IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 April 2025, accepted 8 May 2025, date of publication 12 May 2025, date of current version 20 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3569324

RESEARCH ARTICLE

Porous Models for Enhanced Representation of Saturated Curly Hairs: Simulation and Learning

JONG-HYUN KIM[1] AND JUNG LEE[2]

[1]Department of Artificial Intelligence and Design Technology, College of Software and Convergence, Inha University, Michuhol-gu, Incheon 22212, South Korea
[2]Department of Computer Engineering, Hanbat National University, Yuseong-gu, Daejeon 34158, South Korea

Corresponding author: Jung Lee (airjung@hanbat.ac.kr)

This work was supported in part by the Research Fund of Hanbat National University, 30%, in 2023; in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government (MSIT), Artificial Intelligence Convergence Innovation Human Resources Development (Inha University), 30%, under Grant RS-2022-00155915; and in part by Inha University Research Grant, 40%.

ABSTRACT Simulating the cohesion, adhesion, stiffness, and exaggeration of curls of wet curly hair or fur, expressed through the saturation-hair interaction in physics-based simulations, remains a challenging problem. Wet hair or fur tends to clump and stiffen at the ends, a common phenomenon observed in wet hair or animal fur. Additionally, while wet hair should exhibit adhesion when in contact with solids, the uneven distribution of forces in wet curly hair, manifested as noise, complicates an accurate representation of adhesion. Research into detailed porous models for wet curly hair, driven by saturation-hair interaction, is not yet extensively explored. Previous methods have manually represented wet hair or used static hairstyles for wet curly hair and fur, maintaining shape but resulting in unnatural movement due to the lack of simulation. This paper proposes methods for representing wet curly hair features: 1) curl exaggeration using locally transformed helices, 2) deformation-based cohesion that remains stable in wet curly hair, 3) level-set-based adhesion for efficiently depicting the sticky and elongated forms of wet hair, 4) dynamic stiffness for improved simulation stability, and 5) collecting a detailed synthetic dataset of curly hairs and extending the solver to represent particle movements in wet strands through learning. Experiments in various scenes demonstrate that our proposed methods more realistically represent the saturation-hair interaction compared to previous wet curly hair simulations. Unlike previous methods in which saturation caused cohesion or curls to tangle, our method stably represents porous flow at the strand level. Additionally, we propose to extend the learning representation solver through both numerical simulation algorithms and AI-based approaches.

INDEX TERMS Porous model, saturation, wet curly hair, cohesion, adhesion, stiffness, exaggeration of curly hair, local-transformed helix.

I. INTRODUCTION

Hair and fur simulations are crucial in representing the skin of digital humans and animals in detail; hence their widespread use in simulation, animation, games, and virtual reality (VR) [11]. Hair, in particular, moves independently per strand and changes state with heat or saturation. Although recent studies focusing on various physical characteristics manifested by the liquid-hair interaction have gained attention, they are mostly designed with straight hair in mind. This limits their applicability in diverse scenes, and movements like cohesion or adhesion that should be applied per strand face issues of noise and trembling due to the unstable forces of curly strands [1], [2], [3], [5], [6].

To express the various interactions that result from the absorption of moisture of a strand, Rungjiratananon et al. [1] designed a grid format capillary system to model the direction of hair moisture dispersion. Building on this method, Lin [3] utilized boundary handling techniques from SPH (Smoothed Particle Hydrodynamics) to simulate the complex interactions between fluid and hair. In particular, Lin employed a Lagrangian approach to model strand saturation,

The associate editor coordinating the review of this manuscript and approving it for publication was Songwen Pei.

effectively calculating the cohesion effects manifested in the interaction between fluid and hair. However, since previous methods were designed assuming straight hair, applying them to curly hair has limitations [1], [3], [5]. Before developing a new simulation framework in this study, we observed photographs that contained real wet curly hair to identify the subtle characteristics of wet curly hair when the strands are adjacent to each other.

The interaction between hair and liquid exhibits characteristics similar to those of porous materials. When hair is fully submerged in liquid, it acts as a porous medium, resulting in a drag effect due to the surrounding flow. In the hair-liquid interaction, the hair retains a certain amount of liquid, which flows along the hair strand before eventually falling off. Additionally, wet strands exhibit cohesion, represented by clumping, and adhesion, which occurs due to collisions with solids.

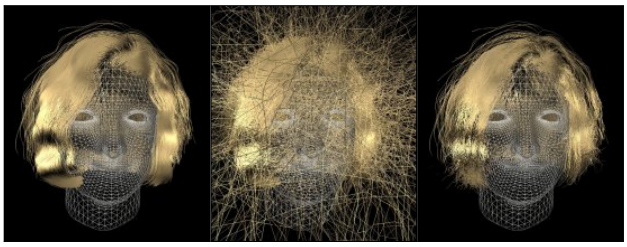
(a) Dry curly hair (b) Wet curly hair

FIGURE 1. The real appearance of dry and wet curly hair.

In Figure 1 comparing actual dry and wet curly hair, key observations include stronger clumping, stiffness, and more pronounced curliness in wet curly strands compared to dry ones. Previous methods used mainly triangular mesh-based or grid-based approaches to model the shape of wet hair [4]. However, predefined hair meshes are insufficient to simulate saturation-induced hair interactions. Kim et al. [6] applied the surface minimization force used in SPH to the wet hair cohesion, effectively representing the clumping effect between the strands. This method, designed with straight hair in mind, faces challenges in curly hair, where hair particles are irregularly distributed, reducing system stability and making it difficult to fully represent the characteristics of wet curly hair.

A. PROBLEM STATEMENT

Hair and fur models are similarly structured, often using mesh structures for simulation. Recent techniques such as MeshCNN [7] and Graph Nets [8], [9], [10] have focused on learning 3D mesh representations by considering model connectivity. However, these approaches primarily target triangular or meshless structures, leaving a gap in simulations for strand-based structures like hair and fur. Fei et al. [5] addressed this by modeling cohesion through surface tension between wet hair, submerged hair movement, and liquid



(a) Dry curly hair (b) Lin et al. [2], [3] (c) Kim et al. [6]

FIGURE 2. Simulation of wet curly hair(*short hair, high res.*) using previous methods.

drag force. They applied the liquid surface tension force for cohesion between strands and adhesion with objects, similar to Kim et al. However, this requires water simulation, increasing computational load and indicating that saturation alone inadequately models cohesion [6].

Figure 2 illustrates the results of applying the characteristics of wet hair to the simulation of curly hair using previous approaches. Due to most methods assuming straight hair, forces are applied unstably on curly strands, leading to noisy movements and the simulations not converging. In the technique used by Lin [2], [3], which uses boundary handling, instability issues arise from the early stages of the simulation (see Figure 2b). Similarly, Kim et al.'s method [6] encountered instability issues similar to those seen with Lin et al., but additionally, noisy movements propagated from the tips, where curls were strongly represented, ultimately causing the simulation to diverge (see Figure 2c).

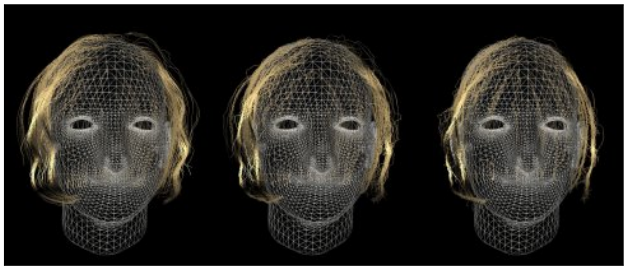
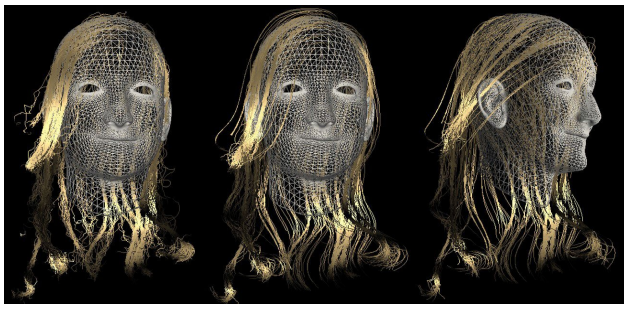


FIGURE 3. Simulation of wet curly hair(*short hair, low res.*) using Lin [2], [3].

To determine whether the issue depends on the number of strands, we conducted experiments with a more sparse hairstyle to simulate wet curly hair. As shown in Figure 3, Lin technique [2], [3] demonstrated that minor tremors in curly strands were transmitted as noisy forces, leading to tangled movements. To illustrate that this issue was not unique to this scene, we conducted the same experiment on long hair. Figure 4 shows the results of simulating wet curly hair cohesion in long hair using previous methods. Compared to Figure 3, a relatively stable result appeared; however, because it clumped only thinly, the hair was prone to tangling (see Figure 4a,b). This problem became more severe when applying movement, such as rotating the head, rather than in static scenes (see Figure 4c).



(a) Lin et al. [2], [3] (b) Kim et al. [6] (c) Rotated wet curly hair with Kim et al. [6]

FIGURE 4. Simulation of wet curly hair(*long hair, low res.*) using previous methods.

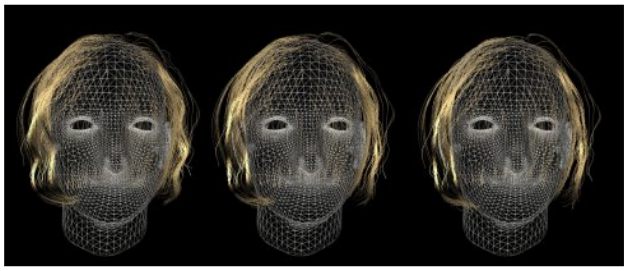


FIGURE 5. Simulation of wet curly hair(*short hair, low res.*) using Kim et al. [6].

The problem described was also present in the method of Kim et al. [6], where the intensity of the strand noise was reduced compared to Lin [2], [3], but, as shown in Figure 5, there was still observable trembling or tangling. As in other studies, this problem also appeared in long-hair simulations. In this paper, we define a new problem to mitigate these issues and our technical contributions to solve it are as follows:

1) Curl exaggeration in wet curly hair when saturated: A locally transformed helix to transfer curl styles from the root to the tip of each strand.
2) Stable cohesion in curly hair due to interaction caused by saturation: Analyzing the saturation relationships between the target strand and adjacent strands to deform positions according to cohesion.
3) Level-set-based adhesion represented by the interaction between objects and wet curly hair: Expressing adhesion and elongated movements in wet curly hair based on the level-set of the objects.
4) Dynamic stiffness based on saturation to enhance the stability of wet curly hair: Typically, various forces are applied to maintain curly hair curl shape, with additional forces for wet hair. The presence of numerous forces can reduce the stability of the simulation. Thus, this paper introduces a method for applying dynamic stiffness to each hair particle, depending on saturation.
5) Solver extensions: This paper proposes extending the solver to construct synthetic data required for learning through the suggested simulation and to represent the learning of cohesion, adhesion, and curl exaggeration in particles existing in wet strands.

II. RELATED WORK

A. HAIR DYNAMICS

One approach to individually simulate hair strands is the mass-spring system [12], which alleviates the issue of hair strands stretching through spring dynamics. However, this method can cause numerical instability during the process of mitigating strand stretching. Recently, methods have been developed to make hair inextensible [13] or to combine Position-based Dynamics (PBD) [14], [17] with the Discrete Elastic Rods (DER) framework [15], [16]. Müller et al. combined PBD with the Follow-the-Leader (FTL) approach to simulate thousands of hair strands in real time, imposing restrictions to prevent excessive stretching of hair strands even at large time steps [13]. Iben et al. proposed a technique for realistically representing curly hair or fur [18]. In this method, the hair-hair collision was managed using the Gauss-Seidel collision solver proposed by Kaufman et al. [16], and Gornowicz and Borac suggested a hybrid algorithm for improved performance and better stability [19]. The proposed method builds on this technique to model the hair interaction based on saturation.

B. POROUS MATERIALS

In the field of simulation based on physical data, methods have been proposed to model the physical mechanisms of wetting effects. These physical models are still utilized in the field of Computer Graphics: The second law of Fick [26] can be used to represent moisture permeation in homogeneous fabric materials [27]. Furthermore, Darcy's law [28] can be used to calculate the velocity of a liquid passing through porous media, using pressure drop, viscosity coefficient, and permeability, and is also applied to calculate viscous drag in numerical simulations [29].

The mixture theory was developed to represent saturated porous media observed in incompressible solids, proving theories regarding the drag and pore pressure exhibited between porous solids and liquids [30]. Recently, Borja generalized the theory of mixtures for unsaturated porous media with compressible solids, presenting a method to formulate it in a mathematical framework [31], [32]. This approach has been used in various methods for simulating two-way coupled porous media. For example, Abe et al. proposed a method to calculate the generalized Darcy equation in the Material Point Method (MPM) to simulate the creeping flow in porous soil [33]. Extending this method, Bandara and Soga introduced a way to represent porous media in large deformation [29]. Daviet and Bertails-Descoubes combined mixture theory with Drucker-Prager rheology to simulate immersed granular flows [34].

Rungjiratananon et al. [1] proposed a method using a particle-based model to absorb liquid in a porous medium, and Lenaerts et al. [20], [21] presented realistic representations of wettable materials through sponge and fabric simulations to demonstrate fluid absorption and flow. Baek et al. [22] introduced a method to realistically

represent mud on various scales using the interaction between particle-based fluids and sand. Chen et al. [23] realistically simulated wet garments considering the frictional forces arising from liquids. Um et al. [24] suggested a double layer technique to simulate the saturation of water flowing in thin shells. Fei et al. [25] realistically modeled the physical phenomena represented in the liquid-cloth interaction. This approach proposed an integrated framework that can represent the theory of mixtures and porous flow in APIC (Affine Particle-in-Cell), developing an anisotropic fabric microstructure model to represent nonlinear drag force and pore pressure.

Most hair research in computer graphics has focused on straight or wavy hair, limiting its applicability to diverse hair styles. Recently, Shi et al. proposed an isotropic hyperelastic model designed to efficiently simulate highly coiled hair with a curvature radius close to 5 mm [41]. This method operates robustly under large bending and torsion and does not require a parallel transport operator, enabling the use of large time steps. Wu et al. introduced a geometric method for generating the characteristic forms of highly coiled hair [40]. When hair is well-approximated by high-frequency helices, its visually significant features change, and their approach models three key phenomena: (1) phase locking, (2) period skipping, and (3) switchbacks or helical perversions. Darke et al. proposed a framework for representing Black hair, specifically kinky or Afro-textured hair [42]. They introduced styling guidelines for 3D models from the Open Source Afro Hair Library and presented Lifted Curls, a strand simulation method specifically designed for Afro-textured hair.

C. WET HAIR MODEL

Rungjiratananon et al. [1] proposed a grid-based approach to simulate micropermeability associated with the absorption of liquid by hair. They discretized a porous flow model within a regular Cartesian grid surrounding the hair bounding box, applying previous SPH-based porous flow techniques [20]. Lin [2], [3] utilized fluid-solid interaction techniques to simulate the absorption and dispersion exhibited in the interaction between hair and water, proposing an algorithm that could represent the adhesion and cohesion that occurs between wet strands. They modeled fluids using SPH and used the DER model for hair to express two-way fluid-hair interactions. Furthermore, they deactivated the hair-hair adhesion force for underwater hair to accurately represent fully submerged areas.

As mentioned above, Fei et al. [5] extended the liquid-cloth interaction algorithm to liquid-hair interactions, realistically representing the interactions exhibited in wet hair. This method introduced inertial forces to express the detailed movement of liquid traveling and diffusing along hair strands, and newly modeled cohesion due to surface tension between wet hair, movement of submerged hair, and drag force exhibited between liquids. However, these methods assume straight hair, and in curly hair, the irregular dispersion of hair particles can result in unstable force calculations, affecting hair's cohesion and adhesion. This paper aims to present an efficient method to address these issues, examining three commonly used approaches to represent characteristics of hair interaction caused by saturation: 1) Lin method [2], [3], which represents wet hair similarly to SPH's boundary handling, 2) Kim et al. and Fei et al.'s methods [5], [6], which utilize SPH's surface tension force to represent wet hair. Among these, Fei et al. explicitly require particle-based fluids, and thus this paper compares with the approach of Kim et al.

III. PROPOSED METHOD

A. CURLY HAIR DYNAMICS

In this paper, we utilize the technique proposed by Iben et al. [18] to represent curly hair, and in this section, we describe three spring-based methods for calculating the dynamics of curly hair.

1) STRETCH SPRING

The stretch spring type typically defines a hair model using a set of particle positions connected by linear springs. Each hair comprises a set of current particle positions $P = \{p_0, \ldots, p_{N-1}\}$ and the initial positions of the particles $\bar{P}$, where the subscript 0 denotes the initial quantities. The current velocity of the particles is represented as $V = \{v_0, \ldots, v_{N-1}\}$, and the edges connecting the hair particles are denoted as $e_i = p_{i+1} - p_i$. The linear spring force for the particle i is calculated as follows:

$$f_s(k_s, c_s) = k_s(\|e_i\| - \|\bar{e}_i\|)\hat{e}_i + c_s\left(\Delta v_i \cdot \hat{e}_i\right)\hat{e}_i \quad (1)$$

where k_s and c_s are the spring constant and damping constant corresponding to stretch, respectively, and Δv_i is defined as $\Delta v_i = v_{i+1} - v_i$. Since a spring connects two particles, each spring force is applied equally but in opposite directions to both particles. To represent hair without using stiff springs, we limit the strain of an edge, similar to the method proposed by Selle et al. [12]. During the calculation of damping, if Δv_i^2 exceeds a threshold, we revert from the root to the tip to limit the velocity of the particles.

2) BENDING SPRING

The bending spring used by Iben et al. [18] is similar to the elastic rod model. $\Lambda = \{\lambda_0, \ldots, \lambda_{N-1}\}$ represents elements related to hair, such as the positions and velocities of particles. In this method, a smoothing function based on the Infinite Impulse Response (IIR) is defined as follows: $d_i = \varsigma(\Lambda, \alpha)_i$. This is a recursive filtering function that combines input results with previous results to produce new results. Here, α, the amount of smoothing defined along the curve, is set to be greater than zero, and the vector d_i is recursively calculated as follows.

$$d_i = 2(1-\beta)d_{i-1} - (1-\beta)^2 d_{i-2} + \beta^2(\lambda_{i+1} - \lambda_i) \quad (2)$$

β is defined as $\beta = \min(1, 1 - \exp(-l/\alpha))$, where l is the rest length of the hair strand. In this process, if the coefficients

are initialized as follows: $d_2 = d_1 = \lambda_1 - \lambda_0$, then the equation is reduced to $d_0 = \lambda_1 - \lambda_0$ at $i = 0$.

ς is a set of positions, by recursively adding vectors from the fixed root hair particle, the hair strand can be recalculated, and the new positions are calculated as follows.

$$p_i' = p_{i-1}' + d_{i-1} \tag{3}$$

where $p_0' = \lambda_0$, which represents the root particle of the hair strand. To maintain the helical shape of curly hair, it is necessary to stably control the bending between the rest of the hair and its current pose, which requires the calculation of bending springs. Additionally, to calculate the force for bending, spring forces are added between the current shape of the strand e_i and the target vector t_i.

$$f_b(k_b, c_b) = k_b(e_i - t_i) + c_b\left(\Delta v_i - \left(\Delta v_i \cdot \hat{e}_i\right)\hat{e}_i\right) \tag{4}$$

where k_b and c_b represent the spring and damping constants for bending, respectively, and t_i is the target position calculated considering the initial curly shape of the hair strand. It is recommended to refer to the previous method [18] for a detailed calculation process of t_i.

3) CORE SPRING

When simulating curly hair, it is challenging to meet the designer's requirements with only stretch and bending springs. Stretch and bending springs can maintain curly hair, but they are not sufficient to control the form in which the hair bends. To ensure that curly hair remains naturally curly even when external forces are applied, we introduce core springs (see Equation 5).

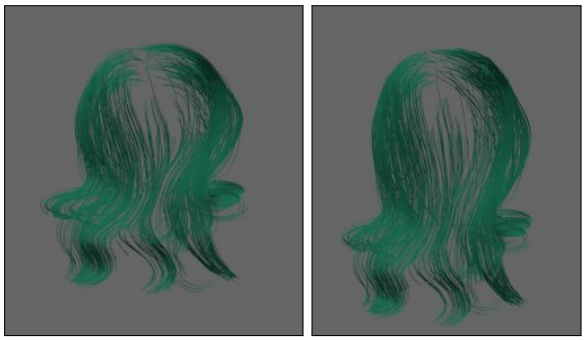
$$f_c(k_c, c_c) = k_c\left(\|b_i\| - \left\|\bar{b}_i\right\|\right)\hat{b}_i + c_c\left(v_i \cdot \hat{b}_i\right)\hat{b}_i \tag{5}$$

where k_c and c_c represent the spring and damping constants for the core, respectively, and v_i is the velocity after applying a smoothing function: $v_i = \zeta(V, \alpha_c)$. $\bar{b}_i$ is the core force of the curly strand calculated from the initial position of hair particles: $\bar{b}_i = \zeta\left(\bar{P}, \alpha_c\right)$, and b_i is the core force calculated from the current position of hair particles: $b_i = \zeta(P, \alpha_c)$.

Figure 6 shows the results of representing curly hair using stretch, bending, and core springs as previously described. It demonstrates that the curls remain intact even in scenes with vertical movement. In the next section, we will explain how to extend the solver to the dynamics of wet curly hair, allowing for a detailed representation of the characteristics of wet strands.

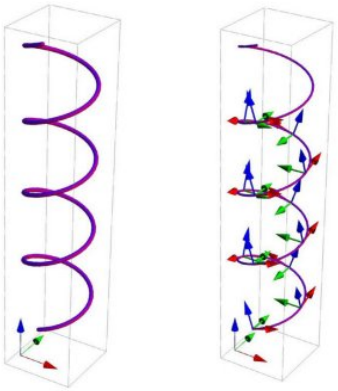
B. EXAGGERATED CURLY MOTION IN WET HAIR

The hair model used in this paper uses the simulation of curly hair described previously. Similarly to the previous method, it constructs three types of spring model and designs a new force to control the curl shape in a helix structure model, thus modeling the state change of curls. To model the state changes due to moisture absorption, which are not represented in the hair's original curl shape, a 3D helix structure model is calculated in the preprocessing stage (see Equation 6). It is



(a) (b)

FIGURE 6. Curly hair simulations.



(a) (b)

FIGURE 7. A helix defined by the parametric equation.

crucial that the number of particles in the hair particles and the helix structure model match.

$$x(t) = r\cos(t), y(t) = r\sin(t), z(t) = pt \tag{6}$$

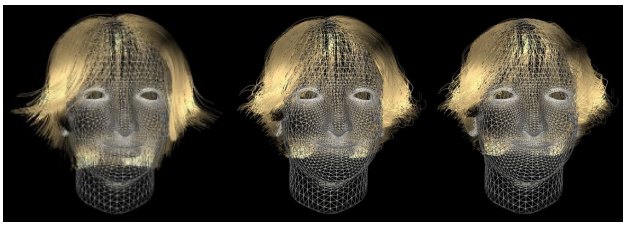
The above equation represents the result of generating a helix with radius r in the Z-axis direction as the parameter t increases (see Figure 7).

As shown in Figure 1, wet curly hair tends to express curls more strongly, so this paper represents the characteristics of wet hair based on helix patterns. The current position of the hair particles is defined as $P = \{p_0, \ldots, p_{N-1}\}$, and the position of the particles in the helix structure model is defined as $H = \{h_0, \ldots, h_{N-1}\}$. During simulation, the vectors from the root to the tip of hair and helix particles, $p^{\forall} = p_{N-1} - p_0$, $h^{\forall} = h_{N-1} - h_0$, are normalized to calculate the matrix M, which transforms the helix model to the current direction of the hair (see Equation 7).

$$M = I + \Psi + 2\Psi\left(\frac{1-c}{\text{pow}(\|\gamma\|, 2)}\right) \tag{7}$$

$$\text{where, } \Psi = \begin{bmatrix} 0 & -\gamma.z & \gamma.y \\ \gamma.z & 0 & -\gamma.x \\ -\gamma.y & \gamma.x & 0 \end{bmatrix} \tag{8}$$

where $\gamma = p^{\forall} \times h^{\forall}$ and $c = \gamma = p^{\forall} \cdot h^{\forall}$. Using this rotation matrix M, the edges of the helix model align with the current



(a) Original curl (b) Weak saturation (c) Strong saturation

FIGURE 8. Comparing the exaggerated forms of curls (b, c) according to saturation.

(a) Original curly hair (b) Static pose (c) Rotated wet curly hair

FIGURE 9. Testing cohesion in wet curly hair using the method of Lin [2], [3].

pose of the strand. Consequently, the force f_w that models the change in the curly state is calculated as Equation 9.

$$f_w = M\left(h_{i+1} - h_i\right) w_h w_s \tag{9}$$

where $(h_{i+1} - h_i)$ represents the local helix direction calculated based on the helix's particles, and M is the transformation matrix for rotating the current strand direction to align with the helix direction. Additionally, w_s and w_h represent the saturation and weight for the helix model, respectively. By adjusting these values, a variety of results can be achieved.

Figure 8 compares the original form of curly hair with the results of the curl exaggeration applied using our method. The saturation was strongly set at the tip of the strand, resulting in an exaggerated expression of the curl's form. Figure 8b shows exaggerated curls with relatively weak saturation, and Figure 8c presents curl exaggeration experimented with strong saturation. Compared to the original curl form, it effectively demonstrates the natural appearance of wet curls.

C. STABLE COHESION FORCE IN WET CURLY HAIR

In this section, we discuss methods for stably representing cohesion in wet curly hair. As described in the problem statement, previous methods attempted to stably represent the cohesion exhibited in wet hair [2], [3], [6]. These methods modeled the clumping that occurs when hair gets wet, based on the distribution and saturation of adjacent hair particles. However, as most were designed based on straight hair, they do not adequately represent cohesion in curly hair due to the irregular distribution of adjacent hair particles.

(a) Original curly hair (b) Static pose (c) Rotated wet curly hair

FIGURE 10. Testing cohesion in wet curly hair using the method of Kim et al. [6].

Figures 9 and 10 show the results of testing cohesion in wet curly hair using previous methods. In comparison to the input hair's form, cohesion is relatively well represented in wet hair. However, there is a tendency for hair to clump in thin forms, often leading to tangling issues, especially when the hair rotates, causing the tangles to tighten, and ultimately making the simulation unstable. The fundamental reason for these issues is as follows: The distribution of adjacent hair particles is irregular, causing forces to be applied as irregular noise, leading to unstable movements. This problem worsens when the hair is wet, especially when forces such as rotation are involved, where the noise becomes more pronounced. To address this problem, we calculate the position of wet hair particles, p_i^w, using Equation 10.

$$p_i^w = \left[\frac{\sum_j \left(p_j\sigma_j + p_i\left(1-\sigma_j\right)\right)\frac{m_j}{\rho_j} W\left(p_i - p_j, h\right)}{\sum_j W\left(p_i - p_j, h\right)}\right]\lambda \tag{10}$$

where p_i and p_j represent the positions of the target hair particle and adjacent hair particles, respectively, while m_j and ρ_j denote their mass and density. Additionally, σ_j is the saturation of each particle, and finally, λ is a value representing tightness, which is set to 0.85 in this paper. As a result, by separately storing positions for cohesion for each strand's particles, a stable clumping effect due to saturation was created. Since p_i^w does not affect the movement of the original curly hair, cohesion is stably represented. Equation 10 is designed to represent wet curly hair, determining the position for expressing wet hair in a position-based manner similar to PBD, rather than updating from force. A larger value of σ is designed to apply a stronger cohesion force, and as the saturation σ_j of adjacent hair particles increases, p_i^w is placed closer to p_j. When σ_j is small, p_i^w tends to remain closer to p_i, resulting in a weaker cohesion force. Consequently, the position p_i^w represents the attractive force between hair particles depending on saturation, effectively creating a clumping effect where wet hair particles gather. The magnitude of this clumping effect is controlled by λ.

Figure 11 shows the results of representing cohesion using the method proposed in this paper. Compared to the original hair model, cohesion was naturally represented without tangling (see Figure 11b), and when compared to previous

methods, wet curly hair was well represented without strands tangling into thin clumps. In particular, even when the head rotates, our method successfully and stably represented cohesion due to saturation, unlike the previous method in which tangled hair appeared (see Figure 11c).

(a) Original curly hair **(b)** Static pose **(c)** Rotated wet curly hair

FIGURE 11. Testing cohesion in wet curly hair using our method.

D. STABLE ADHESION FORCE BETWEEN WET CURLY HAIR AND OBJECTS

In this section, we introduce a method for stably representing adhesion between wet curly hair and objects. While the previous method added boundary particles to objects to represent force in a kernel form, we use a level-set approach to more efficiently calculate adhesion. Handling of hair collisions was processed using the previous method [35], and the adhesion force F_i^{ad} is calculated using the following equation under the condition that satisfies the level-set value $\phi < 0$ for collisions (see Equations 11 and 12).

$$F_i^{ad} = \sigma_i m_i \upsilon A\left(|\phi|\,\nabla\phi\,|\phi|\right) \tag{11}$$

$$A\left(r\right) = \frac{\frac{0.007}{h^{3.25}}\left(-\frac{4r^2}{h} + 6r - 2h\right)}{|\phi|} \tag{12}$$

where σ_i represents the saturation of the hair particle, and υ is a weight, which in this paper is set to 1,300. By adjusting these values, the extent to which strands elongate due to adhesion can be controlled. The form of the adhesion kernel $A\left(r\right)$ is designed to reflect the manner in which strands stretch due to adhesion from saturation when wet curly hair come into contact with an object.

E. DYNAMIC STIFFNESS IN WET CURLY HAIR

To represent curly hair, various constraint forces are included, and the additional forces added to depict wet effects make the simulation unstable. To mitigate these numerical issues and stably represent curly hair, this section introduces a dynamic stiffness method considering saturation. The stiffness model is applied in two forms: one is a global stiffness, ξ, applied to the entire hair, and the other is a saturation-considered stiffness model, ζ_i. ξ is relatively easy to apply as a constant ($v_i \leftarrow v_i\xi$), but the stiffness based on saturation is calculated using Equation 13.

$$v_i = v_i\varsigma_i, \text{ where } \zeta_i = \sigma_i\Upsilon_0 + (1-\sigma_i)\,\Upsilon_1 \tag{13}$$

(a) Without dynamic stiffness

(b) With dynamic stiffness (our method)

FIGURE 12. Comparison results with dynamic stiffness.

where v_i and σ_i represent the velocity and saturation of the hair particle, respectively, while Υ_0 and Υ_1 are the minimum and maximum magnitudes of stiffness defined by the user. In this paper, the value of ξ is set to 0.999995, and Υ_0 and Υ_1 are set to 0.9999 and 1.0, respectively.

Figure 12 shows a scene in which a head with wet curly hair is rotated left and right to experiment how the dynamic stiffness proposed in this paper affects simulation stability. While the previous method by Iben et al. [18] depicted hair scattered by strong external forces (see Figure 12a), our approach, which applies dynamic stiffness, effectively represents the stiffness induced by saturation and also improves the stability of the simulation against external forces (see Figure 12b).

IV. SOLVER EXTENSIONS

In this section, we introduce a method to extend the solver by training a neural network to represent wet curly hair, as previously discussed: 1) Train artificial neural networks to transform the geometry image of dry curly hair (strands with applied cohesion, stiffness, and curl exaggeration) into the geometry image of wet curly hair. 2) Considering saturation, calculate a nonlinear transform that can convert the 2D calculated geometry image of hair into 3D, thereby restoring it to a 3D strand shape. 3) Finally, experiment to determine whether it is possible to efficiently represent wet curly hair based on learning representation by training simple neural networks to learn the level-set-based adhesion force expressed in the hair-solid interaction. As a result, we build a dataset for wet curly hair using the simulation techniques described earlier, demonstrating the capability to easily represent the features of wet hair without the need for complex mathematical or physical equations. The neural networks are trained independently to infer cohesion, stiffness, and curl exaggeration, and to infer adhesion from

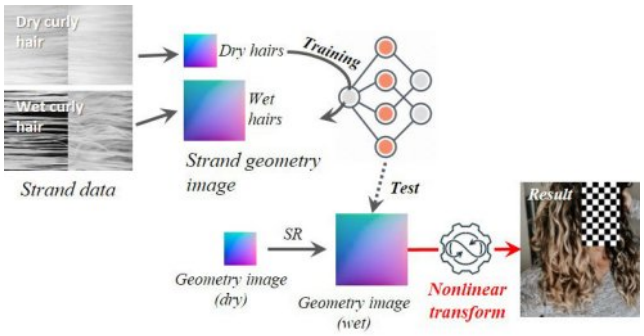


FIGURE 13. Learning representation overview.

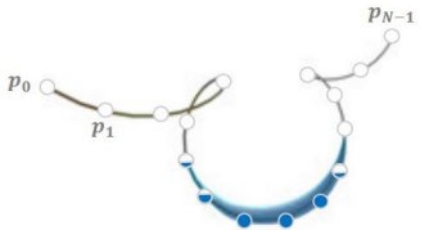


FIGURE 14. Saturation state in a curly strand (p_0: root hair particle, p_{N-1}: tip hair particle).

wet hair particles. Training data undergoes preprocessing suitable for learning before being applied to the network, and after completion of training, the test results for each model are visualized. As mentioned earlier, the irregular distribution of particles in curly strands also affects the learning representation, therefore training is carried out according to the recently proposed previous method [36].

In this paper, we adopt a previous approach that applies super-resolution (SR) to hair by converting only the root particles of each strand into a geometry image and using a nonlinear transform, rather than learning from all particles of each strand [36]. The algorithm overview proposed in this paper is shown in Figure 13. During the training stage, data pairs consisting of dry and wet curly hair are created and converted into geometry images for use in training with a Convolutional Neural Network (CNN). In the run-time stage, a CNN-based SR network is used to synthesize geometry images composed of wet curly hair, and a nonlinear transform is applied to reconstruct the hair data from 2D to 3D. The detailed process will be further discussed in the following subsections.

A. LEARNING FOR CURL EXAGGERATION AND HAIR-HAIR COHESION

The proposed method generates a strand geometry image that stores the positional information of other particles relative to the fixed root particle (the first particle on the strand) of the scalp hair. As mentioned in the previous method, the movement of hair and fur is nonlinear, making it difficult for conventional SR networks to fully capture their motion, especially in the case of wet curly hair. In 2D image space, the color exhibits continuity among neighboring pixels, appearing as similar colors. Therefore, training all particles' positions of hair using a neural network can lead to inaccurate movements due to noise. To alleviate this problem, we propose a novel approach of applying a nonlinear transform when reconstructing a 3D strand from a 2D image. This method is similar to the approach of Kim et al. [36], and in this paper we extend the solver by generating geometry images and modifying the nonlinear transform process to learn and represent characteristics of wet curly hair such as cohesion and curl exaggeration.

To mitigate the mentioned noise issue, this paper proposes a solution that includes only the root particle of each strand, considering its saturation, in the network training operations. By focusing on the root particle, which serves as a representative anchor point, the network training process becomes more robust and less sensitive to pixel-level discrepancies. Due to the nature of hair, the movement of the strands is predominantly influenced by the root particle, which is connected to the scalp. In this paper, the following equations are used to calculate the geometric image of the strand (see Equations 14 and 15): 1) A function $\gamma_{h\to g}$ that converts the set of root particles of hair into a geometry image, where the approximated position of the root particle, $\bar{p}$, is calculated by using weight values proposed in this paper. 2) A function $\gamma_{g\to h}$ that performs the reverse operation, converting the geometry image back into hair strand representations.

$$\gamma_{h\to g} = \frac{\bar{p} - b_{min}}{\|b_{max} - b_{min}\|} \tag{14}$$

$$\gamma_{g\to h} = c_i^{rgb} \|b_{max} - b_{min}\| + b_{min} \tag{15}$$

where $\bar{p}$, b_{min}, and b_{max} represent the approximated position of the root particle of the strand and the minimum and maximum positions of the simulation space, respectively. c_i^{rgb} represents the color $[r, g, b]$ obtained by converting a 3D coordinate $[x, y, z]$ to RGB. $\bar{p}$ is calculated considering the weight of saturation, and unlike the previous method that only handles the position of the root hair particle [36], this paper approximates the saturation of the root particle by taking a weighted average of the entire saturation constituting each strand.

Figure 14 depicts the saturation of a single strand, illustrating that saturation can occur not only at the root particle but also throughout other parts of the strand. Consequently, in this paper, we calculate the saturation of all the hair particles that make up the strand as follows to calculate $\bar{p}$ (see Equation 16).

$$\bar{p}_i = p + \frac{\sum_j \nabla W\left(p_i - p_j^s\right)\sigma_j}{\sum_j W\left(p_j - p_i^s, h\right)} \tag{16}$$

where σ_j represents the saturation of the hair particle ($\sigma \in [0, 1]$). The equation ultimately simplifies to only considering the position of the root particle in the case of dry hair, whereas for wet hair ($\sigma > 0$), it uses the position of hair particles p_i^s

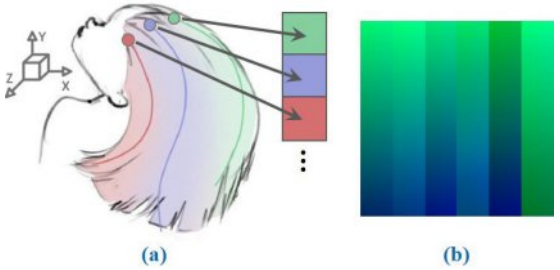


FIGURE 15. An example of a strand geometry image with $\gamma_{h\to g}$: (a) 3D to 2D transform, (b) a strand geometry image obtained by calculation.

located in adjacent strands to calculate $\bar{p}$, the approximated position of the root particle as influenced by saturation.

By applying Equation 14, all root particles of the strand can be transformed into RGB color space, resulting in a smoothly represented geometry image that appears to have a color gradation (see Figure 15). Consequently, since the position of the root particle is calculated within the simulation space defined by $b_{min,max}$, it is limited to values between 0 and 1. Figure 15b shows the color representation obtained by multiplying the transformed coordinates (x, y, z) by 255. In this method, the color is represented as integers for visualization purposes only, while real values were used for actual calculations.

B. NEURAL NETWORK

For training, we utilized a neural network technique from a previous method [36], and in this section, we briefly describe that process. After obtaining the data on the root particles of the hair, indicated as $\{F^0, F^1, \ldots\}$ using physics-based simulation, the method described in the previous section is used to generate the geometry images for dry curly hair, represented as $\{\delta_d^0, \delta_d^1, \ldots\}$, and for wet curly hair, represented as $\{\delta_w^0, \delta_w^1, \ldots\}$. δ represents the strand geometry image described above, and these images are divided into patches before being fed into the training network. Given the training data, the goal of this paper is to find a mapping function $f(x)$ that minimizes the loss between the predicted values ν_s and the ground truth (GT) δ_w. The loss function used in this paper is the Mean Squared Error (MSE) between the predicted geometry image and the GT data. Our goal is to minimize the value of $\nu_s = f(x)$.

In Super-resolution CNN (SRCNN) [37], the use of many weighted layers requires a considerable amount of memory, which limits the construction of deep networks. To alleviate this issue, we use residual learning to train and test the geometry image data of wet curly hair. The residual map of the geometry image is calculated as follows: $r = \delta_w - \delta_d$. In the original SRCNN method, the loss function is $\frac{1}{2}\|\delta_w - f(x)\|$. However, in this paper, since we aim to predict the residual map, the final loss L is calculated using Equation 17.

$$L(r, x) = \frac{1}{2}\|r - x\|^2 \tag{17}$$

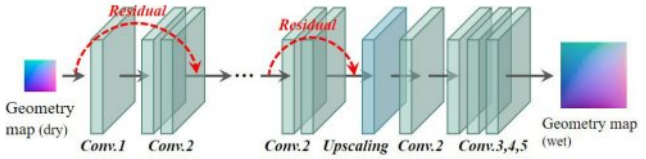


FIGURE 16. VGG19 neural network structure (red arrow: residual process).

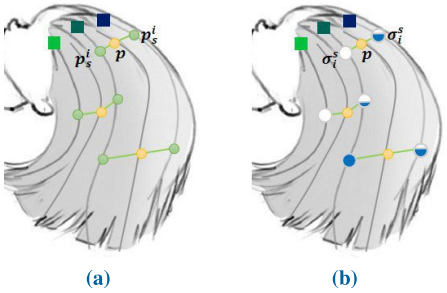


FIGURE 17. Restoring strand: (a) structure between target hair particles and neighboring hiar particles, (b) structure between target hair particles and neighboring saturations.

where r represents the residual, and x is the value obtained from $f(x)$. During network training, the loss layer used to measure loss incorporates three components: residual estimation, δ_w, and δ_d. The loss is calculated as the Euclidean distance between the restored map and δ_w, where the restored map is the sum of the network's input map and output map.

The neural network is modeled on CNN and is structured as depicted in Figure 16. This network architecture is designed using a residual learning approach, where the feature map obtained after the first convolution operation is added back into the results of two subsequent convolutional operations. This approach mitigates the error lost due to convolution operations through residual learning. In this paper, this process was repeated 10 times, with each iteration performing two convolutions, resulting in a total of 20 convolution operations performed throughout the process. Initially, the output of the first convolution is added, but in subsequent iterations, the previous output is recursively added. Subsequently, the size is increased by a factor of two through upscaling, followed by four additional convolution operations before completing the process. The resulting geometry image of a strand obtained in this way is used to represent wet curly hair.

C. NONLINEAR RECONSTRUCTION OF GEOMETRY IMAGE TO HAIR

In this section, we describe the process of reconstructing 3D hair from a geometry image. As mentioned earlier, due to the nonlinear movement of hair, accurately representing wet curly hair solely with a network-based approach is extremely challenging. In the field of artificial intelligence, various activation functions and Graph Neural Networks (GNNs) are being chosen to tackle nonlinear problems. However, as we have seen, noise can occur in the movement of hair particles, which is why this paper uses a geometry image combined

with a nonlinear transform to construct a super-resolution (SR) process based on wet curly hair, thereby resolving this issue. In this process, the upscaled geometry image obtained through the network is used to generate the SR of wet curly hair and also acts as a weight for representing nonlinear motion.

In this paper, the super-resolution (SR) process involves representing newly generated wet hair particles paired with a geometry image and four input curly strands. Essentially, the network generates one virtual strand per quad, each quad consisting of four hair samples or input hairs. Consequently, a quad-shaped network configuration is utilized to generate one virtual strand for each set of four hair samples. The particles of the new hair are computed as Equations 18 and 19.

$$p = p^* + \frac{\sum_i^N \sigma_i^s \nabla W\left(p_i - p_i^s, h\right) \gamma_i^s}{N}, \text{where } p^* = \frac{\sum_i^N p_i^s}{N} \tag{18}$$

$$\gamma_i = \frac{\left\| c^* - c_j^s \right\|}{N} \tag{19}$$

In this paper, N represents the number of adjacent strands, and given that hair strands are modeled in a quad shape, it is set to 4. p^*, c^*, and σ_i^s, respectively, denote the average position of adjacent hair particles, the average color of the geometry image, and the saturation. In the above equations, γ_i is a weight used in the interpolation calculation, which can be derived from the geometry strand image computed by the network. Managing adjacent hair particles appropriately is crucial in this process. Hair particles maintain line-shaped connectivity information for each strand. Hence, this paper uses the information on the particles from adjacent hair strands, denoted as p_i^s. Here, p^s denotes an adjacent hair strand, and in this context, includes four neighboring hair strands. Consequently, p_i^s represents the ith hair particle that constitutes the strand s. The formula utilizes the geometry image associated with the root particle and leverages the geometry information and saturation of adjacent hair particles. In this process, σ_i^s influences ∇W, which means that higher saturation levels intensify the clumping effect during the reconstruction phase.

Figure 17 illustrates the process described earlier, where the green and yellow particles shown in Figure 17a represent the adjacent hair particles and the hair particles generated by the network, respectively. The small squares at the top of the figure depict the colors of the strand geometry images calculated from each root particle. Essentially, the movement of the newly generated strand through super-resolution (SR) is designed to be influenced by adjacent strands in the same line. Furthermore, the saturation held by the adjacent hair particles is also interpolated in a similar manner to determine their positions (see Figure 17b). Real hair is affected by physical interactions such as friction and cohesion with neighboring strands. In this paper, we consider these factors and reconstruct strands from the strand geometry images of adjacent hair.

D. LEARNING FOR HAIR-OBJECT ADHESION FORCE

In this section, we introduce a method for learning the representation of adhesion forces expressed in interactions between wet hair and objects using a neural network. The proposed model does not directly infer the velocity v_i of the hair particles for the current frame but rather uses artificial intelligence to infer the adhesion force F_i^{*ad} of the hair particles. The input feature vector includes four data points: a flag indicating whether there is a collision between the hair particle and the object, v_i, ϕ, and F_i^{ad}, as well as information about the directly connected particles p_{i+1} and p_{i-1}. Consequently, a total of 12 pieces of data per particle are used as training data. Although the training data includes all hair particles, only a small fraction of these particles actually exhibit adhesion forces. Using these data as-is could potentially bias the learning process, making it difficult to accurately infer the adhesion forces of hair particles. To mitigate this issue, the paper proposes equalizing the number of particles that have experienced collisions with those that have not, through a 1:1 sampling process. This sampling aims to balance the data between the classes in the classification model, preventing poor learning outcomes due to imbalanced data counts. For example, if there are 350 particles that exhibit adhesion and 900 particles that do not, we would adjust the number by randomly sampling the 900 to match the 350, thus achieving a 1:1 ratio. This balanced approach helps ensure more effective and fair training outcomes for the neural network model.

Using the refined data from the process mentioned earlier, we train a network to model hair particle adhesion (see Figure 18). The hair-object adhesion network model is a simple neural network consisting of five layers. We utilize the Leaky ReLU activation function to enhance the model's ability to learn nonlinearities effectively. For updating weights, the cost function employed is the Least Squared Error (LSE), which is commonly used for regression problems (see Equation 20).

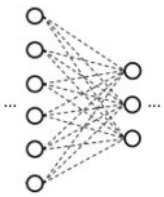


FIGURE 18. Adhesion network.

The training data is processed through a network architecture that begins with an input layer and is followed by three hidden layers and an output layer. The first hidden layer contains 6 neurons, the second has 5, and the third consists of 4 neurons. The output layer is designed to infer the adhesion force, which is the final goal of the network. Adam was used as the optimizer and a 9,000 epoch optimization was performed. During this extensive training process, the loss

TABLE 1. Details of the strand convolutional neural network architecture.

Attribute	ConvNet r1	ConvNet r2	ConvNet r3	ConvNet r4	ConvNet r5	ConvNet r6	ConvNet r7
Transpose(...)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	–
#x(w,h,d)	(8,8,64)	(16,16,64)	(32,32,64)	(64,64,64)	(128,128,64)	(256,256,64)	–
concat(...)	(input x, ConvNet5)	(input x, ConvNet4)	(input x, ConvNet3)	(input x, ConvNet2)	(input x, ConvNet1)	–	–
#x(w,h,d)	(8,8,64+512)	–	–	–	–	–	–
[weight],[bias]	5×5×64,64	5×5×64,64	5×5×64,64	5×5×64,64	5×5×64,64	5×5×64,64	5×5×3,3
Num. ConvNet	ConvNet r1-4	ConvNet r2-4	ConvNet r3-4	ConvNet r4-2	ConvNet r5-2	ConvNet r6-2	ConvNet r7-1
#x(w,h,d)	(8,8,64)	(16,16,64)	(32,32,64)	(64,64,64)	(128,128,64)	(256,256,64)	(256,256,3)

was measured at 0.15

$$L = \sum_{i=0} \left\| F_i^{ad} - M_i \right\|^2 \tag{20}$$

where M_i represents the output of the adhesion network. This means that the network infers by applying the differences in adhesion force to the cost function to update the weights accordingly. After obtaining the adhesion force F_i^{*ad} for the hair particle through the adhesion network, the position of the hair particle is updated using Euler integration.

E. IMPLEMENTATION DETAILS

1) CURLY HAIR DYNAMICS

The experiments detailed in this paper leveraged a robust computing environment: an Intel i7-7700k CPU at 4.20GHz, 32GB of RAM, and an NVIDIA GeForce GTX 1080 Ti Graphics Card. The dynamics of curly hair was simulated and accelerated using a GPU-based approach from a previous method [18], allowing the generation of data representing both straight and curly hair. Additionally, the USC-HairSalon dataset was utilized to represent a variety of hairstyles, enhancing the diversity and realism of the simulations. To implement solid-hair collision handling, a level-set-based collision detection and response technique [35] was employed. Using the techniques mentioned above, the necessary data was collected and, on the basis of these data, both the training and testing phases were conducted.

2) VGG19 NETWORK ARCHITECTURE FOR SOLVER EXTENSIONS

We utilized a neural network to learn the strand geometry image and proposed a method to transform the inferred geometry image into a 3D hair strand using a nonlinear transformation, thus effectively representing wet curly hair. The hyperparameters chosen for the training process were meticulously selected to optimize the learning outcomes: the network architecture was designed with a batch size of 128, a learning rate of 1e-4, and a stride of 1. In this research, we used the VGG19 neural network architecture to design our network. The details of developing the VGG19 network will be elaborated in this section. The VGGNet model is structured in two major phases: the VGG network and the reconstruction phase. For example, if we consider an input image as a 3-channel image with a resolution of 128 × 128, the input **x** would be represented as follows: **x** (128, 128, 3).

ConvNet1, as previously described, consists of two ConvNet layers, using weights of size (5 × 5 × 64) and biases of 64. The output dimensions of ConvNet1 are (64 × 64 × 64) in terms of width, height, and depth. The activation function used here is ReLU. The output from ConvNet1 serves as the input to ConvNet2, allowing for a sequential and structured flow of data through the network. The general architecture of the VGGNet model is depicted in Figure 19. In this figure, the labeling such as ''ConvNet 1-2'' indicates that ConvNet1 is composed of two ConvNet layers, and ''ConvNet 5-4'' indicates that ConvNet5 consists of four ConvNet layers.

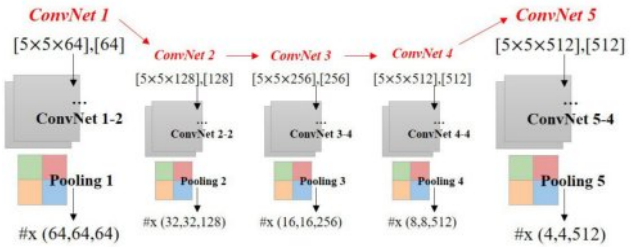


FIGURE 19. Strand convolutional neural network architecture (input: x(128,128,128), output: x(4,4,512), [weight][bias], #x(width,height,depth)).

The reconstruction phase involves restoring the output of the VGG network using transpose convolution operations (see Figure 20). Detailed specifications from ConvNet r1 to ConvNet r7 during the reconstruction phase are set as indicated in Table 1. The input to the neural network is half the size of the original map. Future additions to this reduced map, particularly the residual map, are scaled up to match the output size of the reconstruction, ensuring consistency and continuity in the final image dimensions. This network was implemented using TensorFlow [38] and the Adam optimizer (Adaptive Moment Estimation) is used.

3) NONLINEAR TRANSFORM

Nonlinear transform is required when converting a 3D hair strand into a 2D geometry image and reconstructing it back to a 3D strand. In the reconstruction process of the geometry image, which has undergone the SR step, into 3D spatial data, unlike previous cloth simulation [44], [45], hair

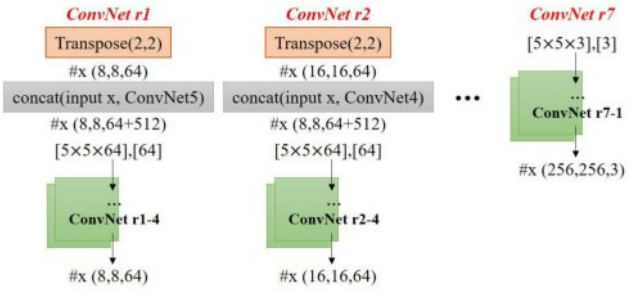
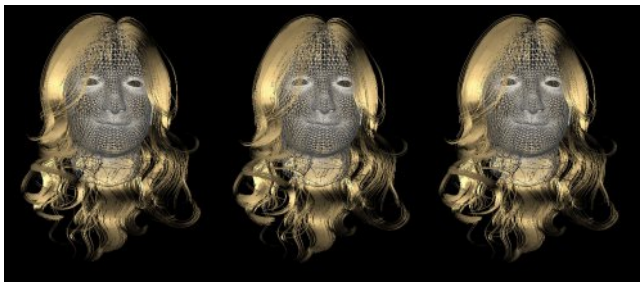
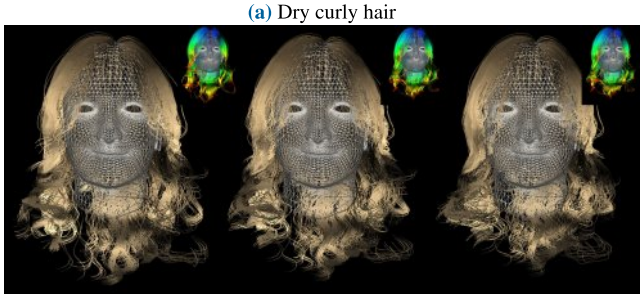


FIGURE 20. Feature reconstruction (input: x(4, 4, 512), output: x(256, 256, 3)).

exhibits elastic movement per strand and is also influenced by the motion of surrounding hair particles. The nonlinear transform reconstructs 3D positions from the geometry image by considering not only the nearest neighbor information of hair particles but also the positions of hair particles whose strand root positions are close. Without this process, noise in the hair particles may lead to entangled movements. In this study, we adopt a nonlinear transform process previously shown to successfully represent hair through learning-based methods.



(a) Dry curly hair



(b) Wet curly hair (our method)

FIGURE 21. Quality comparison between (a) dry hair and (b) our method. Rainbow color(red: strong saturation, blue: weak saturation) represents the saturation of wet hair (scene1).

V. EXPERIMENT AND RESULTS

In our study, we proposed methods to reliably represent the porous flow of wet curly hair caused by saturation, including mechanisms such as hair-hair cohesion, hair-object adhesion, curl exaggeration, and dynamic stiffness, all of which perform well even in curly hair simulations. Additionally, we extend the solver to enable the learning representation of these phenomena through artificial intelligence. The proposed methods are designed to enhance both the efficiency and visual quality of the hair simulations. To validate the effectiveness of our proposed methods, several scenarios were tested, examining the results from both simulation and artificial intelligence perspectives.

A. SIMULATION RESULTS

In this section, we will analyze the results produced by the simulation techniques proposed in this paper and compare them to previous methods to evaluate their superiority.

Figure 21 illustrates a comparison between dry curly hair and wet curly hair. Our method distinctly captures the effect of the increased weight of wet curly hair, which results in a more pronounced drooping effect compared to dry hair, and also accurately represents the cohesion among strands. In particular, the method enhances the curly shape when the hair is wet, which is more pronounced because of the exaggerated curly motion proposed in our paper.

Figure 22 shows the results of simulating wet curly hair using our method as the head rotates side to side. In the case of dry curly hair, the curly shapes of the strands are maintained not only by constraint forces, but also, as the head rotates sideways, the dry strands disperse due to gravity (see Figure 22a). In contrast, our method effectively represents the characteristics of wet curly hair, which do not scatter even when rotating side to side, due to saturation-induced clumping among the strands.

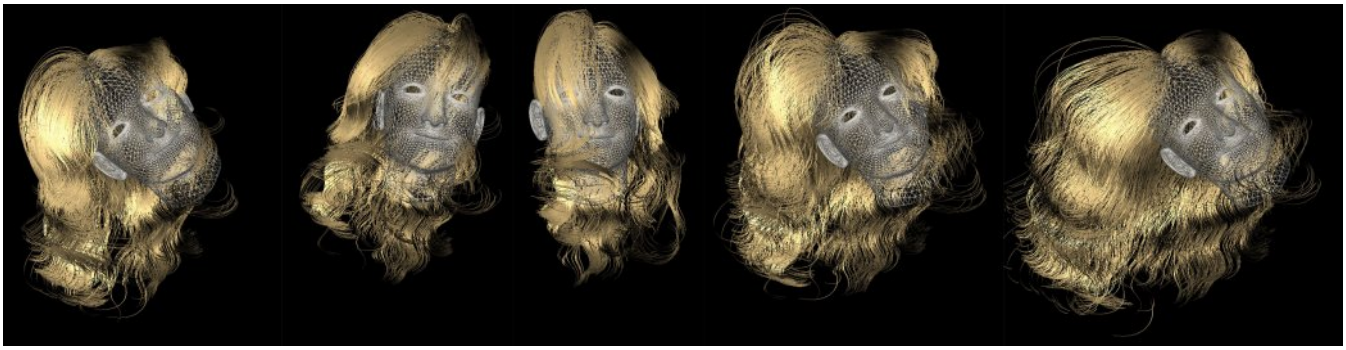
Figure 23 displays the results of wet curly hair simulations created using previous methods in the same scene. The work of Lin [2], [3] demonstrated that the cohesion forces expressed in curly hair appeared unstable, leading to unstable simulation results when applied to curly hair rather than straight hair. In contrast, the method by Kim et al. [6] improved stability but exhibited tremulous motions in the simulation, which ultimately failed to accurately represent the cohesion observed in wet hair.

Figure 24 shows the results of simulating wet curly hair using our method as the head rotates from side to side. This demonstrates our method's stability in preventing the hair from tangling, not only in static scenes but also under dynamic conditions.

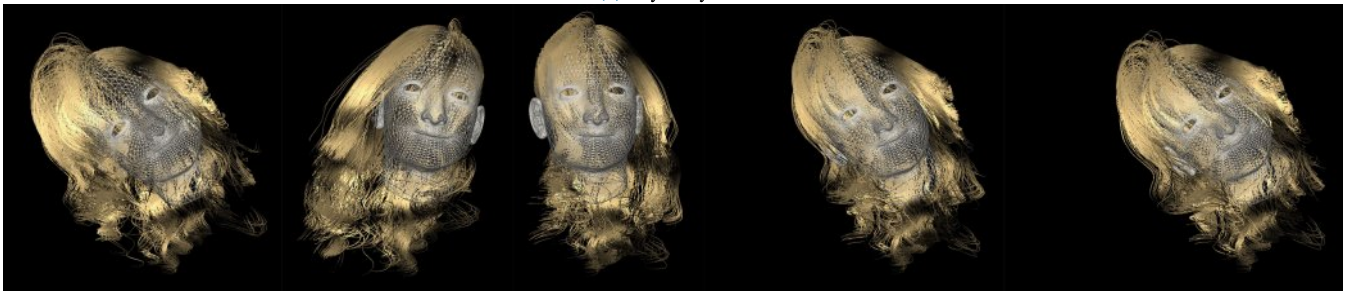
Figure 25 shows the results of the experiments carried out on short hair. Compared to dry hair, our method not only demonstrates the cohesion and curl typical of wet curly hair, but also shows a clear distinction in visual quality in the rendering results when compared to dry hair.

As previously mentioned, our method can reliably represent wet features not only in straight hair but also in curly hair. Figure 26 presents the results of comparing curl exaggeration between dry and wet strands in long curly hair. Using our method, wet curly hair exhibits pronounced shape exaggeration due to the use of weights based on helix. Users can modify the weight w_h to control the influence of the helix model.

Figure 27 shows how the shape of wet curly hair varies with changes in the saturation area and magnitude. In Figure 27a,

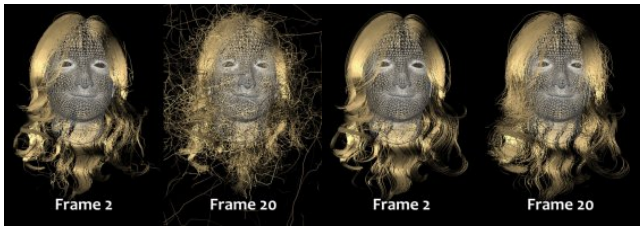


(a) Dry curly hair



(b) Wet curly hair (our method)

FIGURE 22. Quality comparison between (a) dry hair and (b) our method (scene3).



(a) Lin et al. [2], [3] (b) Kim et al. [6]

FIGURE 23. Wet curly hair simulations generated using the previous methods.

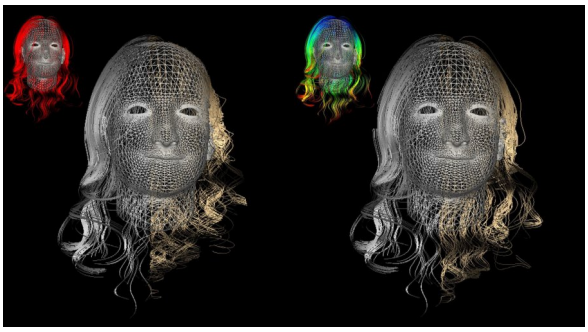
FIGURE 24. Turning head from side to side (scene2).

(a) Dry curly hair (b) Wet curly hair

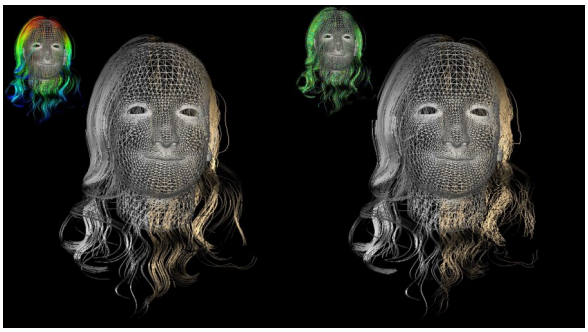
FIGURE 25. Quality comparison between (a) dry hair and (b) our method (scene4).

FIGURE 26. Quality comparison (only curl exaggeration) between (gray) dry hair and (yellow) our method (scene5).

the saturation of all the strands is set to maximum for the experiment, showing wet curly hair with a more pronounced curly form compared to dry curly hair. In Figure 27b, the saturation is relatively stronger at the tips of the strands than at the roots. In this figure, the effects of curl and clumping are applied more subtly than in Figure 27a. Since the tips are more saturated, the areas near the roots maintain characteristics closer to the original form of the hair, resulting in curly wet hair that still resembles the original length of the hair compared to Figure 27a. Figure 27c displays a scene where wetness is more pronounced at the root than

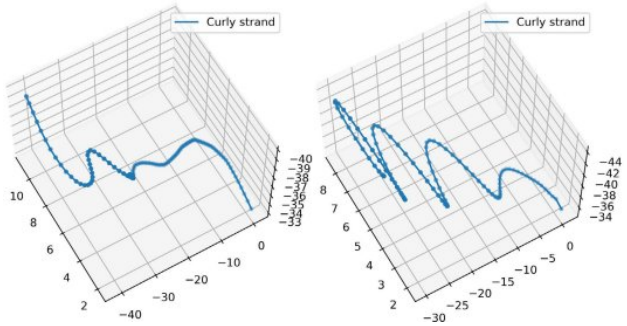


(a) Entire : strong saturation (b) Root(weak), tip(strong)



(c) Root(strong), tip(weak) (d) Entire : random saturation

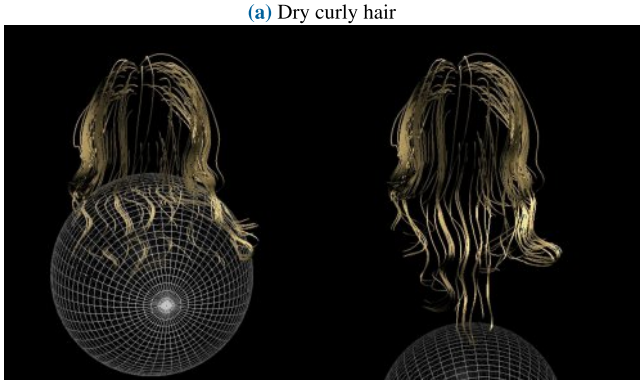
FIGURE 27. Change of wet curly hair according to saturation area and magnitude (inset image). The left and right images show dry and wet hair, respectively (scene6).



(a) A dry curly strand (b) A wet curly strand

FIGURE 28. Shape comparison between dry hair and our method (scene7).

(a) Dry curly hair



(b) Wet curly hair with our method

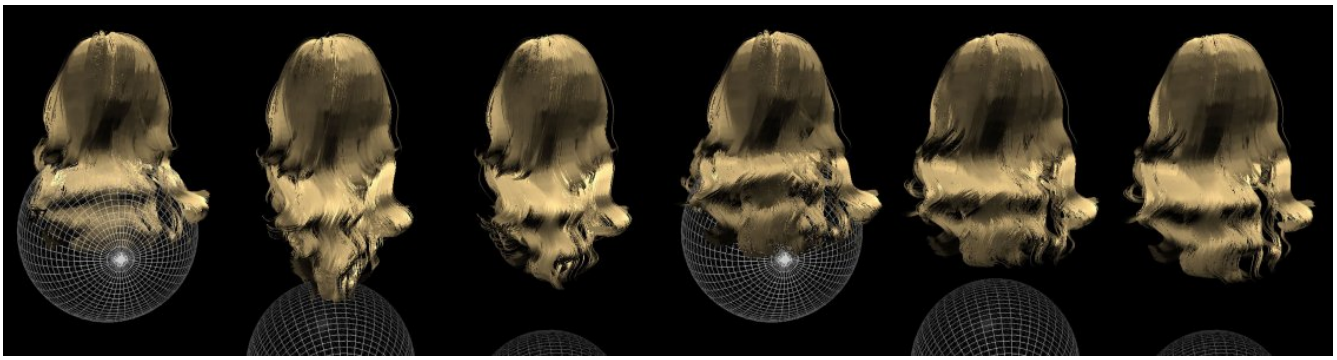
FIGURE 29. Scene with collision of low-resolution wet curly hair and sphere moving up and down (scene8).

at the tip. Because wet curly hair is expressed near the root, clumping effects are more prominently displayed in strands near the forehead and the left ear. Our method stably applies wet curly hair based on saturation, and to more naturally represent tangled strands in wet hair, jitter can be added to the saturation. Figure 27d shows results with random added jitter, which presents an additional appearance of tangled strands compared to previous results.

In Figure 28, our method was applied to test the morphological characteristics of a single curly strand in both its wet and dry states. Figure 28a displays the dry curly strand, which fundamentally maintains its curly shape throughout the length range of approximately (2∼10,0∼40,33∼40). When our method is applied to wet curly hair, as shown in Figure 28b, not only is the curl more pronounced compared to Figure 28a, but there is also a visible length contraction characteristic of wet curly hair. The total length of the wet curly strand is (2∼8,0∼30,34∼44), and comparing the values of the Y -axis indicates that the length of the dry curly strand has been reduced by approximately 25%.

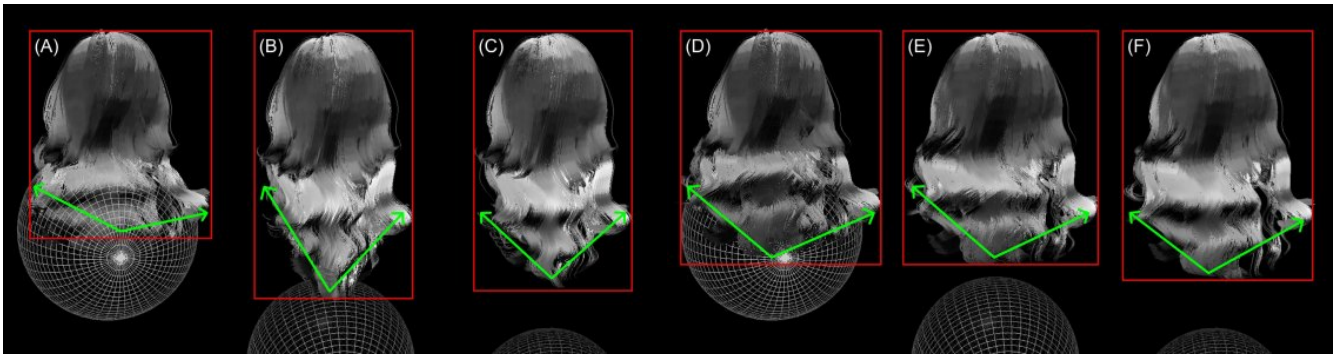
Figure 29 presents the results of experiments that tested the adhesion resulting from the interaction between wet curly hair and a solid sphere. Our method effectively represented adhesion effects stably even without the boundary particles. Figure 29a shows the encounter between the sphere and the dry curly hair, where the tips of the strands are displayed without any adhesion effects. However, in Figure 29b, the adhesion causes the tips of the strands to stretch significantly. The boundary particles approach efficiently handles the issue of objects passing through the boundary particles, and our method performs well not only in maintaining the wet curly shape but also in expressing adhesion, thus ensuring stable performance in both curly and straight hair types.

Figure 30 shows the results of experiments carried out at high resolution considering only adhesion. Our method effectively represented the shape of adhesion-induced wet strands even at high resolution in wet curly hair. In these tests, wet curly hair that collided with a sphere spread less widely compared to dry hair (see (A) in Figure 31), whereas dry hair spreads more broadly upon collision (see (D) in Figure 31). The actual width ratios were measured as 700 for (A) and 770 for (D) (see Table 2). (B) shows a clear representation



(a) Wet curly hair with our method **(b)** Dry curly hair

FIGURE 30. Scene with collision of high-resolution wet curly hair and sphere moving up and down (scene9).



(a) Wet curly hair with our method (A)∼(C) **(b)** Dry curly hair (D)∼(F)

FIGURE 31. Changes in the shape of a strand expressed by the cohesion of wet curly hair.

TABLE 2. Analytics table for Figure 31.

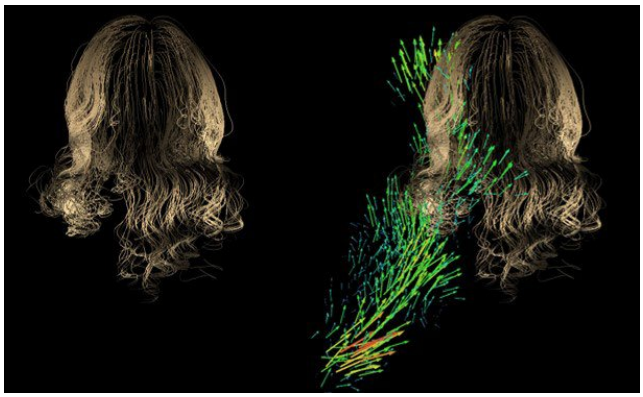
Figure	Red box (width/height)	Green arrow (degree)
31a(A)	700/800	140°
31a(B)	610/1030	70°
31a(C)	610/1000	100°
31b(D)	770/900	115°
31b(E)	750/900	115°
31b(F)	670/960	115°

of the shape of wet curly hair due to adhesion, with the hair stretching upward due to strong adhesion at the top (see (B) in Figure 31), while the dry hair does not stretch upward because it shows no adhesion (see (E) in Figure 31). The actual height ratios were 1030 for (B) and 900 for (E), indicating that the strands stretched upward (see Table 2). In this scene, not only does the strand stretch due to adhesion, but the angle between the adhered and non-adhered sections is distinctly different compared to dry hair (see green arrow in (B)). The actual angles were 70° for (B) and 115° for (E), clearly showing the features of the hair due to adhesion (see Table 2). As adhesion weakens, the angle slightly relaxes, but the stiffness and curl exaggeration due to saturation are maintained. As shown in the measured results, our method showed a smaller angle and a higher height compared to dry hair when adhesion weakened. These features confirm that the hair stretched by saturation and adhesion is maintained, whereas in dry curly hair, it spreads widely without these characteristics. In summary, as seen previously, our method not only captures the characteristics of wet curly hair but also stably represents cohesion, adhesion, and curl exaggeration in interactions with solid objects.

Figure 32 presents an experiment that involves the interaction between an air field and wet curly hair. While previous tests assessed the stability of wet curly hair representation under various external forces, this scene tests whether our method can effectively represent the characteristics of wet curly hair when interacting with an air field calculated using the Navier-Stokes equation. The air field was calculated using the Stable Fluids solver [39], with a simulation grid resolution of $64 \times 64 \times 64$. The wet curly hair rendered by our method maintained its curl and cohesion due to saturation without unraveling, even as the strands moved in response to the air field interaction (see Figure 32).

B. LEARNING RESULTS

In this section, the results produced by extending the simulation method with a neural network solver are presented. We examine whether the learning-based approach can



(a) Only strands (b) Air field and strands

FIGURE 32. Interaction of an air field with wet curly hair. The arrow indicates the direction of the air field, and the size/color is the magnitude of the air field, expressed as a rainbow color depending on the magnitude (scene10).

(a) Low resolution hair (scene11) (b) High resolution hair (scene12)

FIGURE 33. Simulating wet curly hair with our learning representation (gray: dry, color: wet).

(a) Other scene (scene13) (b) Other scene (scene14)

FIGURE 34. Various styles of wet curly hair with our learning representation (gray: dry, color: wet).

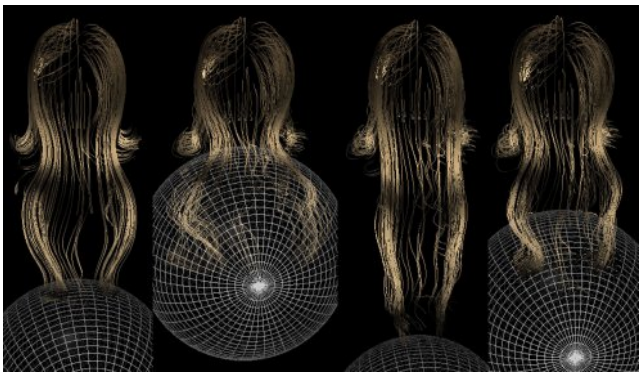


FIGURE 35. Interaction of wet curly hair and a solid using our learning representation (scene15).

(a) Dry curly hair (b) Wet curly hair

FIGURE 36. Quality comparison between (a) dry hair and (b) our method based on learning representation (scene16).

(a) (b) (c)

FIGURE 37. Results of wet hair under different levels of wetness.

represent wet curly hair at a level similar to the results shown previously.

Figure 33 shows the results of representing wet curly hair using our method extended with a neural network solver. Our method effectively expresses cohesion and curl exaggeration compared to dry curly hair, and it performs well not only at low resolution, but also at high resolution.

Figure 34 shows the results of simulating various styles of wet curly hair through learning representation. Similarly to the previously shown results, the method effectively represents wet features compared to dry curly hair, even using a neural network. In particular, the curly form observed at the tips of the wet strands is exaggerated, showing the cohesion effect well. Due to higher saturation at the tips of the strands, they are represented more prominently than other parts, making the difference compared to dry curly hair clearly evident.

Figure 35 shows the results of representing adhesion through learning representation when a solid interacts with wet curly hair. When the strands touch the surface of the sphere, they adhere to it, and as the sphere moves downward, the strands stretch downward because of the adhesion force. If the sphere moves downward beyond the force maintaining adhesion, the adhesion becomes stronger, and the curls are represented relatively weakly.

Figure 36 shows the results of representing wet curly hair by learning representation on short hair. The wet features were well represented even in short hair. The curl shape appeared to be more pronounced in short hair compared to long hair (see inset image in Figure 36). This characteristic was not observed in other scenes and can

(a) Strong adhesion (υ : 1,600)

(b) Weak adhesion (υ : 700)

(c) Normal adhesion (υ : 1,300)

FIGURE 38. Results of wet hair under different levels of adhesion.

be controlled by saturation. In addition, when measuring only the inference speed, the performance was sufficient for real-time processing, as shown in Figure 36b, where inference alone achieved 50 FPS. However, since factors such as scene complexity and hairstyle can affect performance, further optimization is considered necessary. The execution times measured for results generated via neural networks include the wet curly hair simulation process, which naturally incurs additional time. As previously explained, the learning representation simultaneously learns the difference between the geometry images of dry and wet curly hair and the SR process to obtain weights, which are then used for 3D reconstruction. Thus, the system is not executed as a fully learning-based model. Real-time and interactive applications generally target over 30 FPS, while games aim for over 60 FPS, and VR applications require over 90 FPS; therefore, the proposed learning model still needs further improvement.

Figure 37 shows the results of wet hair rendered under varying wetness levels. Figures∼37a and 37b all represent wet conditions, with Figure 3b exhibiting the highest wetness. To represent different wetness levels, one can adjust the value of σ and the range of neighboring hair particles j used in computing p_i^w.

Figure 38 presents the results of wet hair rendered under different adhesion levels. Both Figures 38a and 38b represent wet conditions, with Figure 38a exhibiting the strongest adhesion. To depict varying adhesion levels, one can adjust the magnitude of υ used in the calculation of F_i^{ad}.

Figure 39 shows the results of wet hair rendered under different stiffness levels. Both Figures 39a and 39b represent wet conditions, with Figure 39a exhibiting the highest stiffness. To represent results under varying stiffness levels, one can adjust the values of ξ and Υ used in the calculation of $F_i^a d$.

VI. DISCUSSION

Figure 40 is a chart that measures the performance of the results created using simulation and the learning representation. In the case of dry hair, the computational load is not

(a) Strong stiffness

(b) Weak stiffness

(c) Normal stiffness

FIGURE 39. Results of wet hair under different levels of stiffness.

TABLE 3. Scene configuration and performance.

Scene	Frames	# of strands	# of particles (per strand)	Computational time (min, max, mean)	Computational time (standard deviation)
1a	280	512	100	1.035s, 1.36s, 1.093s	0.05
1b	280	512	100	2.265s, 3.023s, 2.513s	0.176(wet)
2	280	512	100	2.229s, 2.781s, 2.452s	0.095
3a	280	512	100	1.03s, 1.21s, 1.057s	0.024
3b	280	512	100	2.133s, 3.296s, 2.34s	0.215(wet)
4a	280	9900	100	4.715s, 5.658s, 4.877s	0.136
4b	280	9900	100	17.406s, 28.926s, 22.036s	2.71(wet)
5(wet)	280	2000	100	2.093s, 2.664s, 2.23s	0.074
8a	280	256	100	0.254s, 0.308s, 0.271s	0.008
8b	280	256	100	0.27s, 0.321s, 0.284s	0.012(wet)
10	280	512	100	0.555s, 0.63s, 0.586s	0.012
16b	280	1900	100	0.352s, 0.389s, 0.36s	0.006

significant. However, the proposed method adds a relatively large computational load because it needs to calculate clumping and cohesion for each strand due to saturation even without external forces. Figure 40(l) shows the results created using the learning representation, where the same scene took an average of 1.084 seconds in the simulation results, but

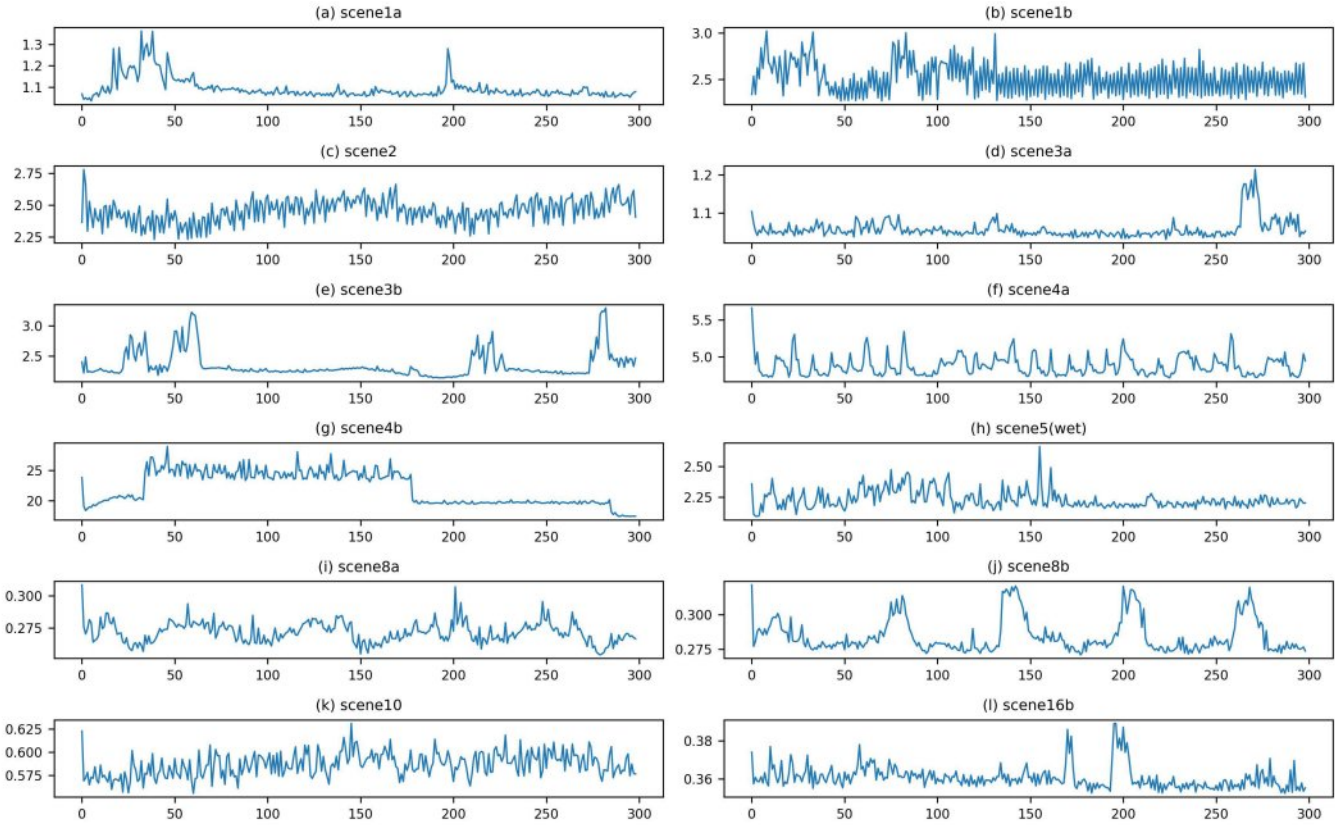


FIGURE 40. Comparison of execution times.

TABLE 4. Comparison of computational complexity.

Method	Neighbor Search	Energy Computation	Total Complexity
Brute-force	$O(N^2)$	$O(1)$	$O(N^2)$
Uniform Grid	$O(N)$	$O(1)$	$O(N)$
Kd-Tree / BVH	$O(NlogN)$	$O(1)$	$O(NlogN)$

TABLE 5. Comparing calculation times for dry and wet curly hairs (our mthod based on learning representation).

Comparison scenes	Computational time (mean)	Performance (dry/wet)
1a(dry) and 1b(wet)	1.096(1a), 2.513s(1b)	×0.436
3a(dry) and 3b(wet)	1.057(3a), 2.34s(3b)	×0.451
4a(dry) and 4b(wet)	4.877(3a), 22.036s(3b)	×0.221
8a(dry) and 8b(wet)	0.271(8a), 0.284s(8b)	×0.954

improved to 0.36 seconds with the learning representation, enhancing the performance by approximately 3 times. Table 3 presents the configurations and corresponding analysis results for each scenario experimented with in this paper. The computational time was measured for the minimum, maximum, and mean for each scene, and the standard deviation (SD) was also calculated. As shown in the table, the SD for dry curly hair was smaller as it approached the equilibrium state, whereas the SD for wet curly hair was relatively larger due to the continuous calculation of cohesion caused by saturation. Table 5 compares the computation times between our simulation-based method and dry curly hair. Our method increased the computation time by approximately 10∼50% compared to traditional methods due to the additional process needed to represent the wet strands.

A. LIMITATION

However, there are some limitations. Firstly, to represent cohesion stably in curly hair, adjacent strand particles are considered. If the search range h is too small, cohesion is weakly applied, resulting in unsightly wet strands. In contrast, if h is too large, the computational load increases.

This characteristic also affects the learning representation. Secondly, the results represented through the learning representation were not as robust as the simulation results in some scenes. Especially in scenes with strong external forces, there were issues with cohesion breaking apart. This limitation needs to be addressed by using more powerful neural networks or deep learning techniques.

As shown earlier, the proposed method performs stable wetting even on curly hair and represents clumping forces more reliably than previous methods. Moreover, by using a level-set approach to model adhesion with solids, it efficiently captures the adhesion effect observed in wet hair without requiring explicit boundary particles. However, several limitations still remain.

1) Curly hairstyles vary in intensity, including forms such as tightly coiled hair. However, this study focuses on the types of curly and straight hair typically handled in conventional hair simulation, and does not cover coiled hair. To include coiled hair, the hair dynamics would need to be modeled differently. Since the method proposed by Wu et al. can handle highly coiled hair, we plan to extend our algorithm to incorporate their approach in future work [40]
2) Our method computes the algorithm within a sub-iterative solver to calculate cohesion and adhesion, which are essential for representing wet curly hairs. This is necessary because the drooping behavior of curly strands when wet, along with cohesion and adhesion, must be expressed simultaneously with the curly shape. In this study, the number of iterations was set to 10. However, this approach reduces GPU efficiency, and future work should focus on optimizing the method to represent wet curly hair without requiring sub-iterations.
3) To represent stable cohesion, our method computes a new position, p_i^w, for visualizing wet hair. Instead of integrating forces over time to update positions—as done in traditional dynamics—we directly compute p_i^w in a manner similar to PBD to represent saturated curly hair, even under dynamic motion. This approach ensures stable wetting effects without noticeable jitter. However, it remains difficult to capture the curls formed by entangled wet strands. We believe that incorporating this phenomenon in future work could further enhance the detail and realism of wet hair simulation.

B. ALGORITHMIC COMPLEXITY

The algorithmic complexity of the self-cohesion proposed in our method involves energy computation with respect to neighboring particles and proceeds through the following steps:

1) A neighbor search is performed to identify hair particles adjacent to each hair particle.
2) Using the distances between neighboring particles, the wet hair position p_i^w is computed. In this process, an isotropic kernel function $W(r, h)$ is used to compute weights, and depending on the physical model, quantities such as kinetic energy and elastic energy are calculated.

The energy computation itself has a complexity of $O(k)$, where k is the number of neighboring hair particles. Since k is typically limited to around 20∼50, it is treated as a constant, making the energy computation for each hair particle effectively $O(1)$. However, the overall complexity is dominated by the neighbor search process, which takes $O(N)$ or $O(NlogN)$ depending on the spatial data structure used. Thus, the total computational complexity is determined by the neighbor search.

Considering the entire hair particle system:

- Neighbor search: $O(N)$ using a uniform grid or $O(NlogN)$ using a Kd-tree.

Therefore, the final computational complexity for the entire system is as summarized in Table 2.

The level-set-based hair-solid adhesion proposed in our method involves computing the solid's level-set, which proceeds as follows. A Signed Distance Field (SDF) is constructed using a voxel grid, where the 3D space is discretized and distances to the nearest surface are computed. Typically, methods such as the Fast Marching Method (FMM) or Fast Sweeping Method (FSM) are used. The complexity of this preprocessing step is $O(M)$, where M is the resolution of the grid. Since SDF generation is generally performed as a one-time preprocessing step, it only incurs this $O(M)$ cost if real-time updates are not required.

To calculate hair-solid adhesion, each particle must sample the SDF value. In the case of a uniform grid SDF, trilinear interpolation is used, and its computational complexity is $O(1)$ (per interpolation operation for each hair particle). Next, using the sampled SDF value ϕ, adhesion force is computed, and this process requires $\nabla\phi$ to calculate the force direction, which also has a complexity of $O(1)$ (simple calculation per particle). If the SDF is for a single solid, each hair particle only needs to sample once, so the total complexity is $O(N)$. However, if multiple solids exist, the SDF of the closest solid must be selected, resulting in additional search cost. The search cost is $O(logS)$ (where S is the number of solids). In conclusion, the computational complexity of adhesion force calculation per hair particle is $O(1)$, and for the entire hair system, it becomes $O(N)$ (in the case of a single solid) or $O(NlogS)$ (in the case of multiple solids).

C. DIFFERENCE BETWEEN LEVEL-SET-BASED ADHESION AND PARTICLE-BASED BOUNDARY HANDLING

One of the main advantages of the boundary particle method used in particle-based fluids is that it integrates fluid simulation and boundary handling into a unified particle-based framework, making it highly efficient on GPU architectures. However, this method also has several drawbacks. In SPH, symmetric forces are not always accurately computed near boundaries, leading to unstable pressure estimations.

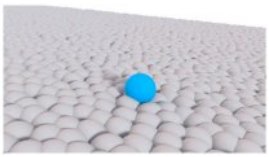


FIGURE 41. A single particle flowing down an inclined plane approximated by boundary particles.

Therefore, one of the primary goals is to approximate the solid boundary using particle structures similar to fluid particles so that symmetric forces can be correctly calculated. Boundary particles must be sampled across the entire object surface, significantly increasing their number. Moreover, when the velocity of fluid particles is high, tunneling problems may occur. To address this, multiple layers of boundary particles or adaptive time-stepping is often employed. Yet, such solutions make simulation handling more complex and require smaller time-steps, which in turn increases computation time. To overcome these limitations, our method employs a level-set approach that allows for preprocessing to express adhesion effects. Thanks to this preprocessing, adhesion can be represented with near real-time performance during execution, making it suitable not only for hair and fur simulations but also for broader interactive applications.

D. BALANCING REALISM AND COMPUTATIONAL EFFICIENCY

The approach used in this study is not limited to static scenes or straight hair but aims to stably represent wetting effects in curly hair even in dynamic scenes. To robustly handle the cohesion process caused by saturation, the target positions where cohesion occurs are directly computed in a manner similar to PBD, rather than being integrated from forces. As this method does not require time integration or velocity, it is not only stable but also computationally efficient; however, the lack of time-dependent numerical integration may lead to a slight loss in accuracy.

Hair-solid adhesion is handled using a volumetric data structure based on the level-set, allowing for pre-processing and requiring only sampling during the online stage, which ensures real-time performance. However, if the solid is a deformable body, the level-set must be recalculated at every timestep, potentially reducing efficiency. This issue is not unique to our method and also arises when using boundary particle approaches. In particular, spherical boundary particles often fail to represent smooth motion on the surface, resulting in bouncing artifacts. To address this issue, some studies have proposed constructing density maps on SPH boundaries for improved collision handling [43] (see Figure 41). In this study, although post-processing is required, we designed hair-solid adhesion using a level-set method that eliminates noisy motion from boundary particles and guarantees real-time performance during the online process. However, this approach requires storing the level-set field in memory, which may lead to increased memory usage.

E. REAL-WORLD APPLICATIONS

The technique developed in this study for stably representing adhesion and cohesion in wet curly hair can play a significant role across various fields, which we explain in detail as follows.

1) In the film and animation industry, the proposed technique can contribute to achieving realistic hair simulation. In particular, naturally representing the adhesion and tangling behavior of wet hair strands—an aspect that has been challenging with conventional methods—can be effectively addressed using our approach. This allows for more realistic reproduction of hair textures and can be applied in CG-based films, 3D animations, commercials, and VFX production workflows.
2) In the game industry, the proposed technique enables realistic physical representation of character hair. In particular, simulating hair behavior under rain or underwater conditions has been difficult to achieve naturally with conventional simple physics-based simulations. By applying the method proposed in this study, more refined hair motion can be realized even in real-time rendering environments. This enhances the overall immersion of the game and maximizes the vividness of the characters.
3) In the virtual fitting and beauty industry, the proposed technique can be applied to hair styling simulation. For example, by integrating this technology into virtual hairstyling apps or beauty software, it becomes possible to more accurately predict shape changes of wet hair and the natural persistence of curls. This can be particularly useful for hairstylists when planning and designing hairstyles.
4) The proposed technique can also be utilized in VR/AR technologies. For instance, it can be applied to ensure that a character's hair in VR-based interactive content responds naturally according to physical laws. This enhances the realism of the user experience and contributes to the development of high-quality immersive content.

F. SYNTHETIC DATA GENERATION

In this study, data were synthetically generated through physically-based hair simulation. To compute the dynamics of curly hair, we adopted the method proposed by Iben et al. However, due to the nonlinear motion of 3D strands, we generate synthetic data by converting geometric information into geometry images for efficient learning. A geometry image pair corresponding to dry and wet curly hair is constructed, and a learning step is performed to find the weight values between them. Super-resolution (SR) is applied to represent high-resolution hair, and a nonlinear transform is used to reconstruct 3D strand data from the geometry image. The network model is structurally similar to a previously successful learning-based hair representation

TABLE 6. Comparison of technical elements between previous methods and our method.

Method	Wet straight hair	Wet curly hair	Static scene	Dynamic scene
Rungjiratananon et al. [1]	✓	✗	✓	✗
Lin et al. [2], [3]	✓	✗	✓	✓
Lenaerts et al. [20]	✗	✗	✓	✓
Kim et al. [6]	✓	✗	✓	✓
Fei et al. [26]	✓	✗	✓	✓
Lee et al. [48]	✓	✗	✓	✗
Our method	✓	✓	✓	✓

framework [36], and it is extended here to handle wet hair. Geometry images are used in the hair-hair cohesion process, but not in hair-object adhesion.

G. EXTENDING TO TRIANGULAR MESH BASED HAIRS

Some studies in physically-based simulation have already explored methods for reconstructing 3D data using geometry images [44], [45], [46]. A common feature of these studies is the encoding and decoding of all positional information into 2D geometry images. However, when there are large color differences between adjacent pixels in a geometry image, the reconstructed vertex positions can be significantly affected. Due to this issue, previous studies have encountered difficulties in generalizing across various scenes and often compensated using different data structures. Although our method is fundamentally designed based on a quadrilateral structure, it can be extended to triangular meshes. When root particles are generated in a non-rectangular shape, a parameterization process is required, and various techniques have been proposed to address this. The As-Rigid-As-Possible (ARAP) method allows the generation of geometry images even for non-rectangular shapes, which was experimentally validated in the work of Chen et al. [44], [45]. In this study, we applied this approach to generate virtual strands based on neural networks when three root particles form a triangle.

H. TECHNICAL POINTS NOT CONSIDERED IN PREVIOUS RESEARCH

This section compares five techniques closely related to our method, focusing on the technical limitations of previous methods and how these limitations affect the simulation of curly hair.

Rungjiratananon et al. [1] proposed a grid-based approach to simulate the physical motion of wet hair. In this method, individual water droplets are represented as particles, and their motion is calculated by considering the adhesive properties of hair. Smoothed Particle Hydrodynamics (SPH) was used to compute fluid dynamics between droplets, and implicit Euler time integration was adopted to ensure simulation stability. This method successfully simulated the natural behavior of water flowing along the hair when wet and the gradual shape change as hair dries. However, due to the use of implicit integration, the computational cost is significantly high. Furthermore, detailed hair-water interaction characteristics were not fully captured. They primarily targeted static scenes, making it difficult to accurately simulate fast hair motion or strong wind effects, and it was designed under the assumption of straight hair only.

Lin [2], [3] modeled water diffusion within hair using SPH-based fluid simulation. They treated hair as a porous medium and applied the concept of anisotropic permeability, allowing water to flow more easily along the fiber direction. This method also modeled wetting and self-adhesion of hair due to saturation and reproduced the cohesion effects among wet hair strands using an SPH-based adhesion model. However, this approach also assumed only straight hair. Unlike the previous method, it extended the simulation to underwater scenarios and effectively illustrated hair motion in submerged conditions, but it primarily focused on static scenes.

Lenaerts et al. [20] proposed a two-way coupling method for simulating materials and fluids interacting through porous deformable media. They reinterpreted Darcy's law within the SPH framework to simulate both fluid and deformable objects. While most of their results focus on deformable bodies and this method is applicable to cloth simulation, it is difficult to apply directly to strand-based structures such as hair.

Kim et al. [6] proposed a method for efficiently and stably representing hair clumping effects using SPH-based surface tension. Notably, they demonstrated that clumping effects remained stable across various dynamic scenes, such as a rotating sphere, running horse, head-banging, and jumping motions, rather than just static scenes. However, all target objects in their study were limited to straight hair or fur.

Fei et al. [5] proposed a multi-scale model capable of capturing the detailed characteristics of wet hair. To solve the motion of liquid along hair strands, they introduced a reduced-dimensional liquid model that accounts for the moving reference frame and the influence of hair dynamics. They also derived a model that faithfully reflects cohesion effects induced by surface tension, based on the geometric structure of liquid bridges formed between hair strands. To handle macroscopic interactions between hair and surrounding fluids, they adopted an experimentally validated drag model and proposed a novel volume-conserving strategy for liquid dripping and absorption. This method enables liquid motion using a reduced-dimensional model combined with a particle-cell framework. While these techniques were demonstrated in dynamic scenes such as hair flipping,

animal fur drying, spinning car wash roller brushes, and hair coalescence effects, the model was primarily designed for straight hair, and thus the unique characteristics and simulation stability of curly hair were not addressed.

Lee et al. [47] proposed a novel framework for hair-water interaction using a tetrahedral volume mesh. After embedding the tetrahedral mesh, they animated the hair by deforming it according to a kinematically skinned mesh. Their approach included methods for storing porosity within the kinematic mesh and applying drag and adhesion forces through the kinematically deformed structure. However, most of their experiments were conducted in static scenes, and their method focused solely on straight hair or fur.

The aforementioned methods have proposed various hair-water interaction techniques, but most of them focused only on straight hair or conducted experiments limited to static scenes. As a result, they are difficult to generalize for use with curly hair in more dynamic or diverse environments. Furthermore, when the subject shifts from straight to curly hair, force instability issues arise, making wet hair simulation more challenging, as discussed in the problem statement. This method proposed in this study addresses these challenges efficiently, and a comparison with previous methods is summarized in Table 6.

I. EVALUATION OF KINETIC/ELASTIC ENERGY DIVERGENCE

The numerical stability addressed in this study has been explained in the problem statement. When transitioning from straight hair to curly hair, numerical instability arises due to the curly geometry, causing noisy forces, and ultimately resulting in jittery movements of strands during simulation. Particularly in curly hair, simulations either fail to converge (see Figure 2b) or exhibit significant jitter due to noise in the positions of hair particles (see Figures 2c, 3, 4, and 5). The primary cause of this issue is numerical instability during the calculation and propagation of forces. To demonstrate convergence and numerical stability, this section performs an Energy Stability Test on the Mass-Spring system to verify whether energy is conserved or stably damped. Although numerical instability visibly manifests in previous methods, our approach quantitatively evaluates stability through energy measurements. Here, the total energy E is computed from kinetic energy ($E_k = \sum \frac{1}{2}mv^2$) and elastic energy ($E_p = \sum \frac{1}{2}k(x - x_0)$): $E = E_k + p$. A convergence test assessing the time-dependent energy variation was conducted to confirm whether the energy remains stable as the simulation progresses. While the previous methods for wet curly hair, which included noticeable visual noise, all demonstrated unstable or divergent patterns, our proposed method exhibited a stable pattern without abrupt energy fluctuations.

J. REASON FOR USING THE VGG19 ARCHITECTURE

The proposed method in this study is not restricted to using only the VGG19 architecture. We employed VGG19 because it has been successfully applied in similar hair simulation research in previous work [36]. VGG19 allows for stable training due to its relatively simple yet deep architecture, effectively extracting fine details using small filters. Additionally, its comparatively simpler network structure makes it more efficient than ResNet or Inception. ResNet, on the other hand, enables stable training of deeper networks than VGG19 and mitigates the vanishing gradient problem through residual connections. Inception employs multiple filters of varying sizes simultaneously, and EfficientNet provides high performance with fewer parameters. Therefore, the choice of each network architecture may yield different trade-offs in terms of computational efficiency and quality. Nevertheless, considering experimental results across various scenarios, VGG19 demonstrated the most satisfactory balance between quality and efficiency and has shown strong performance in prior studies, leading us to determine it as the most appropriate choice.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a simulation technique that efficiently models the cohesion, adhesion, and stiffness of the strands, which are difficult to stably represent in wet curly hair, and a learning representation technique using this simulation method. In wet curly hair, the uneven distribution of forces leads to noisy movements, making it challenging to stably represent cohesion, adhesion, and curl shape. To address this issue, we proposed the following methods: 1) curl exaggeration using a local-transformed helix, 2) deformation-based cohesion that operates stably even in wet curly hair, 3) level-set-based adhesion to represent the stickiness and elongation of wet hair, and 4) dynamic stiffness to improve simulation stability, 5) collecting a synthetic dataset of detailed curly hairs and extending the solver to represent the movement of particles in wet strands through learning representation. We tested our proposed methods in various scenes and confirmed that wet curly hair was well represented in both simulation results and the learning representation. In the future, we plan to simulate the physical phenomena of strands undergoing state changes due to heat or freezing and to extend the algorithm to address the aforementioned limitations.

REFERENCES

[1] W. Rungjiratananon, Y. Kanamori, and T. Nishita, "Wetting effects in hair simulation," *Comput. Graph. Forum*, vol. 31, no. 7, pp. 1993–2002, Sep. 2012.

[2] W.-C. Lin, "Coupling hair with smoothed particle hydrodynamics fluids," in *Proc. Workshop Virtual Reality Interact. Phys. Simul.*, Jan. 2014, pp. 109–117.

[3] W.-C. Lin, "Boundary handling and porous flow for fluid–hair interactions," *Comput. Graph.*, vol. 52, pp. 33–42, Nov. 2015.

[4] C. Yuksel, S. Schaefer, and J. Keyser, "Hair meshes," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–7, Dec. 2009.

[5] Y. (. Fei, H. T. Maia, C. Batty, C. Zheng, and E. Grinspun, "A multi-scale model for simulating liquid-hair interactions," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–17, Aug. 2017.

[6] J.-H. Kim, W. Kim, Y. B. Kim, J. Im, J. Lee, and S.-J. Kim, "Robust handling of clumping and stiffness in wet hair animation," *Comput. Animation Virtual Worlds*, vol. 28, no. 6, pp. e179, Nov. 2017.
[7] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "MeshCNN: A network with an edge," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Aug. 2019.
[8] P. W. Battaglia et al., "Relational inductive biases, deep learning, and graph networks," 2018, *arXiv:1806.01261*.
[9] J. Gilmer, S. S. Schoenholz, P. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2017, pp. 1263–1272.
[10] Á. Sánchez-González, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2018, pp. 4470–4479.
[11] H. Iben, J. Brooks, and C. Bolwyn, "Holding the shape in hair simulation," in *Proc. ACM SIGGRAPH Talks*, Jul. 2019, pp. 1–2.
[12] A. Selle, M. Lentine, and R. Fedkiw, "A mass spring model for hair simulation," in *Proc. ACM SIGGRAPH Papers*, Aug. 2008, pp. 1–11.
[13] M. Müller, T.-Y. Kim, and N. Chentanez, "Fast simulation of inextensible hair and fur," in *Proc. Workshop Virtual Reality Interact. Phys. Simul.*, Jan. 2012, pp. 39–44.
[14] T. Kugelstadt and E. Schömer, "Position and orientation based cosserat rods," in *Proc. Symp. Comput. Animation*, Jul. 2016, pp. 169–178.
[15] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, "Discrete elastic rods," in *Proc. ACM SIGGRAPH Papers*, Aug. 2008, pp. 1–12.
[16] D. M. Kaufman, R. Tamstorf, B. Smith, J.-M. Aubry, and E. Grinspun, "Adaptive nonlinearity for collisions in complex rod assemblies," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 1–12, Jul. 2014.
[17] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *J. Vis. Commun. Image Represent.*, vol. 18, no. 2, pp. 109–118, Feb. 2007.
[18] H. Iben, M. Meyer, L. Petrovic, O. Soares, J. Anderson, and A. Witkin, "Artistic simulation of curly hair," in *Proc. 12th ACM SIGGRAPH/Eurographics Symp. Comput. Animation*, Jul. 2013, pp. 63–71.
[19] G. Gornowicz and S. Borac, "Efficient and stable approach to elasticity and collisions for hair animation," in *Proc. Symp. Digit. Prod.*, Aug. 2015, pp. 41–49.
[20] T. Lenaerts, B. Adams, and P. Dutré, "Porous flow in particle-based fluid simulations," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–8, Aug. 2008.
[21] T. Lenaerts and P. Dutré, "Mixing fluids and granular materials," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 213–218, Apr. 2009.
[22] S. Baek, K. Um, and J. Han, "Muddy water animation with different details," *Comput. Animation Virtual Worlds*, vol. 26, nos. 3–4, pp. 347–355, May 2015.
[23] Y. Chen, N. Magnenat Thalmann, and B. Foster Allen, "Physical simulation of wet clothing for virtual humans," *Vis. Comput.*, vol. 28, nos. 6–8, pp. 765–774, Jun. 2012.
[24] K. Um, T.-Y. Kim, Y. Kwon, and J. Han, "Porous deformable shell simulation with surface water flow and saturation," *Comput. Animation Virtual Worlds*, vol. 24, nos. 3–4, pp. 247–254, May 2013.
[25] Y. (. Fei, C. Batty, E. Grinspun, and C. Zheng, "A multi-scale model for simulating liquid-fabric interactions," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–16, Aug. 2018.
[26] A. Fick, "Ueber diffusion," *Annalen der Physik*, vol. 170, no. 1, pp. 59–86, Jan. 1855.
[27] B. Das, A. Das, V. K. Kothari, R. Fanguiero, and M. D. Araujo, "Moisture transmission through textiles," *Autex Res. J.*, vol. 7, no. 2, pp. 100–110, Jun. 2007.
[28] Darcy and H. P. Gaspard, "Détermination Des lois découlement de l'eau à travers le sable," Tech. Rep., 1856.
[29] S. Bandara and K. Soga, "Coupling of soil deformation and pore fluid flow using material point method," *Comput. Geotechnics*, vol. 63, pp. 199–214, Jan. 2015.
[30] E. B. Pitman and L. Le, "A two-fluid model for avalanche and debris flows," *Phil. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 363, no. 1832, pp. 1573–1601, Jul. 2005.
[31] R. I. Borja, "On the mechanical energy and effective stress in saturated and unsaturated porous continua," *Int. J. Solids Struct.*, vol. 43, no. 6, pp. 1764–1786, Mar. 2006.
[32] X. Song and R. I. Borja, "Mathematical framework for unsaturated flow in the finite deformation range," *Int. J. Numer. Methods Eng.*, vol. 97, no. 9, pp. 658–682, Mar. 2014.
[33] K. Abe, K. Soga, and S. Bandara, "Material point method for coupled hydromechanical problems," *J. Geotechnical Geoenvironmental Eng.*, vol. 140, no. 3, Mar. 2014, Art. no. 04013033.
[34] G. Daviet and F. Bertails-Descoubes, "Simulation of drucker-prager granular flows inside Newtonian fluids," Tech. Rep., 2017.
[35] R. Bridson, S. Marino, and R. Fedkiw, "Simulation of clothing with folds and wrinkles," in *Proc. ACM SIGGRAPH Courses*, Jul. 2003, pp. 28–36.
[36] J.-H. Kim and J. Lee, "Efficient and stable generation of high-resolution hair and fur with ConvNet using adaptive strand geometry images," *IEEE Access*, vol. 11, pp. 81101–81112, 2023.
[37] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[38] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[39] J. Stam, "Stable fluids," in *Seminal Graphics Papers: Pushing the Boundaries*, vol. 2, 2023, pp. 779–786.
[40] H. Wu, A. Shi, A. M. Darke, and T. Kim, "Curly-cue: Geometric methods for highly coiled hair," in *Proc. SIGGRAPH Asia*, Dec. 2024, pp. 1–11.
[41] A. Shi, H. Wu, J. Parr, A. M. Darke, and T. Kim, "Lifted curls: A model for tightly coiled hair simulation," *ACM Trans. Graph.*, vol. 6, no. 3, p. 42, Aug. 2023.
[42] A. M. Darke, I. Olander, and T. Kim, "More than killmonger locs: A style guide for black hair," in *Proc. ACM SIGGRAPH*, 2024, pp. 1–123.
[43] D. Koschier and J. Bender, "Density maps for improved SPH boundary handling," in *Proc. ACM SIGGRAPH / Eurographics Symp. Comput. Animation*, Jul. 2017, pp. 1–10.
[44] L. Chen, J. Ye, L. Jiang, C. Ma, Z. Cheng, and X. Zhang, "Synthesizing cloth wrinkles by CNN-based geometry image superresolution," *Comput. Animation Virtual Worlds*, vol. 29, nos. 3–4, pp. e181, May 2018.
[45] L. Chen, J. Ye, and X. Zhang, "Multi-feature super-resolution network for cloth wrinkle synthesis," *J. Comput. Sci. Technol.*, vol. 36, no. 3, pp. 478–493, Jun. 2021.
[46] J.-H. Kim, S.-J. Kim, and J. Lee, "Geometry image super-resolution with AnisoCBConvNet architecture for efficient cloth modeling," *PLoS ONE*, vol. 17, no. 8, Aug. 2022, Art. no. e0272433.
[47] M. Lee, D. Hyde, M. Bao, and R. Fedkiw, "A skinned tetrahedral mesh for hair animation and hair-water interaction," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 3, pp. 1449–1459, Mar. 2019.