_a}$ applied between adjacent masses on the same component to attract adjacent masses towards each other. The mechanical force is a generalization of Hooke's law, allowing for an arbitrary power of the distance between masses.

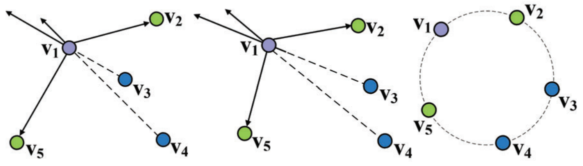
$$\mathbf{F_a}(\mathbf{i}) = H_a||\mathbf{v_{i+1}} - \mathbf{v_i}||^{\beta}(\widehat{\mathbf{v_{i+1}} - \mathbf{v_i}}) + H_a||\mathbf{v_{i-1}} - \mathbf{v_i}||^{\beta}(\widehat{\mathbf{v_{i-1}} - \mathbf{v_i}}) \quad (1)$$

where H_a is a constant.

- Repulsive electrical force $\mathbf{F_r}$ applied between all not-adjacent pairs of masses, following the generalized electrostatic model.

$$\mathbf{F_r}(\mathbf{i}) = \sum_{||i-j||>1} H_r||\mathbf{v_i} - \mathbf{v_j}||^{a}(\widehat{\mathbf{v_i} - \mathbf{v_j}}) \quad (2)$$

where the electric force also allows for a general power of the distance r with the purpose to repel all non-adjacent link components, H_r is a constant. In our studies [12], $\alpha = -6, \beta = 2$.



(a) (b) (c)

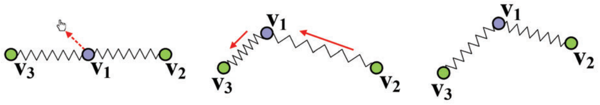
Figure 3: The two forces used for mathematical relaxation: attractive forces applied between adjacent beads and repulsive forces applied between all other pairs will allow the curve to deform and simplify its embedding automatically.

The force model for mathematical relaxation is illustrated in Figure 3. The embedding in this example has only 5 masses. With our defined force model, each mass is attracted by its two neighboring masses, and repelled by two non-adjacent masses. For example, v_1 is attracted by its two neighboring masses, i.e., v_2 and v_5, and repelled by v_3 and v_4. To update the mass location, the aggregated force $\mathbf{F_c} = \mathbf{F_a} + \mathbf{F_r}$ on each mass is calculated to determine the new position of the mass if no topological constraints were active. We use a fairly simple update mechanism — the curve has an invisible intial radius R, and if the distance to update the mass is greater than $0.2 * R$, we clamp the distance moved at $0.2 * R$ in the same direction defined by the aggregated force $\mathbf{F_c}$. If there was topological constraints active during the mass location update, we will simply only update the mass in the same direction but with less magnitude to ensure all components have a distance no less than $2 * R$ from each other. We will discuss this in more detail later.

Physically-based Interaction. Often times we also want to treat part of the mathematical knot as a piece of 3D rope to move it around, or to nudge the knot to change its shape. We use the mass-sprinng model to simulate such physically-based manipulation scenarios. The basic idea is to attach a virtual spring between two adjacent masses $(\mathbf{v_i}, \mathbf{v_j})$. Each spring has its resting lengths defined in the initial layout, and will produce a restoring force when compressed or stretched during the knot manipulation:

$$\mathbf{F_{spring}}(\mathbf{i}, \mathbf{j}) = -K \cdot \mathbf{v_{ij}} \tag{3}$$

where $\mathbf{F_{spring}}(\mathbf{i}, \mathbf{j})$ is the force exerted upon the virtual spring connecting v_i and v_j, $\mathbf{v_{ij}}$ is the amount that the spring stretches relative to its defined resting length, and K is the proportionality constant, often referred to as the spring constant. The spring constant is a positive constant whose value is dependent upon the spring which is being studied. When part of the curve is manipulated, all the springs that form the mathematical curve will generate forces that incrementally update the locations of all the masses until all the springs are restored to their initial lengths (see e.g., Figure 4).



(a) (b) (c)

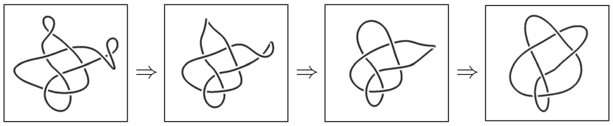
Figure 4: A mass-spring system with three masses. (a)-(c): Manipulation on v_1 will change the lengths of the two springs, which will produce forces to restore to their resting lengths.

Collision Avoidance. During the process of mathematical relaxation and physical simulation, mathematical knots can be stretched or moved around without tearing or self-intersection. At each simulation cycle, a new position for each mass is proposed by our force-driven model, and collision avoidance is strictly performed to detect potential collision between *point and point*, *point and segment*, and *segment and segment*. Most of the computation here is distance calculation. When two components are detected at a distance $\delta < 2 * R$, the pair of closest points on the colliding components are identified to define a 3D vector $\mathbf{v}$ passing through them. Then each component is shifted way by $R - \delta/2$ along v in an opposite direction to be out of collision range.

```
Input: Initial Layout of $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2 ..., \mathbf{v}_n\}$
Output: Updated Layout of $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2 ..., \mathbf{v}_n\}$
while the potential collisions are not handled do
    for $i = 0, ..., n-1$ do
        $\mathbf{F_c}(\mathbf{i}) = \mathbf{F_a}(\mathbf{i}) + \mathbf{F_r}(\mathbf{i})$ ;
        $\mathbf{v_i'} = \mathbf{v_i} + 0.2 * R * \widehat{\mathbf{F_c}} * \frac{||\mathbf{F_c}||}{f}$;
    end
    for $i = 0, ...n-1$ do
        for $j = 0, ...n-1$ do
            if $i \neq j$ then
                if $\delta(\mathbf{v_i}, \mathbf{v_j}) < 2 * R$ then
                    $\mathbf{v_i'} = \mathbf{v_i} + (R - \delta/2)$ ;
                    $\mathbf{v_j'} = \mathbf{v_j} - (R - \delta/2)$ ;
                end
                if $\delta(\mathbf{v_i}, \mathbf{s_j}) < 2 * R$ then
                    $\mathbf{v_i'} = \mathbf{v_i} + (R - \delta/2)$ ;
                    $\mathbf{v_j'} = \mathbf{v_j} - (R - \delta/2)$ ;
                    $\mathbf{v_{j+1}'} = \mathbf{v_{j+1}} - (R - \delta/2)$ ;
                end
                if $\delta(\mathbf{s_i}, \mathbf{s_j}) < 2 * R$ then
                    $\mathbf{v_i'} = \mathbf{v_i} + (R - \delta/2)$ ;
                    $\mathbf{v_{i+1}'} = \mathbf{v_{i+1}} + (R - \delta/2)$ ;
                    $\mathbf{v_j'} = \mathbf{v_j} - (R - \delta/2)$ ;
                    $\mathbf{v_{j+1}'} = \mathbf{v_{j+1}} - (R - \delta/2)$ ;
                end
            end
        end
    end
end
```

Algorithm 1: Collision avoidance mechanism for knot deformation.

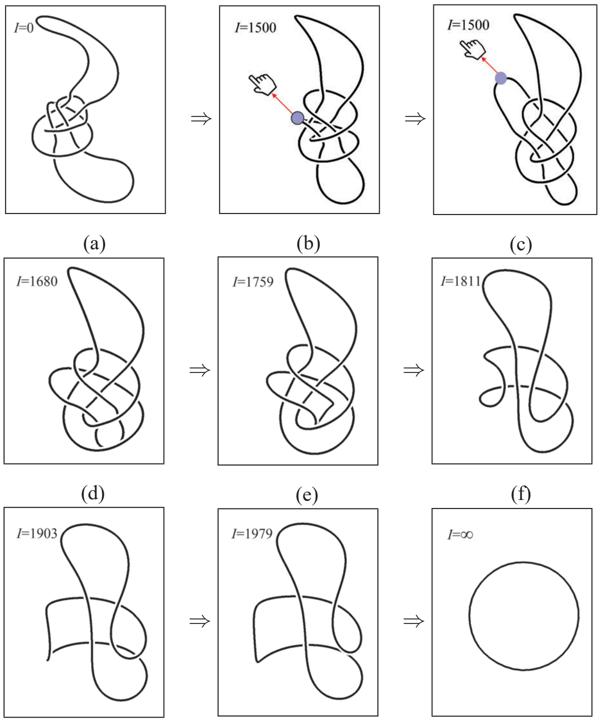
Algorithm 1 describes our collision avoidance mechanism. When mathematically evolving or being physically manipulated, the knots preserve the underlying topological structures while their geometric shapes are refined or simplified. Figure 5 shows a mathematical relaxation for $Knot_6^2$.



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

Figure 5: The relaxation for $Knot_6^2$.

During mathematical relaxations, knots' embeddings are updated by the attractive and repulsive forces applied to the masses, and tend to move to a stable configuration. It is also believed that this force-driven model is likely to only deform knots' embeddings to a local minimum [23]. Figure 6 shows one such situation where an overhand knot can't seem to escape from the local minimum construction with the mathematical relaxation model (see e.g., Figure 6(a)). In Figure 6(b)(c), the user nudge part of the overhand knot (very much like we try to untie a shoelace). This intervention helps "untie" the knot and it continue to evolve by itself and eventually reach a global stable configuration (an unknotted circle, see e.g., Figure 6(i)).



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 6: An overhand knot is relaxed and simplified to an unknot with physical intervention in the middle of the mathematical relaxation.

Plotting The Knot Energy

Knot energy is an important feature to be extracted when knots are deformed. In mathematical relaxation, a highly-tangled knot can evolved by itself to a simplied configuration, This is essentially driven by the knot's energy, charged at each of its masses. The energy is calculated based on the geometry of the knot and it changes when the knot updates its geometric structure during the deformation. A global minimal energy state is expected when the knot achieves its final goal.

There are several energy models being studied in the literature [24]. In our study the Minimum Distance (MD) energy model is used. MD energy model determines the knot energy using the minimum distances, which is consistent with force-directed algorithms where forces are also calculated by distances. The MD energy of a knot K is defined as follows:

$$E(K) = \sum \frac{L_i L_j}{D_{ij}^2} \tag{4}$$

where L_i is the length of edges $\mathbf{e_i}$, D_{ij} is the minimum distance between non-neighboring edges $\mathbf{e_i}$, $\mathbf{e_j}$.

Our visualization tool plots the knot's MD energy in a synchronized display window. In most of the mathematical relaxations, a mathematical knot will simplify its geometric structures driven by the two forces towards a lower-energy state. For example, Figure 7 shows the MD energy plotting for a piece of highly-tanged curve during the mathematical relaxation. At the beginning, the curve untangles by itself and its MD energy declines quickly as plotted. This is mainly because the repulsive forces in our force model can quickly repell most masses from each other to simplify the curve's structure. After around 900 computational iterations, the attractive and repulsive forces on the knot's masses seem to balance out, and the knot is stuck in a local minimum energy state (see e.g., the fairly flat MD energy curve between iteration 900 - 1100.) The knot would stay in this embedding if no intervention was applied to help it escape from the local minimum energy state. Around iteration 1200, a physical intervention is applied to nudge part of the curve, which resulted in a temporarily raised MD energy. However, this intervention helped the curve to break from the local minimum state, and further relax by itself. The curve evetually relax into a trivial knot after 1700 iterations.

Extracting Key Moments from Knot Evolution

Mathematical relaxation under the force-directed algorithm often takes a huge number of computing iterations, and leads to a very long process for one to observe the knot deformation, the interesting moments, and the final state. One interesting question is how to extract from the evolution process only those key moments when critial changes are taking place.

Key moments extracted from the knot evolution can effectly communicate about the topological problem we are interested in. In fact, a large portion of the mathematical evolution are trivial geometric updates and do not matter to the topological question being asked. These moments can be ignored from a mathematical visualization perspective. Secondly, we often need to trace the mathematical evolution to examine and validate each of the changes being produced. Computational or numeric errors in the long evolution might result in illegal moves or changes that do not respect the topological constraints, and eventually an incorrect final state. Therefore another value of key moment extraction is to provide a clear interface for us to trace and validate the entire mathematical evolution being proposed.

The key ideas of the overall scenario should now be clear. The logical series of extracting key moments from knot evolution are as follows:

- *Create a sequence of snapshots from the deformation*. We will capture every moment from the entire knot evolution,

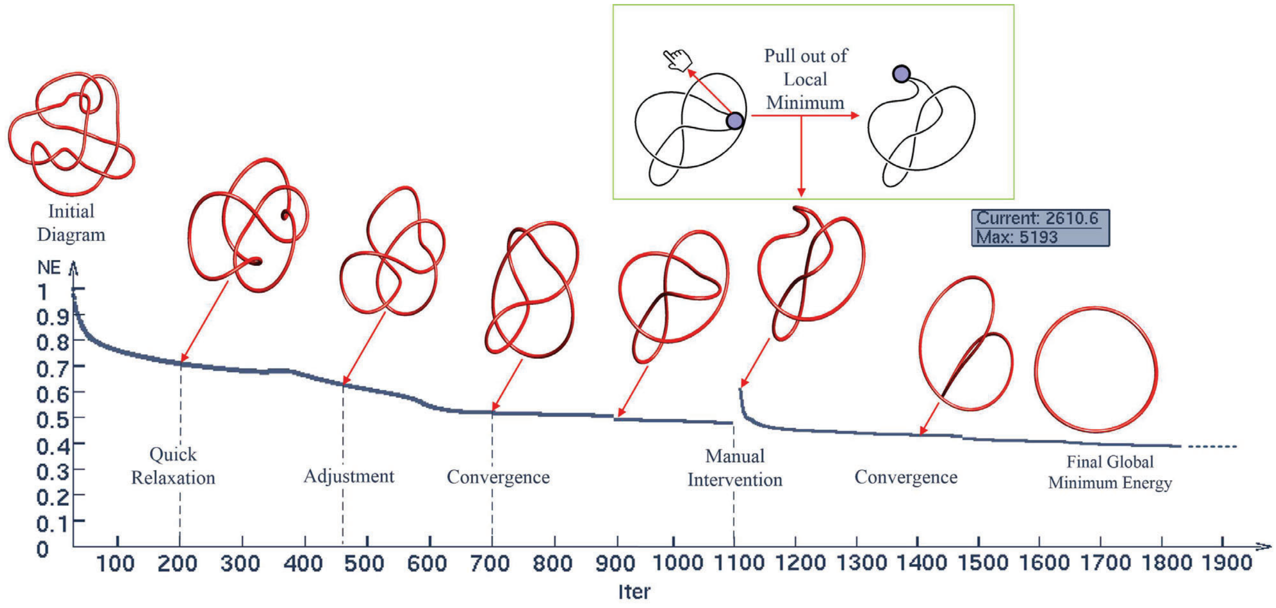
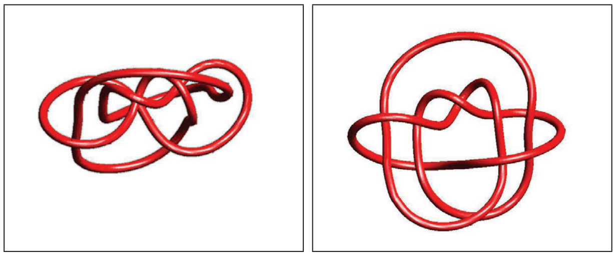


Figure 7: The energy plotting for a “monster” unknot being relaxed.

and we want to optimally aligns the knot across moments for the purpose of identifying changes.

- *Identifying the changes across moments*. For example, crossing number can be calculated and traced across all moments. If the crossing number is changed, a critical change has taken place from one moment to another.
- *Extracting the key moments*. To represent key moments of the evolution, an array of “Snapshots” are caputured from those moments when knot crossing number is changed across the moments (i.e., when critical changes took place in those moments.)

Finding the Best Projections. Mathematical knots do not preserve orientations during the deformation. A knot can appear to be very different when presented with different orientations. For example Figure 8(a) shows a fairly complicated “unknown” knot with a large number of crossing numbers in the presented view. The underlying structure, a $Knot_{10}^{161}$, is revealed in Figure 8(b) as known as the knot’s best projecction.



(a) (b)

Figure 8: Underlying knot structure and comparison of knot structure need to studied with the knot’s best projection. (a) an “unknown” knot with many crossings that are just artifacts of the projected angle. (b) The knot’s best projection view reveal its $Knot_{10}^{161}$ structure with the least crossing numbers in the view.

To identify key moments of the deformation where critical changes have taken place, we need to compare the knot images across a long sequence from an appropriate viewpoint. This type of computational comparison can be done by first orienting the 3D knot to the best projection (so that the number of crossings is minized in the presented view), and then calculating its crossing number changes from one moment to another.

A widely-used measure to evaluate the quality of best views of objects is viewpoint entropy based on Information Theory [25]. The idea of a larger entropy value means more information is included. In our study we implement an optimization measure, which features the area of projection and the visibility of segment in the projection. The formula is described as follows:

$$I(K) = \sum_{i=0}^{N_s} \left(\frac{L_i}{L_j} log \frac{L_i}{L_j} + V(i) \right) \tag{5}$$

where L_i represents the projected length of curve segment $\mathbf{s_i}$, L_t is the total length of the knot curve embedded 3D; $V(i)$ 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 (collisions) in the projection contributes to the entropy value as a penalty. In this way, the best projection is identified as one that contains the longest curve and the least number of crossing numbers in the projection.

With the viewpoint entropy defined above. The knot during the relaxation process is incrementally rotated in the 3-dimensional space, and the entropy value is calculated upon each oriention to identify the best projection that has achieved the largest viewpoint entropy value. We use Figure 9 and Figure 10 to show the result. Figure 9 shows an array of representative “snapshots” from its entire relaxation process, with many snapshots containing a large number of interruptions that could have been avoided if the knot was presented in a better projection. In

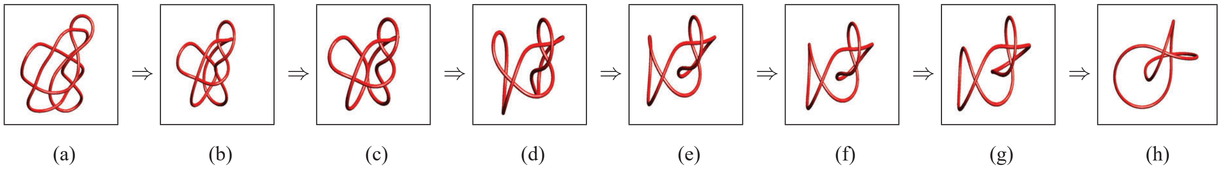


Figure 9: The relaxation for a trefoil knot in 3D space does not present its best view in 2D space.

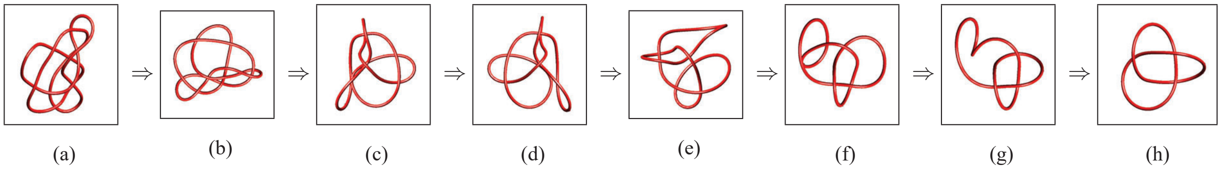


Figure 10: Using the best projection finding method, the relaxation illustrates the knot's best projection over time.

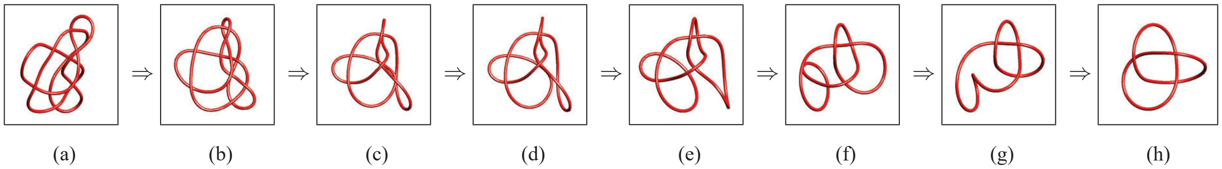


Figure 11: Key moments extracted from the trefoil knot's relaxation where each moment represents a critical change to the knot, and the knot image is optimally aligned from one moment to another.

Figure 10, each of the moments presents the knot structure with the least number of interruptions and provides a clear visualization for us to understand how the knot has been evolved during the entire process.

Key Moment Selection. Although finding knot's best projection at each selected moment can display the knot's underlying structure with the maximal information and the least interruptions, the knot images in successive terms (see Figure 10) do not preserve its orientation. This is because our best projection view is identified by rotating the knot in all angles to identify its greatest possible viewpoint entropy. This introduces visual discontinuities in the array of "snapshots" presented in Figure 10.

The visual experience can be further improved. To maintain visual continuity in the in successive terms of the "snapshots" we generate, we need a rigid transformation that can optimally align the deforming knot over moments in the least square sense. Let $\mathbf{K_1} = (\mathbf{V_1}, \mathbf{E_1})$ represent the knot at moment m_1, where $\mathbf{V_1} = \{\mathbf{v_1}_1, \mathbf{v_1}_2 ..., \mathbf{v_1}_n\}$ is the its vertices and $\mathbf{E_1}$ is the set of its edges. $\mathbf{K_1}$ deforms to $\mathbf{K_2} = (\mathbf{V_2}, \mathbf{E_2})$ at moment m_2, where $\mathbf{V_2} = \{\mathbf{v_2}_1, \mathbf{v_2}_2 ..., \mathbf{v_2}_n\}$. At moment m_2, we add a set of attractive mechanical force $\mathbf{F_a}$ applied between vertices in $\mathbf{V_1}$ and $\mathbf{V_2}$. The set of attractive forces (see Equation 1) will align $\mathbf{K_2}$ with $\mathbf{K_1}$ in the least square sense.

Figure 11 shows an improved key moment extraction where the least interruptions are introduced in each of the snapshot and visual continuity is mained across snapshots.

Depicting Mathematical Links and Braids

In this section, we proceed to extending our basic algorithms and user interfaces to support the depiction of mathematical links and braids. Links and Braids are two important branches of knot theory, and they share many fundamental structures and properties of their knot counterparts.

Link Representation. In mathematical knot theory, a link is a collection of knots that are linked together but do not intersect with each other. A knot can be described as a special link with only one component [11]. Two links are considered to be the equivalent if one can be deformed to another without any tearing or intersections in any component of the link.

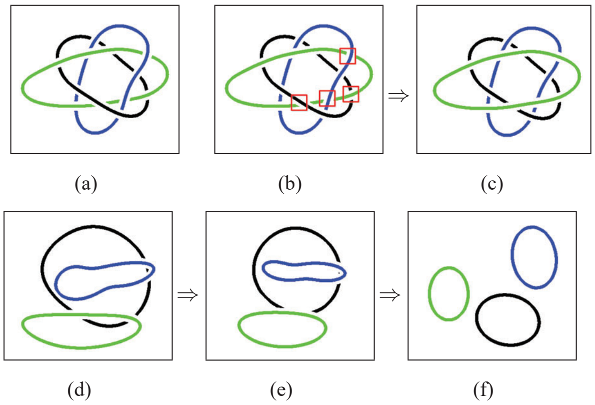


Figure 12: Relaxation of 3-Brunnian link with 12-crossing. (a) Initial diagram. (b) One component in green removed. (c)-(f) The components of the 3-Brunnian Link will deform to three seperate trivial knots.

The force model presented in our knot interface in fact extends trivially to mathematical links which are a collections of knots. The only essential difference is that in a mathematical link, each vertex has still two immediate neighbouring nodes, but will a lot more non-adjacent nodes from its own component and from all other components in the link. Similarly, when calculating distances to identify potential collisions, points and segments from other components should also be included. The following algorithm describes the extended relaxation algorithm for mathematical links:

Figure 12(a) shows a three-component Brunnian link with 12-crossing link scheduled and relaxed in our tool. A Brunnian

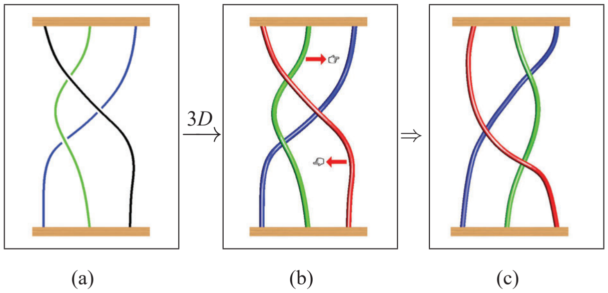
```
Input: Initial Link Layout
Output: Relaxed Link Configuration
while the stop condition is not satisfied do
    for each knot in the link do
        for each mass in the knot do
            Calculate $\mathbf{F_a}$ for each mass;
            Calculate $\mathbf{F_{r1}}$ inside the knot;
            Calculate $\mathbf{F_{r2}}$ from other knots;
            Calculate $\mathbf{F_c} = \mathbf{F_a} + \mathbf{F_{r1}} + \mathbf{F_{r2}}$;
            Generate the new position;
        end
    end
    for each knot in the link do
        for each mass in the knot do
            Calculate the potential collision between
              masses and segments inside the knot;
            Calculate the potential collision between
              masses and segments from other knots;
            Adjust the position;
        end
    end
end
```

Algorithm 2: Relaxation algorithm for link.

link is a nontrivial link that becomes a set of trivial unlinked circles if any one component is removed. In other words, cutting any loop frees all the other loops (so that no two loops can be directly linked [26].) We can experiment with this mathematical phenoma in our link interface. As Figure 12(b)(c) shows, we simply modified the crossings of one component to have that component in green removed. All the components of the Brunnian link will continue to evolve to three trivia knots and separated due to the repulsive forces in our relaxation algorithm.

Braids Representation. Our current implementation can also be trivially extended to depict mathematical braids and their evolution. A mathematical braid is a collection of strands in space with fixed endpoints that are braided around each other. The braid from top to bottom forms a word in the generators$\sigma_1, \sigma_2, ..., -_{n-1}$, where σ_i represents strands i passing over $i+1$. Similar to knot equivalence, two topological braids are isotopic if one can be continuously deformed into the other without causing any of strands to intersect each other [27].

The braid shares the same data structure and relaxation algorithm as in mathematical links. To "fix" the endpoints, we assign a negative value to the masses on the two endpoints of each strand in the braid. These masses with negative value will not be updated (and thus will stay fixed) during deformation and relaxation. The strands of a braid can be presented with 2D/3D diagrams, and it can be rearranged to its equivalent one with mathematical relaxation or the user's manipulation (see e.g., Figure 13).



(a) (b) (c)

Figure 13: An isotopy between two braids. (a) 2D braid ($\sigma_i\sigma_{i+1}\sigma_i$). (b) 3D braid ($\sigma_i\sigma_{i+1}\sigma_i$). (c) 3D braid ($\sigma_{i+1}\sigma_i\sigma_{i+1}$).

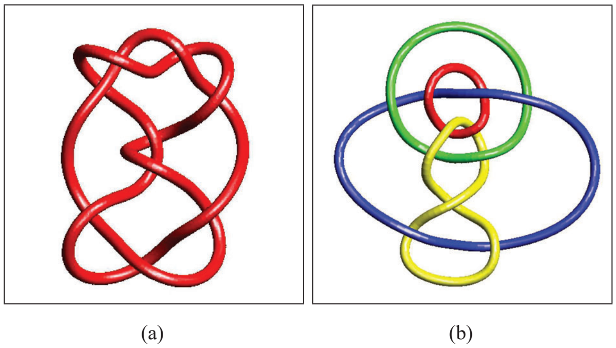
Implementation Environment and User Study

Implementation Environment. Our visualization interface is based on OpenGL and Windows Visual Studio C++. The software runs on a Dell PC desktop with 3. 5GHz Intel Xeon CPU. A preliminary evaluation to gauge user experience with this user interface design was performed.

Participants. A total of 8 participants were recruited, including 3 males and 5 females, their age ranged between 11 to 45. These participants include elementary school students, K-12 mathematical teachers, and graduate students from college. All of the subjects have experiences with personal laptops and sketching interface like Microsoft Paint software. None of them had any prior knowledge of knot theory, except that the mathematics teachers are familiar with topology concepts. Before performing the required tasks, all the subjects were given a brief introduction and demonstration about the interface elements.

The Tasks. The set of tasks to complete with our knot interface were as follows:

1. Sketch a trefoil knot (i.e., $Knot_1^3$);
2. Given $Knot_9^9$ diagram (see Figure14 (a)) on the sheet, sketch it on the computer and refine its structure;
3. Given $Link_{11}^{459}$ diagram (see Figure14 (b)) on the sheet and repeat the above procedure;
4. Deform a given braid (see e.g. Figure 13) to find its isotopy.



(a) (b)

Figure 14: The knot and link diagrams used in the task.

The Measures. In particular, we measured the usability of this user interface following a System Usability Scale (SUS) questionnaire [15] [28], which gives a global view of subjective usability assessments.

Table 1: SUS-like questionnaire scores for user satisfaction.

No.	Survey	Average Scores
1	The overall function is easy to follow.	3.3
2	The knot edit function is unnecessarily complex.	3.6
3	The knot depiction was easy to complete.	3.8
4	The link example was difficult to depict.	3.4
5	The braid diagram was easy to depict.	2.1
6	The relaxation process was difficult to follow.	3
7	The key moments setting is necessary for a better observation of relaxation.	3.5
8	The energy plotting function is useless.	3.6
9	I think that I would like to use this system frequently.	1.8
10	I needed to learn a lot of things before I could get going with this system.	3.4
	Overall Score (sum of the aboved scores *2.5)	78.6

A pool of 10 questionnaire items covers a variety of aspects of the system usability, including the requirement for support, training, the system complexity and the user's subjective reactions to using the system (see e.g., Table 1). They are assembled with 5-score scale and different score contribution ranged from 0 to 4. For the odd items the score contribution is the scale position minus 1. For the even items, the contribution is 5 minus the scale position. The overall score is obtained through multiplying the sum of the scores by 2.5 which is ranged from 0 to 100.

The Results. The result of the questionnaire scores are shown in Table 1, ranged from 67.5 to 87.5, with an average value of 78.6. This value is an acceptable level (65-84) in the SUS scores. The distribution of each score is shown in Figure 15. Most participants gave high scores to easy-using and function support. Few of them needed extra technical support to complete the tasks. Low scores were given to Q_9. Only a graduate student who works in the field of visualization gave a high score, which is acceptable due to the specific topic being explored in this study. our study has had a task to depict mathematical braids, some users found that task was too complicated to complete. The general observation was that users had a positive experience for knot depiction and manipulation. Most participants were able to complete the tasks with minimum instruction and practices.

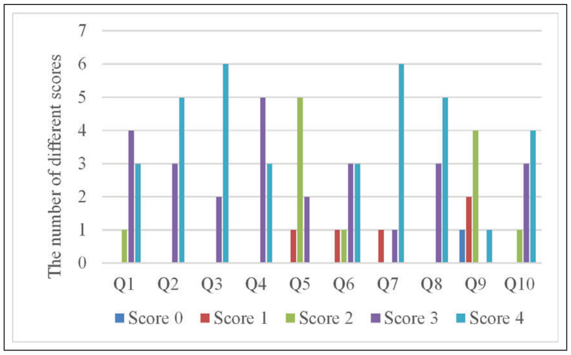


Figure 15: The distribution of scores.

Conclusion and Future Work

We have discussed a family of interactive methods for exploring mathematical knots, links, braids, and their mathematical phenoma such as topological relaxation. We propose a hybrid force-driven model to allow mathematical knots to untangle themselves driven by the energies, while allowing the users to intervene the automatic relaxation to escape from local minimum energy state. While many mathematical phenoma involve a long sequence of deformation, we introduce an innovative key moment extraction method to reduce the long evolution into a minimal number of visual frames that represents the critical changes over time. Our future directions include the study of more intelligent and efficient relaxation methods to accelerate the mathematical knot evolution, and the extension of our algorithms and user interfaces to support a family of mathematical entities such as surfaces and manifolds embedded in 4-dimensional space to understand and visualize their deformations.

Acknowledgments

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

References

[1] C. Livingston, *Knot theory*, vol. 24. Cambridge University Press, 1993.

[2] E. Rowell and Z. Wang, "Mathematics of topological quantum computing," *Bulletin of the American Mathematical Society*, vol. 55, no. 2, pp. 183–238, 2018.

[3] D. Kleckner and W. T. Irvine, "Creation and dynamics of knotted vortices," *Nature physics*, vol. 9, no. 4, p. 253, 2013.

[4] J.-F. Ayme, J. E. Beves, D. A. Leigh, R. T. McBurney, K. Rissanen, and D. Schultz, "A synthetic molecular pentafoil knot," *Nature Chemistry*, vol. 4, no. 1, pp. 15–20, 2012.

[5] K. Alexander, A. J. Taylor, and M. R. Dennis, "Proteins analysed as virtual knots," *Scientific reports*, vol. 7, p. 42300, 2017.

[6] O. Knill and E. Slavkovsky, "Illustrating mathematics using 3d printers," *arXiv preprint arXiv:1306.5599*, 2013.

[7] M. Lackenby, "A polynomial upper bound on reidemeister moves," *Annals of Mathematics*, pp. 491–564, 2015.

[8] A. J. Hanson, T. Munzner, and G. Francis, "Interactive methods for visualizable geometry," *Computer*, vol. 27, no. 7, pp. 73–83, 1994.

[9] J. J. Van Wijk and A. M. Cohen, "Visualization of seifert surfaces," *IEEE Transactions on Visualization and Com-*

puter Graphics, vol. 12, no. 4, pp. 485–496, 2006.

[10] M. Carlen, "Computation and visualization of ideal knot shapes," 2010.

[11] K. Murasugi, *Knot theory and its applications*. Springer Science & Business Media, 2007.

[12] R. G. Scharein, *Interactive topological drawing*. PhD thesis, University of British Columbia, 1998.

[13] Y. Wu, "An md knot energy minimizing program," *Department of Mathematics, University of Iowa*.

[14] 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.

[15] G. Costagliola, M. De Rosa, A. Fish, V. Fuccella, R. Saleh, and S. Swartwood, "Knotsketch: a tool for knot diagram sketching, encoding and re-generation," *Journal of Visual Languages and Sentient Systems*, vol. 2, pp. 16–25, 2016.

[16] H. Zhang, S. Thakur, and A. J. Hanson, "Haptic exploration of mathematical knots," in *International Symposium on Visual Computing*, pp. 745–756, Springer, 2007.

[17] H. Zhang, J. Weng, and G. Ruan, "Visualizing 2-dimensional manifolds with curve handles in 4d," *IEEE Transactions on Visualization & Computer Graphics*, no. 1, pp. 1–1, 2014.

[18] H. Zhang, J. Weng, L. Jing, and Y. Zhong, "Knotpad: Visualizing and exploring knot theory with fluid reidemeister moves," *IEEE transactions on visualization and computer graphics*, vol. 18, no. 12, pp. 2051–2060, 2012.

[19] Y. Hu and L. Shi, "Visualizing large graphs," *WIREs Comput. Stat.*, vol. 7, pp. 115–136, Mar. 2015.

[20] S. C. North and G. Woodhull, "Online hierarchical graph drawing," in *Graph Drawing* (P. Mutzel, M. Jünger, and S. Leipert, eds.), (Berlin, Heidelberg), pp. 232–246, Springer Berlin Heidelberg, 2002.

[21] C. Hall, "Natural cubic and bicubic spline interpolation," *SIAM Journal on Numerical Analysis*, vol. 10, no. 6, pp. 1055–1060, 1973.

[22] U. Shani and D. H. Ballard, "Splines as embeddings for generalized cylinders," *Computer Vision, Graphics, and Image Processing*, vol. 27, no. 2, pp. 129–156, 1984.

[23] M. Huang, R. P. Grzeszczuk, and L. H. Kauffman, "Untangling knots by stochastic energy optimization," in *Proceedings of the 7th conference on Visualization'96*, pp. 279–ff, IEEE Computer Society Press, 1996.

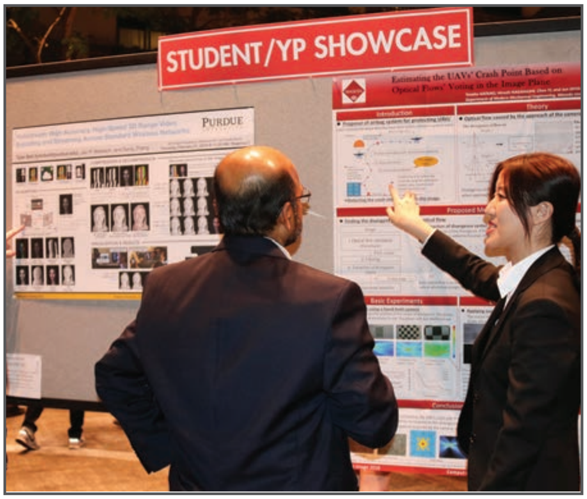
[24] J. Simon, "Energy functions for knots: beginning to predict physical behavior," in *Mathematical Approaches to Biomolecular Structure and Dynamics*, pp. 39–58, Springer, 1996.

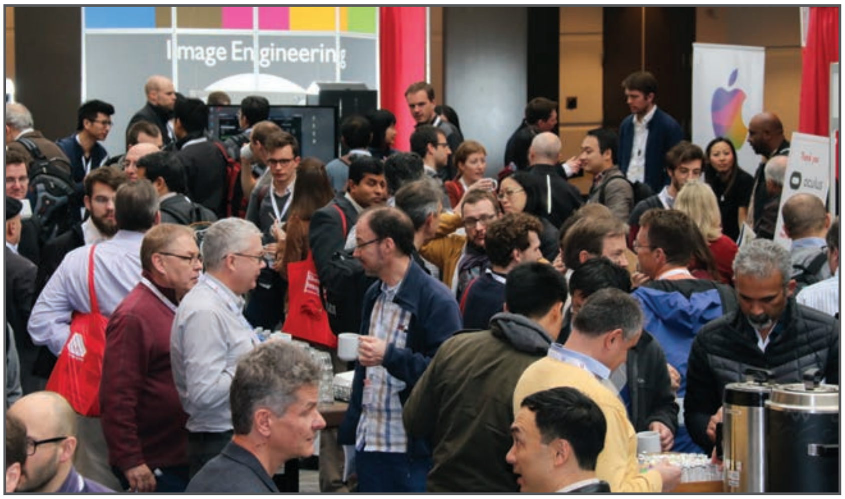
[25] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy.," in *VMV*, vol. 1, pp. 273–280, 2001.

[26] C. Liang and K. Mislow, "On borromean links," *Journal of Mathematical Chemistry*, vol. 16, no. 1, pp. 27–35, 1994.

[27] R. Hoberg, "Knots and braids," *University of Chicago*, 2011.

[28] S. McLellan, A. Muddimer, and S. C. Peres, "The effect of experience on system usability scale ratings," *Journal of usability studies*, vol. 7, no. 2, pp. 56–67, 2012.





AR is a fundamental and transformative interface and platform.

20+ years from my initial working demo to today's AR displays

Optimistic that it won't take another 20 to reach ubiquitous consumer AR!