Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 309–314
DOI: 10.15439/2025F5270
ISSN 2300-5963 ACSIS, Vol. 43

Length-Preserving Hair Simulation via Bézier Curves and Artificial Neural Network

Kali Gurkahraman
0000-0002-0697-125X
Department of Computer Engineering
Sivas Cumhuriyet University
Sivas, Turkey
kgurkahraman@cumhuriyet.edu.tr

Ahmet Firat Yelkuvan
0000-0003-4148-1923
Department of Computer Engineering
Sivas Cumhuriyet University
Sivas, Turkey
aftyelkuvan@cumhuriyet.edu.tr

Rukiye Karakis
0000-0002-1797-3461
Department of Software Engineering
Sivas Cumhuriyet University
Sivas, Turkey
rkarakis@cumhuriyet.edu.tr

***Abstract*—In this study, we present a data-driven method for preserving the length of cubic Bézier curves under external deformations, specifically in the context of hair strand simulation. By sampling millions of original and displaced control point configurations within a 2D bounded grid space, we construct a large dataset of Bézier curve pairs and compute the corresponding t_{final} parameters required to preserve arc length. A lightweight artificial neural network (ANN) architecture is trained on this dataset to predict t_{final} given 16 features representing the coordinates of initial and displaced control points. The model achieves a low mean absolute error (MAE) of 0.00091992 on the test set, ensuring high predictive accuracy. Performance evaluations show that the ANN can predict t_{final} values for up to 200k hair strands in approximately 8 seconds on a standard laptop, aligning with the average number of strands on a human scalp. This approach replaces computationally expensive numerical length-matching operations with a fast, inference-based framework, allowing for efficient simulation of realistic, deformable hair motion while maintaining individual strand lengths.**

***Index Terms*—Hair simulation, hair modelling, Bézier curve, artificial neural network**

I. Introduction

A SINGLE human hair strand, though microscopically thin, represents a significant computational unit in computer graphics due to its complex physical properties. A realistic digital human typically involves simulating tens of thousands of strands—up to 100,000 on average—making hair one of the most resource-intensive components in real-time rendering [1]. In modern video games, virtual reality, and interactive applications, simulating lifelike hair movement under dynamic motion and external forces such as wind is essential but remains a non-trivial challenge. A fundamental and often under-addressed issue in this context is preserving the physical length of each strand, especially during rapid motion or deformation.

Hair simulation traditionally relies on strand-based models, which represent each hair as a discrete chain of mass points connected by springs. These models offer fine-grained control over local deformation such as bending, twisting, and inter-strand collision handling [2,3]. Mass-spring systems, for example, have been widely used due to their simplicity and real-time compatibility. However, ensuring inextensibility—the preservation of strand length—becomes problematic, especially under dynamic motion, where stiff springs introduce instability [3,4]. To improve numerical robustness, implicit integration schemes have been introduced [4,5], but their computational overhead makes them impractical for real-time environments. More recent advancements, such as the Augmented Mass-Spring (AMS) model, introduce ghost rest-shapes and biphasic coupling to improve simulation stability and strand length preservation during movement and collisions [6]. Additionally, Müller et al. proposed the Dynamic Follow-The-Leader (DFTL) method, which enforces inextensibility through a geometric update rule and achieves real-time performance with a single iteration per frame [7]. Another common artifact in strand-based systems is initial sagging caused by gravitational forces at the start of the simulation. To overcome this, Hsu et al. developed a sag-free initialization framework, which establishes a static equilibrium and neutral torque configuration prior to dynamic simulation. This significantly reduces unwanted deformation and helps preserve the intended hairstyle [8]. Furthermore, Position-Based Dynamics (PBD) has become a foundational strand-based method by directly enforcing positional constraints such as segment length. PBD is widely used in interactive simulations due to its robustness and control [9].

In contrast, volume-based methods treat hair as a continuous medium, modeling global movement and shape using tools such as fluid dynamics, volumetric mass models, or free-form deformations [10,11]. These approaches offer better scalability for large hair volumes and can efficiently capture overall flow and motion. However, they often sacrifice individual strand resolution, which is critical for detailed hairstyles and physical realism. For example, continuum models [10] and lattice-based structures [11] can simulate general hair behavior but lack fidelity in capturing high-frequency deformations or enforcing per-strand constraints like inextensibility. As a result, while volume-based methods are advantageous for broad dynamics, they are less suited for applications requiring high detail, such as cinematic effects or close-up character interactions.

Given the respective limitations of strand- and volume-based approaches, researchers have increasingly turned to hybrid simulation methods. These techniques are designed to leverage the physical fidelity of strand-level models while

incorporating the computational efficiency and stability offered by volume-based or constraint-based frameworks. One such approach is Position-Based Dynamics (PBD), which eliminates the need for solving complex differential equations and enforces constraints directly on particle positions [12]. Shape matching methods [9] and meshless deformation frameworks have also been explored to stabilize motion while avoiding stiffness artifacts common in traditional systems. Another line of hybrid research adopts rigid body inspired models using serial chains, enabling accurate modeling of both local strand behavior and global transformations [13,14]. These techniques integrate strand-level articulation within a larger kinematic or physics-based framework to improve simulation realism without sacrificing computational efficiency.

Recent years saw, the emergence of data-driven techniques has opened new possibilities in hair simulation. Neural network-based methods aim to approximate complex physical behaviors without relying on explicit solvers, making them especially suitable for real-time environments. One such approach, Quaffure [15], introduces a quasi-static neural model trained with a self-supervised loss function to predict plausible strand deformations across a variety of poses and hairstyles. By eliminating the need for simulated training data, Quaffure achieves high temporal stability and runs efficiently on consumer hardware, bridging the gap between physical accuracy and performance.

Complementing this trend, Dr.Hair [16] addresses the challenge of strand-level reconstruction from visual input. Leveraging a differentiable rendering framework, the method reconstructs scalp-connected hair strands directly from multi-view images without pretraining, accurately preserving both strand length and directional flow. These techniques exemplify how learning-based methods can supplement or replace traditional solvers for specific simulation tasks.

Alongside advances in simulation, efficient rendering methods have also evolved to meet the demands of real-time graphics. Hair mesh systems, such as the one introduced by Yuksel et al. [17], generate strand geometry procedurally on the GPU at render time. This drastically reduces memory usage while maintaining geometric fidelity, allowing for the real-time rendering of hundreds of uniquely styled characters. Together, these learning- and rendering-focused approaches highlight a growing trend toward hybrid solutions that integrate physical modeling, data-driven inference, and hardware-aware rendering strategies to advance the realism and scalability of hair simulation systems.

Building on the potential of curve-based modeling and hybrid techniques, recent studies have explored combining Bézier representations with physically inspired solvers and mathematical optimization techniques—such as the use of Budan–Fourier analysis for root bounding in curve behavior prediction [18] and contour tracing algorithms for shape reconstruction in structured objects [19]. A notable example is the work which integrated Bézier curves with the Articulated Body Method (ABM) to simulate dynamic hair strand behavior with real-time performance [20]. Their model treats each strand as a Bézier curve and applies ABM to resolve motion and constraint forces efficiently. These approaches reinforce the growing viability of combining geometric curve modeling with physically inspired or learning-based systems—an idea central to our study, which uses Bézier curves in conjunction with artificial neural networks to predict strand deformation under motion and external forces.

This study specifically focuses on the problem of strand length preservation during hair motion and external interactions, such as wind. We critically analyze existing approaches from the perspective of both physical accuracy and real-time performance, highlighting their respective advantages and limitations. Based on this analysis, we propose directions for enhancing simulation fidelity while maintaining computational feasibility. A shallow artificial neural networks (ANN) can accurately and efficiently predict the target final curve parameter for achieving length-preserving transformations of hair strands represented by Bezier curves. Due to their parametric structure, differentiability, and compact representation, Bezier curves inherently support scalable and generalized deformation modeling. This makes them particularly advantageous for learning-based strand manipulation, even under varying strand complexities. This study was initiated with this hypothesis in mind, aiming to validate the effectiveness of lightweight ANN architectures in performing accurate and efficient strand deformations through Bezier-based geometric encoding.

II. Materials and Methods

In this study, we propose a data-driven approach based on ANN to maintain the length of cubic Bezier curves under external deformations, specifically aimed at simulating individual hair strands affected by physical forces while preserving their natural length. This section details the process of data generation, Bezier curve modeling, and ANN-based prediction methodology.

A. Bézier Curve and Hair Modelling

A cubic Bézier curve is defined by four control points: A, B, C, and D, and is expressed parametrically as in equation (1):

$$P(t) = (1-t)^3 A + 3(1-t)^2 tB + 3(1-t)t^2 C + t^3 D \quad (1)$$

Where $t \in [0, 1]$. As the parameter t varies from 0 to 1, the curve smoothly transitions from A to D, influenced in shape by intermediate control points B and C. This property makes Bézier curves particularly well-suited for modeling flexible and natural curves such as hair strands. For example, let us consider a hair strand modeled by a Bézier curve with the control points at locations (x, y) : $A = (0, 0), B = (0.25, 1.0), C = (0.75, -1.0)$ and $D = (1, 0)$. This configuration produces a slightly wavy shape representative of a natural hair strand. Figure 1 illustrates this cubic Bézier curve constructed from the defined control points.

An important advantage of using Bézier curves in hair simulation is that each point on the curve can be independently

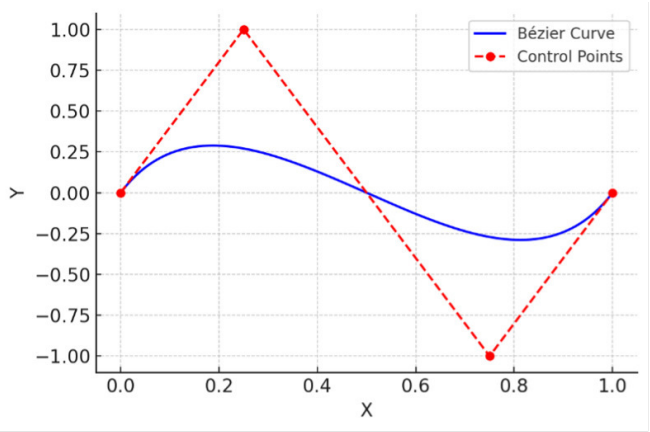

Fig. 1. Bézier Curve for a Wavy Hair Strand

computed by evaluating the Bézier formula at any value of t. For instance, stepping through the interval $[0, 1]$ in increments of 0.1 allows for the generation of discrete, evenly spaced points along the hair strand. This makes the approach inherently parallelizable and computationally efficient for simulating thousands of strands simultaneously.

B. Bézier Curve Representation and Length Estimation for Data Generation

We consider a cubic Bézier curve defined by four control points: A,B,C, and D. Points A and D are fixed at coordinates (0, 0) and (1, 0), respectively, defining the endpoints of a hair strand. Control points B and C are allowed to locate within a region $[0, 1] \times [-1, 1]$ (point space region shown in Figure 1 with 1 unit width and 2 unit height) with a spatial resolution of 0.1. This resolution results in a discrete point cloud of 231 (11x21) possible positions for each of B and C. Although 231 grid points are defined, combining different configurations of initial and moved control points can yield tens of millions of curve pairs. This vast combinatorial space ensures that the dataset captures a wide range of plausible deformations necessary for training a reliable predictive model. However, since many initial and subsequent curves (moved curves) would be undesirably close to each other at a resolution of 0.1, two distinct region location maps were constructed. The first one is a point space with 0.2 resolution at even position values (e.g., 0, 0.2, 0.4), and the second one is a point space with 0.2 resolution at odd position values (e.g., 0.1, 0.3, 0.5). Using each position space separately, initial and subsequent curve locations were determined, and some unsuitable curve pairs were filtered out. Then, to ensure an acceptable data loading time and avoid memory issues, a dataset consisting of 7.5 million samples was constructed through random selection. Each sample consists of 16 input features: the x and y coordinates of the original and moved control points (8 control points with sixteen x and y components), and one output value: the corresponding t_{final} that preserves the curve length. Figure 2 illustrates a unit square point cloud at 0.1 resolution, with an example of original and subsequent cubic Bézier curves generated using a specific configuration of the control points. For each data sample, the corresponding t_{final} values were computed to ensure that the curve length at the subsequent position is equal to that at the initial position.

To simulate external influences, new displaced positions B' and C' are sampled independently from the same discrete space. For each $A-B-C-D$ configuration and its subsequent curve counterpart $A - B' - C' - D$, the curve length of the original and deformed curve is computed. The first curve is parameterized over the interval $t \in [0, 1]$, and the moved curve length is evaluated using numerical integration (e.g., Simpson's rule). Then, the moved curve $A - B' - C' - D$ is sampled over $t \in [0, t_{final}]$ such that its arc length is numerically equal to the original curve.

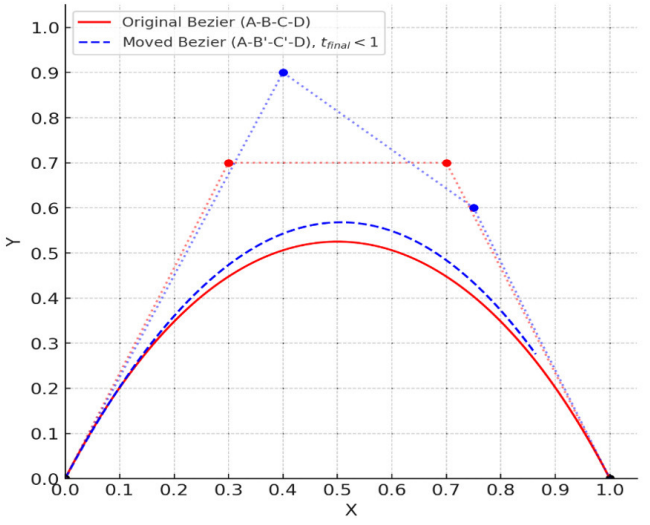

Fig. 2. Example of original and moved Bézier curves

C. Scaling Bézier Curves to Any Rectangular Domains

Although the dataset is generated within a $[0, 1] \times [-1, 1]$ region, Bézier curves can be reliably applied to arbitrarily positioned rectangular regions. This is achieved through a combination of translation and axis-wise scaling that aligns the rectangular configuration with the ANN's reference point space domain. Let a rectangular region be defined by horizontal bounds $[x_{min}, x_{max}]$ and vertical bounds $[y_{min}, y_{max}]$. To align this region with the reference point space, we first translation and then scaling operations as in (2), (3) and (4):

Translation: Shift the lower-left corner of the rectangle to the origin by subtracting (x_{min}, y_{min}) from all control points.

Scaling: Normalize the width and height of the rectangle using the scale factors:

$$s_x = x_{\max} - x_{\min} \tag{2}$$

$$s_y = \frac{y_{\max} - y_{\min}}{2} \tag{3}$$

Each control point $P = (x, y)$ is then transformed to reference region coordinates by:

$$P_{\text{unit}} = \left(\frac{x - x_{\min}}{s_x}, \frac{y - y_{\min}}{s_y} \right) \tag{4}$$

This transformation allows both original and displaced control points to be converted into a form compatible with the ANN trained on reference region samples. Once the ANN predicts the t_{final} value, the Bézier curve can be redrawn directly in the rectangular domain using the actual (moved) control points A, B', C', and D. This method preserves the accuracy of the curve length while adapting the model to diverse geometric contexts.

D. Neural Network Model

To approximate the mapping from control point displacement to the required value, a feed-forward artificial neural network (ANN) was constructed. The ANN model receives a 16-dimensional feature vector as input—comprising the x and y coordinates of original and displaced control points (total of 8 points) —and outputs a single scalar. The network architecture consists of four fully connected layers implemented using the Keras Sequential API:

```
model = Sequential([
    Dense(128, activation='relu',input_shape=(16,)),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1, activation='sigmoid')
])
```

This shallow and lightweight model architecture is intentionally selected to enable extremely fast predictions, which is essential when simulating thousands of hair strands in real-time applications. Although the model depth is limited, its predictive power is effectively enhanced by using a very large and diverse training set of more than 7 million Bézier curve configurations.

To train the network, the dataset was split into 80% training and 20% test sets. The target values were normalized using a $MinMaxScaler$. The model was trained with a batch size of 1000 for up to 350 epochs using the Adam optimizer. Early stopping and model checkpointing strategies were used to prevent overfitting and retain the best-performing model. After training, the best model achieved low error on unseen data, with the mean absolute error and sum of absolute errors computed during evaluation. The overall strategy demonstrates that by combining a simple ANN structure with a large and carefully prepared dataset, highly accurate and computationally efficient predictions can be achieved.

This ANN framework enables the rapid estimation of values for potentially thousands of hair strands affected by external factors, allowing for efficient regeneration of hair geometry while maintaining individual strand lengths. The proposed solution thus replaces computationally expensive numerical searches with a lightweight inference mechanism. The network is trained using the mean absolute error (MAE) as the loss function. MAE is a commonly used regression metric that quantifies the average magnitude of the errors between predicted values and the actual ground truth values. It is defined by equation (5).

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |t_{\text{final\_gt}} - t_{\text{final\_pr}}| \tag{5}$$

Where n, t_{final_gt} and t_{final_pr} are number of samples, the ground truth and predicted time values, respectively. MAE is expressed in the same units as the target variable and provides an intuitive measure of model accuracy. It is especially useful when all errors are equally weighted.

III. Results and Discussion

All experiments were conducted on a laptop running Windows 10 Pro, equipped with a 12th Gen Intel® Core™ i7-12700H processor (2.30 GHz, 14 cores, 20 threads), 32 GB of RAM and NVIDIA GeForce RTX 3060 Laptop GPU. The training and testing of the proposed ANN model were implemented using the Keras library (Python 3.9) with TensorFlow. Hair strand movements under external forces were simulated using MATLAB R2021a.

Table 1 gives the training, testing and prediction details of the study. This study used a training dataset of 6 million samples (80% of the full dataset), and 1.5 million samples were reserved for testing. The model achieved a test set MAE of 0.00091992, indicating high predictive accuracy in estimating the parameter for length-preserving curve transformations. In this context, the MAE corresponds to the average deviation in the predicted t_{final} value per hair strand. It does not directly represent a geometric length error but rather a deviation in the Bézier time parameter that determines where the curve should be truncated to preserve length. For example, a MAE of 0.00091992 implies that, on average, the predicted truncation point deviates from the true value by less than one-thousandth of the normalized parameter range $[0, 1]$. Assuming that a single hair strand may undergo deformation multiple times, such a deviation would only accumulate to a 1% error after approximately 5.48 independent displacements for 0.5 of t_{final} true value $(5.48 \times 0.00091992/0.5 \approx 0.01)$. This indicates that even under sustained movement, the predicted t_{final} remains sufficiently accurate to maintain a nearly constant hair length. This also suggests that increasing the amount of training data and using more powerful computational hardware could further reduce the MAE, improving the model's suitability for long-duration hair simulations under dynamic conditions. Table 1 also presents prediction runtime for varying batch sizes. The model predicted values for 1, 1k, 10k, 100k, and 200k samples (batch size) in approximately 0.0579, 0.0994, 0.5296, 4.09, and 8.11 seconds, respectively. The inclusion of 200k samples simulates a full-head hair scenario, as this figure closely matches the average number of hair strands on a human scalp. While the current implementation achieves prediction times of approximately 8 seconds for 200k samples, this duration may be considered borderline for real-time applications. However, this benchmark was obtained on a standard laptop

without GPU acceleration. Since the prediction task involves a shallow ANN with only a few dense layers, it is likely that the inference time could be significantly reduced on optimized hardware or with additional engineering effort. Employing GPU-accelerated inference and incorporating preprocessing strategies such as dimensionality reduction or data batching may potentially improve performance and make the model more suitable for time-sensitive applications. The results in Table 1 pertain to the prediction of t_{final} values rather than complete physical simulation, and thus no direct baseline comparison is available in the literature.

TABLE I
TRAINING, TESTING AND PREDICTION DETAILS OF THE STUDY

Number of train data (0.8 split ratio)		6 milion
Number of test data (0.2 split ratio)		1.5 million
MEA test error		0.00091992
Prediction time (second) [mean (std) for 5 trials]	1 sample	0.05795540 (0.002448)
	1k samples	0.09941540 (0.013011)
	10k samples	0.52964460 (0.039699)
	100k samples	4.08589480 (0.188377)
	200k samples	8.10715880 (0.45811)

Figure 3 illustrates an initial hair strand (blue curve) and its multiple deformed configurations (red curves) resulting from external influences applied to its control points. Despite visual variations in trajectory, all red curves are generated by adjusting the t_{final} parameter through the trained ANN model such that their arc lengths are preserved relative to the initial curve. This figure demonstrates the model's effectiveness in maintaining strand length consistency under dynamic conditions, validating the proposed method for realistic hair motion simulation. Based on the curve analysis illustrated in the figure, the original strand represented by the blue curve has a computed length of 4.803 units. However, the final (sixth) red strand—generated with a cumulative t_{final} deviation of 0.00091992 per step—exhibits a length of 4.876 units. This results in an absolute error of 0.073 units, corresponding to a relative error of approximately 1.52% compared to the original. These findings demonstrate that even minor per-step deviations in t_{final} can accumulate and lead to noticeable differences in physical attributes such as strand length. Consequently, preserving the original length with high fidelity becomes essential, especially in applications requiring physical realism, such as animation and real-time hair simulation.

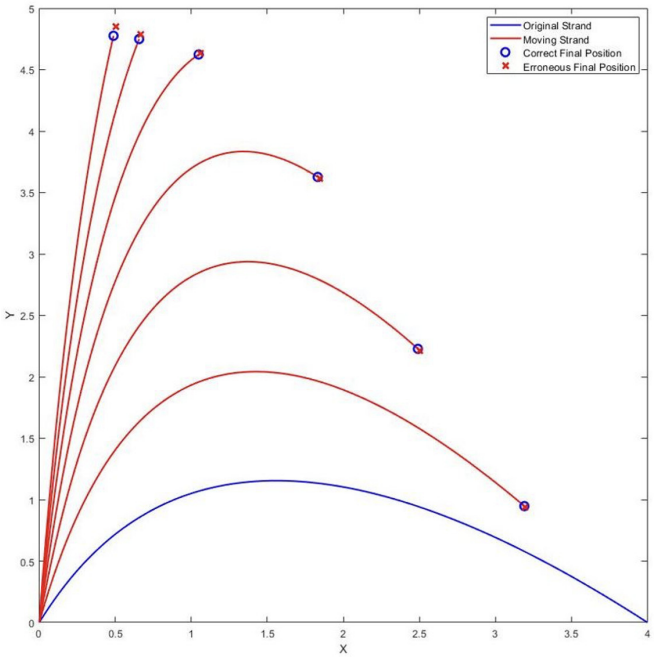

Fig. 3. An initial strand (blue) and its six subsequent movements (red). The cumulative spatial error resulting from an average MAE of 0.00091992 in the predicted t_{final} values is illustrated. Blue circles mark the true terminal positions, while red crosses indicate the erroneous positions due to cumulative prediction error.

IV. CONCLUSION

This work proposes a neural network-based approach to estimate the parameter t_{final} for cubic Bézier curves subjected to external deformation, with the goal of preserving hair strand length in dynamic 2D simulations. By generating a dataset of 7.5 million curve configurations and training a shallow ANN with high generalization capability, the study demonstrates that accurate t_{final} predictions can be achieved rapidly. The resulting MAE, although referring to the time domain rather than direct spatial error, accumulates to only 1% deviation after approximately 5 sequential displacements—suggesting that strand length is effectively preserved over long-term movement. The ANN's prediction times for up to 200k strands—comparable to the number of hairs on a typical human scalp—are under 10 seconds, and can be further reduced with more powerful hardware or model optimization. These findings support the viability of ANN-based inference for realistic and computationally efficient hair simulations.

In future work, this framework will be extended to simulate the dynamic motion of entire hair assemblies composed of Bézier strands under various external forces, where each strand's length is regulated via ANN-predicted t_{final} values. Additionally, further training on a larger dataset is planned to reduce prediction error even more, ensuring higher fidelity in long-duration and high-frequency deformation scenarios. Although this study is currently limited to 2D simulations, the architecture of the dataset generation and ANN model are fully generalizable to 3D Bezier strand representations. Future work will explore 3D vector space adaptation for each control point coordinate.

REFERENCES

[1] R. E. Rosenblum, W. E. Carlson, and E. Tripp, "Simulating the structure and dynamics of human hair," *J. Vis. Comput. Anim.*, vol. 2, no. 4, pp. 141–148, 1991.
[2] A. Selle, M. Lentine, and R. Fedkiw, "A mass spring model for hair simulation," in *ACM SIGGRAPH 2008 Papers*, 2008, pp. 64:1–64:11.
[3] B. Choe, M. G. Choi, and H.-S. Ko, "Simulating complex hair with robust collision handling," in *Proc. Symp. Comput. Anim.*, 2005, pp. 153–160.
[4] R. Bridson, R. Fedkiw, and J. Anderson, "Robust treatment of collisions, contact and friction for cloth animation," in *ACM SIGGRAPH*, 2002, pp. 594–603.

[5] D. Baraff and A. Witkin, “Large steps in cloth simulation,” in *ACM SIGGRAPH*, 1998, pp. 43–54.
[6] J. A. Amador Herrera *et al.*, “Augmented mass-spring model for real-time dense hair simulation,” *arXiv preprint arXiv:2412.17144v2*, 2024.
[7] M. Müller, T.-Y. Kim, and N. Chentanez, “Fast simulation of inextensible hair and fur,” in *Proc. VRIPHYS*, 2012.
[8] J. Hsu *et al.*, “Sag-free initialization for strand-based hybrid hair simulation,” *ACM Trans. Graph.*, vol. 42, no. 4, 2023.
[9] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, “Position based dynamics,” *J. Vis. Commun. Image Represent.*, vol. 18, no. 2, pp. 109–118, 2007.
[10] S. Hadap and N. Magnenat-Thalmann, “Modeling dynamic hair as a continuum,” *Comput. Graph. Forum*, vol. 20, no. 3, pp. 329–338, 2001.
[11] P. Volino and N. Magnenat-Thalmann, “Animating complex hairstyles in real-time,” in *Proc. ACM Symp. Virtual Reality Software Technol. (VRST)*, 2004, pp. 41–48.
[12] M. Müller, B. Heidelberger, M. Teschner, and M. Gross, “Meshless deformations based on shape matching,” in *ACM SIGGRAPH*, 2005, pp. 471–478.
[13] D.-W. Lee and H.-S. Ko, “Natural hairstyle modeling and animation,” *Graph. Models*, vol. 63, no. 2, pp. 67–85, 2001.
[14] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2007.
[15] T. Stuyck *et al.*, “Quaffure: Real-time quasi-static neural hair simulation,” *arXiv preprint arXiv:2412.10061v1*, 2024.
[16] Y. Takimoto, H. Takehara, H. Sato, Z. Zhu, and B. Zheng, “Dr.Hair: Reconstructing scalp-connected hair strands without pre-training via differentiable rendering of line segments,” *arXiv preprint arXiv:2403.17496v2*, 2024.
[17] G. Bhokare, E. Montalvo, E. Diaz, and C. Yuksel, “Real-time hair rendering with hair meshes,” in *SIGGRAPH Conf. Papers*, 2024.
[18] Q. Zhang, S. Thomas, R. Ramamoorthi, and K. Zhou, “Exploiting Budan-Fourier and Bernstein theorems for real-time collision culling of Bézier curves,” in *ACM SIGGRAPH / Eurographics High Perform. Graph. (HPG)*, 2017.
[19] N. Arslan and K. Gurkahraman, “A novel contour tracing algorithm for object shape reconstruction using parametric curves,” *Comput. Mater. Continua*, vol. 75, no. 1, pp. 331–350, 2023. doi: 10.32604/cmc.2023.035087
[20] K. Lee and S. Lee, “Hair simulation using articulated body method and Bézier curve,” in *SIGGRAPH Asia Tech. Commun.*, 2024.